QUANTIUM FORAGE PROJECT: TASK 1

The objective of task 1 is to analyze the potato chips sales for a Quantium client. They want to gain insights on sales data they have for the past year. The sales period is from July 2018 to June 2019

In [210…
```
# Importing necessary libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

LOAD DATASET 1

In [211…
```
chips_demog = pd.read_csv('QVI_purchase_behaviour.csv')
chips_demog.head()
```

Out[211…

	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 [212…
```
chips_demog.shape
```

Out[212… (72637, 3)

In [213…
```
chips_demog.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 [214…
```
chips_demog.isna().sum()
```

Out[214…
```
LYLTY_CARD_NBR      0
LIFESTAGE           0
PREMIUM_CUSTOMER    0
dtype: int64
```

In [215…
```
chips_demog.duplicated().sum()
```

Out[215… np.int64(0)

LOAD DATASET 2

In [216…
```
chips_transac = pd.read_excel('QVI_transaction_data.xlsx')
chips_transac.head()
```

Out[216…

	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

In [218…
```
chips_transac.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 [219…
```
# Excel stores dates as the number of days since January 1, 1900. Convert it to date and change data type

chips_transac['DATE'] = pd.to_datetime(chips_transac['DATE'], origin = '1899-12-30', unit = 'D')
chips_transac.head(5)
```

Out[219…

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

In [220…
```
chips_transac.dtypes
```

Out[220…
```
DATE              datetime64[ns]
STORE_NBR                  int64
LYLTY_CARD_NBR             int64
TXN_ID                     int64
PROD_NBR                   int64
PROD_NAME                 object
PROD_QTY                   int64
TOT_SALES                float64
dtype: object
```

In [221…
```
chips_transac.isna().sum()
```

Out[221…
```
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 [222…
```
chips_transac[['TOT_SALES']].describe()
```

Out[222…

	TOT_SALES
count	264836.000000
mean	7.304200
std	3.083226
min	1.500000
25%	5.400000
50%	7.400000
75%	9.200000
max	650.000000

In [223…
```
chips_transac['PROD_NAME'].unique()
```

Out[223…
```
array(['Natural Chip        Compny SeaSalt175g',
       'CCs Nacho Cheese    175g',
       'Smiths Crinkle Cut  Chips Chicken 170g',
       'Smiths Chip Thinly  S/Cream&Onion 175g',
       'Kettle Tortilla ChpsHny&Jlpno Chili 150g',
       'Old El Paso Salsa   Dip Tomato Mild 300g',
       'Smiths Crinkle Chips Salt & Vinegar 330g',
       'Grain Waves         Sweet Chilli 210g',
       'Doritos Corn Chip Mexican Jalapeno 150g',
       'Grain Waves Sour    Cream&Chives 210G',
       'Kettle Sensations   Siracha Lime 150g',
       'Twisties Cheese     270g', 'WW Crinkle Cut      Chicken 175g',
       'Thins Chips Light&  Tangy 175g', 'CCs Original 175g',
       'Burger Rings 220g', 'NCC Sour Cream &    Garden Chives 175g',
       'Doritos Corn Chip Southern Chicken 150g',
       'Cheezels Cheese Box 125g', 'Smiths Crinkle      Original 330g',
       'Infzns Crn Crnchers Tangy Gcamole 110g',
       'Kettle Sea Salt     And Vinegar 175g',
       'Smiths Chip Thinly  Cut Original 175g', 'Kettle Original 175g',
       'Red Rock Deli Thai  Chilli&Lime 150g',
       'Pringles Sthrn FriedChicken 134g', 'Pringles Sweet&Spcy BBQ 134g',
       'Red Rock Deli SR    Salsa & Mzzrlla 150g',
       'Thins Chips         Originl saltd 175g',
       'Red Rock Deli Sp    Salt & Truffle 150G',
       'Smiths Thinly       Swt Chli&S/Cream175G', 'Kettle Chilli 175g',
       'Doritos Mexicana    170g',
       'Smiths Crinkle Cut  French OnionDip 150g',
       'Natural ChipCo      Hony Soy Chckn175g',
       'Dorito Corn Chp     Supreme 380g', 'Twisties Chicken270g',
       'Smiths Thinly Cut   Roast Chicken 175g',
       'Smiths Crinkle Cut  Tomato Salsa 150g',
       'Kettle Mozzarella   Basil & Pesto 175g',
       'Infuzions Thai SweetChili PotatoMix 110g',
       'Kettle Sensations   Camembert & Fig 150g',
       'Smith Crinkle Cut   Mac N Cheese 150g',
       'Kettle Honey Soy    Chicken 175g',
       'Thins Chips Seasonedchicken 175g',
       'Smiths Crinkle Cut  Salt & Vinegar 170g',
       'Infuzions BBQ Rib   Prawn Crackers 110g',
       'GrnWves Plus Btroot & Chilli Jam 180g',
       'Tyrrells Crisps     Lightly Salted 165g',
       'Kettle Sweet Chilli And Sour Cream 175g',
       'Doritos Salsa       Medium 300g', 'Kettle 135g Swt Pot Sea Salt',
       'Pringles SourCream  Onion 134g',
       'Doritos Corn Chips  Original 170g',
       'Twisties Cheese     Burger 250g',
       'Old El Paso Salsa   Dip Chnky Tom Ht300g',
       'Cobs Popd Swt/Chlli &Sr/Cream Chips 110g',
       'Woolworths Mild     Salsa 300g',
       'Natural Chip Co     Tmato Hrb&Spce 175g',
       'Smiths Crinkle Cut  Chips Original 170g',
       'Cobs Popd Sea Salt  Chips 110g',
       'Smiths Crinkle Cut  Chips Chs&Onion170g',
       'French Fries Potato Chips 175g',
       'Old El Paso Salsa   Dip Tomato Med 300g',
       'Doritos Corn Chips  Cheese Supreme 170g',
       'Pringles Original   Crisps 134g',
       'RRD Chilli&         Coconut 150g',
       'WW Original Corn    Chips 200g',
       'Thins Potato Chips  Hot & Spicy 175g',
```

```
'Cobs Popd Sour Crm  &Chives Chips 110g',
'Smiths Crnkle Chip  Orgnl Big Bag 380g',
'Doritos Corn Chips  Nacho Cheese 170g',
'Kettle Sensations   BBQ&Maple 150g',
'WW D/Style Chip     Sea Salt 200g',
'Pringles Chicken    Salt Crips 134g',
'WW Original Stacked Chips 160g',
'Smiths Chip Thinly  CutSalt/Vinegr175g', 'Cheezels Cheese 330g',
'Tostitos Lightly    Salted 175g',
'Thins Chips Salt &  Vinegar 175g',
'Smiths Crinkle Cut  Chips Barbecue 170g', 'Cheetos Puffs 165g',
'RRD Sweet Chilli &  Sour Cream 165g',
'WW Crinkle Cut      Original 175g',
'Tostitos Splash Of  Lime 175g', 'Woolworths Medium   Salsa 300g',
'Kettle Tortilla ChpsBtroot&Ricotta 150g',
'CCs Tasty Cheese    175g', 'Woolworths Cheese   Rings 190g',
'Tostitos Smoked     Chipotle 175g', 'Pringles Barbeque   134g',
'WW Supreme Cheese   Corn Chips 200g',
'Pringles Mystery    Flavour 134g',
'Tyrrells Crisps     Ched & Chives 165g',
'Snbts Whlgrn Crisps Cheddr&Mstrd 90g',
'Cheetos Chs & Bacon Balls 190g', 'Pringles Slt Vingar 134g',
'Infuzions SourCream&Herbs Veg Strws 110g',
'Kettle Tortilla ChpsFeta&Garlic 150g',
'Infuzions Mango     Chutny Papadums 70g',
'RRD Steak &         Chimuchurri 150g',
'RRD Honey Soy       Chicken 165g',
'Sunbites Whlegrn    Crisps Frch/Onin 90g',
'RRD Salt & Vinegar  165g', 'Doritos Cheese      Supreme 330g',
'Smiths Crinkle Cut  Snag&Sauce 150g',
'WW Sour Cream &OnionStacked Chips 160g',
'RRD Lime & Pepper   165g',
'Natural ChipCo Sea  Salt & Vinegr 175g',
'Red Rock Deli Chikn&Garlic Aioli 150g',
'RRD SR Slow Rst     Pork Belly 150g', 'RRD Pc Sea Salt     165g',
'Smith Crinkle Cut   Bolognese 150g', 'Doritos Salsa Mild  300g'],
dtype=object)
```

In [224…

```
# Remove Unwanted Spaces

chips_transac['PROD_NAME'] = chips_transac['PROD_NAME'].str.replace(r'\s+', ' ', regex=True).str.strip()
chips_transac['PROD_NAME']
```

Out[224…

```
0                 Natural Chip Compny SeaSalt175g
1                           CCs Nacho Cheese 175g
2            Smiths Crinkle Cut Chips Chicken 170g
3            Smiths Chip Thinly S/Cream&Onion 175g
4         Kettle Tortilla ChpsHny&Jlpno Chili 150g
                          ...
264831     Kettle Sweet Chilli And Sour Cream 175g
264832                Tostitos Splash Of Lime 175g
264833                      Doritos Mexicana 170g
264834     Doritos Corn Chip Mexican Jalapeno 150g
264835                Tostitos Splash Of Lime 175g
Name: PROD_NAME, Length: 264836, dtype: object
```

In [225…

```
# Splitting into chip's weight

chips_transac['Weight'] = chips_transac['PROD_NAME'].str[-4:]
chips_transac['Weight'].head(4)
```

Out[225…
```
0    175g
1    175g
2    170g
3    175g
Name: Weight, dtype: object
```

In [226…
```
chips_transac['Weight'].unique()
```

Out[226…
```
array(['175g', '170g', '150g', '300g', '330g', '210g', '210G', '270g',
       '220g', '125g', '110g', '134g', '150G', '175G', '380g', '180g',
       '165g', 'Salt', '250g', '200g', '160g', '190g', ' 90g', ' 70g'],
      dtype=object)
```

In [227…
```
chips_transac['Weight'].value_counts()
```

Out[227…
```
Weight
175g    64929
150g    41633
134g    25102
110g    22387
170g    19983
165g    15297
300g    15166
330g    12540
380g     6418
270g     6285
200g     4473
Salt     3257
250g     3169
210g     3167
210G     3105
 90g     3008
190g     2995
160g     2970
220g     1564
 70g     1507
150G     1498
180g     1468
175G     1461
125g     1454
Name: count, dtype: int64
```

In [228…
```
# Clean Weight data

chips_transac['Weight'] = chips_transac['Weight'].replace({"Salt":"135g", "210G": "210g", "150G":"150g", "175G":"175g"})
```

In [229…
```
chips_transac['Weight'].unique()
```

Out[229…
```
array(['175g', '170g', '150g', '300g', '330g', '210g', '270g', '220g',
       '125g', '110g', '134g', '380g', '180g', '165g', '135g', '250g',
       '200g', '160g', '190g', ' 90g', ' 70g'], dtype=object)
```

In [230…
```
chips_transac['Weight in g'] = chips_transac['Weight'].str[-4:-1]
```

In [231…
```
chips_transac['Weight in g'] = chips_transac['Weight in g'].astype(int)
```

In [232…
```
chips_transac['Weight in g'].unique()
```

Out[232…
```
array([175, 170, 150, 300, 330, 210, 270, 220, 125, 110, 134, 380, 180,
       165, 135, 250, 200, 160, 190,  90,  70])
```

In [233…
```
chips_transac.head()
```

Out[233…

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

In [234…
```
chips_transac.dtypes
```

Out[234…
```
DATE              datetime64[ns]
STORE_NBR                  int64
LYLTY_CARD_NBR             int64
TXN_ID                     int64
PROD_NBR                   int64
PROD_NAME                 object
PROD_QTY                   int64
TOT_SALES                float64
Weight                    object
Weight in g                int64
dtype: object
```

In [235…
```
chips_transac['Weight in g'].value_counts()
```

Out[235…
```
Weight in g
175    66390
150    43131
134    25102
110    22387
170    19983
165    15297
300    15166
330    12540
380     6418
270     6285
210     6272
200     4473
135     3257
250     3169
90      3008
190     2995
160     2970
220     1564
70      1507
180     1468
125     1454
Name: count, dtype: int64
```

In [236…
```
chips_transac['PROD_NAME'].reset_index()
```

Out[236...

	index	PROD_NAME
0	0	Natural Chip Compny SeaSalt175g
1	1	CCs Nacho Cheese 175g
2	2	Smiths Crinkle Cut Chips Chicken 170g
3	3	Smiths Chip Thinly S/Cream&Onion 175g
4	4	Kettle Tortilla ChpsHny&Jlpno Chili 150g
...	...	...
264831	264831	Kettle Sweet Chilli And Sour Cream 175g
264832	264832	Tostitos Splash Of Lime 175g
264833	264833	Doritos Mexicana 170g
264834	264834	Doritos Corn Chip Mexican Jalapeno 150g
264835	264835	Tostitos Splash Of Lime 175g

264836 rows × 2 columns

In [237...

```
# Only CHIPS data

chips_filtered = [item for item in chips_transac['PROD_NAME'] if 'salsa' in item.lower() or 'dip' in item.lower()]
```

In [238...

```
np.unique(chips_filtered)
```

Out[238...

```
array(['Doritos Salsa Medium 300g', 'Doritos Salsa Mild 300g',
       'Old El Paso Salsa Dip Chnky Tom Ht300g',
       'Old El Paso Salsa Dip Tomato Med 300g',
       'Old El Paso Salsa Dip Tomato Mild 300g',
       'Red Rock Deli SR Salsa & Mzzrlla 150g',
       'Smiths Crinkle Cut French OnionDip 150g',
       'Smiths Crinkle Cut Tomato Salsa 150g',
       'Woolworths Medium Salsa 300g', 'Woolworths Mild Salsa 300g'],
      dtype='<U39')
```

In [239...

```
chips_transac_filter = chips_transac[chips_transac['PROD_NAME'].str.contains('salsa|dip', case=False, na=False)]
chips_transac_filter
```

Out[239...

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	Weight	Weight in g
5	2019-05-19	4	4074	2982	57	Old El Paso Salsa Dip Tomato Mild 300g	1	5.1	300g	300
25	2019-05-15	39	39144	35506	57	Old El Paso Salsa Dip Tomato Mild 300g	1	5.1	300g	300
32	2019-05-20	45	45127	41122	64	Red Rock Deli SR Salsa & Mzzrlla 150g	2	5.4	150g	150
39	2018-08-18	55	55072	48878	107	Smiths Crinkle Cut French OnionDip 150g	1	2.6	150g	150
44	2018-08-18	56	56013	50090	39	Smiths Crinkle Cut Tomato Salsa 150g	1	2.6	150g	150
...	...	...	...	...	...	...	...	...	...	...
264719	2018-10-28	266	266278	264104	39	Smiths Crinkle Cut Tomato Salsa 150g	1	2.6	150g	150
264734	2019-01-11	267	267324	264374	41	Doritos Salsa Mild 300g	1	2.6	300g	300
264746	2018-10-18	268	268200	264616	107	Smiths Crinkle Cut French OnionDip 150g	1	2.6	150g	150
264759	2019-02-03	269	269125	265775	107	Smiths Crinkle Cut French OnionDip 150g	2	5.2	150g	150
264780	2019-01-10	269	269222	266382	64	Red Rock Deli SR Salsa & Mzzrlla 150g	2	5.4	150g	150

19532 rows × 10 columns

In [240...
```
chips_transac.shape
```

Out[240... (264836, 10)

In [241...
```
# Drop all Non Chips Column

chips_transac = chips_transac.drop(chips_transac_filter.index)
chips_transac
```

Out[241...

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	Weight	Weight in g
0	2018-10-17	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0	175g	175
1	2019-05-14	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3	175g	175
2	2019-05-20	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9	170g	170
3	2018-08-17	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	15.0	175g	175
4	2018-08-18	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	13.8	150g	150
...	...	...	...	...	...	...	...	...	...	...
264831	2019-03-09	272	272319	270088	89	Kettle Sweet Chilli And Sour Cream 175g	2	10.8	175g	175
264832	2018-08-13	272	272358	270154	74	Tostitos Splash Of Lime 175g	1	4.4	175g	175
264833	2018-11-06	272	272379	270187	51	Doritos Mexicana 170g	2	8.8	170g	170
264834	2018-12-27	272	272379	270188	42	Doritos Corn Chip Mexican Jalapeno 150g	2	7.8	150g	150
264835	2018-09-22	272	272380	270189	74	Tostitos Splash Of Lime 175g	2	8.8	175g	175

245304 rows × 10 columns

In [242…
```
# Check if all non chip items are removed

chips_transac[chips_transac['PROD_NAME'].str.contains('salsa|dip', case=False)]
```

Out[242…

DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	Weight	Weight in g

In [243…
```
# Create a Brand Column

chips_transac['PROD_BRAND'] = chips_transac['PROD_NAME'].str.split().str.get(0)
chips_transac.head()
```

Out[243…

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

In [244…
```
chips_transac['PROD_BRAND'].unique()
```

Out[244…
```
array(['Natural', 'CCs', 'Smiths', 'Kettle', 'Grain', 'Doritos',
       'Twisties', 'WW', 'Thins', 'Burger', 'NCC', 'Cheezels', 'Infzns',
       'Red', 'Pringles', 'Dorito', 'Infuzions', 'Smith', 'GrnWves',
       'Tyrrells', 'Cobs', 'French', 'RRD', 'Tostitos', 'Cheetos',
       'Woolworths', 'Snbts', 'Sunbites'], dtype=object)
```

In [245…
```
chips_transac['PROD_BRAND'].value_counts()
```

Out[245…

```
PROD_BRAND
Kettle        41288
Smiths        25952
Pringles      25102
Doritos       22041
Thins         14075
RRD           11894
Infuzions     11057
WW            10320
Cobs           9693
Tostitos       9471
Twisties       9454
Tyrrells       6442
Grain          6272
Natural        6050
Cheezels       4603
CCs            4551
Red            4427
Dorito         3185
Infzns         3144
Smith          2963
Cheetos        2927
Snbts          1576
Burger         1564
Woolworths     1516
GrnWves        1468
Sunbites       1432
NCC            1419
French         1418
Name: count, dtype: int64
```

In [246…

```
# Correcting the names

chips_transac['PROD_BRAND'] = chips_transac['PROD_BRAND'].replace({"RRD":"Red", "Smith":"Smiths", "Dorito":"Doritos", "Infzns":"Infuzions"})
```

In [247…

```
chips_transac['PROD_BRAND'].value_counts()
```

Out[247…

```
PROD_BRAND
Kettle        41288
Smiths        28915
Doritos       25226
Pringles      25102
Red           16321
Infuzions     14201
Thins         14075
WW            10320
Cobs           9693
Tostitos       9471
Twisties       9454
Tyrrells       6442
Grain          6272
Natural        6050
Cheezels       4603
CCs            4551
Cheetos        2927
Snbts          1576
Burger         1564
Woolworths     1516
GrnWves        1468
Sunbites       1432
NCC            1419
French         1418
Name: count, dtype: int64
```

In [248…

```
chips_transac.head()
```

Out[248…

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

In [249…

```
highest_purchases = chips_transac.groupby('LYLTY_CARD_NBR')['PROD_QTY'].sum().reset_index()
highest_purchases
```

Out[249…

	LYLTY_CARD_NBR	PROD_QTY
0	1000	2
1	1002	1
2	1003	2
3	1004	1
4	1005	1
...	...	...
71181	2370651	2
71182	2370701	2
71183	2370751	2
71184	2370961	2
71185	2373711	2

71186 rows × 2 columns

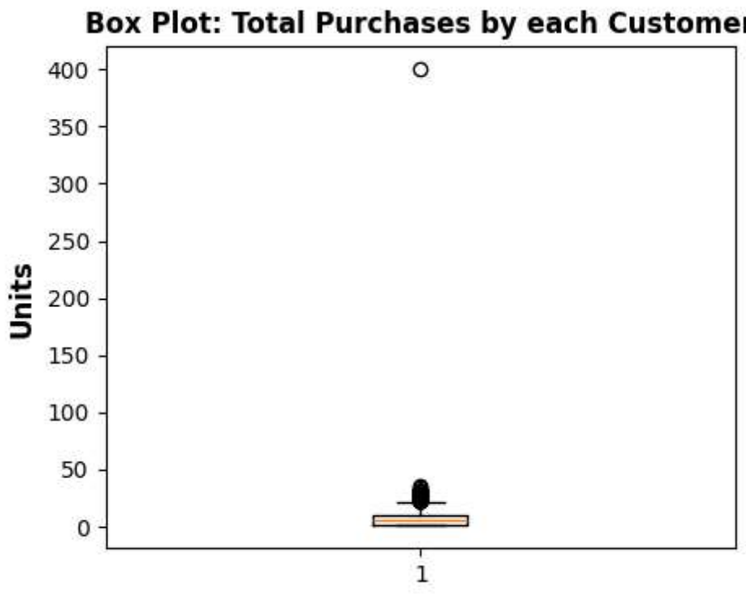
In [250…

```
# Check for outlier

plt.figure(figsize=(5,4))

plt.boxplot(highest_purchases['PROD_QTY'])
plt.ylabel("Units", fontsize=12, fontweight='bold')
plt.title("Box Plot: Total Purchases by each Customer", fontsize=12, fontweight='bold')

plt.show()
```



THIS SHOWS AN OUTLIER IS PRESENT

In [251…
```
# Drop the outlier

chips_transac = chips_transac.drop(chips_transac[chips_transac['LYLTY_CARD_NBR'] == 226000].index)
```

In [252…
```
cleaned_purchases = chips_transac.groupby('LYLTY_CARD_NBR')['PROD_QTY'].sum().reset_index()
cleaned_purchases.head()
```

Out[252…

	LYLTY_CARD_NBR	PROD_QTY
0	1000	2
1	1002	1
2	1003	2
3	1004	1
4	1005	1

In [261…
```
chips_merged = pd.merge(chips_transac, chips_demog, on='LYLTY_CARD_NBR', how='left')
chips_merged.head()
```

Out[261…

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

In [260…
```
chips_transac.shape
```

Out[260…
```
(245302, 11)
```

In [262…
```
chips_merged.shape
```

Out[262…
```
(245302, 13)
```

In [264…
```
chips_merged['LIFESTAGE'].unique()
```

Out[264…
```
array(['YOUNG SINGLES/COUPLES', 'MIDAGE SINGLES/COUPLES', 'NEW FAMILIES',
       'OLDER FAMILIES', 'OLDER SINGLES/COUPLES', 'RETIREES',
       'YOUNG FAMILIES'], dtype=object)
```

In [265…
```
chips_merged['PREMIUM_CUSTOMER'].unique()
```

Out[265…
```
array(['Premium', 'Budget', 'Mainstream'], dtype=object)
```

In [305…
```
# Save the merged file

chips_merged.to_csv('final_data.csv', index= False)
```

VISUALIZATIONS

In [369…
```
import warnings
warnings.filterwarnings("ignore")
```

a) Customer Segment wise Count

In [289…
```
customer_segment = chips_merged['PREMIUM_CUSTOMER'].value_counts().reset_index()
customer_segment
```

Out[289…

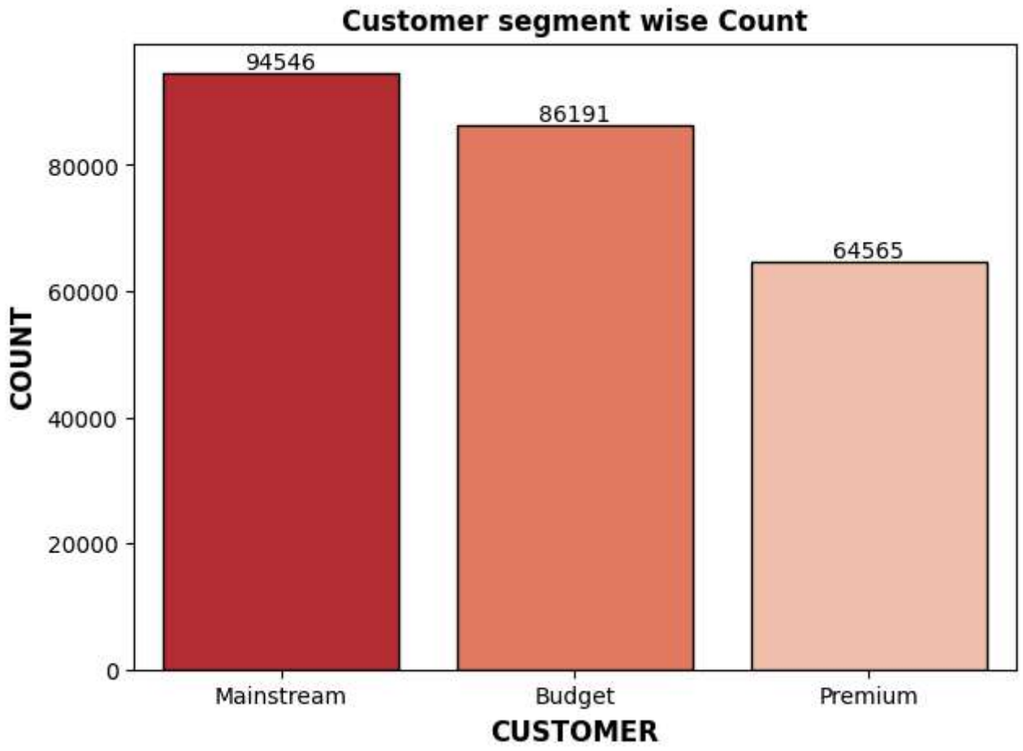
	PREMIUM_CUSTOMER	count
0	Mainstream	94546
1	Budget	86191
2	Premium	64565

In [418…
```
plt.figure(figsize=(7,5))

ax = sns.barplot(x = 'PREMIUM_CUSTOMER', y = 'count', data=customer_segment, palette='Reds_r', edgecolor='black')
plt.xlabel("CUSTOMER", fontsize=12, fontweight='bold')
plt.ylabel("COUNT", fontsize=12, fontweight='bold')
plt.title("Customer segment wise Count", fontsize=12, fontweight='bold')

for bars in ax.containers:
    ax.bar_label(bars)

plt.show()
```



In [336…
```
chips_merged.duplicated(subset="LYLTY_CARD_NBR").sum()
```

Out[336…
```
np.int64(174117)
```

In [343…
```
unique_members = chips_merged.drop_duplicates(subset="LYLTY_CARD_NBR")
```

In [345…
```
unique_members.shape
```

Out[345…
```
(71185, 13)
```

b) Unique Customer Count

In [540…
```
plot_um1 = unique_members['PREMIUM_CUSTOMER'].value_counts().reset_index()
plot_um1['Percentage'] = (plot_um1['count'] / plot_um1['count'].sum() * 100).round(3)
plot_um1
```

Out[540…

	PREMIUM_CUSTOMER	count	Percentage
0	Mainstream	28696	40.312
1	Budget	23960	33.659
2	Premium	18529	26.029

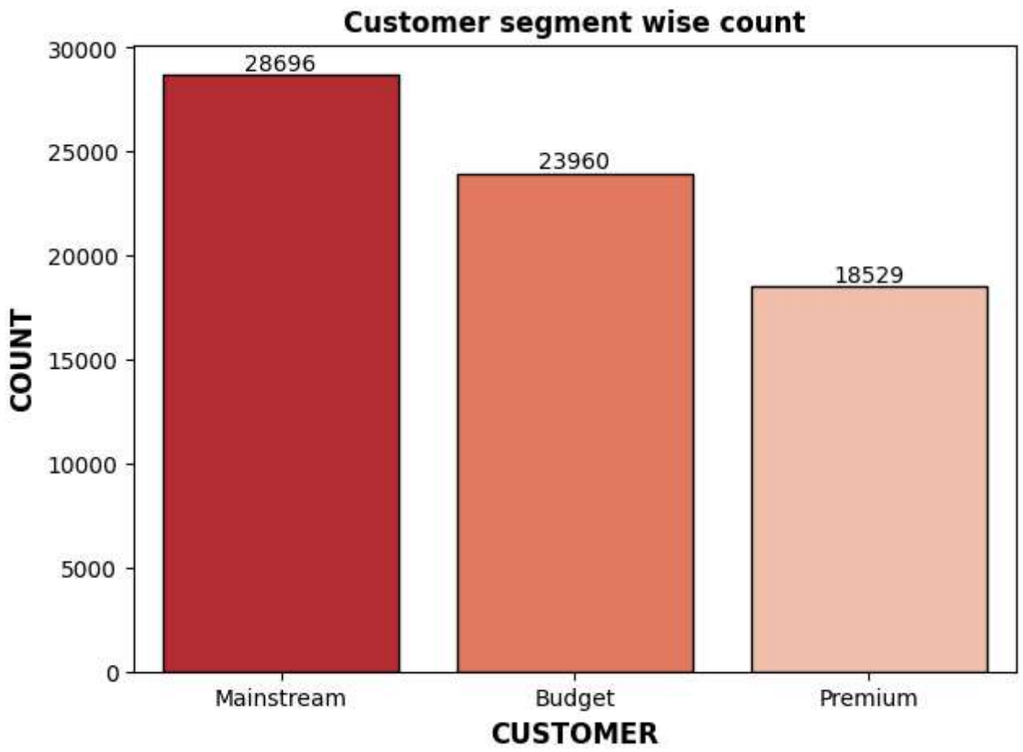
In [389…
```
plt.figure(figsize=(7,5))

ax = sns.barplot(x = 'PREMIUM_CUSTOMER', y = 'count', data=plot_um1, palette='Reds_r', edgecolor='black')
plt.xlabel("CUSTOMER", fontsize=12, fontweight='bold')
plt.ylabel("COUNT", fontsize=12, fontweight='bold')
```

```
plt.title("Customer segment wise count", fontsize=12, fontweight='bold')

for bars in ax.containers:
    ax.bar_label(bars)

plt.show()
```



In [422…
```
unique_members.columns
```

Out[422…
```
Index(['DATE', 'STORE_NBR', 'LYLTY_CARD_NBR', 'TXN_ID', 'PROD_NBR',
       'PROD_NAME', 'PROD_QTY', 'TOT_SALES', 'Weight', 'Weight in g',
       'PROD_BRAND', 'LIFESTAGE', 'PREMIUM_CUSTOMER'],
      dtype='object')
```

c) Customer segment wise Top 2 Brand Purchase

In [487…
```
customer_brand = unique_members.groupby('PREMIUM_CUSTOMER')['PROD_BRAND'].apply(lambda x: x.value_counts().nlargest(2)).reset_index()
```

In [488…
```
customer_brand.columns = ['PREMIUM CUSTOMER', 'BRAND', 'COUNT']
customer_brand
```

Out[488…

	PREMIUM CUSTOMER	BRAND	COUNT
0	Budget	Kettle	4027
1	Budget	Smiths	2821
2	Mainstream	Kettle	5222
3	Mainstream	Doritos	3165
4	Premium	Kettle	3112
5	Premium	Smiths	2116

In [503…
```
customer_brand.dtypes
```

Out[503…
```
PREMIUM CUSTOMER    object
BRAND               object
COUNT                int64
dtype: object
```

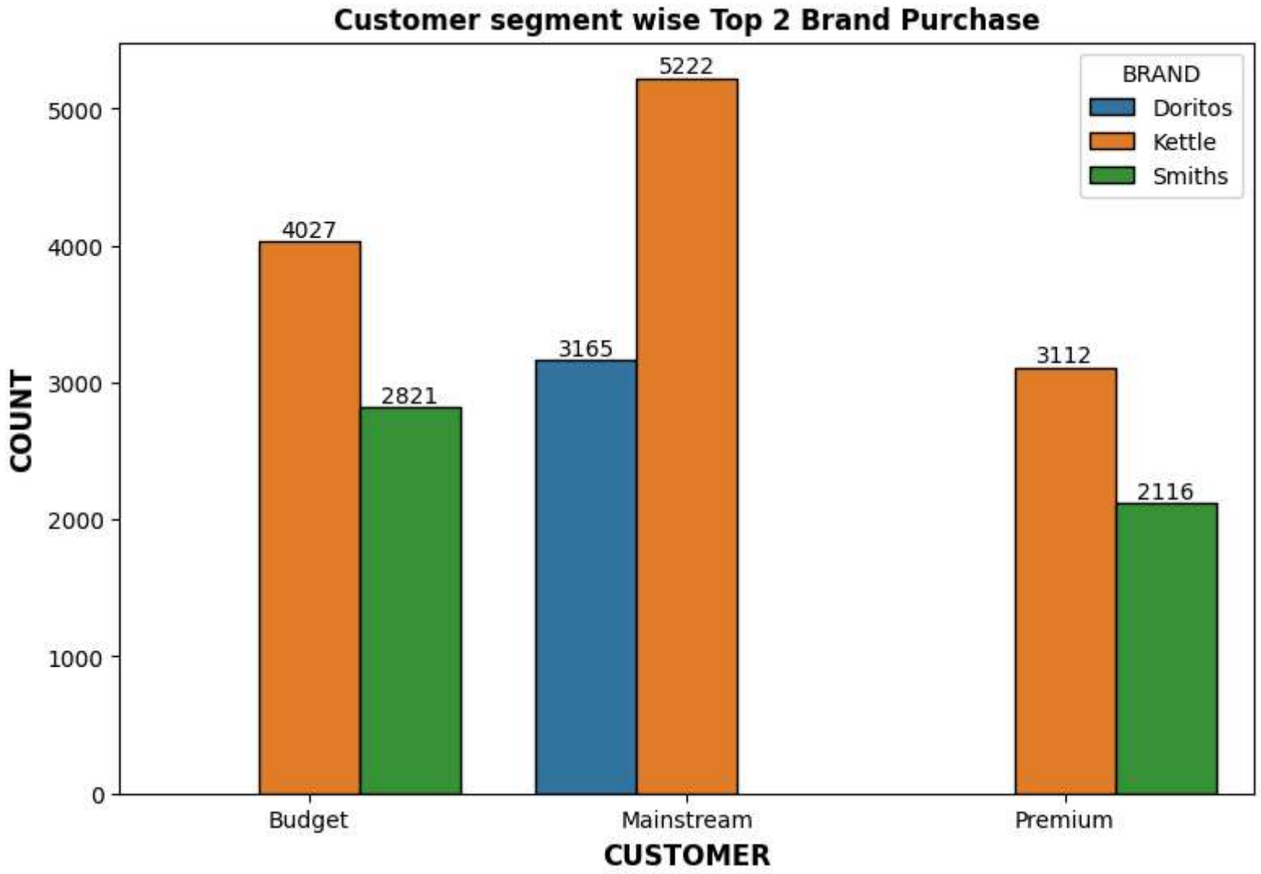
In [504…
```
customer_brand['PREMIUM CUSTOMER'] = customer_brand['PREMIUM CUSTOMER'].astype('category')
customer_brand['BRAND'] = customer_brand['BRAND'].astype('category')
```

In [517…
```
plt.figure(figsize=(9,6))

ax = sns.barplot(x = 'PREMIUM CUSTOMER', y = 'COUNT', hue='BRAND', data = customer_brand, edgecolor='black', dodge=True)
plt.xlabel("CUSTOMER", fontsize=12, fontweight='bold')
plt.ylabel("COUNT", fontsize=12, fontweight='bold')
plt.title("Customer segment wise Top 2 Brand Purchase", fontsize=12, fontweight='bold')

for bars in ax.containers:
    ax.bar_label(bars)

plt.show()
```



d) Unique Lifestage wise Count

In [546...
```
plot_um2 = unique_members['LIFESTAGE'].value_counts().reset_index()
plot_um2['Percentage'] = (plot_um2['count'] / plot_um2['count'].sum() * 100).round(3)
plot_um2
```

Out[546...

	LIFESTAGE	count	Percentage
0	RETIREES	14529	20.410
1	OLDER SINGLES/COUPLES	14372	20.190
2	YOUNG SINGLES/COUPLES	14014	19.687
3	OLDER FAMILIES	9617	13.510
4	YOUNG FAMILIES	9031	12.687
5	MIDAGE SINGLES/COUPLES	7133	10.020
6	NEW FAMILIES	2489	3.497

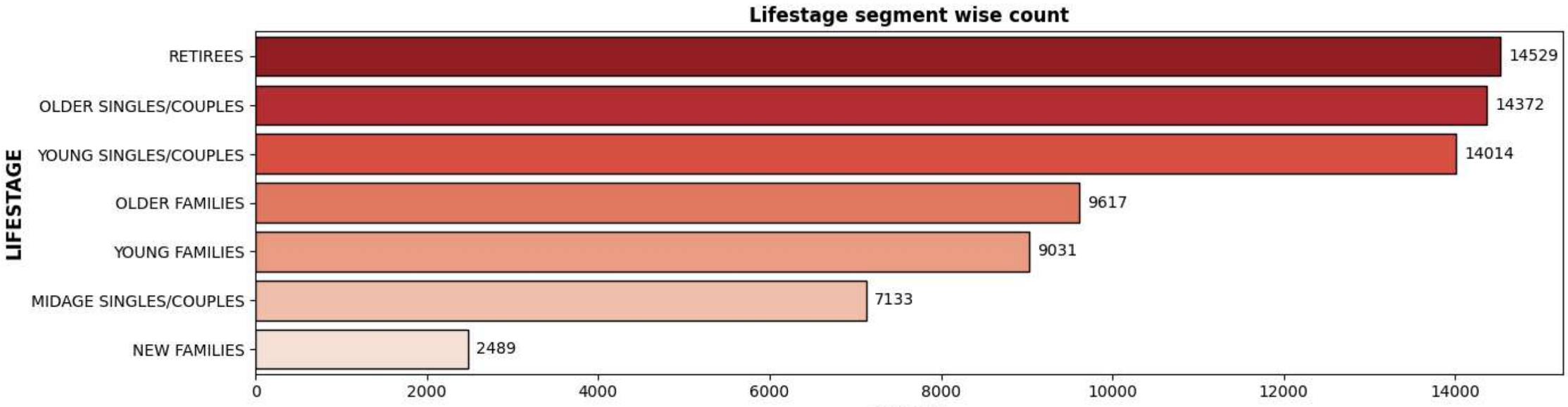
In [399...
```
plt.figure(figsize=(14,4))

ax = sns.barplot(x = 'count', y = 'LIFESTAGE', data=plot_um2, palette='Reds_r', edgecolor='black')
plt.xlabel("COUNT", fontsize=12, fontweight='bold')
```

```
plt.ylabel("LIFESTAGE", fontsize=12, fontweight='bold')
plt.title("Lifestage segment wise count", fontsize=12, fontweight='bold')

for bars in ax.containers:
    ax.bar_label(bars, padding=5)

plt.tight_layout()
plt.show()
```



In [357…
```
chips_merged.head(2)
```

Out[357…

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	Weight	Weight in g	PROD_BRAND	LIFESTAGE	PREMIUM_CUSTOMER
0	2018-10-17	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0	175g	175	Natural	YOUNG SINGLES/COUPLES	Premium
1	2019-05-14	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3	175g	175	CCs	MIDAGE SINGLES/COUPLES	Budget

e) Brand wise Top Sales

In [547…
```
highest_sales = chips_merged.groupby('PROD_BRAND')['TOT_SALES'].sum().reset_index().sort_values(by='TOT_SALES', ascending=False)
highest_sales['Percentage'] = (highest_sales['TOT_SALES'] / highest_sales['TOT_SALES'].sum() * 100).round(3)
highest_sales
```

Out[547...

	PROD_BRAND	TOT_SALES	Percentage
10	Kettle	111250.60	22.313
5	Doritos	64557.05	12.948
15	Smiths	55860.70	11.204
13	Pringles	51389.30	10.307
9	Infuzions	27702.40	5.556
18	Thins	25621.20	5.139
20	Twisties	23431.40	4.700
19	Tostitos	23144.00	4.642
14	Red	20701.50	4.152
4	Cobs	20033.60	4.018
21	Tyrrells	14569.80	2.922
7	Grain	12092.40	2.425
3	Cheezels	11342.70	2.275
22	WW	8569.90	1.719
12	Natural	8172.00	1.639
1	CCs	4462.50	0.895
2	Cheetos	4316.70	0.866
6	French	2109.00	0.423
8	GrnWves	1993.30	0.400
11	NCC	1974.00	0.396
0	Burger	1653.70	0.332
23	Woolworths	1224.00	0.245
16	Snbts	1212.10	0.243
17	Sunbites	1205.30	0.242

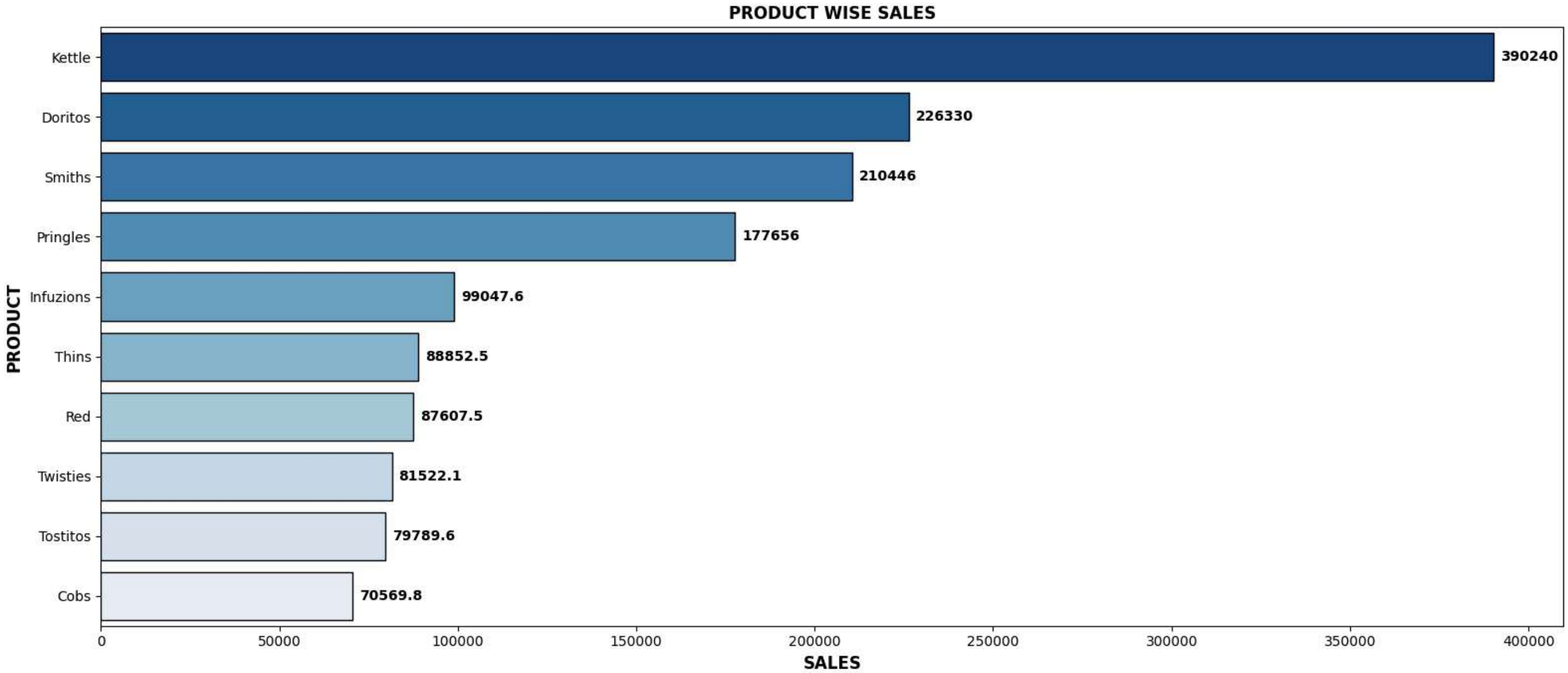
In [378...

```
plt.figure(figsize=(16,7))

ax = sns.barplot(x = 'TOT_SALES', y = 'PROD_BRAND', data=highest_sales.head(10), palette='Blues_r', edgecolor='black')
plt.xlabel("SALES", fontsize=12, fontweight='bold')
plt.ylabel("PRODUCT", fontsize=12, fontweight='bold')
plt.title("PRODUCT WISE SALES", fontsize=12, fontweight='bold')

for bars in ax.containers:
    ax.bar_label(bars, padding=5, fontweight = 'bold')

plt.tight_layout()
plt.show()
```



f) Lifestage wise Purchases

In [548...
```
lifestage_purchases = unique_members.groupby('LIFESTAGE')['PROD_QTY'].sum().sort_values(ascending=False).reset_index()
lifestage_purchases['Percentage'] = (lifestage_purchases['PROD_QTY'] / lifestage_purchases['PROD_QTY'].sum() * 100).round(3)
lifestage_purchases
```

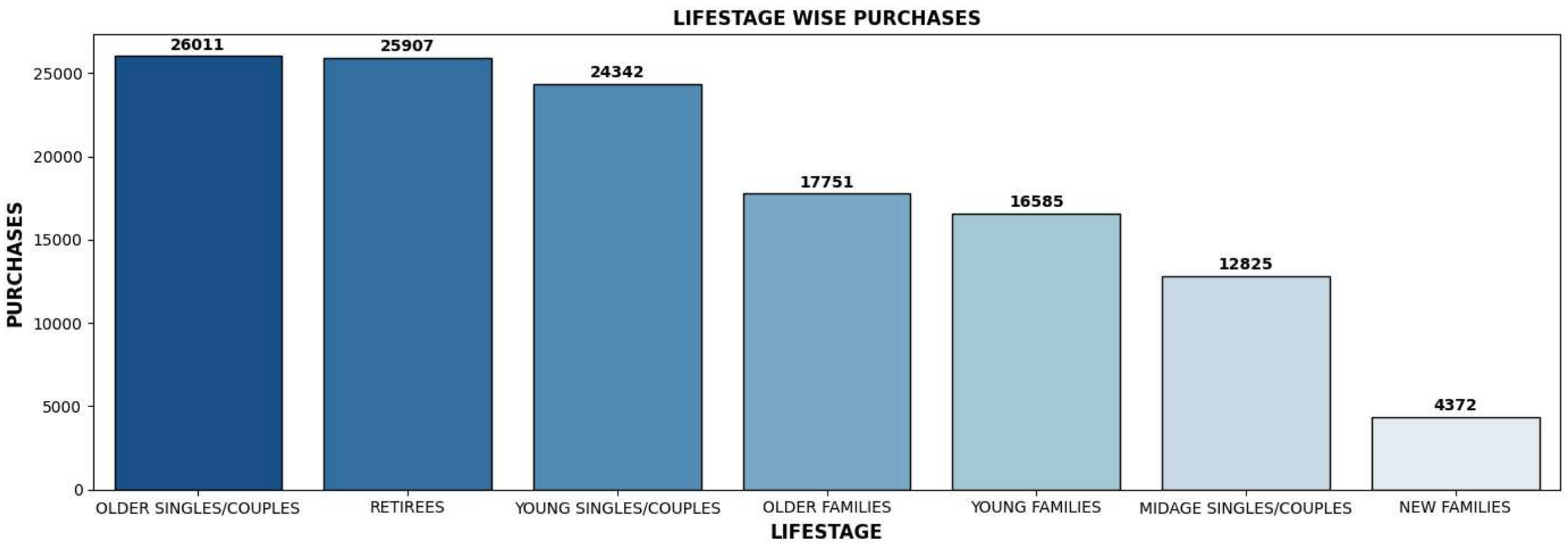
Out[548...

	LIFESTAGE	PROD_QTY	Percentage
0	OLDER SINGLES/COUPLES	26011	20.354
1	RETIREES	25907	20.273
2	YOUNG SINGLES/COUPLES	24342	19.048
3	OLDER FAMILIES	17751	13.890
4	YOUNG FAMILIES	16585	12.978
5	MIDAGE SINGLES/COUPLES	12825	10.036
6	NEW FAMILIES	4372	3.421

In [416…
```
plt.figure(figsize=(14,5))

ax = sns.barplot(x = 'LIFESTAGE', y = 'PROD_QTY', data=lifestage_purchases, palette='Blues_r', edgecolor='black')
plt.xlabel("LIFESTAGE", fontsize=12, fontweight='bold')
plt.ylabel("PURCHASES", fontsize=12, fontweight='bold')
plt.title("LIFESTAGE WISE PURCHASES", fontsize=12, fontweight='bold')

for bars in ax.containers:
    ax.bar_label(bars, padding=1.8, fontweight = 'bold')
plt.tight_layout()
plt.show()
```



In [419…
```
chips_merged.columns
```

Out[419…
```
Index(['DATE', 'STORE_NBR', 'LYLTY_CARD_NBR', 'TXN_ID', 'PROD_NBR',
       'PROD_NAME', 'PROD_QTY', 'TOT_SALES', 'Weight', 'Weight in g',
       'PROD_BRAND', 'LIFESTAGE', 'PREMIUM_CUSTOMER'],
      dtype='object')
```

g) LIFESTAGE WISE TOTAL SALES

In [550…
```
lifestage_sales = chips_merged.groupby('LIFESTAGE')['TOT_SALES'].sum().sort_values(ascending=False).reset_index()
lifestage_sales['Percentage'] = (lifestage_sales['TOT_SALES'] / lifestage_sales['TOT_SALES'].sum() * 100).round(3)
lifestage_sales
```

Out[550…

	LIFESTAGE	TOT_SALES	Percentage
0	OLDER SINGLES/COUPLES	101930.70	20.444
1	RETIREES	101729.30	20.403
2	YOUNG SINGLES/COUPLES	96159.40	19.286
3	OLDER FAMILIES	67754.00	13.589
4	YOUNG FAMILIES	63629.30	12.762
5	MIDAGE SINGLES/COUPLES	50191.10	10.067
6	NEW FAMILIES	17195.35	3.449

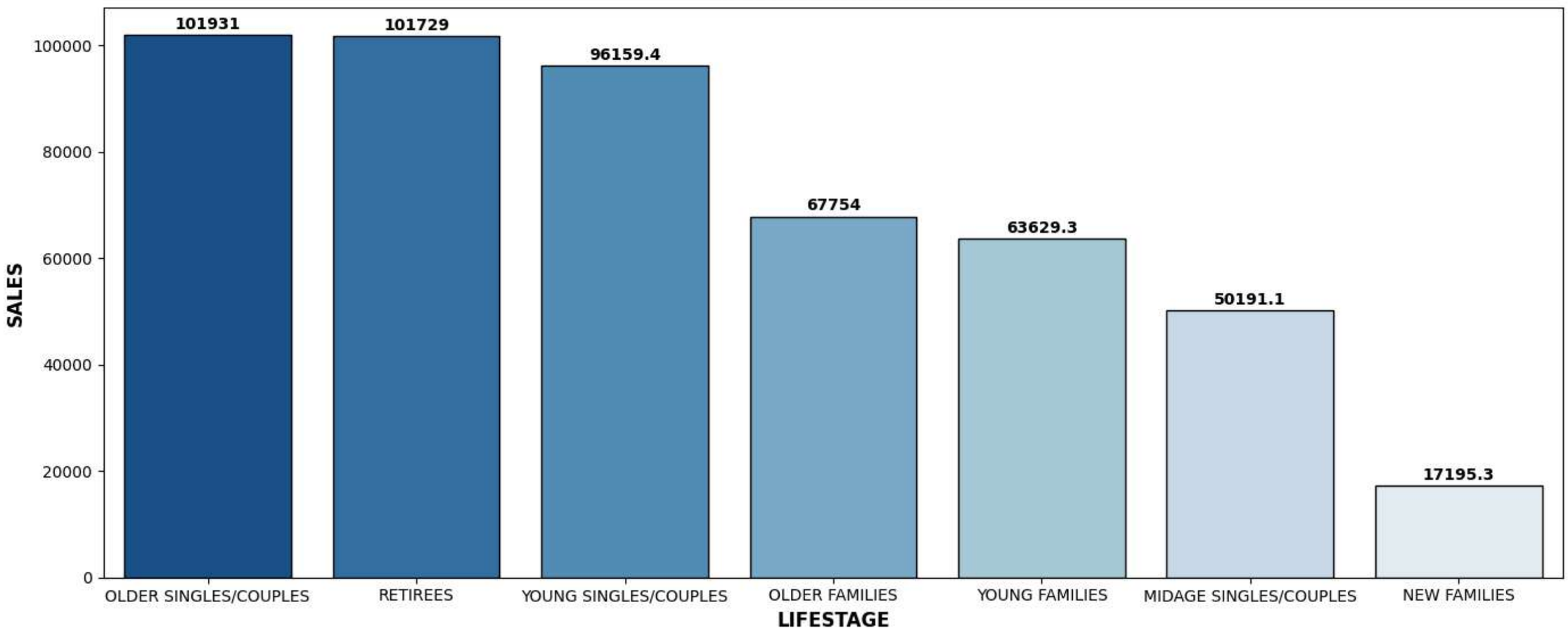
In [524…

```
plt.figure(figsize=(14,6))

ax = sns.barplot(x = 'LIFESTAGE', y = 'TOT_SALES', data=lifestage_sales, palette='Blues_r', edgecolor='black')
plt.xlabel("LIFESTAGE", fontsize=12, fontweight='bold')
plt.ylabel("SALES", fontsize=12, fontweight='bold')
plt.title("LIFESTAGE WISE SALES", fontsize=12, fontweight='bold')

for bars in ax.containers:
    ax.bar_label(bars, padding=1.8, fontweight = 'bold')

plt.tight_layout()
plt.show()
```



In [525…
```
chips_merged.head(2)
```

Out[525…

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	Weight	Weight in g	PROD_BRAND	LIFESTAGE	PREMIUM_CUSTOMER
0	2018-10-17	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0	175g	175	Natural	YOUNG SINGLES/COUPLES	Premium
1	2019-05-14	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3	175g	175	CCs	MIDAGE SINGLES/COUPLES	Budget

h) Average Sales by Customer

In [530…
```
avg_sales_customer = unique_members.groupby('PREMIUM_CUSTOMER')['TOT_SALES'].mean().round(3).reset_index()
avg_sales_customer
```

Out[530…

	PREMIUM_CUSTOMER	TOT_SALES
0	Budget	6.936
1	Mainstream	7.100
2	Premium	6.945

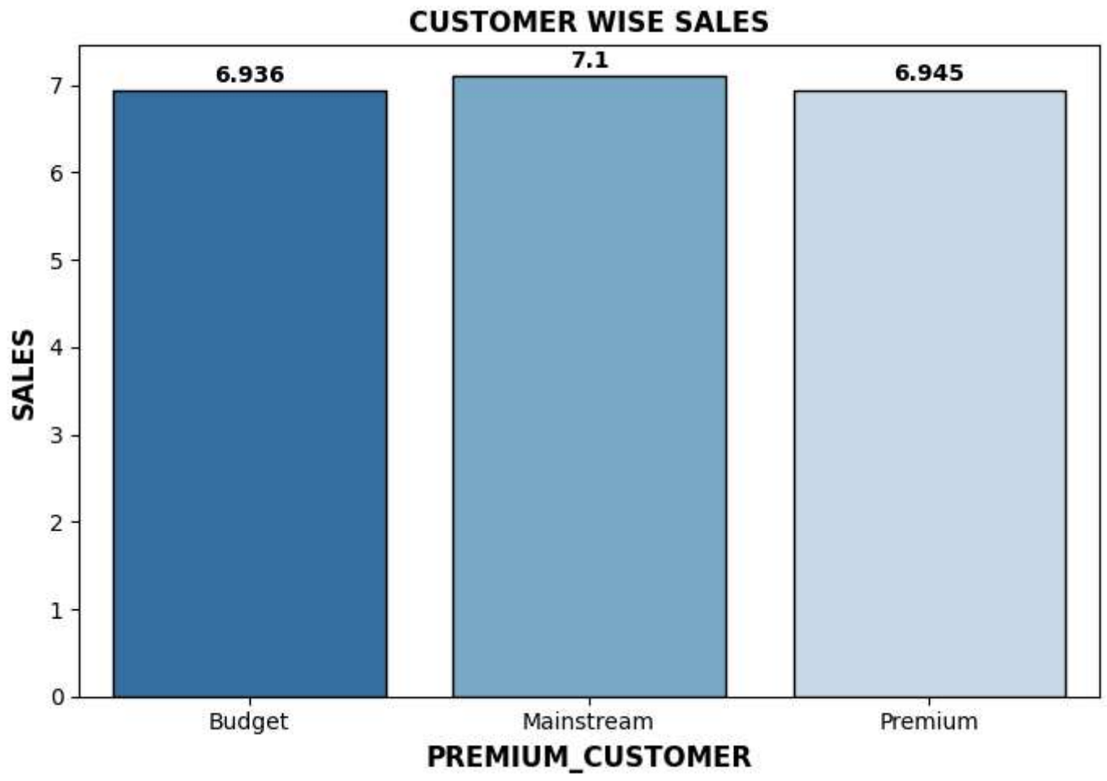
In [557…

```
plt.figure(figsize=(7,5))

ax = sns.barplot(x = 'PREMIUM_CUSTOMER', y = 'TOT_SALES', data=avg_sales_customer, palette='Blues_r', edgecolor='black')
plt.xlabel("PREMIUM_CUSTOMER", fontsize=12, fontweight='bold')
plt.ylabel("SALES", fontsize=12, fontweight='bold')
plt.title("CUSTOMER WISE SALES", fontsize=12, fontweight='bold')

for bars in ax.containers:
    ax.bar_label(bars, padding=1.8, fontweight = 'bold')

plt.tight_layout()
plt.show()
```



In [551…

```
chips_merged.columns
```

Out[551…

```
Index(['DATE', 'STORE_NBR', 'LYLTY_CARD_NBR', 'TXN_ID', 'PROD_NBR',
       'PROD_NAME', 'PROD_QTY', 'TOT_SALES', 'Weight', 'Weight in g',
       'PROD_BRAND', 'LIFESTAGE', 'PREMIUM_CUSTOMER'],
      dtype='object')
```

i) Customer wise Total Sales

In [554…
```
customer_sales = unique_members.groupby('PREMIUM_CUSTOMER')['TOT_SALES'].sum().reset_index()
customer_sales['Percentage'] = (customer_sales['TOT_SALES'] / customer_sales['TOT_SALES'].sum() * 100).round(3)
customer_sales
```

Out[554…

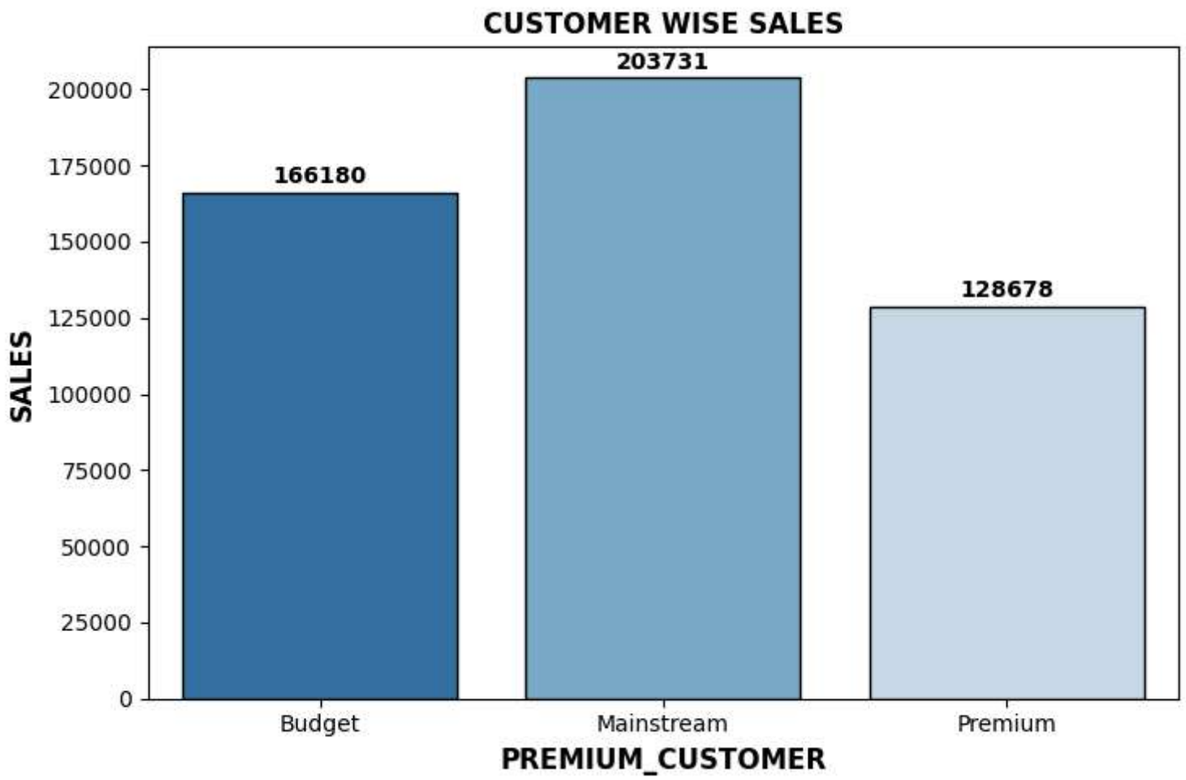
	PREMIUM_CUSTOMER	TOT_SALES	Percentage
0	Budget	166180.45	33.330
1	Mainstream	203730.85	40.861
2	Premium	128677.85	25.808

In [558…
```
plt.figure(figsize=(7.5,5))

ax = sns.barplot(x = 'PREMIUM_CUSTOMER', y = 'TOT_SALES', data=customer_sales, palette='Blues_r', edgecolor='black')
plt.xlabel("PREMIUM_CUSTOMER", fontsize=12, fontweight='bold')
plt.ylabel("SALES", fontsize=12, fontweight='bold')
plt.title("CUSTOMER WISE SALES", fontsize=12, fontweight='bold')

for bars in ax.containers:
    ax.bar_label(bars, padding=1.8, fontweight = 'bold')

plt.tight_layout()
plt.show()
```



j) Bag Size Purchases

In [573…
```
def categorize_bag_size(weight):
    if 70 <= weight <= 90:
```

```
        return 'Extra Small'
    elif 91 <= weight <= 190:
        return 'Small'
    elif 191 <= weight <= 270:
        return 'Medium'
    else:
        return 'Large'

# Apply the function to assign 'Bag Size'
chips_transac['Bag Size'] = chips_transac['Weight in g'].apply(categorize_bag_size)
```

In [576…

```
bag_size = chips_transac['Bag Size'].value_counts().reset_index()
bag_size
```

Out[576…

	Bag Size	count
0	Small	200068
1	Medium	21763
2	Large	18956
3	Extra Small	4515

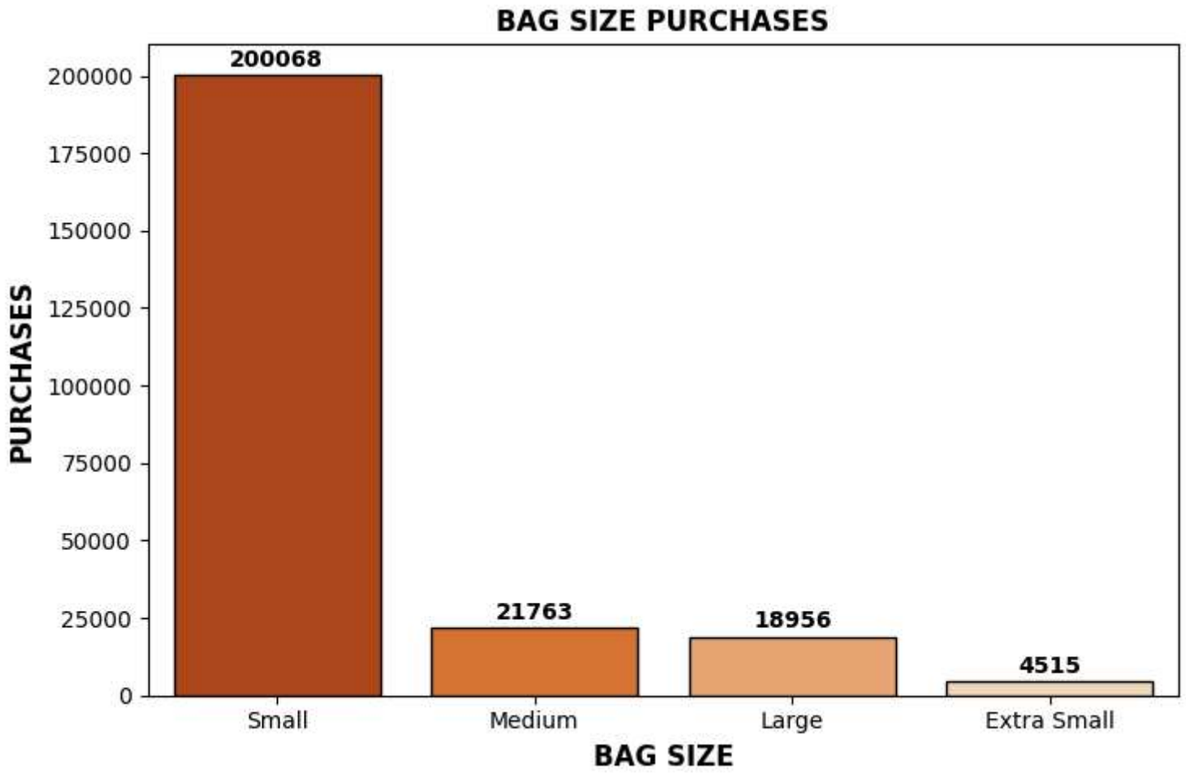
In [580…

```
plt.figure(figsize=(7.5,5))

ax = sns.barplot(x = 'Bag Size', y = 'count', data=bag_size, palette='Oranges_r', edgecolor='black')
plt.xlabel("BAG SIZE", fontsize=12, fontweight='bold')
plt.ylabel("PURCHASES", fontsize=12, fontweight='bold')
plt.title("BAG SIZE PURCHASES", fontsize=12, fontweight='bold')

for bars in ax.containers:
    ax.bar_label(bars, padding=1.8, fontweight = 'bold')

plt.tight_layout()
plt.show()
```



OBSERVATIONS

- 40.3% of the customers belong to Mainstream
- Maximum revenue contrubution is from the mainstream customers i.e 40.8%
- Top 2 selling brands for Budget Customers are Kettle and Smith, for Mainstream are Kettle and Doritos and for Premium Customers are Kettle and Smith
- 20.4% of customers are Retirees
- Brand KETTLE has highest number of sales revenue i.e about 22.4%
- Older/Singles Couples makes the most purchases of about 20.4%
- Older/Singles Couples contributes to the most sales revenue of about 20.44%
- Average Sales by customer segment is almost the same for all 3 segments i.e around 7$
- Small Bag Sizes are most in demand i.e 91g to 190g