In [48]:
```
#importing the necessary libraries needed
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import os
```

In [2]:
```
# confirming the current working directory where the datasets are.
print(os.getcwd())
```

```
C:\Users\USER\Desktop\quantium data analysis project\Untitled Folder
```

In [49]:
```
#Creating a function to read the first dataset
def readat(file_path):
    df = pd.read_csv(file_path)
    return df
```

In [50]:
```
df = readat("C:\\Users\\USER\\Desktop\\quantium data analysis project\\Untitled
```

In [51]:
```
#checking for the column types in the dataset
df.dtypes
```

Out[51]:
```
LYLTY_CARD_NBR       int64
LIFESTAGE           object
PREMIUM_CUSTOMER    object
dtype: object
```

In [52]:
```
#making a copy of the data set to manipulate
purch = df.copy()
```

In [53]:
```
#checking for null values
purch.info()
purch.isna().sum()
```

```
<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
```

Out[53]:
```
LYLTY_CARD_NBR      0
LIFESTAGE           0
PREMIUM_CUSTOMER    0
dtype: int64
```

In [54]:
```
#dropping null values
purch.dropna(inplace = True)
```

In [55]:
```
#number of rows by columns we are dealing with
purch.shape
```

Out[55]:
```
(72637, 3)
```

In [56]:
```
#displaying the data
purch.head()
```

Out[56]:

	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 [57]:
```
#renaming the 'PREMIUM_CUSTOMER' to 'CUSTOMER TYPE'
purch.rename(columns={'PREMIUM_CUSTOMER': 'CUSTOMER TYPE'}, inplace=True)
```

In [58]:
```
#find the unique types of customers
purch['CUSTOMER TYPE'].unique()
```

Out[58]:
```
array(['Premium', 'Mainstream', 'Budget'], dtype=object)
```

In [59]:
```
#finding the number of premium customers
premdf = purch[purch['CUSTOMER TYPE'] == 'Premium']
premdf['CUSTOMER TYPE'].count()
```

Out[59]:
```
18922
```

In [60]:
```
#finding the number of Budget customers
bugdf = purch[purch['CUSTOMER TYPE'] == 'Budget']
bugdf['CUSTOMER TYPE'].count()
```

Out[60]:
```
24470
```

In [61]:
```
#finding the number of Mainstream customers
maindf = purch[purch['CUSTOMER TYPE'] == 'Mainstream']
maindf['CUSTOMER TYPE'].count()
```

Out[61]:
```
29245
```

In [62]:
```
#we can rseparate the values in 'LIFESTAGE' removing the '/'
purch[['LIFESTAGE', 'SPLIT']] = purch['LIFESTAGE'].str.split('/',expand = True
```

In [63]:
```
#we therefore remove the Column 'SPLIT' for we dont need it
purch.drop(columns = 'SPLIT', inplace = True)
```

In [64]:
```
#newly renamed dataset
purch.head(19)
```

Out[64]:

	LYLTY_CARD_NBR	LIFESTAGE	CUSTOMER TYPE
0	1000	YOUNG SINGLES	Premium
1	1002	YOUNG SINGLES	Mainstream
2	1003	YOUNG FAMILIES	Budget
3	1004	OLDER SINGLES	Mainstream
4	1005	MIDAGE SINGLES	Mainstream
5	1007	YOUNG SINGLES	Budget
6	1009	NEW FAMILIES	Premium
7	1010	YOUNG SINGLES	Mainstream
8	1011	OLDER SINGLES	Mainstream
9	1012	OLDER FAMILIES	Mainstream
10	1013	RETIREES	Budget
11	1016	OLDER FAMILIES	Mainstream
12	1018	YOUNG SINGLES	Mainstream
13	1019	OLDER SINGLES	Premium
14	1020	YOUNG SINGLES	Mainstream
15	1022	OLDER FAMILIES	Budget
16	1023	MIDAGE SINGLES	Premium
17	1024	YOUNG SINGLES	Premium
18	1025	YOUNG FAMILIES	Budget

In [4]:
```
#Creating a funvtion to read the second dataset
def readat1(file_path):
    df2 = pd.read_csv(file_path)


    return df2
```

In [5]:
```
#loading the dataframe
df2 = readat1("C:\\Users\\USER\\Desktop\\quantium data analysis project\\Untitl
```

In [6]:
```
#making a copy of the dataset
transc = df2.copy()
```

In [10]:
```
#display the data
transc.head(60)
```

Out[10]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_Q
0	43390	1	1000	1	5	Natural Chip Compny SeaSalt175g	
1	43599	1	1307	348	66	CCs Nacho Cheese 175g	
2	43605	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	
3	43329	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	
4	43330	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	
5	43604	4	4074	2982	57	Old El Paso Salsa Dip Tomato Mild	

In [12]:
```
#checking for null values
transc.info()
transc.isnull().sum()
```

```
<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
```

Out[12]:

```
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 [13]:
```
# Convert 'DATE' column to 'yyyy-mm-dd' format
transc['DATE'] = pd.to_datetime(transc['DATE'], origin='1899-12-30', unit='D')
```

In [15]:
```
#display to see date change
transc.head(20)
```

Out[15]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	T
0	2018-10-17	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	
1	2019-05-14	1	1307	348	66	CCs Nacho Cheese 175g	3	
2	2019-05-20	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	
3	2018-08-17	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	
4	2018-08-18	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	
5	2019-05-19	4	4074	2982	57	Old El Paso Salsa Dip Tomato Mild 300g	1	
6	2019-05-16	4	4149	3333	16	Smiths Crinkle Chips Salt & Vinegar 330g	1	
7	2019-05-16	4	4196	3539	24	Grain Waves Sweet Chilli 210g	1	
8	2018-08-20	5	5026	4525	42	Doritos Corn Chip Mexican Jalapeno 150g	1	
9	2018-08-18	7	7150	6900	52	Grain Waves Sour Cream&Chives 210G	2	
10	2019-05-17	7	7215	7176	16	Smiths Crinkle Chips Salt & Vinegar 330g	1	
11	2018-08-20	8	8294	8221	114	Kettle Sensations Siracha Lime 150g	5	
12	2019-05-18	9	9208	8634	15	Twisties Cheese 270g	2	
13	2018-08-17	13	13213	12447	92	WW Crinkle Cut Chicken 175g	1	
14	2019-05-15	19	19272	16686	44	Thins Chips Light& Tangy 175g	1	
15	2019-05-19	20	20164	17136	54	CCs Original 175g	1	
16	2018-08-18	20	20418	17413	94	Burger Rings 220g	4	

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	T
17	2018-08-14	22	22411	18646	98	NCC Sour Cream & Garden Chives 175g	1	
18	2018-08-17	22	22456	18696	93	Doritos Corn Chip Southern Chicken 150g	1	
19	2019-05-16	23	23067	19162	56	Cheezels Cheese Box 125g	1	

In [18]:
```
# change the 'PROD_NAME' to  the same case
transc['PROD_NAME'] = transc['PROD_NAME'].str.lower()
```

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

Out[20]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TO
0	2018-10-17	1	1000	1	5	natural chip compny seasalt175g	2	
1	2019-05-14	1	1307	348	66	ccs nacho cheese 175g	3	
2	2019-05-20	1	1343	383	61	smiths crinkle cut chips chicken 170g	2	
3	2018-08-17	2	2373	974	69	smiths chip thinly s/cream&onion 175g	5	
4	2018-08-18	2	2426	1038	108	kettle tortilla chpshny&jlpno chili 150g	3	
5	2019-05-19	4	4074	2982	57	old el paso salsa dip tomato mild 300g	1	
6	2019-05-16	4	4149	3333	16	smiths crinkle chips salt & vinegar 330g	1	
7	2019-05-16	4	4196	3539	24	grain waves sweet chilli 210g	1	
8	2018-08-20	5	5026	4525	42	doritos corn chip mexican jalapeno 150g	1	
9	2018-08-18	7	7150	6900	52	grain waves sour cream&chives 210g	2	
10	2019-05-17	7	7215	7176	16	smiths crinkle chips salt & vinegar 330g	1	
11	2018-08-20	8	8294	8221	114	kettle sensations siracha lime 150g	5	
12	2019-05-18	9	9208	8634	15	twisties cheese 270g	2	
13	2018-08-17	13	13213	12447	92	ww crinkle cut chicken 175g	1	
14	2019-05-15	19	19272	16686	44	thins chips light& tangy 175g	1	
15	2019-05-19	20	20164	17136	54	ccs original 175g	1	
16	2018-08-18	20	20418	17413	94	burger rings 220g	4	

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TO
17	2018-08-14	22	22411	18646	98	ncc sour cream & garden chives 175g	1	
18	2018-08-17	22	22456	18696	93	doritos corn chip southern chicken 150g	1	
19	2019-05-16	23	23067	19162	56	cheezels cheese box 125g	1	

In [26]:
```
#separating the string and the digits to be in the same formating as the rest c
import re

transc['PROD_NAME'] = transc['PROD_NAME'].apply(lambda x: re.sub(r'(?<=[a-zA-Z)
```

In [30]:
```
#display to see the changes
transc.head()
```

Out[30]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT
0	2018-10-17	1	1000	1	5	natural chip compny seasalt 175g	2	
1	2019-05-14	1	1307	348	66	ccs nacho cheese 175g	3	
2	2019-05-20	1	1343	383	61	smiths crinkle cut chips chicken 170g	2	
3	2018-08-17	2	2373	974	69	smiths chip thinly s/cream&onion 175g	5	
4	2018-08-18	2	2426	1038	108	kettle tortilla chpshny&jlpno chili 150g	3	

In [33]:
```
#splitting the column 'PROD_NAME'
# Extract the quantity information using regular expressions
transc['QTY-GRAMS'] = transc['PROD_NAME'].str.extract(r'(\d+g)')
transc.head()
```

Out[33]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT
0	2018-10-17	1	1000	1	5	natural chip compny seasalt 175g	2	
1	2019-05-14	1	1307	348	66	ccs nacho cheese 175g	3	
2	2019-05-20	1	1343	383	61	smiths crinkle cut chips chicken 170g	2	
3	2018-08-17	2	2373	974	69	smiths chip thinly s/cream&onion 175g	5	
4	2018-08-18	2	2426	1038	108	kettle tortilla chpshny&jlpno chili 150g	3	

In [44]:
```
#removing the digit + 'g' in the 'PROD_NAME'
transc['PROD_NAME'] = transc['PROD_NAME'].str[:-4].str.strip()
# Print the modified dataframe
transc.head()
```

Out[44]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT
0	2018-10-17	1	1000	1	5	natural chip compny seasalt	2	
1	2019-05-14	1	1307	348	66	ccs nacho cheese	3	
2	2019-05-20	1	1343	383	61	smiths crinkle cut chips chicken	2	
3	2018-08-17	2	2373	974	69	smiths chip thinly s/cream&onion	5	
4	2018-08-18	2	2426	1038	108	kettle tortilla chpshny&jlpno chili	3	

In [65]:
```
#joining the first dataset with the second data set using the primary key
merged_data = pd.merge(left=purch, right=transc, on='LYLTY_CARD_NBR', how='righ
```

In [68]:
```
#check the total number of rows and columns after merging in comparison to the
print('merged_data = ', merged_data.shape)
print('purch =', purch.shape)
print('transc =', transc.shape)
```

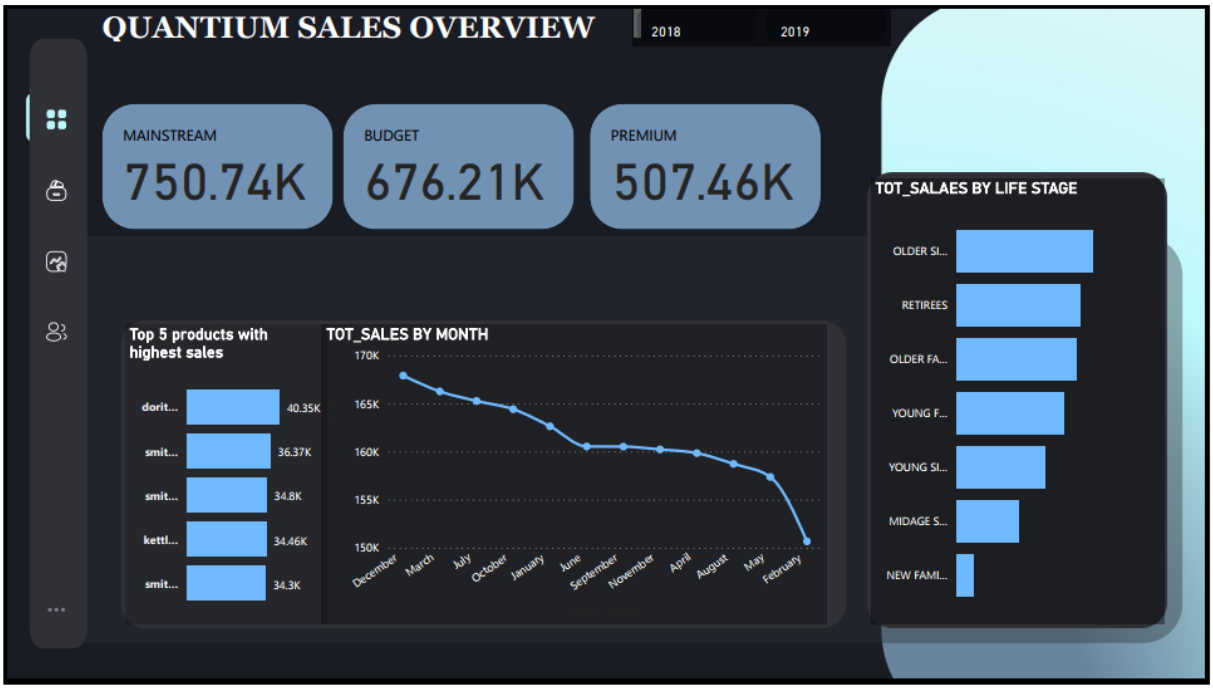
```
merged_data =  (264836, 11)
purch = (72637, 3)
transc = (264836, 9)
```

In [70]:
```
#save the merged data set to csv
merged_data.to_csv('merged_dataset.csv', index=False)
```

In [ ]:

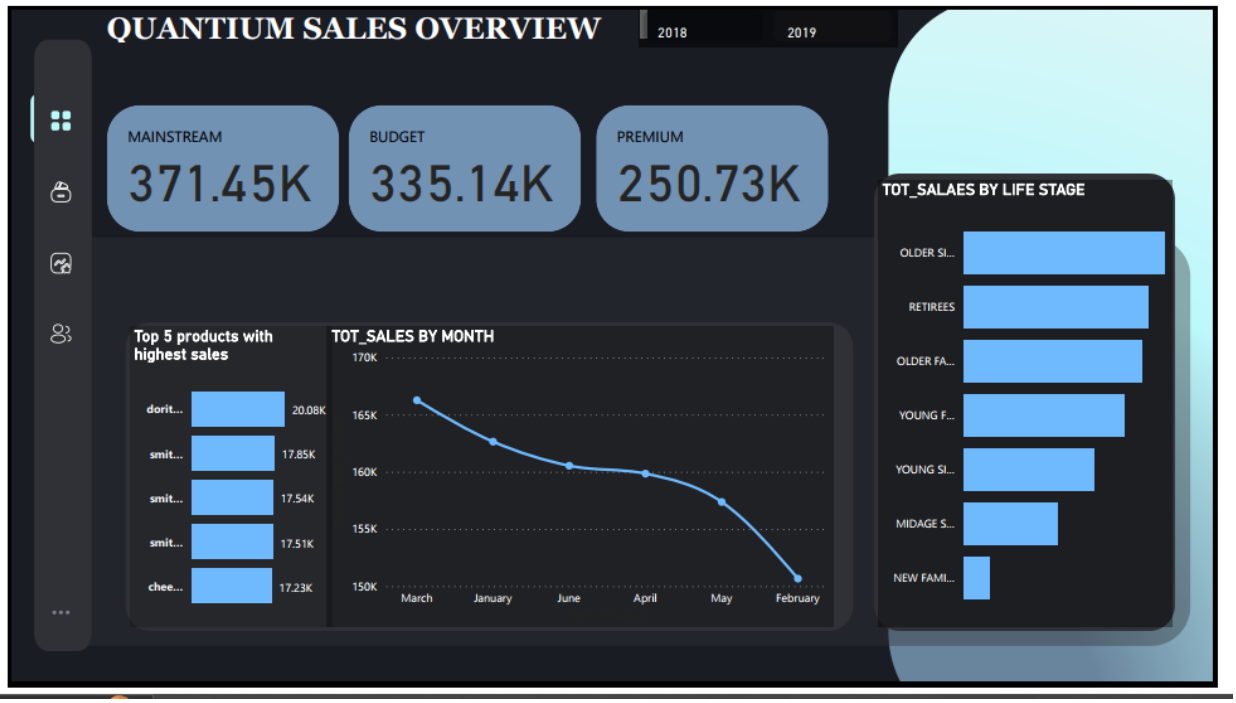
BRADLEY DAUDI - PROJECT SUMMARY

1. Overall summary for the 2 years



2. 2018 Analysis

3. 2019 Analysis



FINDINGS

1. Dorito corn chp supreme is the most sold product for all the 2 years

2. December has been the month of the years where most sales are made.

3. Customer type Mainstream are the main driving force of the sales for they propel most sales according to the analysis.

4. Older singles in the lifestage are the highest drivers of sales as well.

RECOMMENDATIONS

1. **Promote Top 5 Products:**
    - Focus on marketing and promotional activities centered around the top 5 product names. Highlight their features, benefits, and any special promotions or discounts associated with them.
    - Consider bundling these products together or creating exclusive deals to encourage customers to purchase them as a package.
2. **Target Older Singles Demographic:**
    - Tailor marketing campaigns specifically to older singles. Understand their preferences, needs, and purchasing behavior to create targeted and effective advertising messages.
    - Leverage social media platforms, email campaigns, and other channels frequented by older singles to reach them effectively.
3. **Customized Offers for Older Singles:**
    - Create personalized offers or loyalty programs for the older singles demographic. This could include discounts, exclusive access to new products, or loyalty points that can be redeemed for future purchases.
4. **Mainstream Customer Engagement:**
    - Strengthen engagement with mainstream customers by conducting surveys or collecting feedback to understand their preferences.
    - Implement customer retention strategies, such as loyalty programs, to encourage repeat business from mainstream customers.
5. **Seasonal Promotions in December:**
    - Leverage the rising trend towards December by planning special promotions, holiday-themed marketing campaigns, or limited-time offers.

- Consider collaborating with influencers or running social media contests to generate buzz and attract more customers during the festive season.

6. **Data-Driven Decision-Making:**
    - Continue to monitor and analyze sales data regularly to identify any evolving trends or shifts in customer behavior.
    - Use data analytics tools to gain deeper insights into customer preferences and adjust strategies accordingly.
7. **Cross-Sell and Up-Sell Opportunities:**
    - Identify opportunities for cross-selling and up-selling, especially among the top 5 products. Recommend complementary items to customers during the purchasing process to increase the average transaction value.
8. **Customer Segmentation:**
    - Further segment the customer base to identify specific needs and preferences within the mainstream customer type. This allows for more targeted marketing strategies.
9. **Employee Training:**
    - Ensure that sales and customer service teams are well-informed about the top-selling products and are trained to effectively engage with older singles and mainstream customers.
10. **Continuous Improvement:**
    - Regularly review the effectiveness of implemented strategies and make adjustments as needed. Stay agile in responding to market dynamics and customer feedback.