quantium task 1

August 6, 2021

Task 1

0.1 Data preparation and customer analytics

We need to present a strategic recommendation to Julia that is supported by data which she can then use for the upcoming category review however to do so we need to analyse the data to understand the current purchasing trends and behaviours. The client is particularly interested in customer segments and their chip purchasing behaviour. Consider what metrics would help describe the customers' purchasing behaviour.

We have chosen to complete this task in R, however you will also find Python to be a useful tool in this piece of analytics. If you aren't familiar with R or Python we would recommend searching a few online courses to help get you started. We have also provided an R solution template if you want some assistance in getting through this Task. Whilst its possible to complete the task in Excel you may find the size of the data and the nature of the tasks is such that it is more difficult to complete in Excel.

To get started, download the resource csv data files below and begin performing high level data checks such as:

- Creating and interpreting high level summaries of the data
- Finding outliers and removing these (if applicable)
- Checking data formats and correcting (if applicable)

You will also want to derive **extra features** such as **pack size** and **brand name** from the data and **define metrics** of interest to enable you to draw insights on **who spends on chips** and **what drives spends for each customer segment**.

Remember **our end goal is to form a strategy based on the findings to provide a clear recommendation** to Julia the Category Manager so make sure your insights can have a commercial application.

As we are in the early stages of this analysis Zilinka has asked us to submit our initial findings, so please save your code as a .pdf file and upload it to unlock the model answer.

LIFESTAGE: Customer attribute that identifies whether a customer has a family or not and what point in life they are at e.g. are their children in pre-school/primary/secondary school.

PREMIUM_CUSTOMER: Customer segmentation used to differentiate shoppers by the price point of products they buy and the types of products they buy. It is used to identify whether customers may spend more for quality or brand or whether they will purchase the cheapest options.

- **Pro analytics Tip:** While the data set would not normally be considered large some operations may still take some time to run.

0.2 Importing libraries

[1]:
```
import pandas as pd
import numpy as np
```

[2]:
```
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

0.3 Importing data

[3]:
```
purchase = pd.read_csv("QVI_purchase_behaviour.csv")
```

[4]:
```
purchase.head()
```

[4]:

	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

[5]:
```
purchase.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
```

[6]:
```
purchase.isna().sum()
```

[6]:
```
LYLTY_CARD_NBR      0
LIFESTAGE           0
PREMIUM_CUSTOMER    0
dtype: int64
```

[7]:
```
purchase.columns
```

[7]:
```
Index(['LYLTY_CARD_NBR', 'LIFESTAGE', 'PREMIUM_CUSTOMER'], dtype='object')
```

[8]:
```
purchase.describe()
```

[8]:
```
       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
```

0.3.1 Import xlsx dataset and convert it to csv

[9]:
```
## let's covert "QVI_transaction_data.xlsx" to "QVI_transaction_data.csv" and
 ↪import it
read_file = pd.read_excel("QVI_transaction_data.xlsx")
read_file.to_csv("QVI_transaction_data.csv", index = None, header=True)
```

[10]:
```
transaction = pd.read_csv("QVI_transaction_data.csv")
```

[11]:
```
transaction.head()
```

[11]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	\
0	43390	1	1000	1	5	
1	43599	1	1307	348	66	
2	43605	1	1343	383	61	
3	43329	2	2373	974	69	
4	43330	2	2426	1038	108	

	PROD_NAME	PROD_QTY	TOT_SALES
0	Natural Chip Compny SeaSalt175g	2	6.0
1	CCs Nacho Cheese 175g	3	6.3
2	Smiths Crinkle Cut Chips Chicken 170g	2	2.9
3	Smiths Chip Thinly S/Cream&Onion 175g	5	15.0
4	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	13.8

[12]:
```
transaction.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
```

There we notice the date column is an int64 we will convert it to datetime type

[13]:
```
transaction.describe()
```

[13]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	\
count	264836.000000	264836.00000	2.648360e+05	2.648360e+05	
mean	43464.036260	135.08011	1.355495e+05	1.351583e+05	
std	105.389282	76.78418	8.057998e+04	7.813303e+04	
min	43282.000000	1.00000	1.000000e+03	1.000000e+00	
25%	43373.000000	70.00000	7.002100e+04	6.760150e+04	
50%	43464.000000	130.00000	1.303575e+05	1.351375e+05	
75%	43555.000000	203.00000	2.030942e+05	2.027012e+05	
max	43646.000000	272.00000	2.373711e+06	2.415841e+06	

	PROD_NBR	PROD_QTY	TOT_SALES
count	264836.000000	264836.000000	264836.000000
mean	56.583157	1.907309	7.304200
std	32.826638	0.643654	3.083226
min	1.000000	1.000000	1.500000
25%	28.000000	2.000000	5.400000
50%	56.000000	2.000000	7.400000
75%	85.000000	2.000000	9.200000
max	114.000000	200.000000	650.000000

[14]:
```
transaction.columns
```

[14]:
```
Index(['DATE', 'STORE_NBR', 'LYLTY_CARD_NBR', 'TXN_ID', 'PROD_NBR',
       'PROD_NAME', 'PROD_QTY', 'TOT_SALES'],
      dtype='object')
```

[15]:
```
# here we will be transforming date column from julian to datetime type
transaction["DATE"]=pd.to_datetime(transaction["DATE"], origin =␣
 ↪"1899-12-30",unit="D")
```

[16]:
```
transaction.head()
```

[16]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	\
0	2018-10-17	1	1000	1	5	
1	2019-05-14	1	1307	348	66	
2	2019-05-20	1	1343	383	61	
3	2018-08-17	2	2373	974	69	

```
4 2018-08-18          2            2426    1038       108

                                  PROD_NAME  PROD_QTY  TOT_SALES
0    Natural Chip        Compny SeaSalt175g         2        6.0
1                  CCs Nacho Cheese    175g         3        6.3
2    Smiths Crinkle Cut  Chips Chicken 170g         2        2.9
3    Smiths Chip Thinly  S/Cream&Onion 175g         5       15.0
4  Kettle Tortilla ChpsHny&Jlpno Chili 150g         3       13.8
```

We also can see PROD_NAME column has to split it's values into multiple columns

We have what the customer have bought and how much does it weight

[17]:
```
transaction.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  datetime64[ns]
 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: datetime64[ns](1), float64(1), int64(5), object(1)
memory usage: 16.2+ MB
```

[18]:
```
transaction.isna().sum()
```

[18]:
```
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
```

Whoaaa!!! no missing values in both datasets let's check the unique values of each column from both datasets

But first let's check some values

[19]:
```
transaction["PROD_NAME"].describe()
```

[19]:
```
count                                   264836
unique                                     114
top       Kettle Mozzarella   Basil & Pesto 175g
freq                                      3304
Name: PROD_NAME, dtype: object
```

From **264836** values there's just **114** unique values

The highest sold merchandise is **Kettle Mozzarella** from **Basil & Pesto** which weights **175g**

To find the most frequent words our approach will be from this link

Approach :

- Import Counter class from collections module.
- Split the string into list using split(), it will return the lists of words.
- Now pass the list to the instance of Counter class.
- The function 'most-common()' inside Counter will return the list of most frequent words from list and its count.

[20]:
```
## frequent words
from collections import Counter
freq = Counter([j  for s in transaction["PROD_NAME"] for j in s.split()])
```

[21]:
```
#sorting in decreasing order of the frequency of words
frequ = pd.DataFrame([freq.keys(),freq.values()],
                     index=['Word','Frequency']).transpose().
 ↪sort_values(by='Frequency',ascending=False)
```

[22]:
```
# removing useless words like '170g'
frequ=frequ[[ s[0] not in  ['0','1','2','3','4','5','6','7','8','9','&'] for s␣
 ↪in frequ['Word'] ]]
# most frequent words
frequ.head()
```

[22]:

	Word	Frequency
11	Chips	49770
16	Kettle	41288
8	Smiths	28860
29	Salt	27976
6	Cheese	27890

[23]:
```
transaction[[("Salsa" in s)  for s in transaction['PROD_NAME']]]
```

[23]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	\
5	2019-05-19	4	4074	2982	57	
25	2019-05-15	39	39144	35506	57	
32	2019-05-20	45	45127	41122	64	
44	2018-08-18	56	56013	50090	39	
63	2019-05-15	82	82480	82047	101	

```
…            …         …              …       …       …
264675 2019-04-20        265          265103  263419        59
264678 2019-03-30        265          265111  263428        35
264719 2018-10-28        266          266278  264104        39
264734 2019-01-11        267          267324  264374        41
264780 2019-01-10        269          269222  266382        64

                                      PROD_NAME  PROD_QTY  TOT_SALES
5       Old El Paso Salsa   Dip Tomato Mild 300g         1        5.1
25      Old El Paso Salsa   Dip Tomato Mild 300g         1        5.1
32      Red Rock Deli SR    Salsa & Mzzrlla 150g         2        5.4
44         Smiths Crinkle Cut  Tomato Salsa 150g         1        2.6
63               Doritos Salsa       Medium 300g         1        2.6
…                                          …       …         …
264675   Old El Paso Salsa   Dip Tomato Med 300g         1        5.1
264678            Woolworths Mild     Salsa 300g         1        1.5
264719     Smiths Crinkle Cut  Tomato Salsa 150g         1        2.6
264734                  Doritos Salsa Mild  300g         1        2.6
264780  Red Rock Deli SR    Salsa & Mzzrlla 150g         2        5.4

[18094 rows x 8 columns]
```

[24]:
```
i = 0
while i < 6 in transaction['TOT_SALES']:
    i += 1
    print(len(transaction['PROD_NAME'].unique()))
    print(pd.unique(transaction['PROD_NAME']))
```

```
114
['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']
114
['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']
114
['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']
114
['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']
114
['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']
114
['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']
```

[25]:
```
# dropping salsa items
transaction.drop(transaction[[("Salsa" in s)  for s in␣
 ↪transaction['PROD_NAME']]].index,inplace=True)
transaction[[("Salsa" in s)  for s in transaction['PROD_NAME']]]
```

[25]:
```
Empty DataFrame
Columns: [DATE, STORE_NBR, LYLTY_CARD_NBR, TXN_ID, PROD_NBR, PROD_NAME,
PROD_QTY, TOT_SALES]
Index: []
```

Let's check after modification

[26]:
```
transaction.describe()
```

[26]:
```
            STORE_NBR  LYLTY_CARD_NBR        TXN_ID       PROD_NBR  \
count  246742.000000    2.467420e+05  2.467420e+05  246742.000000
mean      135.051098    1.355310e+05  1.351311e+05      56.351789
std        76.787096    8.071528e+04  7.814772e+04      33.695428
min         1.000000    1.000000e+03  1.000000e+00       1.000000
25%        70.000000    7.001500e+04  6.756925e+04      26.000000
50%       130.000000    1.303670e+05  1.351830e+05      53.000000
75%       203.000000    2.030840e+05  2.026538e+05      87.000000
max       272.000000    2.373711e+06  2.415841e+06     114.000000

            PROD_QTY      TOT_SALES
count  246742.000000  246742.000000
mean        1.908062       7.321322
std         0.659831       3.077828
min         1.000000       1.700000
25%         2.000000       5.800000
50%         2.000000       7.400000
75%         2.000000       8.800000
max       200.000000     650.000000
```

[27]:
```
transaction.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 246742 entries, 0 to 264835
Data columns (total 8 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   DATE            246742 non-null  datetime64[ns]
 1   STORE_NBR       246742 non-null  int64
 2   LYLTY_CARD_NBR  246742 non-null  int64
 3   TXN_ID          246742 non-null  int64
 4   PROD_NBR        246742 non-null  int64
 5   PROD_NAME       246742 non-null  object
 6   PROD_QTY        246742 non-null  int64
 7   TOT_SALES       246742 non-null  float64
dtypes: datetime64[ns](1), float64(1), int64(5), object(1)
memory usage: 16.9+ MB
```

[28]:
```
transaction.isna().sum()
```

[28]:
```
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
```

0.4 Removing Outliers

[29]:
```
transaction['PROD_QTY'].describe()
```

[29]:
```
count    246742.000000
mean          1.908062
std           0.659831
min           1.000000
25%           2.000000
50%           2.000000
75%           2.000000
max         200.000000
Name: PROD_QTY, dtype: float64
```

[30]:
```
transaction[transaction['PROD_QTY']>5]
```

[30]:
```
            DATE  STORE_NBR  LYLTY_CARD_NBR  TXN_ID  PROD_NBR  \
69762 2018-08-19        226          226000  226201         4
69763 2019-05-20        226          226000  226210         4

                            PROD_NAME  PROD_QTY  TOT_SALES
69762  Dorito Corn Chp     Supreme 380g       200      650.0
69763  Dorito Corn Chp     Supreme 380g       200      650.0
```

[31]:
```
transaction[transaction['TOT_SALES'] > 600]
```

[31]:
```
            DATE  STORE_NBR  LYLTY_CARD_NBR  TXN_ID  PROD_NBR  \
69762 2018-08-19        226          226000  226201         4
69763 2019-05-20        226          226000  226210         4

                            PROD_NAME  PROD_QTY  TOT_SALES
69762  Dorito Corn Chp     Supreme 380g       200      650.0
69763  Dorito Corn Chp     Supreme 380g       200      650.0
```

[32]:
```
# transaction.drop(labels=transaction[transaction['PROD_QTY']==200].
 ↪index,inplace=True)
transaction.drop(labels=transaction[transaction['TOT_SALES']>600].
 ↪index,inplace=True)
# transaction.drop(labels=transaction[transaction['TXN_ID']>1500000].
 ↪index,inplace=True)
```

[33]:
```
transaction.describe()
```

[33]:
```
           STORE_NBR  LYLTY_CARD_NBR        TXN_ID       PROD_NBR  \
count  246740.000000    2.467400e+05  2.467400e+05  246740.000000
```

```
mean      135.050361    1.355303e+05  1.351304e+05      56.352213
std        76.786971    8.071520e+04  7.814760e+04      33.695235
min         1.000000    1.000000e+03  1.000000e+00       1.000000
25%        70.000000    7.001500e+04  6.756875e+04      26.000000
50%       130.000000    1.303670e+05  1.351815e+05      53.000000
75%       203.000000    2.030832e+05  2.026522e+05      87.000000
max       272.000000    2.373711e+06  2.415841e+06     114.000000

            PROD_QTY      TOT_SALES
count  246740.000000  246740.000000
mean        1.906456       7.316113
std         0.342499       2.474897
min         1.000000       1.700000
25%         2.000000       5.800000
50%         2.000000       7.400000
75%         2.000000       8.800000
max         5.000000      29.500000
```

Transaction maximum value now is 29.5 after it was 650 total sales of only a specific product **"Dorito Corn Chp Supreme 380g"**

[34]:
```
## missing dates
transac=transaction.groupby('DATE').count()
```

[35]:
```
transac.head(20)
```

[35]:

DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	\
2018-07-01	663	663	663	663	663	663	
2018-07-02	650	650	650	650	650	650	
2018-07-03	674	674	674	674	674	674	
2018-07-04	669	669	669	669	669	669	
2018-07-05	660	660	660	660	660	660	
2018-07-06	711	711	711	711	711	711	
2018-07-07	695	695	695	695	695	695	
2018-07-08	653	653	653	653	653	653	
2018-07-09	692	692	692	692	692	692	
2018-07-10	650	650	650	650	650	650	
2018-07-11	701	701	701	701	701	701	
2018-07-12	717	717	717	717	717	717	
2018-07-13	727	727	727	727	727	727	
2018-07-14	661	661	661	661	661	661	
2018-07-15	712	712	712	712	712	712	
2018-07-16	678	678	678	678	678	678	
2018-07-17	694	694	694	694	694	694	
2018-07-18	689	689	689	689	689	689	
2018-07-19	637	637	637	637	637	637	
2018-07-20	684	684	684	684	684	684	

```
            TOT_SALES
DATE
2018-07-01        663
2018-07-02        650
2018-07-03        674
2018-07-04        669
2018-07-05        660
2018-07-06        711
2018-07-07        695
2018-07-08        653
2018-07-09        692
2018-07-10        650
2018-07-11        701
2018-07-12        717
2018-07-13        727
2018-07-14        661
2018-07-15        712
2018-07-16        678
2018-07-17        694
2018-07-18        689
2018-07-19        637
2018-07-20        684
```

[36]:
```
#missing date
set(pd.date_range('2018-07-01', end='2019-06-30',freq='D'))-set((transac.index))
```

[36]:
```
{Timestamp('2018-12-25 00:00:00', freq='D')}
```

[37]:
```
transac.loc['2018-12-25']=np.nan
# =transac.mean().apply(int)
```

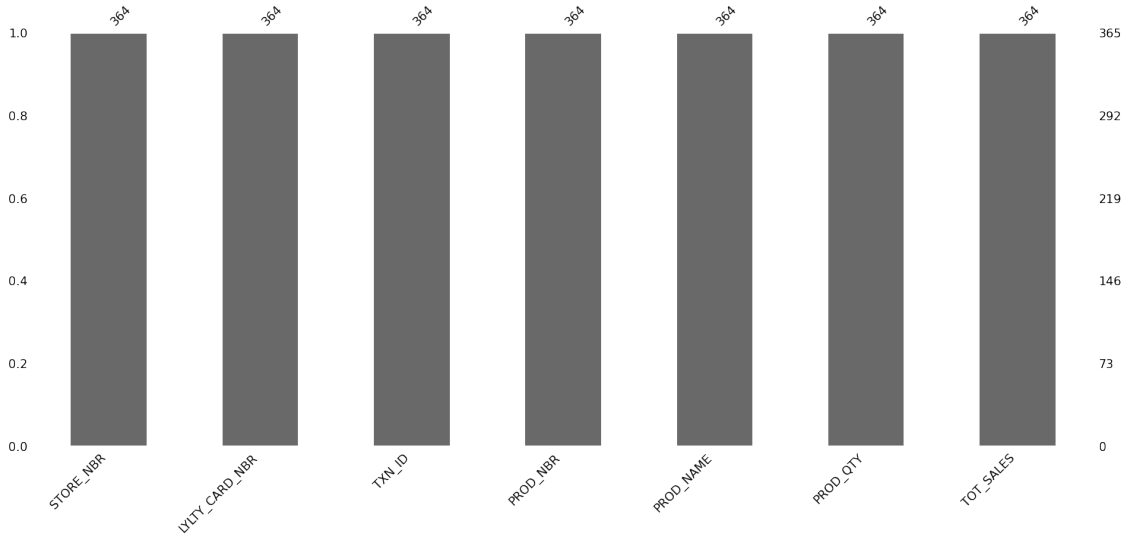
[38]:
```
transac[transac.index=='2018-12-25']
```

[38]:
```
            STORE_NBR  LYLTY_CARD_NBR  TXN_ID  PROD_NBR  PROD_NAME  PROD_QTY  \
DATE
2018-12-25        NaN             NaN     NaN       NaN        NaN       NaN

            TOT_SALES
DATE
2018-12-25        NaN
```

[39]:
```
#plot showing missing date
import missingno as msno
```

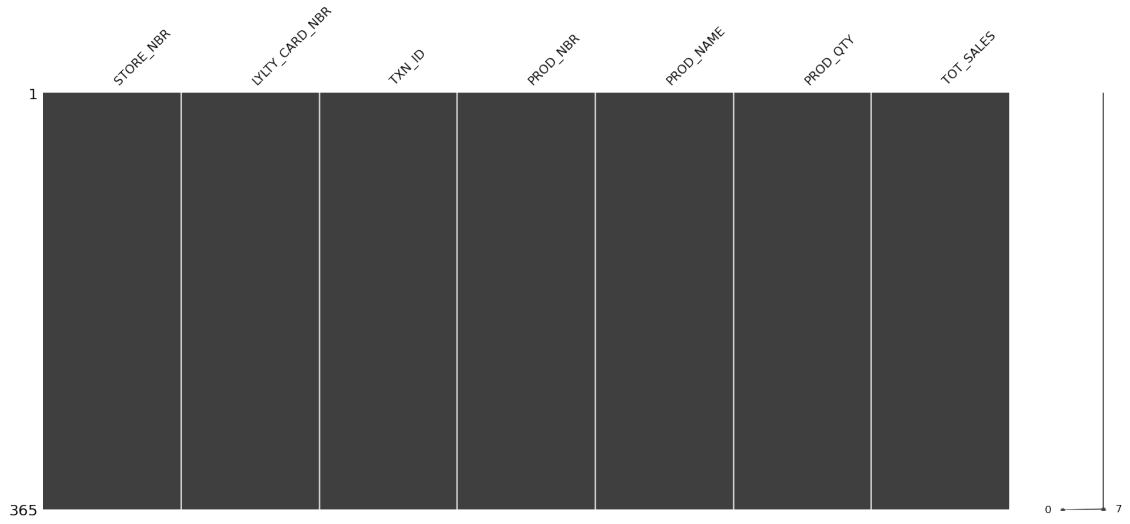
[40]:
```
# Visualize the completeness of the dataframe
msno.bar(transac)
```

[40]: <AxesSubplot:>



[41]:
```
# Visualize the locations of the missing values of the dataset
sorted = transac.sort_values(by = ['TXN_ID'])
msno.matrix(sorted)
```

[41]: <AxesSubplot:>

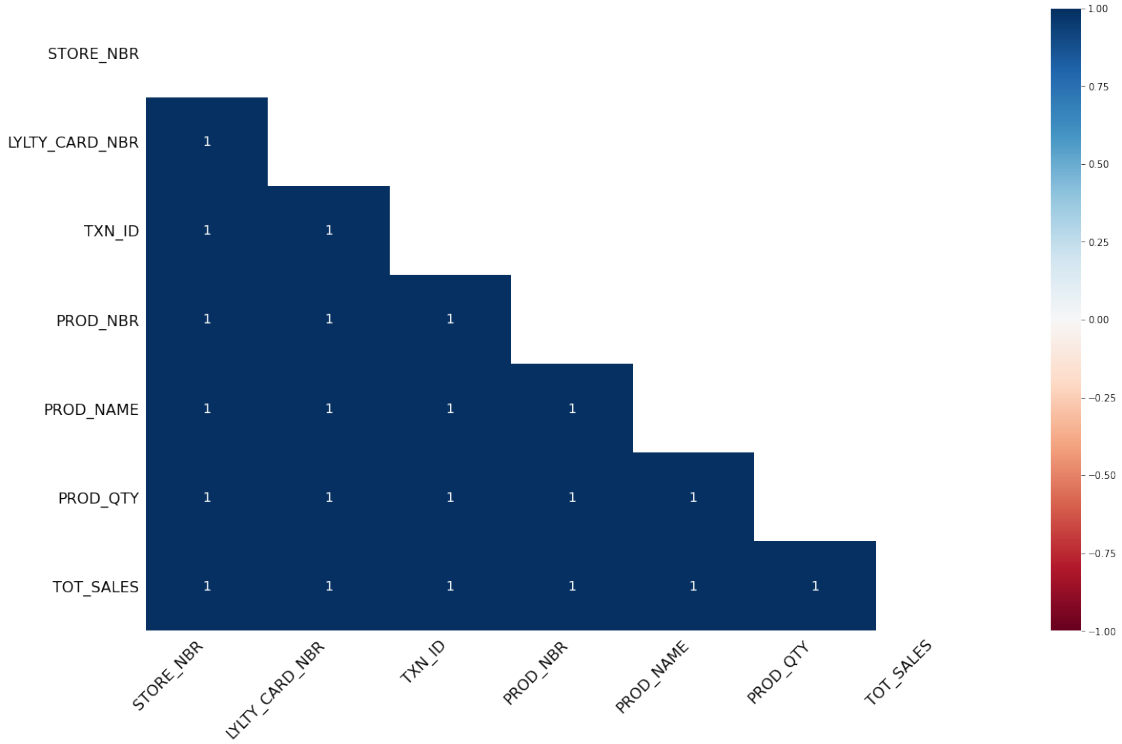


- The nullity matrix describes the nullity of the dataset and appears blank wherever there are missing values.
- The column on the very right summarizes the general shape of the data completeness and points out the row.

- Total count of columns at the bottom right.

[42]:
```
# Visualize the correlation between the numeric variables of the dataframe
msno.heatmap(transac)
```

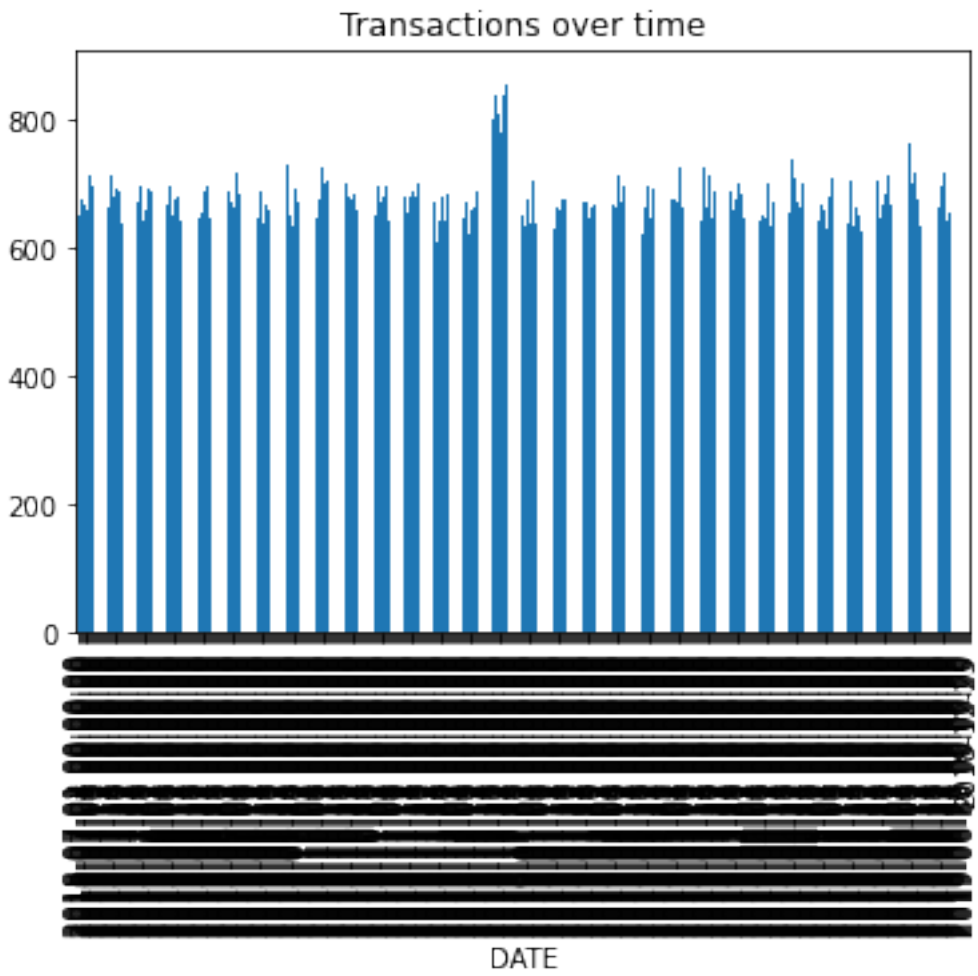
[42]: <AxesSubplot:>



Let's plot **TXN_ID** and see how it's changed it's transactions over time

[43]:
```
transac['TXN_ID'].plot(kind='bar',x ='Day',y = "Number of transactions", title␣
↪= "Transactions over time")
```

[43]: <AxesSubplot:title={'center':'Transactions over time'}, xlabel='DATE'>



[44]:
```
#Adding features

## it collects numbers in PROD_NAME rows and put them in different column
def fun(s):
    a=[]
    for i in s:
        if i in ['0','1','2','3','4','5','6','7','8','9']:
            a.append(i)
    return int("".join(a))
```

[45]:
```
transaction['PACKAGE_SIZE']=transaction['PROD_NAME'].apply(fun)
transaction.head(20)
```

[45]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	\
0	2018-10-17	1	1000	1	5	
1	2019-05-14	1	1307	348	66	

```
2  2019-05-20          1            1343     383        61
3  2018-08-17          2            2373     974        69
4  2018-08-18          2            2426    1038       108
6  2019-05-16          4            4149    3333        16
7  2019-05-16          4            4196    3539        24
8  2018-08-20          5            5026    4525        42
9  2018-08-18          7            7150    6900        52
10 2019-05-17          7            7215    7176        16
11 2018-08-20          8            8294    8221       114
12 2019-05-18          9            9208    8634        15
13 2018-08-17         13           13213   12447        92
14 2019-05-15         19           19272   16686        44
15 2019-05-19         20           20164   17136        54
16 2018-08-18         20           20418   17413        94
17 2018-08-14         22           22411   18646        98
18 2018-08-17         22           22456   18696        93
19 2019-05-16         23           23067   19162        56
20 2019-05-19         25           25105   21815         7

                                    PROD_NAME  PROD_QTY  TOT_SALES  \
0     Natural Chip        Compny SeaSalt175g         2        6.0
1                   CCs Nacho Cheese    175g         3        6.3
2     Smiths Crinkle Cut  Chips Chicken 170g         2        2.9
3     Smiths Chip Thinly  S/Cream&Onion 175g         5       15.0
4   Kettle Tortilla ChpsHny&Jlpno Chili 150g         3       13.8
6   Smiths Crinkle Chips Salt & Vinegar 330g         1        5.7
7      Grain Waves         Sweet Chilli 210g         1        3.6
8    Doritos Corn Chip Mexican Jalapeno 150g         1        3.9
9      Grain Waves Sour    Cream&Chives 210G         2        7.2
10  Smiths Crinkle Chips Salt & Vinegar 330g         1        5.7
11     Kettle Sensations   Siracha Lime 150g         5       23.0
12                  Twisties Cheese     270g         2        9.2
13          WW Crinkle Cut      Chicken 175g         1        1.7
14            Thins Chips Light&  Tangy 175g         1        3.3
15                        CCs Original 175g         1        2.1
16                        Burger Rings 220g         4        9.2
17    NCC Sour Cream &    Garden Chives 175g         1        3.0
18   Doritos Corn Chip Southern Chicken 150g         1        3.9
19                  Cheezels Cheese Box 125g         1        2.1
20         Smiths Crinkle      Original 330g         1        5.7

    PACKAGE_SIZE
0            175
1            175
2            170
3            175
4            150
```

```
6              330
7              210
8              150
9              210
10             330
11             150
12             270
13             175
14             175
15             175
16             220
17             175
18             150
19             125
20             330
```

Now let's rename brands in PROD_NAME column

[46]:
```
transaction['BRAND']=[s.split()[0] for s in transaction['PROD_NAME']]
```

[47]:
```
# transaction['BRAND'].replace('Dorito','Doritos',inplace=True)
transaction['BRAND'].replace('Infzns','Infuzions',inplace=True)
# transaction['BRAND'].replace('Smith','Smiths',inplace=True)
transaction['BRAND'].replace('Snbts','Sunbites',inplace=True)
transaction['BRAND'].replace('Red','RRD',inplace=True)
transaction['BRAND'].replace('Old','Old El Paso',inplace=True)
transaction['BRAND'].replace('WW','Woolworths',inplace=True)
transaction['BRAND'].replace('Natural','NCC',inplace=True)
```

[48]:
```
#Histogram for brands
# import pandas as pd
# df = pd.DataFrame({'A':[1,1,3,2,6,2,8]})
# a = df['A'].unique()
# print a.sort()
# transaction['BRAND'].plot(kind='hist',x='Brand', y='Packets␣
 ↪sold',title='Popular brands')
transaction.head(20)
```

[48]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	\
0	2018-10-17	1	1000	1	5	
1	2019-05-14	1	1307	348	66	
2	2019-05-20	1	1343	383	61	
3	2018-08-17	2	2373	974	69	
4	2018-08-18	2	2426	1038	108	
6	2019-05-16	4	4149	3333	16	
7	2019-05-16	4	4196	3539	24	
8	2018-08-20	5	5026	4525	42	
9	2018-08-18	7	7150	6900	52	

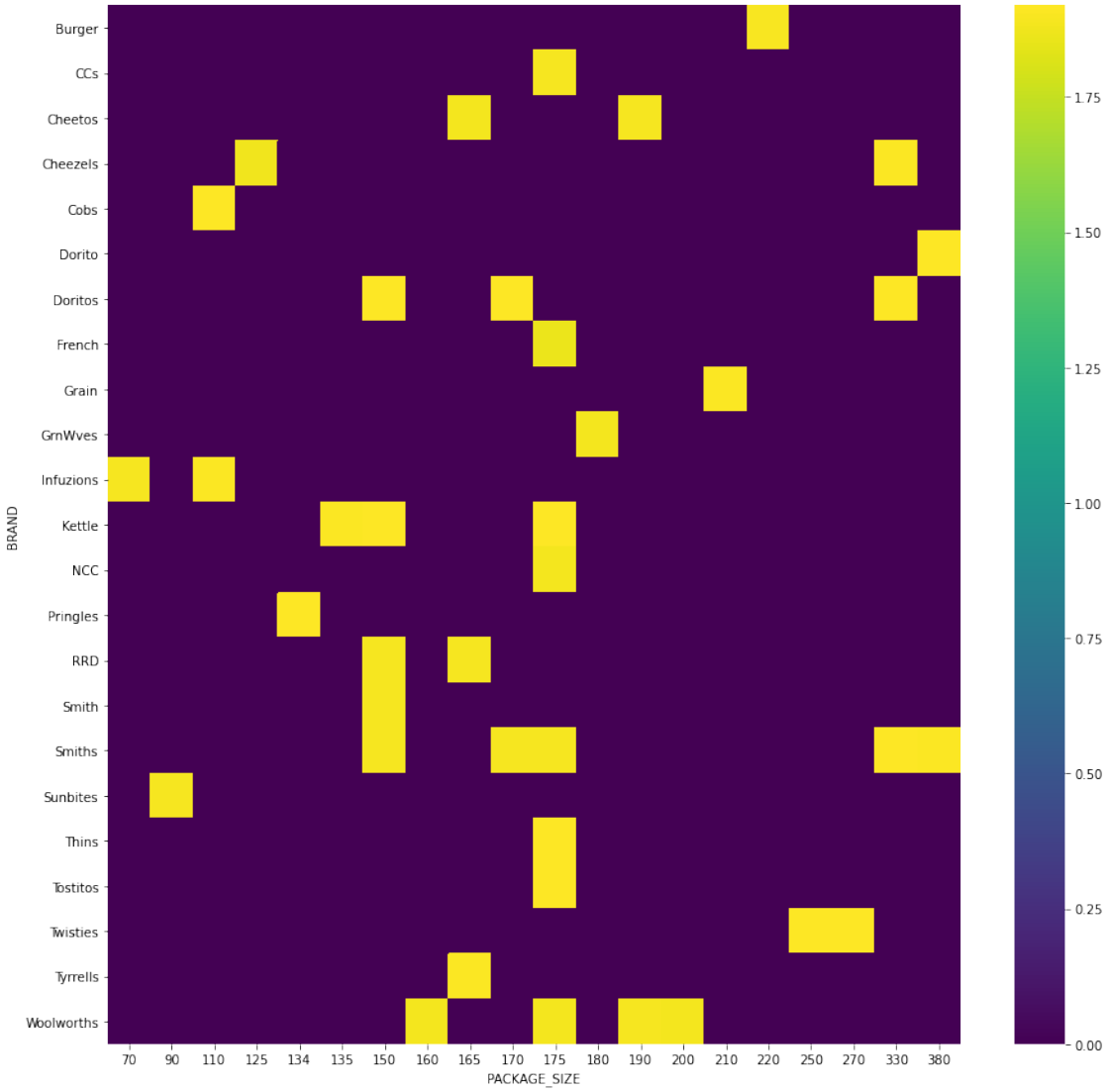
```
10 2019-05-17          7            7215    7176        16
11 2018-08-20          8            8294    8221       114
12 2019-05-18          9            9208    8634        15
13 2018-08-17         13           13213   12447        92
14 2019-05-15         19           19272   16686        44
15 2019-05-19         20           20164   17136        54
16 2018-08-18         20           20418   17413        94
17 2018-08-14         22           22411   18646        98
18 2018-08-17         22           22456   18696        93
19 2019-05-16         23           23067   19162        56
20 2019-05-19         25           25105   21815         7
```

```
                                  PROD_NAME  PROD_QTY  TOT_SALES  \
0     Natural Chip        Compny SeaSalt175g         2        6.0
1                   CCs Nacho Cheese    175g         3        6.3
2     Smiths Crinkle Cut  Chips Chicken 170g         2        2.9
3     Smiths Chip Thinly  S/Cream&Onion 175g         5       15.0
4   Kettle Tortilla ChpsHny&Jlpno Chili 150g         3       13.8
6   Smiths Crinkle Chips Salt & Vinegar 330g         1        5.7
7      Grain Waves         Sweet Chilli 210g         1        3.6
8    Doritos Corn Chip Mexican Jalapeno 150g         1        3.9
9      Grain Waves Sour    Cream&Chives 210G         2        7.2
10  Smiths Crinkle Chips Salt & Vinegar 330g         1        5.7
11     Kettle Sensations   Siracha Lime 150g         5       23.0
12                  Twisties Cheese     270g         2        9.2
13          WW Crinkle Cut      Chicken 175g         1        1.7
14            Thins Chips Light&  Tangy 175g         1        3.3
15                        CCs Original 175g         1        2.1
16                        Burger Rings 220g         4        9.2
17    NCC Sour Cream &    Garden Chives 175g         1        3.0
18   Doritos Corn Chip Southern Chicken 150g         1        3.9
19                  Cheezels Cheese Box 125g         1        2.1
20         Smiths Crinkle      Original 330g         1        5.7

    PACKAGE_SIZE       BRAND
0            175         NCC
1            175         CCs
2            170      Smiths
3            175      Smiths
4            150      Kettle
6            330      Smiths
7            210       Grain
8            150     Doritos
9            210       Grain
10           330      Smiths
11           150      Kettle
12           270    Twisties
```

```
13           175  Woolworths
14           175        Thins
15           175          CCs
16           220       Burger
17           175          NCC
18           150      Doritos
19           125     Cheezels
20           330       Smiths
```

[49]:
```
# Heatmap showing packet quantity

plt.figure(figsize=(15,15))
sns.heatmap(pd.pivot_table(data=transaction
                           , index='BRAND'
                           , columns='PACKAGE_SIZE',values='PROD_QTY').fillna(0)
                           , cmap='viridis')
```

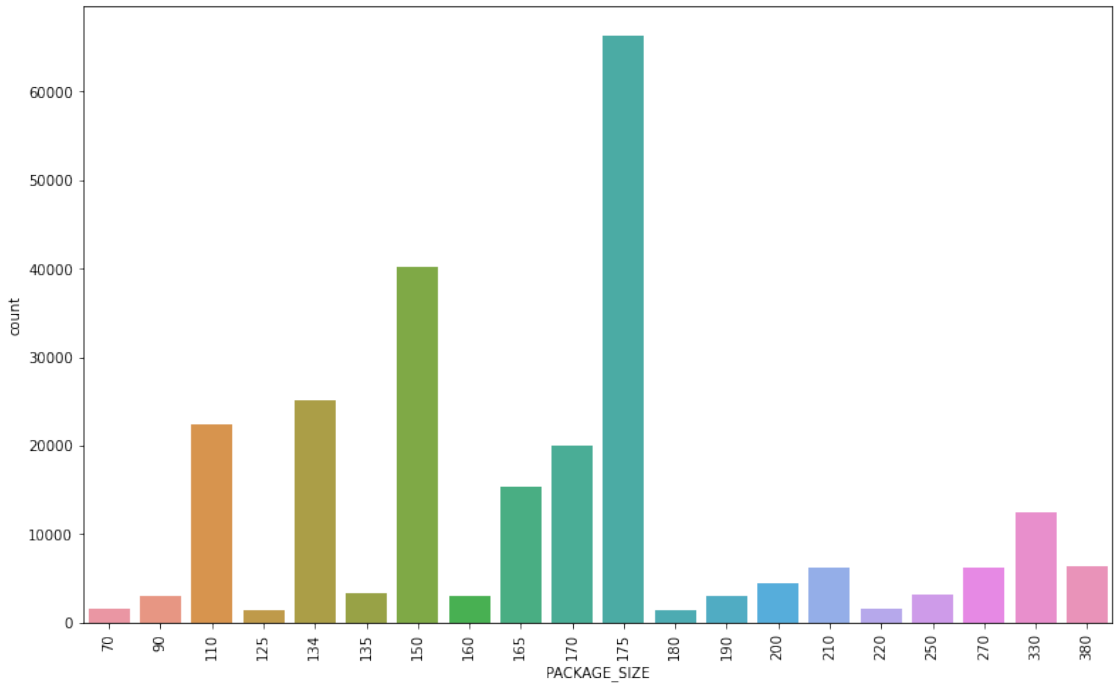
[49]:
```
<AxesSubplot:xlabel='PACKAGE_SIZE', ylabel='BRAND'>
```



[51]:
```
import warnings
warnings.filterwarnings("ignore")
```

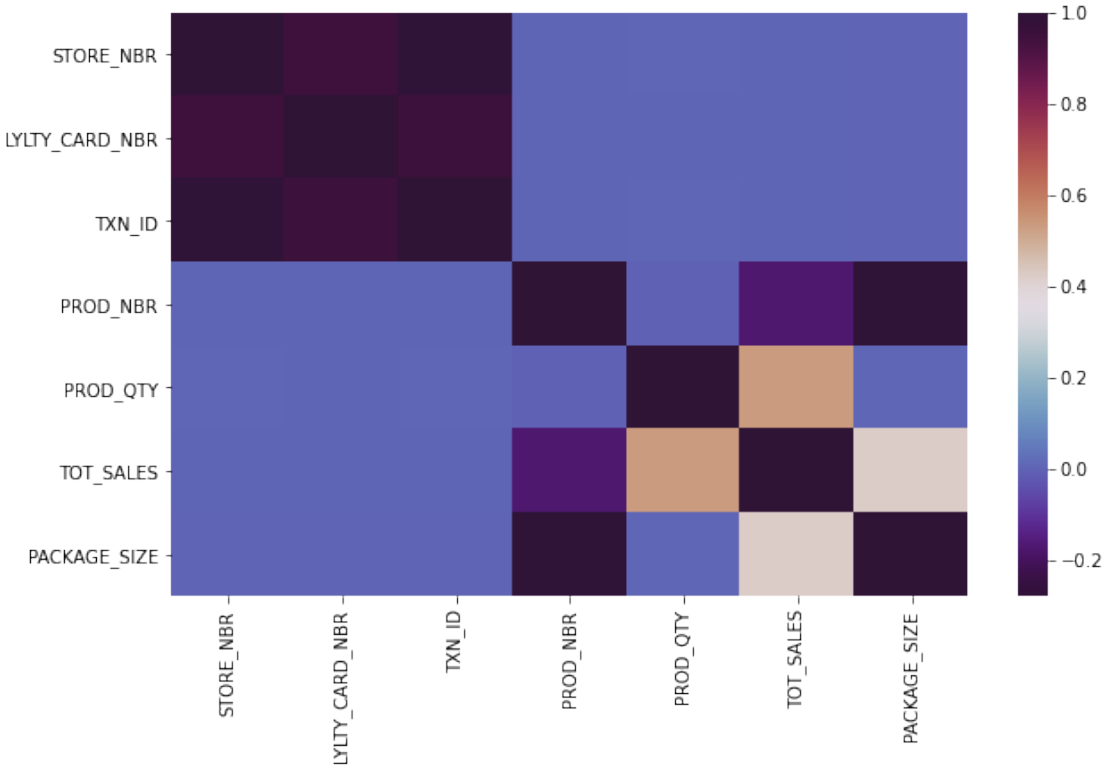
[52]:
```
#histogram of packet size
plt.figure(figsize=(13,8))
plt.xticks(rotation=90)
sns.countplot(transaction['PACKAGE_SIZE'])
```

[52]: <AxesSubplot:xlabel='PACKAGE_SIZE', ylabel='count'>



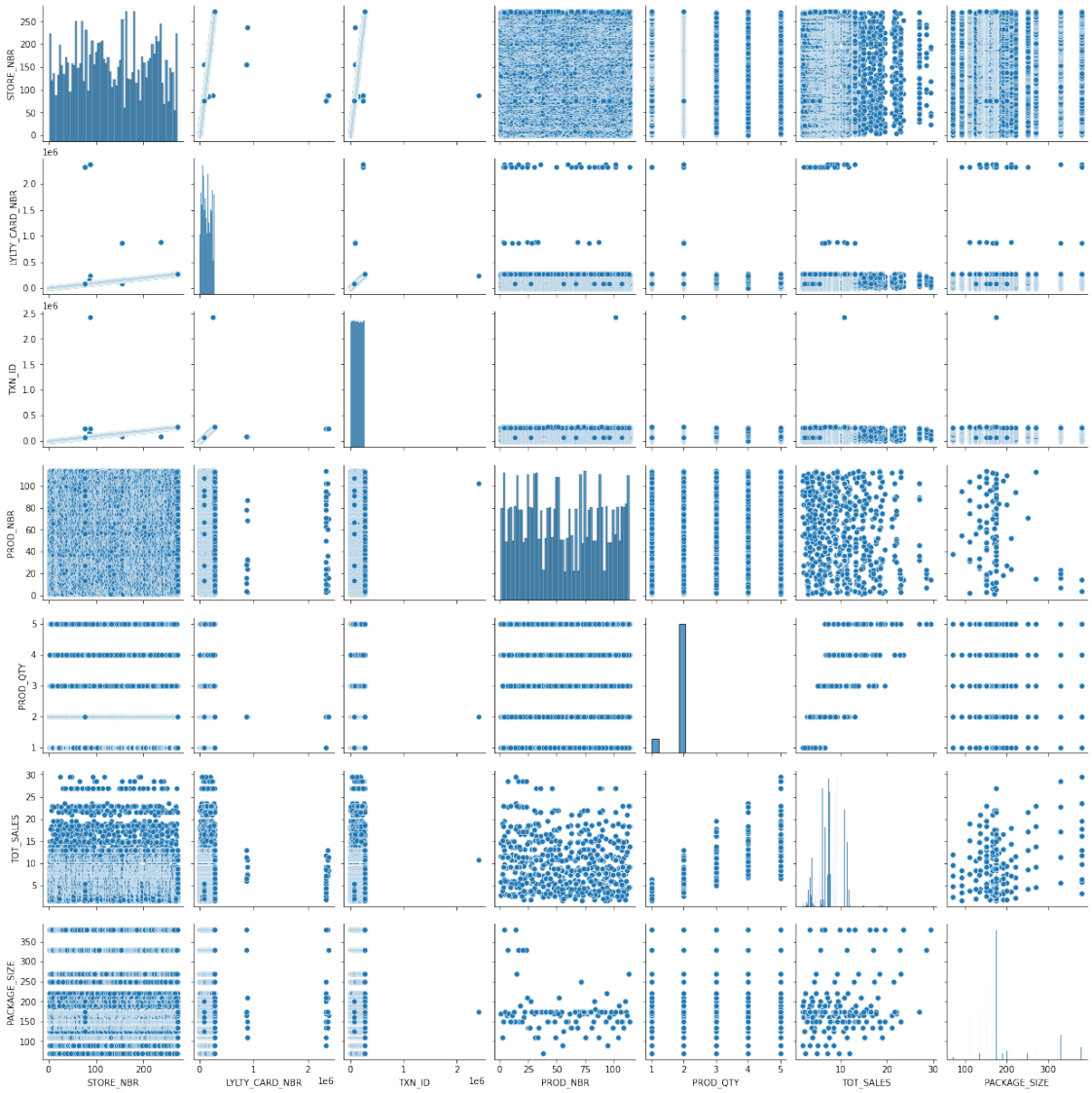
[53]:
```
#correlation heatmap
plt.figure(figsize=(10,6))
sns.heatmap(transaction.corr(),cmap='twilight_shifted')
```

[53]: <AxesSubplot:>



[54]:
```
#pairplot
sns.pairplot(data=transaction[transaction.columns.drop('PROD_NAME')])
```

[54]: <seaborn.axisgrid.PairGrid at 0x14db3940>



0.5 Purchase Data

[57]:
```
purchase.head()
```

[57]:

	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

[58]:
```
purchase.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
```

[60]:
```
purchase.describe(include='all')
```

[60]:

	LYLTY_CARD_NBR	LIFESTAGE	PREMIUM_CUSTOMER
count	7.263700e+04	72637	72637
unique	NaN	7	3
top	NaN	RETIREES	Mainstream
freq	NaN	14805	29245
mean	1.361859e+05	NaN	NaN
std	8.989293e+04	NaN	NaN
min	1.000000e+03	NaN	NaN
25%	6.620200e+04	NaN	NaN
50%	1.340400e+05	NaN	NaN
75%	2.033750e+05	NaN	NaN
max	2.373711e+06	NaN	NaN

we will plot the lifestage content in a graph but since it's not numerical so we can't show it so our approach will be

- check the unique values contained in the column
- map those values into numbers

[63]:
```
purchase['LIFESTAGE'].value_counts()
```

[63]:
```
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
```

[67]:
```
## next we will use encoding to encode those values into numbers
# replace_map = { 'LIFESTAGE':{'RETIREES' : 0, 'OLDER SINGLES/COUPLES': 1,␣
 ↪'YOUNG SINGLES/COUPLES': 2, 'OLDER FAMILIES': 3
#                             , 'YOUNG FAMILIES': 4, 'MIDAGE SINGLES/COUPLES':␣
 ↪5, 'NEW FAMILIES': 6}}
```

```
#this is a better approach no need to manual change the categories :)
labels = purchase['LIFESTAGE'].astype('category').cat.categories.tolist()
replace_map_comp = {'LIFESTAGE' : {k: v for k,v in␣
 ↪zip(labels,list(range(0,len(labels)+1)))}}

print(replace_map_comp)
```

```
{'LIFESTAGE': {'MIDAGE SINGLES/COUPLES': 0, 'NEW FAMILIES': 1, 'OLDER FAMILIES':
2, 'OLDER SINGLES/COUPLES': 3, 'RETIREES': 4, 'YOUNG FAMILIES': 5, 'YOUNG
SINGLES/COUPLES': 6}}
```

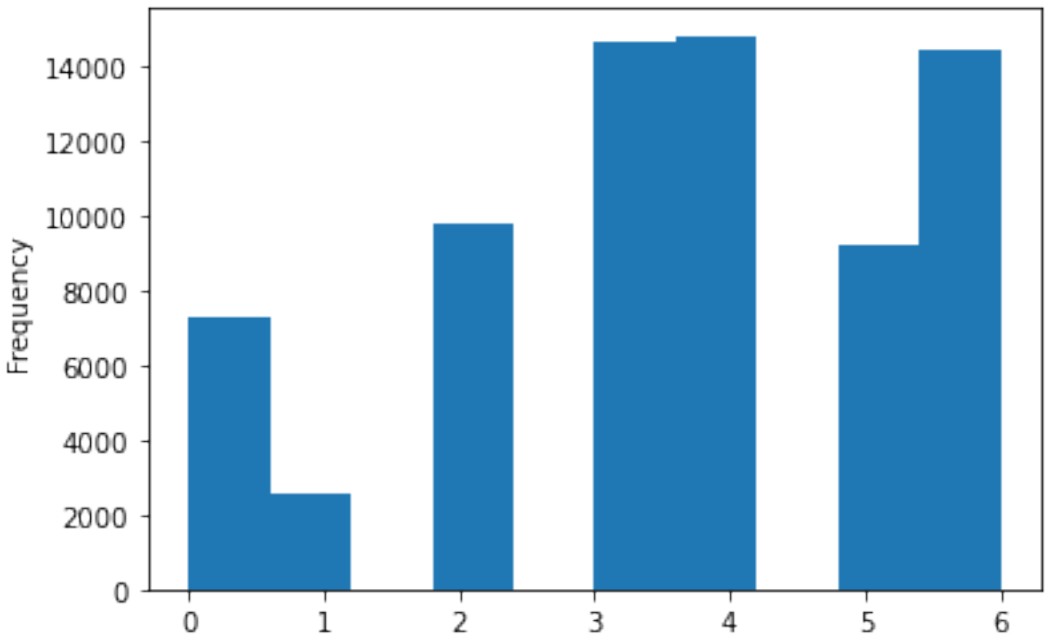
[68]:
```
purchase_replace = purchase.copy()
```

[69]:
```
## now let's pass the replacement elements to our df

purchase_replace.replace(replace_map_comp, inplace=True)
print(purchase_replace.head())
```

```
   LYLTY_CARD_NBR  LIFESTAGE PREMIUM_CUSTOMER
0            1000          6          Premium
1            1002          6       Mainstream
2            1003          5           Budget
3            1004          3       Mainstream
4            1005          0       Mainstream
```

[77]:
```
#lifestage distribution among customers
purchase_replace['LIFESTAGE'].plot(kind='hist')
```

[77]: <AxesSubplot:ylabel='Frequency'>



[72]:
```
## Next we will do the same with PREMIUM_CUSTOMER Variable
```

[75]:
```
PREM_CUST_labels = purchase_replace['PREMIUM_CUSTOMER'].astype('category').cat.
 ↪categories.tolist()
PREM_CUST = {'PREMIUM_CUSTOMER' : {k: v for k,v in␣
 ↪zip(PREM_CUST_labels,list(range(0,len(labels)+1)))}}

print(PREM_CUST)
```

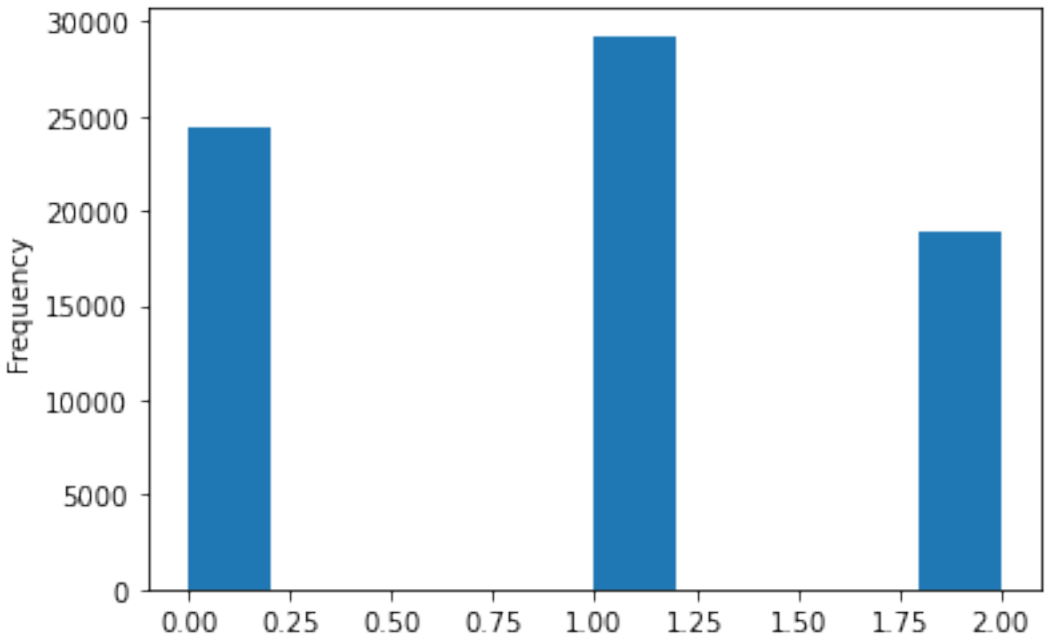
```
{'PREMIUM_CUSTOMER': {'Budget': 0, 'Mainstream': 1, 'Premium': 2}}
```

[76]:
```
## now let's pass the replacement elements to our df

purchase_replace.replace(PREM_CUST, inplace=True)
print(purchase_replace.head())
```

```
   LYLTY_CARD_NBR  LIFESTAGE  PREMIUM_CUSTOMER
0            1000          6                 2
1            1002          6                 1
2            1003          5                 0
3            1004          3                 1
4            1005          0                 1
```

[78]:
```
purchase_replace['PREMIUM_CUSTOMER'].plot(kind='hist')
```

[78]:
```
<AxesSubplot:ylabel='Frequency'>
```



0.6 Joining Datasets

[89]:
```
final_df=pd.merge(transaction,purchase,on='LYLTY_CARD_NBR')
final_df.head(10)
```

[89]:

	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	2018-11-10	1	1307	346	96	WW Original Stacked Chips 160g	2	3.8
3	2019-03-09	1	1307	347	54	CCs Original 175g	1	2.1
4	2019-05-20	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9
5	2018-08-17	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	15.0
6	2018-08-18	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	13.8
7	2019-05-16	4	4149	3333	16			
8	2018-07-06	4	4149	3330	46			
9	2018-07-18	4	4149	3331	112			

```
7  Smiths Crinkle Chips Salt & Vinegar 330g         1        5.7
8                      Kettle Original 175g         2       10.8
9    Tyrrells Crisps     Ched & Chives 165g         2        8.4

   PACKAGE_SIZE       BRAND              LIFESTAGE PREMIUM_CUSTOMER
0           175         NCC   YOUNG SINGLES/COUPLES          Premium
1           175         CCs  MIDAGE SINGLES/COUPLES           Budget
2           160  Woolworths  MIDAGE SINGLES/COUPLES           Budget
3           175         CCs  MIDAGE SINGLES/COUPLES           Budget
4           170      Smiths  MIDAGE SINGLES/COUPLES           Budget
5           175      Smiths  MIDAGE SINGLES/COUPLES           Budget
6           150      Kettle  MIDAGE SINGLES/COUPLES           Budget
7           330      Smiths  MIDAGE SINGLES/COUPLES           Budget
8           175      Kettle  MIDAGE SINGLES/COUPLES           Budget
9           165    Tyrrells  MIDAGE SINGLES/COUPLES           Budget
```

[90]:
```
final_df.isna().sum()
```

[90]:
```
DATE              0
STORE_NBR         0
LYLTY_CARD_NBR    0
TXN_ID            0
PROD_NBR          0
PROD_NAME         0
PROD_QTY          0
TOT_SALES         0
PACKAGE_SIZE      0
BRAND             0
LIFESTAGE         0
PREMIUM_CUSTOMER  0
dtype: int64
```

[83]:
```
final_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 246740 entries, 0 to 246739
Data columns (total 12 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   DATE              246740 non-null  datetime64[ns]
 1   STORE_NBR         246740 non-null  int64
 2   LYLTY_CARD_NBR    246740 non-null  int64
 3   TXN_ID            246740 non-null  int64
 4   PROD_NBR          246740 non-null  int64
 5   PROD_NAME         246740 non-null  object
 6   PROD_QTY          246740 non-null  int64
 7   TOT_SALES         246740 non-null  float64
 8   PACKAGE_SIZE      246740 non-null  int64
```

```
 9   BRAND             246740 non-null  object
 10  LIFESTAGE         246740 non-null  int64
 11  PREMIUM_CUSTOMER  246740 non-null  int64
dtypes: datetime64[ns](1), float64(1), int64(8), object(2)
memory usage: 24.5+ MB
```

[91]:
```
final_df.describe(include='all')
```

[91]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID
count	246740	246740.000000	2.467400e+05	2.467400e+05
unique	364	NaN	NaN	NaN
top	2018-12-24 00:00:00	NaN	NaN	NaN
freq	865	NaN	NaN	NaN
first	2018-07-01 00:00:00	NaN	NaN	NaN
last	2019-06-30 00:00:00	NaN	NaN	NaN
mean	NaN	135.050361	1.355303e+05	1.351304e+05
std	NaN	76.786971	8.071520e+04	7.814760e+04
min	NaN	1.000000	1.000000e+03	1.000000e+00
25%	NaN	70.000000	7.001500e+04	6.756875e+04
50%	NaN	130.000000	1.303670e+05	1.351815e+05
75%	NaN	203.000000	2.030832e+05	2.026522e+05
max	NaN	272.000000	2.373711e+06	2.415841e+06

	PROD_NBR	PROD_NAME	PROD_QTY
count	246740.000000	246740	246740.000000
unique	NaN	105	NaN
top	NaN	Kettle Mozzarella Basil & Pesto 175g	NaN
freq	NaN	3304	NaN
first	NaN	NaN	NaN
last	NaN	NaN	NaN
mean	56.352213	NaN	1.906456
std	33.695235	NaN	0.342499
min	1.000000	NaN	1.000000
25%	26.000000	NaN	2.000000
50%	53.000000	NaN	2.000000
75%	87.000000	NaN	2.000000
max	114.000000	NaN	5.000000

	TOT_SALES	PACKAGE_SIZE	BRAND	LIFESTAGE
count	246740.000000	246740.000000	246740	246740
unique	NaN	NaN	23	7
top	NaN	NaN	Kettle	OLDER SINGLES/COUPLES
freq	NaN	NaN	41288	50793
first	NaN	NaN	NaN	NaN
last	NaN	NaN	NaN	NaN
mean	7.316113	175.583521	NaN	NaN
std	2.474897	59.432118	NaN	NaN

```
min          1.700000      70.000000     NaN                    NaN
25%          5.800000     150.000000     NaN                    NaN
50%          7.400000     170.000000     NaN                    NaN
75%          8.800000     175.000000     NaN                    NaN
max         29.500000     380.000000     NaN                    NaN
```

```
       PREMIUM_CUSTOMER
count            246740
unique                3
top          Mainstream
freq              95043
first               NaN
last                NaN
mean                NaN
std                 NaN
min                 NaN
25%                 NaN
50%                 NaN
75%                 NaN
max                 NaN
```

[92]:
```
final_df.to_csv('Finals.csv')
```

[93]:
```
final_df[['TOT_SALES','PREMIUM_CUSTOMER']].groupby('PREMIUM_CUSTOMER').sum().
↪sort_values('TOT_SALES',ascending=False)
```

[93]:
```
                   TOT_SALES
PREMIUM_CUSTOMER
Mainstream        700865.40
Budget            631406.85
Premium           472905.45
```

In the first place of our premuim customers is **mainstream** with **700865.40** of total sales

In the second place of our premium customers is **Budget** with **631406.85** of total sales

In the third place of our premium customers is **Premium** with **472905.45** of total sales

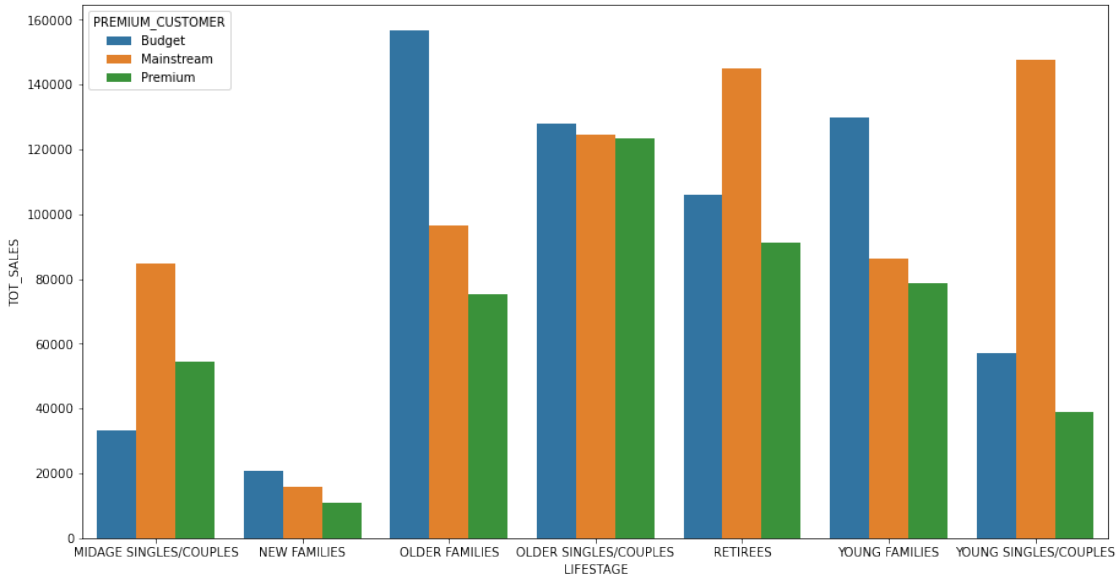
[94]:
```
## Next we will see who spends most in LIFESTAGE and their Premium behaviour
a=final_df[['LIFESTAGE','PREMIUM_CUSTOMER','TOT_SALES']].
↪groupby(['PREMIUM_CUSTOMER','LIFESTAGE']).sum()
a.sort_values('TOT_SALES',ascending=False)
```

[94]:
```
                                        TOT_SALES
PREMIUM_CUSTOMER LIFESTAGE
Budget           OLDER FAMILIES          156863.75
Mainstream       YOUNG SINGLES/COUPLES   147582.20
                 RETIREES                145168.95
Budget           YOUNG FAMILIES          129717.95
```

```
                    OLDER SINGLES/COUPLES   127833.60
Mainstream          OLDER SINGLES/COUPLES   124648.50
Premium             OLDER SINGLES/COUPLES   123537.55
Budget              RETIREES                105916.30
Mainstream          OLDER FAMILIES           96413.55
Premium             RETIREES                 91296.65
Mainstream          YOUNG FAMILIES           86338.25
                    MIDAGE SINGLES/COUPLES   84734.25
Premium             YOUNG FAMILIES           78571.70
                    OLDER FAMILIES           75242.60
Budget              YOUNG SINGLES/COUPLES    57122.10
Premium             MIDAGE SINGLES/COUPLES   54443.85
                    YOUNG SINGLES/COUPLES    39052.30
Budget              MIDAGE SINGLES/COUPLES   33345.70
                    NEW FAMILIES             20607.45
Mainstream          NEW FAMILIES             15979.70
Premium             NEW FAMILIES             10760.80
```

[95]:
```
plt.figure(figsize=(15,8))
sns.barplot(y=a.reset_index()['TOT_SALES'],x=a.reset_index()['LIFESTAGE'],hue=a.
 ↪reset_index()['PREMIUM_CUSTOMER'])
```

[95]:
```
<AxesSubplot:xlabel='LIFESTAGE', ylabel='TOT_SALES'>
```



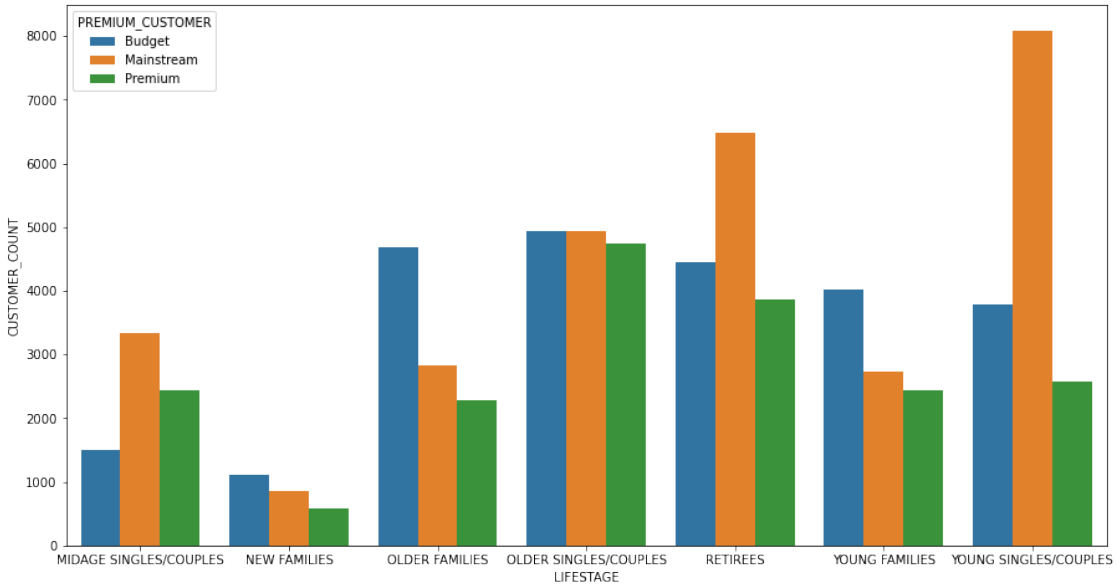
[101]:
```
# Next we will answer this question : how many customers are in each segment
b=purchase.groupby(['PREMIUM_CUSTOMER','LIFESTAGE']).count()
b.columns=['CUSTOMER_COUNT']
b.sort_values('CUSTOMER_COUNT',ascending=False)
```

[101]:

PREMIUM_CUSTOMER	LIFESTAGE	CUSTOMER_COUNT
Mainstream	YOUNG SINGLES/COUPLES	8088
	RETIREES	6479
	OLDER SINGLES/COUPLES	4930
Budget	OLDER SINGLES/COUPLES	4929
Premium	OLDER SINGLES/COUPLES	4750
Budget	OLDER FAMILIES	4675
	RETIREES	4454
	YOUNG FAMILIES	4017
Premium	RETIREES	3872
Budget	YOUNG SINGLES/COUPLES	3779
Mainstream	MIDAGE SINGLES/COUPLES	3340
	OLDER FAMILIES	2831
	YOUNG FAMILIES	2728
Premium	YOUNG SINGLES/COUPLES	2574
	YOUNG FAMILIES	2433
	MIDAGE SINGLES/COUPLES	2431
	OLDER FAMILIES	2274
Budget	MIDAGE SINGLES/COUPLES	1504
	NEW FAMILIES	1112
Mainstream	NEW FAMILIES	849
Premium	NEW FAMILIES	588

[107]:
```
plt.figure(figsize=(15,8))
sns.barplot(y=b.reset_index()['CUSTOMER_COUNT'],x=b.
 ↪reset_index()['LIFESTAGE'],hue=b.reset_index()['PREMIUM_CUSTOMER'])
```

[107]: <AxesSubplot:xlabel='LIFESTAGE', ylabel='CUSTOMER_COUNT'>



We will repeat this with sold chips

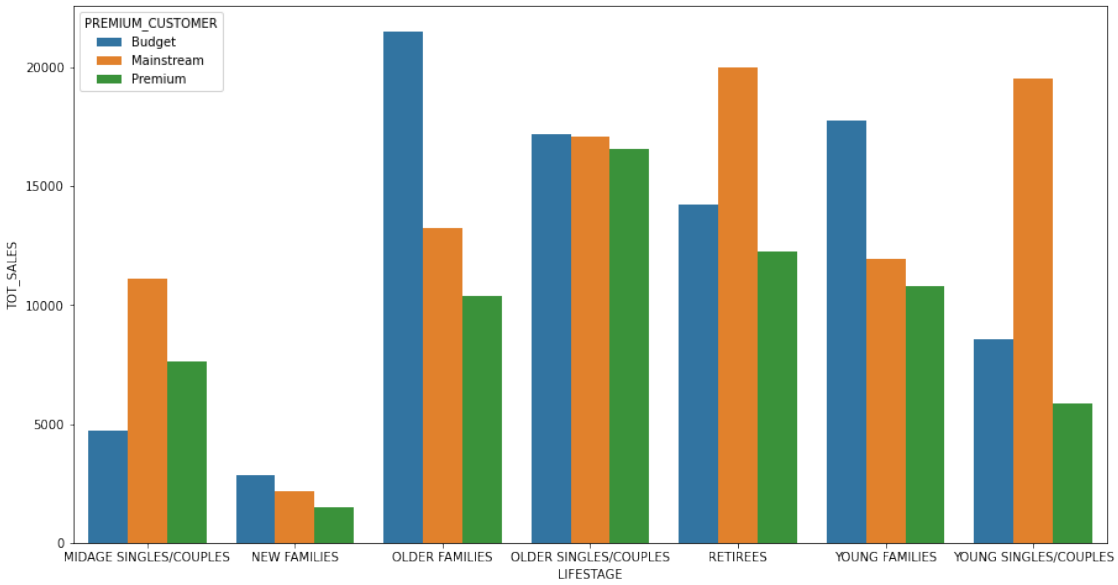
[108]:
```
#How many chips are bought per customer by segment
c=final_df[['LIFESTAGE','PREMIUM_CUSTOMER','TOT_SALES']].
 ↪groupby(['LIFESTAGE','PREMIUM_CUSTOMER']).count()
c.sort_values('TOT_SALES',ascending=False).head(5)
```

[108]:

		TOT_SALES
LIFESTAGE	PREMIUM_CUSTOMER	
OLDER FAMILIES	Budget	21514
RETIREES	Mainstream	19970
YOUNG SINGLES/COUPLES	Mainstream	19544
YOUNG FAMILIES	Budget	17763
OLDER SINGLES/COUPLES	Budget	17172

[109]:
```
plt.figure(figsize=(15,8))
sns.barplot(y=c.reset_index()['TOT_SALES'],x=c.reset_index()['LIFESTAGE'],hue=c.
 ↪reset_index()['PREMIUM_CUSTOMER'])
```

[109]: <AxesSubplot:xlabel='LIFESTAGE', ylabel='TOT_SALES'>



[110]:
```
# here we will understand what proportion of their grocery spend is on chips

transaction_a = pd.read_csv("QVI_transaction_data.csv")
totsalespercust=transaction_a[['LYLTY_CARD_NBR','TOT_SALES']].
 ↪groupby(['LYLTY_CARD_NBR']).sum().reset_index()
```

```
ratio=final_df[['LYLTY_CARD_NBR','TOT_SALES']].
 ↪merge(totsalespercust,on='LYLTY_CARD_NBR').rename(columns={'TOT_SALES_x':
 ↪'TRAN_SALE','TOT_SALES_y':'CUST_TOT_SALE'})
ratio['RATIO']=ratio['TRAN_SALE']/ratio['CUST_TOT_SALE']
ratio.sort_values('RATIO')
```

[110]:

	LYLTY_CARD_NBR	TRAN_SALE	CUST_TOT_SALE	RATIO
174208	152094	1.9	112.1	0.016949
75460	48155	1.9	100.7	0.018868
174557	168140	1.7	86.5	0.019653
16284	104061	1.7	85.9	0.019790
30772	55244	1.7	85.7	0.019837
…	…	…	…	…
163956	49312	11.4	11.4	1.000000
163855	47486	7.4	7.4	1.000000
163852	47465	10.8	10.8	1.000000
162683	12139	8.6	8.6	1.000000
246739	272380	8.8	8.8	1.000000

[246740 rows x 4 columns]

[111]:
```
## we will do the same for chips

e=final_df[['LIFESTAGE','PREMIUM_CUSTOMER','TOT_SALES']].
 ↪groupby(['PREMIUM_CUSTOMER','LIFESTAGE']).count()
e["TOT_SALES"]/(e['TOT_SALES'].sum())
```

[111]:

PREMIUM_CUSTOMER	LIFESTAGE	
Budget	MIDAGE SINGLES/COUPLES	0.019012
	NEW FAMILIES	0.011445
	OLDER FAMILIES	0.087193
	OLDER SINGLES/COUPLES	0.069596
	RETIREES	0.057652
	YOUNG FAMILIES	0.071991
	YOUNG SINGLES/COUPLES	0.034745
Mainstream	MIDAGE SINGLES/COUPLES	0.044966
	NEW FAMILIES	0.008855
	OLDER FAMILIES	0.053664
	OLDER SINGLES/COUPLES	0.069146
	RETIREES	0.080935
	YOUNG FAMILIES	0.048419
	YOUNG SINGLES/COUPLES	0.079209
Premium	MIDAGE SINGLES/COUPLES	0.030850
	NEW FAMILIES	0.006031
	OLDER FAMILIES	0.042162
	OLDER SINGLES/COUPLES	0.067115
	RETIREES	0.049591

```
                    YOUNG FAMILIES            0.043706
                    YOUNG SINGLES/COUPLES     0.023717
Name: TOT_SALES, dtype: float64
```

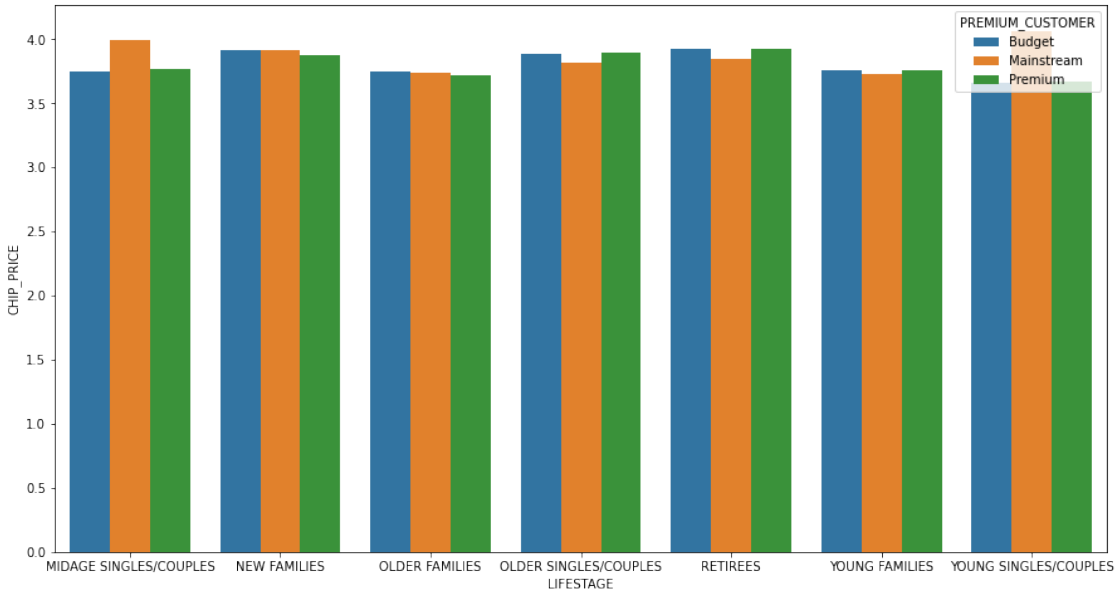
[112]:
```
# What's the average chip price by customer segment
final_df['CHIP_PRICE']=final_df['TOT_SALES']/final_df['PROD_QTY']
d=final_df[['LIFESTAGE','PREMIUM_CUSTOMER','CHIP_PRICE']].
 ↪groupby(['PREMIUM_CUSTOMER','LIFESTAGE']).mean()
d.sort_values("CHIP_PRICE",ascending=False)
```

[112]:

PREMIUM_CUSTOMER	LIFESTAGE	CHIP_PRICE
Mainstream	YOUNG SINGLES/COUPLES	4.065642
	MIDAGE SINGLES/COUPLES	3.994241
Budget	RETIREES	3.924404
Premium	RETIREES	3.920942
Budget	NEW FAMILIES	3.917688
Mainstream	NEW FAMILIES	3.916133
Premium	OLDER SINGLES/COUPLES	3.893182
Budget	OLDER SINGLES/COUPLES	3.882096
Premium	NEW FAMILIES	3.872110
Mainstream	RETIREES	3.844294
	OLDER SINGLES/COUPLES	3.814665
Premium	MIDAGE SINGLES/COUPLES	3.770698
	YOUNG FAMILIES	3.762150
Budget	YOUNG FAMILIES	3.760737
	OLDER FAMILIES	3.745340
	MIDAGE SINGLES/COUPLES	3.743328
Mainstream	OLDER FAMILIES	3.737077
	YOUNG FAMILIES	3.724533
Premium	OLDER FAMILIES	3.717000
	YOUNG SINGLES/COUPLES	3.665414
Budget	YOUNG SINGLES/COUPLES	3.657366

[113]:
```
plt.figure(figsize=(15,8))
sns.barplot(y=d.reset_index()['CHIP_PRICE'],x=d.
 ↪reset_index()['LIFESTAGE'],hue=d.reset_index()['PREMIUM_CUSTOMER'])
```

[113]:
```
<AxesSubplot:xlabel='LIFESTAGE', ylabel='CHIP_PRICE'>
```



[114]:
```
from scipy import stats
#Mainstream vs premium
stats.ttest_ind([4.065642,3.994241],[3.770698,3.665414])
```

[114]:
```
Ttest_indResult(statistic=4.903408005498769, pvalue=0.039164352682153285)
```

[115]:
```
#Mainstream vs budget
stats.ttest_ind([4.065642,3.994241],[3.657366,3.743328])
```

[115]:
```
Ttest_indResult(statistic=5.898899732826305, pvalue=0.027555775534860754)
```

The t-test results in a p-value of 0.04 and 0.03 , i.e. the unit price for mainstream, young and mid-age singles and couples are **significantly higher** than that of budget or premium, young and midage singles and couples.

Now we are focussing on the mainstream, young and mid-age singles and couples brands that these two customer segments prefer more than others

[120]:
```
midage=final_df[(final_df['PREMIUM_CUSTOMER']=='Mainstream') &␣
 ↪(final_df['LIFESTAGE']=='MIDAGE SINGLES/COUPLES')]
print(f"MIDAGE SINGLES/COUPLES\n{midage['BRAND'].value_counts().head(5)}")
```

```
MIDAGE SINGLES/COUPLES
Kettle       2136
Smiths       1176
Pringles     1159
Doritos      1072
Infuzions     679
Name: BRAND, dtype: int64
```

[121]:
```
young=final_df[(final_df['PREMIUM_CUSTOMER']=='Mainstream') &␣
 ↪(final_df['LIFESTAGE']=='YOUNG SINGLES/COUPLES')]
print(f"YOUNG SINGLES/COUPLES\n{young['BRAND'].value_counts().head(5)}")
```

```
YOUNG SINGLES/COUPLES
Kettle       3844
Pringles     2315
Doritos      2076
Smiths       1790
Infuzions    1250
Name: BRAND, dtype: int64
```

As we can see **kettle** is the most popular between those who are mid-age and young and came in the first place.

After that we can see **Smiths** for mid-age and **pringles** for young

At third place **pringles** for mid-age and **Doritos** for young

At fourth place **Doritos** for mid-age and **Smiths** for young

At fifth place **Infuzions** for mid-age and also for young

We also wanted to know which is the most sold packet size for both mid-age and young so we will also use the previous print syntax but we will change the column name

[123]:
```
print(f"MIDAGE SINGLES/COUPLES\n{midage['PACKAGE_SIZE'].value_counts().
 ↪head(5)}")
```

```
MIDAGE SINGLES/COUPLES
175    2975
150    1777
134    1159
110    1124
170     882
Name: PACKAGE_SIZE, dtype: int64
```

[124]:
```
print(f"YOUNG SINGLES/COUPLES\n{young['PACKAGE_SIZE'].value_counts().head(5)}")
```

```
YOUNG SINGLES/COUPLES
175    4997
150    3080
134    2315
110    2051
170    1575
Name: PACKAGE_SIZE, dtype: int64
```

The most sold packet size is 175g and the least packet size is 170g

So now since we will use recommendation system that depends on **One who buy will also buy** so we will use Association rule

0.7 Association Rule

[128]:
```
from apyori import apriori
```

[152]:
```
dataset = pd.read_csv('Finals.csv')
dataset.shape
```

[152]:
```
(246740, 13)
```

[153]:
```
dataset.head()
```

[153]:
```
   Unnamed: 0        DATE  STORE_NBR  LYLTY_CARD_NBR  TXN_ID  PROD_NBR  \
0           0  2018-10-17          1            1000       1         5
1           1  2019-05-14          1            1307     348        66
2           2  2018-11-10          1            1307     346        96
3           3  2019-03-09          1            1307     347        54
4           4  2019-05-20          1            1343     383        61

                                PROD_NAME  PROD_QTY  TOT_SALES  PACKAGE_SIZE  \
0  Natural Chip        Compny SeaSalt175g         2        6.0           175
1                CCs Nacho Cheese    175g         3        6.3           175
2          WW Original Stacked Chips 160g         2        3.8           160
3                       CCs Original 175g         1        2.1           175
4  Smiths Crinkle Cut  Chips Chicken 170g         2        2.9           170

        BRAND               LIFESTAGE PREMIUM_CUSTOMER
0         NCC   YOUNG SINGLES/COUPLES          Premium
1         CCs  MIDAGE SINGLES/COUPLES           Budget
2  Woolworths  MIDAGE SINGLES/COUPLES           Budget
3         CCs  MIDAGE SINGLES/COUPLES           Budget
4      Smiths  MIDAGE SINGLES/COUPLES           Budget
```

[154]:
```
dataset.drop(['Unnamed: 0'], axis=1)
```

[154]:
```
              DATE  STORE_NBR  LYLTY_CARD_NBR  TXN_ID  PROD_NBR  \
0       2018-10-17          1            1000       1         5
1       2019-05-14          1            1307     348        66
2       2018-11-10          1            1307     346        96
3       2019-03-09          1            1307     347        54
4       2019-05-20          1            1343     383        61
…              …        …              …      …       …
246735  2019-03-09        272          272319  270088        89
246736  2018-08-13        272          272358  270154        74
246737  2018-11-06        272          272379  270187        51
246738  2018-12-27        272          272379  270188        42
246739  2018-09-22        272          272380  270189        74

                                PROD_NAME  PROD_QTY  TOT_SALES  \
```

```
0        Natural Chip        Compny SeaSalt175g         2        6.0
1                      CCs Nacho Cheese    175g         3        6.3
2               WW Original Stacked Chips 160g         2        3.8
3                            CCs Original 175g         1        2.1
4        Smiths Crinkle Cut  Chips Chicken 170g         2        2.9
…                                          …       …         …
246735  Kettle Sweet Chilli And Sour Cream 175g         2       10.8
246736           Tostitos Splash Of  Lime 175g         1        4.4
246737                Doritos Mexicana    170g         2        8.8
246738  Doritos Corn Chip Mexican Jalapeno 150g         2        7.8
246739           Tostitos Splash Of  Lime 175g         2        8.8

        PACKAGE_SIZE       BRAND              LIFESTAGE PREMIUM_CUSTOMER
0                175         NCC   YOUNG SINGLES/COUPLES          Premium
1                175         CCs  MIDAGE SINGLES/COUPLES           Budget
2                160  Woolworths  MIDAGE SINGLES/COUPLES           Budget
3                175         CCs  MIDAGE SINGLES/COUPLES           Budget
4                170      Smiths  MIDAGE SINGLES/COUPLES           Budget
…              …          …                        …              …
246735           175      Kettle   YOUNG SINGLES/COUPLES          Premium
246736           175    Tostitos   YOUNG SINGLES/COUPLES          Premium
246737           170     Doritos   YOUNG SINGLES/COUPLES          Premium
246738           150     Doritos   YOUNG SINGLES/COUPLES          Premium
246739           175    Tostitos   YOUNG SINGLES/COUPLES          Premium

[246740 rows x 12 columns]
```

For this solution we will take only a little bit of our humoungous data

- Even when i take 8674 out of 246k rows this will take hours (old laptops with weak processing issues)

[175]:
```
# transactions1 = []
# for i in range(0, 8674):
#     transactions1.append([str(dataset.values[i,j]) for j in range(0, 12)])
# rules = apriori(transactions = transactions1, min_support = 0.003,␣
 ↪min_confidence = 0.2, min_lift = 3, min_length = 2, max_length = 2)
```

[174]:
```
# results = list(rules)
```

[173]:
```
# results
```

Now let's put the result in well organised pandas dataframe

[172]:
```
# def inspect(results):
#     lhs         = [tuple(result[2][0][0])[0] for result in results]
#     rhs         = [tuple(result[2][0][1])[0] for result in results]
#     supports    = [result[1] for result in results]
#     confidences = [result[2][0][2] for result in results]
```

```
#     lifts       = [result[2][0][3] for result in results]
#     return list(zip(lhs, rhs, supports, confidences, lifts))
# resultsinDataFrame = pd.DataFrame(inspect(results), columns = ['Left Hand␣
 ↪Side', 'Right Hand Side', 'Support', 'Confidence', 'Lift'])
```

[167]:
```
## Now let's display the result
```

[171]:
```
# resultsinDataFrame
```

[170]:
```
# resultsinDataFrame.nlargest(n = 10, columns = 'Lift')
```

0.8 Another most clear solution

[143]:
```
from mlxtend.frequent_patterns import apriori, association_rules,fpgrowth
```

[144]:
```
basket=final_df.groupby(['LYLTY_CARD_NBR','BRAND'])['PROD_QTY'].sum().unstack().
 ↪fillna(0)
basket
```

[144]:

BRAND	Burger	CCs	Cheetos	Cheezels	Cobs	Dorito	Doritos	French	\
LYLTY_CARD_NBR									
1000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1002	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1003	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1004	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1005	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	
…	…	…	…	…	…	…	…	…	
2370651	0.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	
2370701	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2370751	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2370961	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2373711	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

BRAND	Grain	GrnWves	…	Pringles	RRD	Smith	Smiths	Sunbites	\
LYLTY_CARD_NBR			…						
1000	0.0	0.0	…	0.0	0.0	0.0	0.0	0.0	
1002	0.0	0.0	…	0.0	1.0	0.0	0.0	0.0	
1003	1.0	0.0	…	0.0	0.0	0.0	0.0	0.0	
1004	0.0	0.0	…	0.0	0.0	0.0	0.0	0.0	
1005	0.0	0.0	…	0.0	0.0	0.0	0.0	0.0	
…	…	…	…	…	…	…	…	…	
2370651	0.0	0.0	…	0.0	0.0	0.0	0.0	0.0	
2370701	2.0	0.0	…	0.0	0.0	0.0	0.0	0.0	
2370751	0.0	0.0	…	0.0	0.0	0.0	0.0	0.0	
2370961	0.0	0.0	…	0.0	0.0	0.0	0.0	0.0	
2373711	0.0	0.0	…	0.0	0.0	0.0	2.0	0.0	

```
BRAND           Thins  Tostitos  Twisties  Tyrrells  Woolworths
LYLTY_CARD_NBR
1000              0.0       0.0       0.0       0.0         0.0
1002              0.0       0.0       0.0       0.0         0.0
1003              0.0       0.0       0.0       0.0         0.0
1004              0.0       0.0       0.0       0.0         1.0
1005              0.0       0.0       0.0       0.0         0.0
…                 …         …         …         …           …
2370651           0.0       0.0       0.0       0.0         0.0
2370701           0.0       0.0       0.0       0.0         0.0
2370751           0.0       0.0       0.0       0.0         0.0
2370961           0.0       0.0       0.0       2.0         0.0
2373711           0.0       0.0       0.0       0.0         0.0

[71287 rows x 23 columns]
```

Now it's time to make all values 0 or 1

[145]:
```
def reducer(x):
    if x <= 0:
        return 0
    else:
        return 1
basket=basket.applymap(reducer)
basket
```

[145]:
```
BRAND           Burger  CCs  Cheetos  Cheezels  Cobs  Dorito  Doritos  French  \
LYLTY_CARD_NBR
1000                 0    0        0         0     0       0        0       0
1002                 0    0        0         0     0       0        0       0
1003                 0    0        0         0     0       0        0       0
1004                 0    0        0         0     0       0        0       0
1005                 0    0        1         0     0       0        0       0
…                    …    …        …         …     …       …        …       …
2370651              0    0        0         0     0       1        0       0
2370701              0    0        0         0     0       0        0       0
2370751              0    0        0         0     0       0        0       0
2370961              0    0        0         0     0       0        0       0
2373711              0    0        0         0     0       0        0       0

BRAND           Grain  GrnWves  …  Pringles  RRD  Smith  Smiths  Sunbites  \
LYLTY_CARD_NBR                  …
1000                0        0  …         0    0      0       0         0
1002                0        0  …         0    1      0       0         0
1003                1        0  …         0    0      0       0         0
1004                0        0  …         0    0      0       0         0
1005                0        0  …         0    0      0       0         0
…                   …        …  …         …    …      …       …         …
```

```
2370651             0        0  …         0    0      0       0         0
2370701             1        0  …         0    0      0       0         0
2370751             0        0  …         0    0      0       0         0
2370961             0        0  …         0    0      0       0         0
2373711             0        0  …         0    0      0       1         0

BRAND           Thins  Tostitos  Twisties  Tyrrells  Woolworths
LYLTY_CARD_NBR
1000                0         0         0         0           0
1002                0         0         0         0           0
1003                0         0         0         0           0
1004                0         0         0         0           1
1005                0         0         0         0           0
…                   …         …         …         …           …
2370651             0         0         0         0           0
2370701             0         0         0         0           0
2370751             0         0         0         0           0
2370961             0         0         0         1           0
2373711             0         0         0         0           0

[71287 rows x 23 columns]
```

[146]:
```
frequent=apriori(basket,0.1,use_colnames=True)
frequent
```

[146]:
```
     support            itemsets
0   0.125745              (Cobs)
1   0.259809           (Doritos)
2   0.177311         (Infuzions)
3   0.423303            (Kettle)
4   0.289772           (Pringles)
5   0.180103                (RRD)
6   0.293686            (Smiths)
7   0.176624             (Thins)
8   0.122884           (Tostitos)
9   0.122449           (Twisties)
10  0.139661         (Woolworths)
11  0.121972   (Kettle, Doritos)
12  0.135452  (Kettle, Pringles)
13  0.126966    (Kettle, Smiths)
```

[147]:
```
association_rules(frequent,metric='lift',min_threshold=1).
↪sort_values(['support','confidence'],ascending=False)
```

[147]:
```
   antecedents consequents  antecedent support  consequent support   support  \
3  (Pringles)    (Kettle)            0.289772            0.423303  0.135452
2    (Kettle)  (Pringles)            0.423303            0.289772  0.135452
```

```
5    (Smiths)    (Kettle)            0.293686            0.423303  0.126966
4    (Kettle)    (Smiths)            0.423303            0.293686  0.126966
1   (Doritos)    (Kettle)            0.259809            0.423303  0.121972
0    (Kettle)   (Doritos)            0.423303            0.259809  0.121972

   confidence      lift  leverage  conviction
3    0.467444  1.104279  0.012791    1.082886
2    0.319989  1.104279  0.012791    1.044436
5    0.432318  1.021296  0.002647    1.015880
4    0.299940  1.021296  0.002647    1.008934
1    0.469467  1.109057  0.011994    1.087015
0    0.288143  1.109057  0.011994    1.039803
```

So as we can see anyone will buy kettle can be recommended with Doritos, Smiths or Pringles and vice-versa

[ ]: