```
In [2]:  import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         import seaborn as sns
```

```
In [3]:  behaviour_df = pd.read_csv(r"C:\Users\Admin\Desktop\QVI_purchase_behaviour (1).csv")
         behaviour_df.head()
```

Out[3]:

	LYLTY_CARD_NBR	LIFESTAGE	PREMIUM_CUSTOMER
0	1000	YOUNG SINGLES/COUPLES	Premium
1	1002	YOUNG SINGLES/COUPLES	Mainstream
2	1003	YOUNG FAMILIES	Budget
3	1004	OLDER SINGLES/COUPLES	Mainstream
4	1005	MIDAGE SINGLES/COUPLES	Mainstream

```
In [4]:  transaction_df = pd.read_csv(r"C:\Users\Admin\Desktop\QVI_transaction_data.csv")
         transaction_df.head()
```

Out[4]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES
0	43390	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0
1	43599	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3
2	43605	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9
3	43329	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	15.0
4	43330	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	13.8

Explore the data(EDA)

```
In [5]:    behaviour_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 72637 entries, 0 to 72636
Data columns (total 3 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   LYLTY_CARD_NBR    72637 non-null  int64
 1   LIFESTAGE         72637 non-null  object
 2   PREMIUM_CUSTOMER  72637 non-null  object
dtypes: int64(1), object(2)
memory usage: 1.7+ MB
```

```
In [6]:    transaction_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 264836 entries, 0 to 264835
Data columns (total 8 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   DATE            264836 non-null  int64
 1   STORE_NBR       264836 non-null  int64
 2   LYLTY_CARD_NBR  264836 non-null  int64
 3   TXN_ID          264836 non-null  int64
 4   PROD_NBR        264836 non-null  int64
 5   PROD_NAME       264836 non-null  object
 6   PROD_QTY        264836 non-null  int64
 7   TOT_SALES       264836 non-null  float64
dtypes: float64(1), int64(6), object(1)
memory usage: 16.2+ MB
```

```
In [7]:    behaviour_df.describe()
```

	LYLTY_CARD_NBR
count	7.263700e+04
mean	1.361859e+05
std	8.989293e+04
min	1.000000e+03
25%	6.620200e+04
50%	1.340400e+05
75%	2.033750e+05
max	2.373711e+06

```python
transaction_df.describe()
```

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_QTY	TOT_SALES
count	264836.000000	264836.00000	2.648360e+05	2.648360e+05	264836.000000	264836.000000	264836.000000
mean	43464.036260	135.08011	1.355495e+05	1.351583e+05	56.583157	1.907309	7.304200
std	105.389282	76.78418	8.057998e+04	7.813303e+04	32.826638	0.643654	3.083226
min	43282.000000	1.00000	1.000000e+03	1.000000e+00	1.000000	1.000000	1.500000
25%	43373.000000	70.00000	7.002100e+04	6.760150e+04	28.000000	2.000000	5.400000
50%	43464.000000	130.00000	1.303575e+05	1.351375e+05	56.000000	2.000000	7.400000
75%	43555.000000	203.00000	2.030942e+05	2.027012e+05	85.000000	2.000000	9.200000
max	43646.000000	272.00000	2.373711e+06	2.415841e+06	114.000000	200.000000	650.000000

```python
print(transaction_df.isnull().sum())
```

```
DATE                 0
STORE_NBR            0
LYLTY_CARD_NBR       0
TXN_ID               0
PROD_NBR             0
PROD_NAME            0
PROD_QTY             0
TOT_SALES            0
dtype: int64
```

In [10]: `print(behaviour_df.isnull().sum())`

```
LYLTY_CARD_NBR       0
LIFESTAGE            0
PREMIUM_CUSTOMER     0
dtype: int64
```

Merged the Datasets

In [11]: `merged_df = pd.merge(transaction_df, behaviour_df, on='LYLTY_CARD_NBR' , how='inner')`

In [12]: `merged_df.head()`

Out[12]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	LIFESTAGE	PREM
0	43390	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0	YOUNG SINGLES/COUPLES	
1	43599	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3	MIDAGE SINGLES/COUPLES	
2	43605	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9	MIDAGE SINGLES/COUPLES	
3	43329	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	15.0	MIDAGE SINGLES/COUPLES	
4	43330	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	13.8	MIDAGE SINGLES/COUPLES	

In [13]:
```python
merged_df['DATE']= pd.to_datetime('1899-12-30')+ pd.to_timedelta(merged_df['DATE'],unit='D')
```
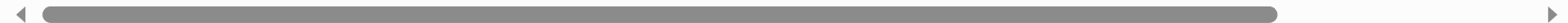
In [14]:
```python
merged_df.head()
```

Out[14]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	LIFESTAGE	PREM
0	2018-10-17	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0	YOUNG SINGLES/COUPLES	
1	2019-05-14	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3	MIDAGE SINGLES/COUPLES	
2	2019-05-20	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9	MIDAGE SINGLES/COUPLES	
3	2018-08-17	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	15.0	MIDAGE SINGLES/COUPLES	
4	2018-08-18	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	13.8	MIDAGE SINGLES/COUPLES	

In [15]:
```python
merged_df['Month'] = merged_df['DATE'].dt.month_name()
```

In [16]:
```python
merged_df.head()
```

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	LIFESTAGE	PREN
0	2018-10-17	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0	YOUNG SINGLES/COUPLES	
1	2019-05-14	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3	MIDAGE SINGLES/COUPLES	
2	2019-05-20	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9	MIDAGE SINGLES/COUPLES	
3	2018-08-17	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	15.0	MIDAGE SINGLES/COUPLES	
4	2018-08-18	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	13.8	MIDAGE SINGLES/COUPLES	

Data Cleaning

```
In [17]: merged_df.drop_duplicates(inplace = True)
```
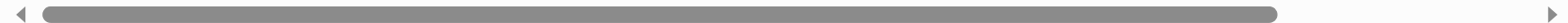
```
In [18]: merged_df.dropna(inplace = True)
```

```
In [19]: merged_df['DATE'] = pd.to_datetime(merged_df['DATE'])
```

```
In [20]: merged_df.head()
```

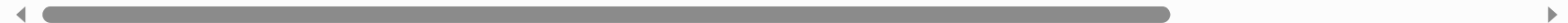|	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	LIFESTAGE	PREM
0	2018-10-17	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0	YOUNG SINGLES/COUPLES	
1	2019-05-14	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3	MIDAGE SINGLES/COUPLES	
2	2019-05-20	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9	MIDAGE SINGLES/COUPLES	
3	2018-08-17	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	15.0	MIDAGE SINGLES/COUPLES	
4	2018-08-18	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	13.8	MIDAGE SINGLES/COUPLES	

In [21]: `merged_df['Total_revenue'] = merged_df['PROD_QTY'] * merged_df['TOT_SALES']`

In [22]: `merged_df.head()`

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	LIFESTAGE	PREN
0	2018-10-17	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0	YOUNG SINGLES/COUPLES	
1	2019-05-14	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3	MIDAGE SINGLES/COUPLES	
2	2019-05-20	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9	MIDAGE SINGLES/COUPLES	
3	2018-08-17	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	15.0	MIDAGE SINGLES/COUPLES	
4	2018-08-18	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	13.8	MIDAGE SINGLES/COUPLES	

Total sales by revenue

In [23]:
```python
import plotly.io as pio
pio.renderers.default = 'notebook'
```

In [24]:
```python
monthly_revenue = merged_df.groupby('Month')['Total_revenue'].sum().reset_index()
monthly_revenue
```

Out[24]:

	Month	Total_revenue
0	April	313360.30
1	August	449475.65
2	December	329150.80
3	February	295523.40
4	January	318608.70
5	July	323612.10
6	June	314343.20
7	March	325716.00
8	May	444728.75
9	November	313813.70
10	October	322330.50
11	September	314797.60

In [25]:
```python
fig = px.line(monthly_revenue,
              x='Month',
              y='Total_revenue',
              title='Monthly Sales Analysis',
          markers=True)
fig.show()
```

Monthly Sales Analysis

Overall sales performance by customer category

```
Sales_performance_by_customer_type = merged_df.groupby('LIFESTAGE')['Total_revenue'].sum().reset_index()
Sales_performance_by_customer_type
```

	LIFESTAGE	Total_revenue
0	MIDAGE SINGLES/COUPLES	363231.50
1	NEW FAMILIES	97681.85
2	OLDER FAMILIES	962667.50
3	OLDER SINGLES/COUPLES	793612.15
4	RETIREES	718219.90
5	YOUNG FAMILIES	628632.40
6	YOUNG SINGLES/COUPLES	501415.40

```python
fig = px.bar(Sales_performance_by_customer_type,
            x = 'LIFESTAGE',
            y = 'Total_revenue',
title = 'Sales performance by Customer Type')
fig.show()
```

Sales performance by Customer Type
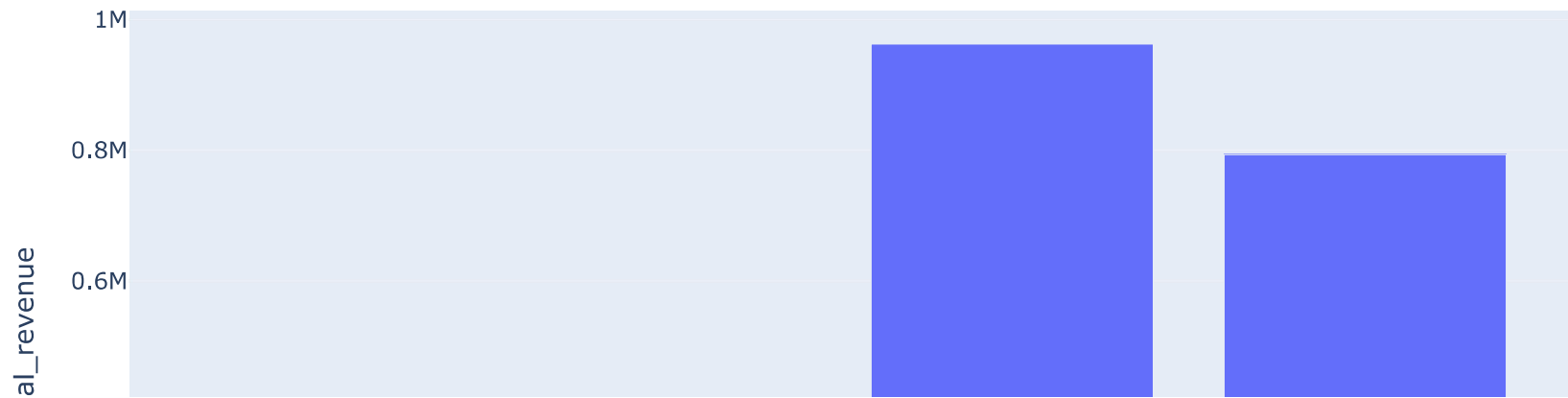
```python
Sales_by_customer_category = merged_df.groupby('PREMIUM_CUSTOMER')['Total_revenue'].sum().reset_index()
Sales_by_customer_category
```

	PREMIUM_CUSTOMER	Total_revenue
0	Budget	1333383.15
1	Mainstream	1475154.70
2	Premium	1256922.85

```python
fig = px.pie(Sales_by_customer_category,
             values = 'Total_revenue',
             names= 'PREMIUM_CUSTOMER',
             hole= 0.5,
             color_discrete_sequence=px.colors.qualitative.Pastel)
fig.update_traces(textposition='inside', textinfo='percent+label')
fig.update_layout(title_text='Sales analysis by Customer Type', title_font=dict(size=24))
fig.show()
```

Sales analysis by Customer Type
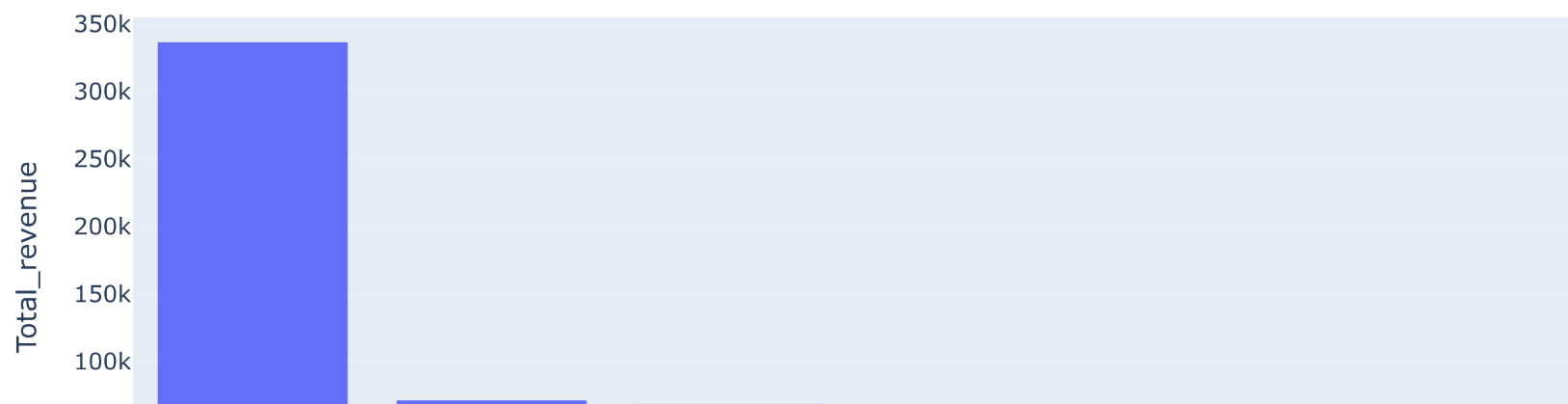

Budget
32.8%

Mainstr
36.3%
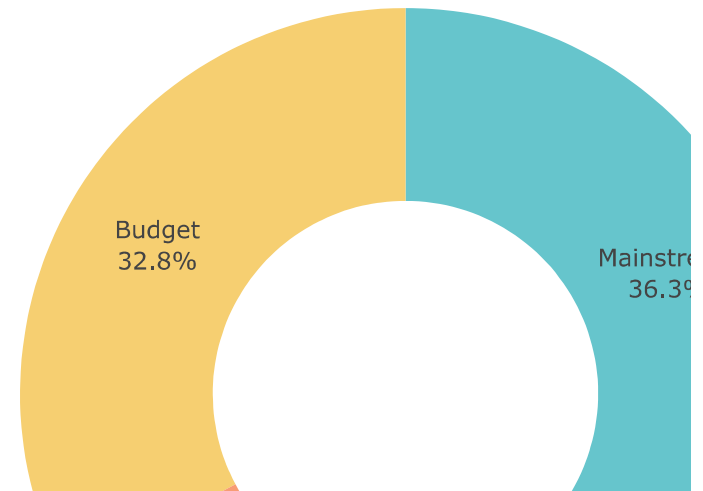
```
In [43]:  Sales_by_Product_category = merged_df.groupby('PROD_NAME')['Total_revenue'].sum().reset_index()
          Sales_by_Product_category= Sales_by_Product_category.sort_values(by='Total_revenue',ascending=False)
          Top_10_products= Sales_by_Product_category.head(10)
          print(Top_10_products)
```

```
                                PROD_NAME    Total_revenue
11          Dorito Corn Chp     Supreme 380g        336992.5
86     Smiths Crnkle Chip   Orgnl Big Bag 380g       71696.8
33      Kettle Mozzarella    Basil & Pesto 175g       68585.4
77  Smiths Crinkle Chips Salt & Vinegar 330g          68365.8
6                   Cheezels Cheese 330g              67653.3
76          Smiths Crinkle      Original 330g          67488.0
12          Doritos Cheese      Supreme 330g          65880.6
34                   Kettle Original 175g             65145.6
39  Kettle Sweet Chilli And Sour Cream 175g           64913.4
32          Kettle Honey Soy    Chicken 175g          64341.0
```

In [44]:
```python
fig = px.bar(Top_10_products, x='PROD_NAME',y='Total_revenue', title= 'Top 10 products by Revenue')
fig.show()
```

Top 10 products by Revenue