## Performing EDA on the data provided from Quantium

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)

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

qvi_purchase = pd.read_csv('QVI_purchase_behaviour.csv')
qvi_purchase
```

```
       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
...               ...                     ...              ...
72632         2370651  MIDAGE SINGLES/COUPLES       Mainstream
72633         2370701          YOUNG FAMILIES       Mainstream
72634         2370751          YOUNG FAMILIES          Premium
72635         2370961          OLDER FAMILIES           Budget
72636         2373711   YOUNG SINGLES/COUPLES       Mainstream

[72637 rows x 3 columns]
```

```python
qvi_transaction = pd.read_excel('QVI_transaction_data.xlsx')
qvi_transaction
```

```
         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
...       ...        ...             ...     ...       ...
264831  43533        272          272319  270088        89
264832  43325        272          272358  270154        74
264833  43410        272          272379  270187        51
264834  43461        272          272379  270188        42
264835  43365        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         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
...                                          ...       ...        ...
264831   Kettle Sweet Chilli And Sour Cream 175g         2       10.8
264832             Tostitos Splash Of  Lime 175g         1        4.4
264833                  Doritos Mexicana    170g         2        8.8
264834   Doritos Corn Chip Mexican Jalapeno 150g         2        7.8
264835             Tostitos Splash Of  Lime 175g         2        8.8

[264836 rows x 8 columns]
```

```
# checking for shape
qvi_purchase.shape
```

```
(72637, 3)
```

```
qvi_transaction.shape
```

```
(264836, 8)
```

```
# checking for nulls
qvi_purchase.isnull().sum()
```

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

```
qvi_transaction.isnull().sum()
```

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

```
# merging the two DataFrames

qvi_merged = pd.merge(qvi_purchase, qvi_transaction, on =
'LYLTY_CARD_NBR', how = 'inner')
qvi_merged
```

```
        LYLTY_CARD_NBR              LIFESTAGE PREMIUM_CUSTOMER   DATE
\
0                 1000  YOUNG SINGLES/COUPLES          Premium  43390

1                 1002  YOUNG SINGLES/COUPLES       Mainstream  43359

2                 1003         YOUNG FAMILIES           Budget  43531

3                 1003         YOUNG FAMILIES           Budget  43532

4                 1004  OLDER SINGLES/COUPLES       Mainstream  43406

...                ...                    ...              ...    ...

264831         2370701         YOUNG FAMILIES       Mainstream  43442

264832         2370751         YOUNG FAMILIES          Premium  43374

264833         2370961         OLDER FAMILIES           Budget  43397

264834         2370961         OLDER FAMILIES           Budget  43400

264835         2373711  YOUNG SINGLES/COUPLES       Mainstream  43448


        STORE_NBR  TXN_ID  PROD_NBR
PROD_NAME  \
0               1       1         5    Natural Chip        Compny
SeaSalt175g
1               1       2        58     Red Rock Deli Chikn&Garlic
Aioli 150g
2               1       3        52     Grain Waves Sour
Cream&Chives 210G
3               1       4       106    Natural ChipCo      Hony Soy
Chckn175g
4               1       5        96            WW Original Stacked
Chips 160g
...           ...     ...       ...
...
264831         88  240378        24     Grain Waves         Sweet
Chilli 210g
264832         88  240394        60      Kettle Tortilla
ChpsFeta&Garlic 150g
264833         88  240480        70   Tyrrells Crisps     Lightly
Salted 165g
```

```
264834         88  240481         65  Old El Paso Salsa   Dip Chnky Tom
Ht300g
264835         88  241815         16  Smiths Crinkle Chips Salt &
Vinegar 330g

        PROD_QTY  TOT_SALES
0              2        6.0
1              1        2.7
2              1        3.6
3              1        3.0
4              1        1.9
...          ...        ...
264831         2        7.2
264832         2        9.2
264833         2        8.4
264834         2       10.2
264835         2       11.4

[264836 rows x 10 columns]
```

```
# checking for data types in qvi_merged
qvi_merged.dtypes
```

```
LYLTY_CARD_NBR        int64
LIFESTAGE            object
PREMIUM_CUSTOMER     object
DATE                  int64
STORE_NBR             int64
TXN_ID                int64
PROD_NBR              int64
PROD_NAME            object
PROD_QTY              int64
TOT_SALES           float64
dtype: object
```

### PROD_NAME

```
qvi_merged['PROD_NAME'].unique()
```

```
array(['Natural Chip        Compny SeaSalt175g',
       'Red Rock Deli Chikn&Garlic Aioli 150g',
       'Grain Waves Sour    Cream&Chives 210G',
       'Natural ChipCo      Hony Soy Chckn175g',
       'WW Original Stacked Chips 160g', 'Cheetos Puffs 165g',
       'Infuzions SourCream&Herbs Veg Strws 110g',
       'RRD SR Slow Rst     Pork Belly 150g',
       'Doritos Cheese      Supreme 330g', 'Doritos Mexicana    170g',
       'Old El Paso Salsa   Dip Tomato Med 300g',
       'GrnWves Plus Btroot & Chilli Jam 180g',
       'Smiths Crinkle Cut  Chips Barbecue 170g',
       'Kettle Sensations   Camembert & Fig 150g',
       'Doritos Corn Chip Southern Chicken 150g',
       'CCs Tasty Cheese    175g', 'Tostitos Splash Of  Lime 175g',
       'Kettle 135g Swt Pot Sea Salt', 'RRD Salt & Vinegar  165g',
       'Infuzions Mango     Chutny Papadums 70g',
       'Smiths Crinkle Cut  Snag&Sauce 150g',
       'Smiths Crinkle      Original 330g',
       'RRD Sweet Chilli &  Sour Cream 165g',
       'Smiths Chip Thinly  S/Cream&Onion 175g',
       'Smiths Crinkle Chips Salt & Vinegar 330g',
       'Red Rock Deli SR    Salsa & Mzzrlla 150g',
       'Cobs Popd Sea Salt  Chips 110g',
       'Natural ChipCo Sea  Salt & Vinegr 175g',
       'Natural Chip Co     Tmato Hrb&Spce 175g', 'Burger Rings 220g',
       'Woolworths Cheese   Rings 190g',
       'Smiths Thinly       Swt Chli&S/Cream175G',
       'Thins Chips Seasonedchicken 175g',
       'Smiths Thinly Cut   Roast Chicken 175g',
       'Tyrrells Crisps     Ched & Chives 165g',
       'Doritos Corn Chips  Cheese Supreme 170g',
       'Smiths Chip Thinly  Cut Original 175g',
       'Smiths Crinkle Cut  Chips Original 170g',
       'Thins Chips Light&  Tangy 175g',
       'Doritos Corn Chips  Original 170g',
       'Kettle Sensations   Siracha Lime 150g',
       'Smiths Crinkle Cut  Salt & Vinegar 170g',
       'Smith Crinkle Cut   Bolognese 150g', 'Cheezels Cheese 330g',
       'Kettle Chilli 175g', 'Tyrrells Crisps     Lightly Salted
165g',
       'Twisties Cheese     270g', 'WW Crinkle Cut      Chicken 175g',
       'RRD Chilli&         Coconut 150g',
       'Infuzions BBQ Rib   Prawn Crackers 110g',
       'Sunbites Whlegrn    Crisps Frch/Onin 90g',
       'Doritos Salsa       Medium 300g',
       'Kettle Tortilla ChpsFeta&Garlic 150g',
       'Smiths Crinkle Cut  French OnionDip 150g',
       'WW D/Style Chip     Sea Salt 200g',
       'Smiths Chip Thinly  CutSalt/Vinegr175g',
       'Kettle Sensations   BBQ&Maple 150g',
       'Old El Paso Salsa   Dip Tomato Mild 300g',
       'Tostitos Smoked     Chipotle 175g', 'RRD Lime & Pepper
165g',
       'CCs Nacho Cheese    175g', 'Snbts Whlgrn Crisps Cheddr&Mstrd
90g',
       'Kettle Tortilla ChpsBtroot&Ricotta 150g',
       'Pringles Sthrn FriedChicken 134g',
       'Pringles Chicken    Salt Crips 134g',
       'French Fries Potato Chips 175g',
       'Kettle Mozzarella   Basil & Pesto 175g', 'CCs Original 175g',
       'Tostitos Lightly    Salted 175g',
       'Smiths Crnkle Chip  Orgnl Big Bag 380g',
       'Smiths Crinkle Cut  Chips Chicken 170g',
       'Smiths Crinkle Cut  Chips Chs&Onion170g', 'Twisties
Chicken270g',
       'Woolworths Medium   Salsa 300g',
       'Red Rock Deli Sp    Salt & Truffle 150G',
       'RRD Pc Sea Salt     165g', 'WW Supreme Cheese   Corn Chips
200g',
       'WW Original Corn    Chips 200g', 'Woolworths Mild     Salsa
300g',
       'Cheezels Cheese Box 125g', 'Doritos Salsa Mild  300g',
       'Cobs Popd Swt/Chlli &Sr/Cream Chips 110g',
       'Infzns Crn Crnchers Tangy Gcamole 110g',
       'WW Sour Cream &OnionStacked Chips 160g',
       'Pringles Mystery    Flavour 134g', 'Pringles Barbeque   134g',
       'Grain Waves         Sweet Chilli 210g',
       'Pringles Sweet&Spcy BBQ 134g', 'Kettle Original 175g',
       'Infuzions Thai SweetChili PotatoMix 110g',
       'Old El Paso Salsa   Dip Chnky Tom Ht300g',
       'Smiths Crinkle Cut  Tomato Salsa 150g',
       'Cheetos Chs & Bacon Balls 190g',
       'Kettle Sweet Chilli And Sour Cream 175g',
       'Doritos Corn Chips  Nacho Cheese 170g',
       'Cobs Popd Sour Crm  &Chives Chips 110g',
       'Red Rock Deli Thai  Chilli&Lime 150g',
       'Twisties Cheese     Burger 250g',
       'Kettle Sea Salt     And Vinegar 175g',
       'WW Crinkle Cut      Original 175g',
       'Dorito Corn Chp     Supreme 380g',
       'Doritos Corn Chip Mexican Jalapeno 150g',
       'Pringles SourCream  Onion 134g',
       'Kettle Tortilla ChpsHny&Jlpno Chili 150g',
       'RRD Steak &         Chimuchurri 150g',
       'Thins Chips Salt &  Vinegar 175g',
       'Thins Chips         Originl saltd 175g',
       'RRD Honey Soy       Chicken 165g',
       'Kettle Honey Soy    Chicken 175g',
       'NCC Sour Cream &    Garden Chives 175g',
       'Pringles Original   Crisps 134g',
       'Smith Crinkle Cut   Mac N Cheese 150g',
       'Thins Potato Chips  Hot & Spicy 175g', 'Pringles Slt Vingar
134g'],
      dtype=object)

# splitting the weight
qvi_merged['WEIGHT'] = qvi_merged['PROD_NAME'].str[-4:]
qvi_merged['WEIGHT']
```

```
0         175g
1         150g
2         210G
3         175g
4         160g
          ...
264831    210g
264832    150g
264833    165g
264834    300g
264835    330g
Name: WEIGHT, Length: 264836, dtype: object
```

```
qvi_merged['WEIGHT'].value_counts()
```

```
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: WEIGHT, dtype: int64
```

```
# correcting the data
qvi_merged['WEIGHT'] = qvi_merged['WEIGHT'].replace({'Salt':'135g',
'210G':'210g', '150G':'150g', '175G':'175g'})

# re checking to see the changes
qvi_merged['WEIGHT'].value_counts()
```

```
175g    66390
150g    43131
```

```
134g    25102
110g    22387
170g    19983
165g    15297
300g    15166
330g    12540
380g     6418
270g     6285
210g     6272
200g     4473
135g     3257
250g     3169
 90g     3008
190g     2995
160g     2970
220g     1564
 70g     1507
180g     1468
125g     1454
Name: WEIGHT, dtype: int64

# removing whitespaces
qvi_merged['PROD_NAME'] = qvi_merged['PROD_NAME'].str.strip()
```

Difference between strip() and split()

.strip(): Cleans up a single string by removing extra whitespace

.split() breaks a string into multiple substrings based on a delimiter

```
# Salsa is in the dataset. Since its not a chip, it needs to be
removed

qvi_merged[qvi_merged['PROD_NAME'] == 'Old El Paso Salsa   Dip Tomato
Med 300g'].count()

LYLTY_CARD_NBR      3114
LIFESTAGE           3114
PREMIUM_CUSTOMER    3114
DATE                3114
STORE_NBR           3114
TXN_ID              3114
PROD_NBR            3114
PROD_NAME           3114
PROD_QTY            3114
TOT_SALES           3114
WEIGHT              3114
dtype: int64
```

```
qvi_merged[qvi_merged['PROD_NAME'] == 'Old El Paso Salsa   Dip Chnky
Tom Ht300g'].count()

LYLTY_CARD_NBR      3125
LIFESTAGE           3125
PREMIUM_CUSTOMER    3125
DATE                3125
STORE_NBR           3125
TXN_ID              3125
PROD_NBR            3125
PROD_NAME           3125
PROD_QTY            3125
TOT_SALES           3125
WEIGHT              3125
dtype: int64

# dropping the 2 above
Salsa1 = qvi_merged[qvi_merged['PROD_NAME'] == 'Old El Paso Salsa
Dip Chnky Tom Ht300g'].index

qvi_merged = qvi_merged.drop(Salsa1)

# confirming

qvi_merged[qvi_merged['PROD_NAME'] == 'Old El Paso Salsa   Dip Chnky
Tom Ht300g'].count()

LYLTY_CARD_NBR      0
LIFESTAGE           0
PREMIUM_CUSTOMER    0
DATE                0
STORE_NBR           0
TXN_ID              0
PROD_NBR            0
PROD_NAME           0
PROD_QTY            0
TOT_SALES           0
WEIGHT              0
dtype: int64

Salsa2 = qvi_merged[qvi_merged['PROD_NAME'] == 'Old El Paso Salsa
Dip Tomato Med 300g'].index

qvi_merged = qvi_merged.drop(Salsa2)

# confirming

qvi_merged[qvi_merged['PROD_NAME'] == 'Old El Paso Salsa   Dip Tomato
Med 300g'].count()
```

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

```
# Creating a column called 'BRAND'

qvi_merged['BRAND'] = qvi_merged['PROD_NAME'].str.split().str.get(0)

qvi_merged['BRAND'].value_counts()
```

```
Kettle        41288
Smiths        28860
Pringles      25102
Doritos       24962
Thins         14075
RRD           11894
Infuzions     11057
WW            10320
Cobs           9693
Tostitos       9471
Twisties       9454
Tyrrells       6442
Grain          6272
Natural        6050
Red            5885
Cheezels       4603
CCs            4551
Woolworths     4437
Dorito         3185
Infzns         3144
Old            3085
Smith          2963
Cheetos        2927
Snbts          1576
Burger         1564
GrnWves        1468
Sunbites       1432
NCC            1419
French         1418
Name: BRAND, dtype: int64
```

```
# So far we have cleansed the column 'PROD_NAME' and created two
columns 'WEIGHT' AND 'BRAND'
# Next check the 'DATE' column next
```

DATE

```
qvi_merged['DATE']
```

```
0         43390
1         43359
2         43531
3         43532
4         43406
          ...
264830    43315
264831    43442
264832    43374
264833    43397
264835    43448
Name: DATE, Length: 258597, dtype: int64
```

```
# 15 Dates
date_offsets = qvi_merged['DATE'].to_list()
base_date = pd.Timestamp("1899-12-30") # Start Date
qvi_merged['DATE'] = [base_date + pd.DateOffset(date_offset) for
date_offset in date_offsets]
qvi_merged['DATE'][0:15]
```

```
0    2018-10-17
1    2018-09-16
2    2019-03-07
3    2019-03-08
4    2018-11-02
5    2018-12-28
6    2018-12-04
7    2018-12-05
8    2018-11-20
9    2018-09-09
11   2018-07-29
13   2018-12-01
14   2018-12-19
15   2019-03-15
16   2019-06-19
Name: DATE, dtype: datetime64[ns]
```

Purpose:

The code aims to convert the DATE column in the qvi_merged DataFrame, which is currently in an integer format (likely representing the number of days since a specific date), into a proper datetime format.

Step-by-Step Explanation:

date_offsets = qvi_merged['DATE'].to_list()

This line extracts the values from the DATE column of the DataFrame and converts them into a Python list. This list now contains the integer representations of the dates.

base_date = pd.Timestamp("1899-12-30")

This creates a pandas Timestamp object representing the base date, which is often used as a reference point for Excel date representations.

qvi_merged['DATE'] = [base_date + pd.DateOffset(date_offset) for date_offset in date_offsets]

This is the core of the conversion logic: It iterates through each date_offset value in the list. For each date_offset, it creates a pd.DateOffset object, which represents a time interval. It adds this pd.DateOffset to the base_date to get the actual date. The resulting list of datetime objects is then assigned back to the DATE column of the DataFrame.

qvi_merged['DATE'][0:15]

This line selects and prints the first 15 rows of the converted DATE column to verify the conversion. In essence, the code translates the integer representation of dates into actual datetime objects by using a base date and adding the appropriate number of days.

This approach is particularly useful when dealing with date data that is stored as integers, which is sometimes encountered in data imported from Excel or other sources.

```
qvi_merged.head()

   LYLTY_CARD_NBR              LIFESTAGE PREMIUM_CUSTOMER
DATE  \
0            1000  YOUNG SINGLES/COUPLES          Premium 2018-10-17

1            1002  YOUNG SINGLES/COUPLES       Mainstream 2018-09-16

2            1003         YOUNG FAMILIES           Budget 2019-03-07

3            1003         YOUNG FAMILIES           Budget 2019-03-08

4            1004  OLDER SINGLES/COUPLES       Mainstream 2018-11-02


   STORE_NBR  TXN_ID  PROD_NBR                               PROD_NAME
\
0          1       1         5  Natural Chip        Compny SeaSalt175g

1          1       2        58   Red Rock Deli Chikn&Garlic Aioli 150g

2          1       3        52   Grain Waves Sour    Cream&Chives 210G

3          1       4       106  Natural ChipCo      Hony Soy Chckn175g
```

```
4          1       5        96         WW Original Stacked Chips 160g


   PROD_QTY  TOT_SALES WEIGHT    BRAND
0         2        6.0   175g  Natural
1         1        2.7   150g      Red
2         1        3.6   210g    Grain
3         1        3.0   175g  Natural
4         1        1.9   160g       WW
```

### BRAND

```
qvi_merged['BRAND'].value_counts()
```

```
Kettle        41288
Smiths        28860
Pringles      25102
Doritos       24962
Thins         14075
RRD           11894
Infuzions     11057
WW            10320
Cobs           9693
Tostitos       9471
Twisties       9454
Tyrrells       6442
Grain          6272
Natural        6050
Red            5885
Cheezels       4603
CCs            4551
Woolworths     4437
Dorito         3185
Infzns         3144
Old            3085
Smith          2963
Cheetos        2927
Snbts          1576
Burger         1564
GrnWves        1468
Sunbites       1432
NCC            1419
French         1418
Name: BRAND, dtype: int64
```

```
# RRD brand is Red brand itself. Merging them

qvi_merged['BRAND'] = qvi_merged['BRAND'].replace({'RRD':'Red'})

qvi_merged['BRAND'].value_counts()
```

```
Kettle        41288
Smiths        28860
Pringles      25102
Doritos       24962
Red           17779
Thins         14075
Infuzions     11057
WW            10320
Cobs           9693
Tostitos       9471
Twisties       9454
Tyrrells       6442
Grain          6272
Natural        6050
Cheezels       4603
CCs            4551
Woolworths     4437
Dorito         3185
Infzns         3144
Old            3085
Smith          2963
Cheetos        2927
Snbts          1576
Burger         1564
GrnWves        1468
Sunbites       1432
NCC            1419
French         1418
Name: BRAND, dtype: int64
```

```
# replacing other brands since they are the same

qvi_merged['BRAND'] = 
qvi_merged['BRAND'].replace({'Dorito':'Doritos','Smith':'Smiths','Infz
ns':'Infuzions'})

# Now, Since the data is cleansed we can get into deriving insights 
through visualizations
```

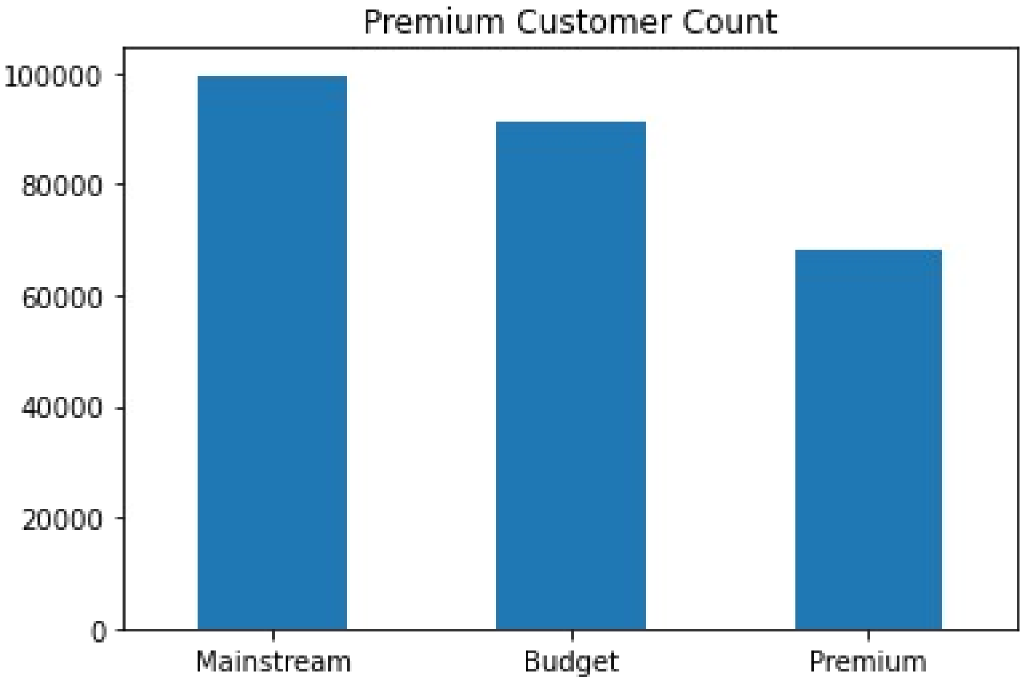
## Visualizations

```
qvi_merged.columns
```

```
Index(['LYLTY_CARD_NBR', 'LIFESTAGE', 'PREMIUM_CUSTOMER', 'DATE', 
'STORE_NBR',
       'TXN_ID', 'PROD_NBR', 'PROD_NAME', 'PROD_QTY', 'TOT_SALES', 
'WEIGHT',
       'BRAND'],
      dtype='object')
```

PREMIUM_CUSTOMER

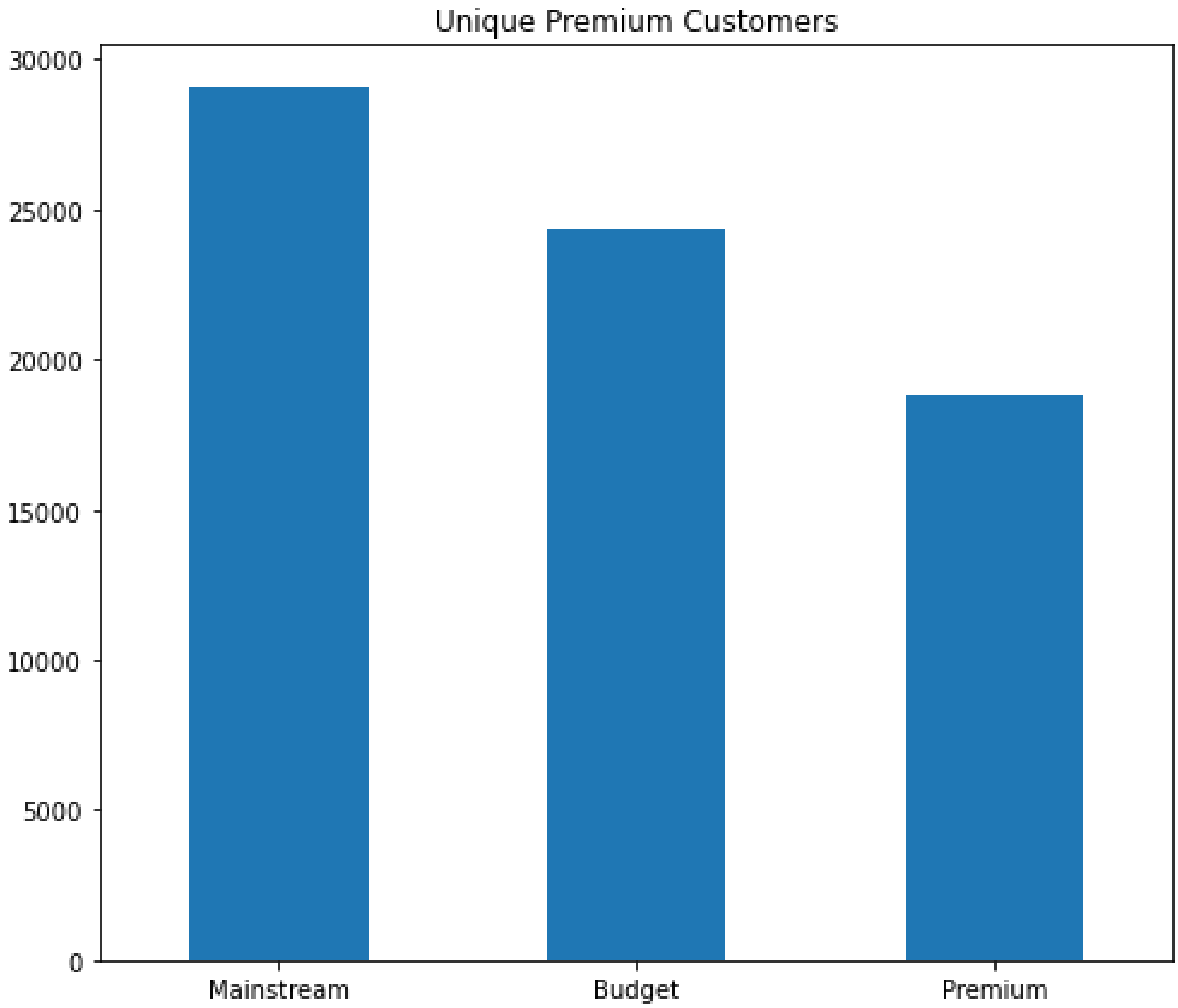
```
qvi_merged['PREMIUM_CUSTOMER'].value_counts().plot.bar()
plt.xticks(rotation = 360,)
plt.title('Premium Customer Count')
plt.show()
```



```
# Creating a graph of just unique member numbers and counting the
unique members by premium customer type

unique_members = qvi_merged.drop_duplicates(subset = 'LYLTY_CARD_NBR')
unique_members['PREMIUM_CUSTOMER'].value_counts()
```

```
Mainstream    29041
Budget        24334
Premium       18814
Name: PREMIUM_CUSTOMER, dtype: int64
```

```
plt.figure(figsize = [8,7])
unique_members['PREMIUM_CUSTOMER'].value_counts().plot.bar()
plt.xticks(rotation = 360)
plt.yticks()
plt.title('Unique Premium Customers')
plt.show()
```

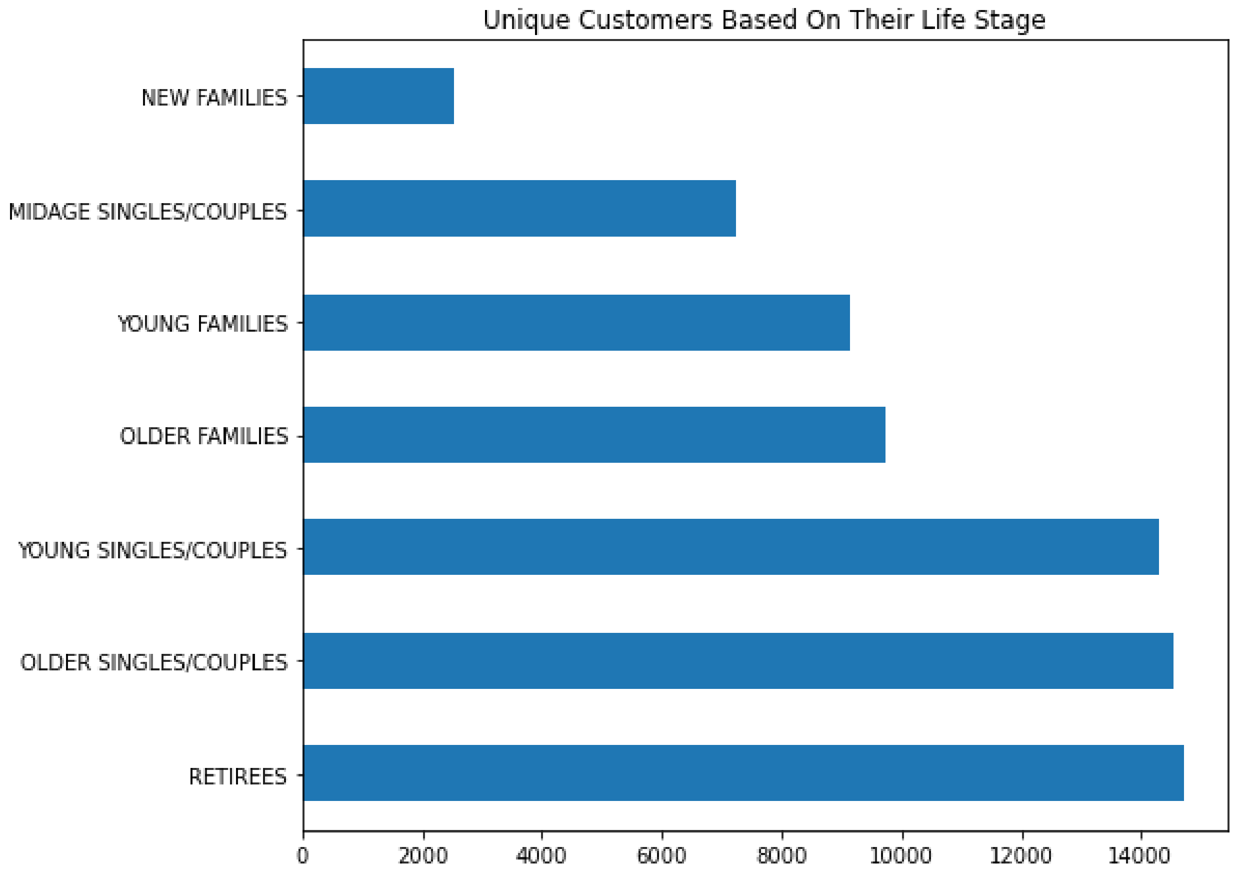


LIFESTAGE

```
unique_members['LIFESTAGE'].value_counts()
```

```
RETIREES                  14721
OLDER SINGLES/COUPLES     14550
YOUNG SINGLES/COUPLES     14309
OLDER FAMILIES             9723
YOUNG FAMILIES             9129
MIDAGE SINGLES/COUPLES     7232
NEW FAMILIES               2525
Name: LIFESTAGE, dtype: int64
```

```
plt.figure(figsize = [8,7])
unique_members['LIFESTAGE'].value_counts().plot.barh()
plt.title('Unique Customers Based On Their Life Stage')
plt.show()
```



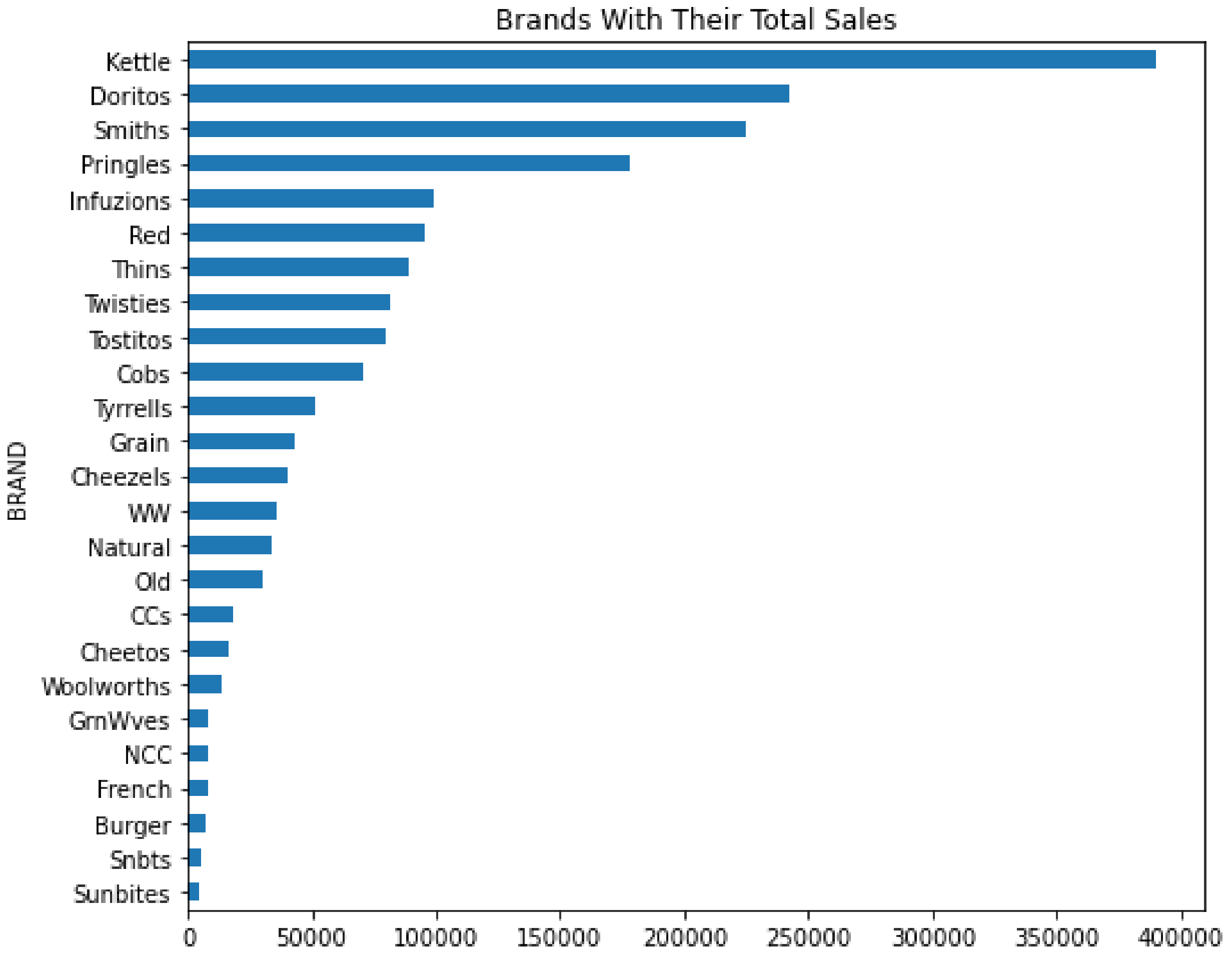
BRAND

```
qvi_merged.groupby('BRAND')['TOT_SALES'].sum().sort_values()
```

```
BRAND
Sunbites       4600.2
Snbts          5076.2
Burger         6831.0
French         7929.0
NCC            8046.0
GrnWves        8568.4
Woolworths    13454.1
Cheetos       16884.5
CCs           18078.9
Old           30033.9
Natural       34272.0
WW            35889.5
Cheezels      40029.9
Grain         43048.8
Tyrrells      51647.4
Cobs          70569.8
Tostitos      79789.6
Twisties      81522.1
Thins         88852.5
```

```
Red               95046.0
Infuzions         99047.6
Pringles         177655.5
Smiths           224660.2
Doritos          241890.9
Kettle           390239.8
Name: TOT_SALES, dtype: float64
```

```
plt.figure(figsize = [8,7])
qvi_merged.groupby('BRAND')
['TOT_SALES'].sum().sort_values().plot.barh()
plt.title('Brands With Their Total Sales')
plt.show()
```



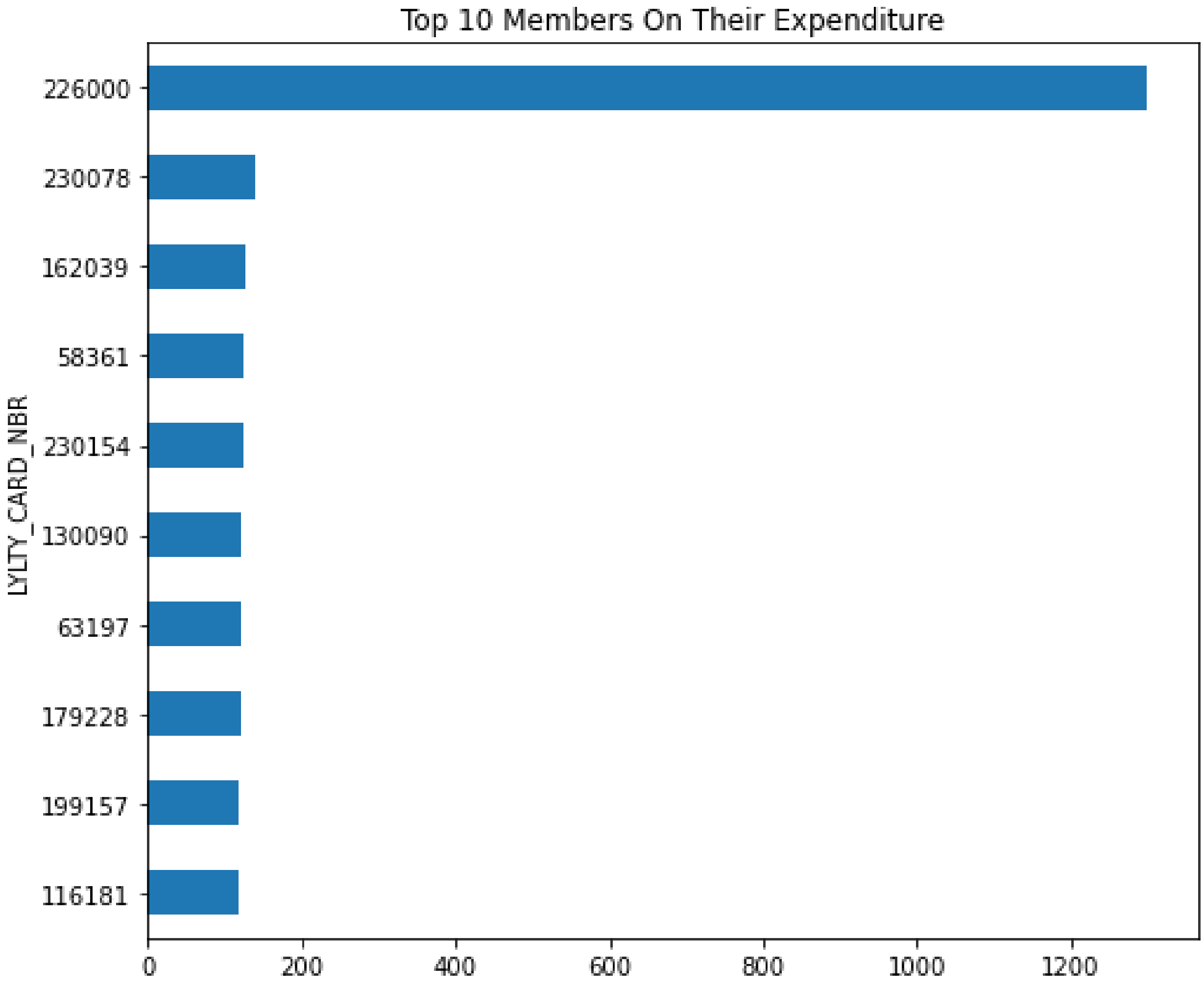
## Top 10 Members That Bought The Most

```
qvi_merged.groupby('LYLTY_CARD_NBR')
['TOT_SALES'].sum().sort_values().tail(10)
```

```
LYLTY_CARD_NBR
116181     117.60
```

```
199157      118.80
179228      120.80
63197       122.60
130090      122.65
230154      124.40
58361       124.80
162039      126.80
230078      138.60
226000     1300.00
Name: TOT_SALES, dtype: float64
```

```
plt.figure(figsize = [8,7])
qvi_merged.groupby('LYLTY_CARD_NBR')
['TOT_SALES'].sum().sort_values().tail(10).plot.barh()
plt.title('Top 10 Members On Their Expenditure')
plt.show()
```

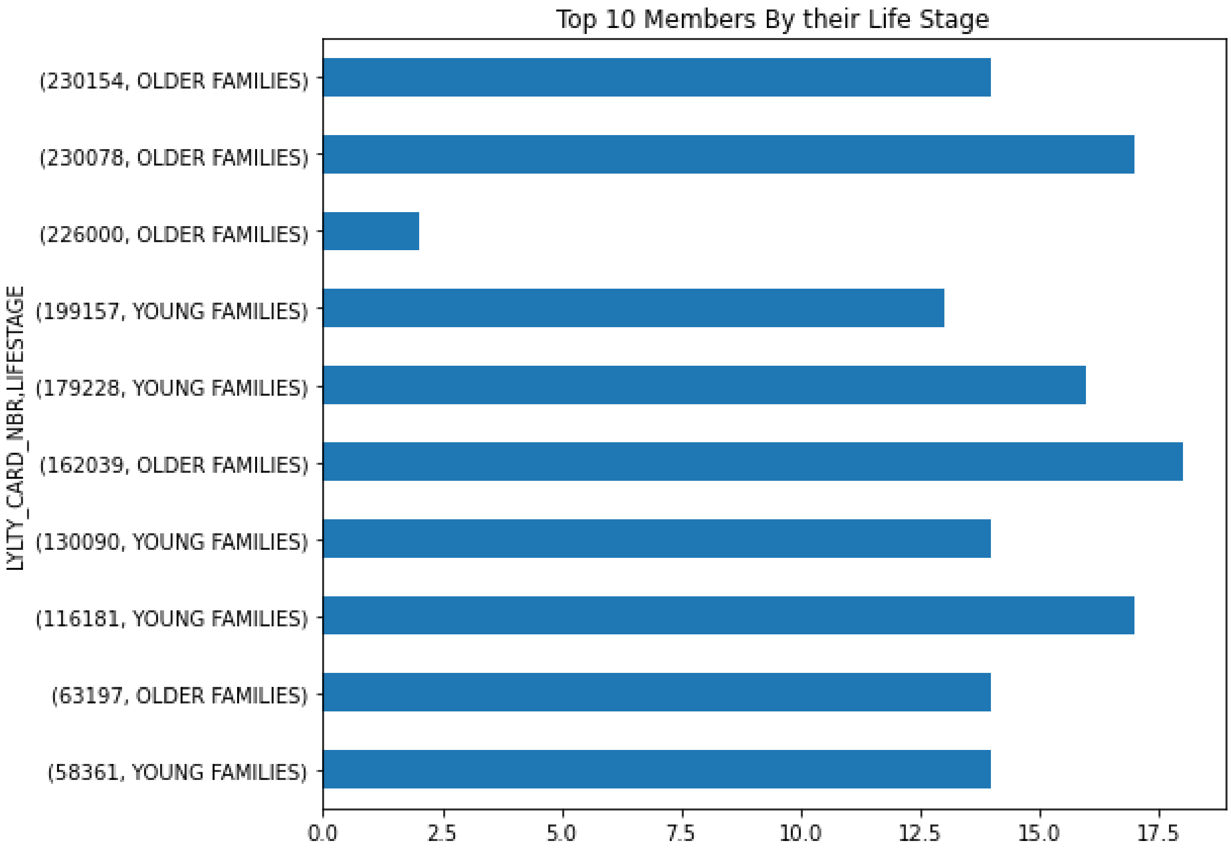


Top 10 Members By their Life Stage Categories

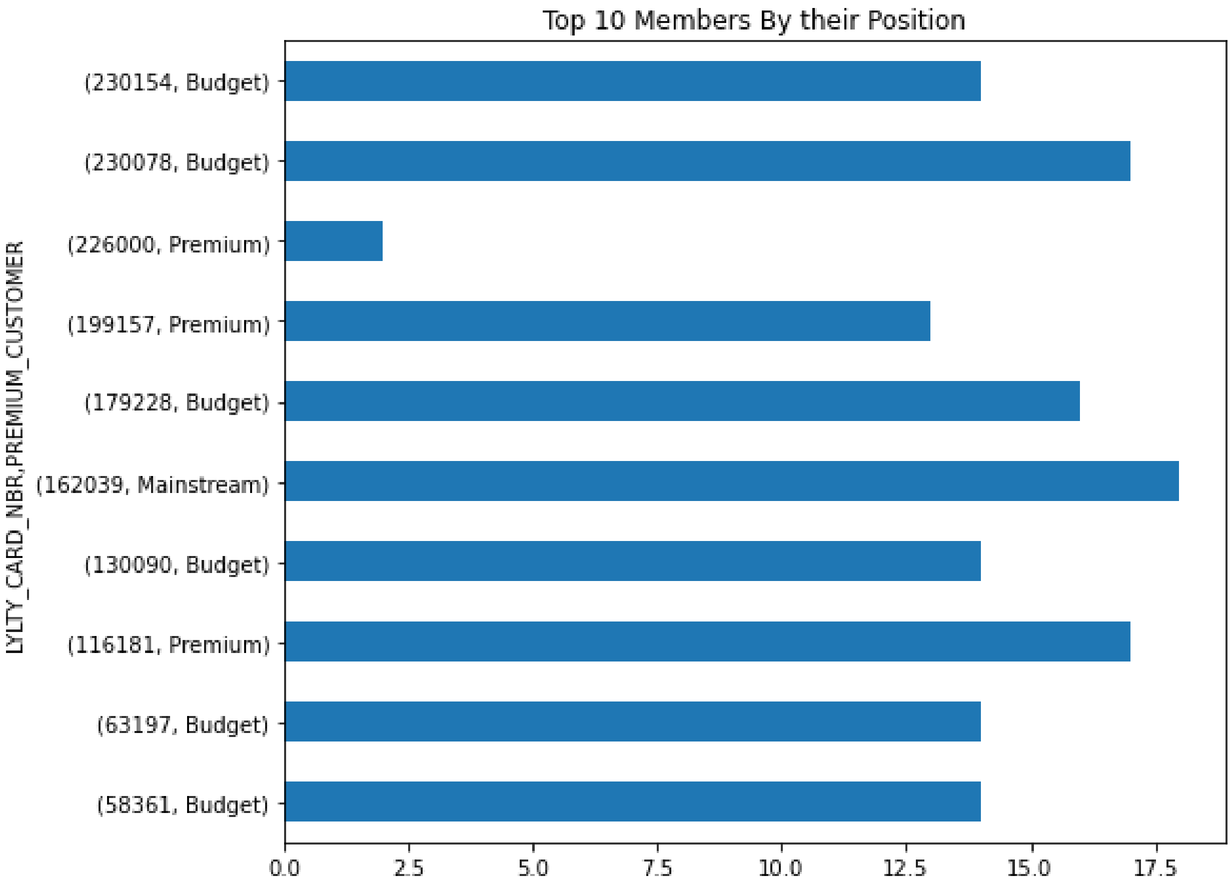
```
Top_10 =
[116181,199157,179228,63197,130090,230154,58361,162039,230078,226000]
Top_10_Members = qvi_merged[qvi_merged['LYLTY_CARD_NBR'].isin(Top_10)]

Top_10_Members.groupby('LYLTY_CARD_NBR')['LIFESTAGE'].value_counts()
```

```
LYLTY_CARD_NBR  LIFESTAGE
58361           YOUNG FAMILIES    14
63197           OLDER FAMILIES    14
116181          YOUNG FAMILIES    17
130090          YOUNG FAMILIES    14
162039          OLDER FAMILIES    18
179228          YOUNG FAMILIES    16
199157          YOUNG FAMILIES    13
226000          OLDER FAMILIES     2
230078          OLDER FAMILIES    17
230154          OLDER FAMILIES    14
Name: LIFESTAGE, dtype: int64
```

```
plt.figure(figsize = [8,7])
Top_10_Members.groupby('LYLTY_CARD_NBR')
['LIFESTAGE'].value_counts().plot.barh()
plt.title('Top 10 Members By their Life Stage')
plt.show()
```

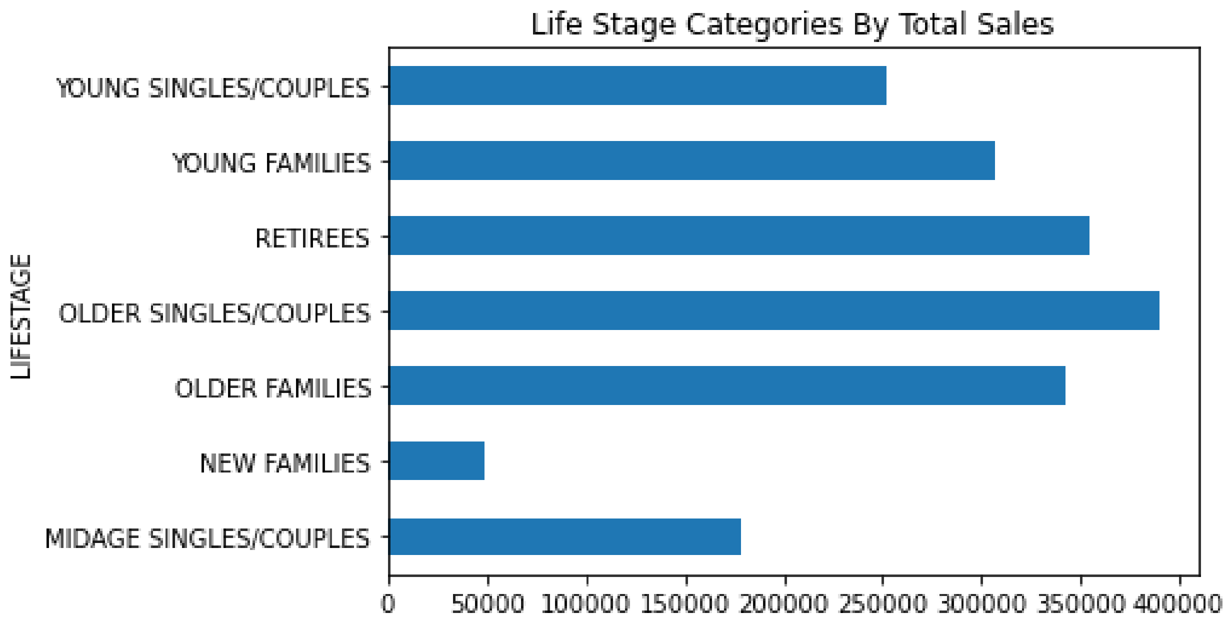


```
Top_10_Members.groupby('LYLTY_CARD_NBR')
['PREMIUM_CUSTOMER'].value_counts()

LYLTY_CARD_NBR  PREMIUM_CUSTOMER
58361           Budget              14
63197           Budget              14
116181          Premium             17
130090          Budget              14
162039          Mainstream          18
179228          Budget              16
199157          Premium             13
226000          Premium              2
230078          Budget              17
230154          Budget              14
Name: PREMIUM_CUSTOMER, dtype: int64

plt.figure(figsize = [8,7])
Top_10_Members.groupby('LYLTY_CARD_NBR')
['PREMIUM_CUSTOMER'].value_counts().plot.barh()
plt.title('Top 10 Members By their Position')
plt.show()
```



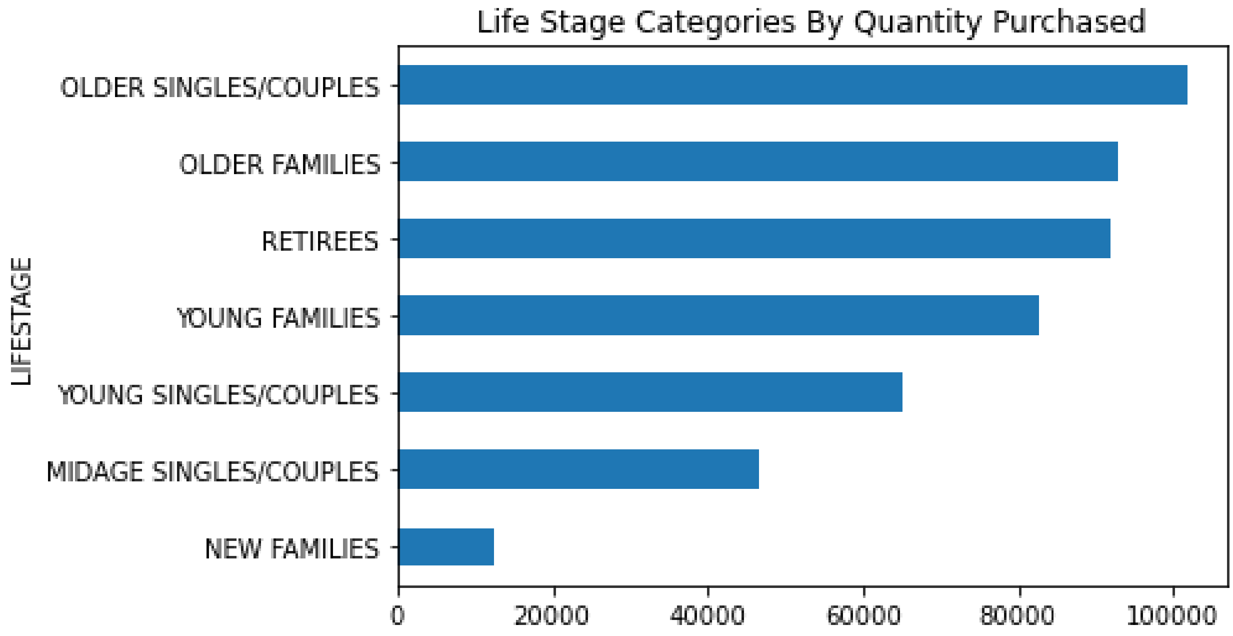
Grouping LifeStage By Total Sales

```
qvi_merged.groupby('LIFESTAGE')['TOT_SALES'].sum()

LIFESTAGE
MIDAGE SINGLES/COUPLES    178927.10
NEW FAMILIES               48755.55
OLDER FAMILIES            342904.20
OLDER SINGLES/COUPLES     389814.45
RETIREES                  354669.50
YOUNG FAMILIES            306434.40
YOUNG SINGLES/COUPLES     252158.60
Name: TOT_SALES, dtype: float64

qvi_merged.groupby('LIFESTAGE')['TOT_SALES'].sum().plot.barh()
plt.title('Life Stage Categories By Total Sales')
plt.show()
```



Grouping Life Stage By Quantity Purchased

```
qvi_merged.groupby('LIFESTAGE')
['PROD_QTY'].sum().sort_values().plot.barh()
plt.title('Life Stage Categories By Quantity Purchased')
plt.show()
```



Viewing Sales By Premium Customer type and Brand

```
qvi_merged.groupby(['PREMIUM_CUSTOMER','BRAND']).TOT_SALES.sum().sort_
values(ascending = False)
```

```
PREMIUM_CUSTOMER  BRAND
Mainstream        Kettle      154477.0
Budget            Kettle      134407.8
Premium           Kettle      101355.0
Mainstream        Doritos      95638.9
                  Smiths       84726.2
                                 ...
                  Snbts         1752.7
Budget            Sunbites      1725.5
Mainstream        Sunbites      1601.4
Premium           Snbts         1380.4
                  Sunbites      1273.3
Name: TOT_SALES, Length: 75, dtype: float64
```

### Exploring Weight And Bag Size Of Chips

```
qvi_merged['WEIGHT'].value_counts()
```

```
175g    66390
150g    43131
134g    25102
110g    22387
170g    19983
165g    15297
330g    12540
300g     8927
380g     6418
270g     6285
210g     6272
200g     4473
135g     3257
250g     3169
 90g     3008
190g     2995
160g     2970
220g     1564
 70g     1507
180g     1468
125g     1454
Name: WEIGHT, dtype: int64
```

```
# Changing the dtype of the 'WEIGHT' column to categorize it easily
qvi_merged.head()
```

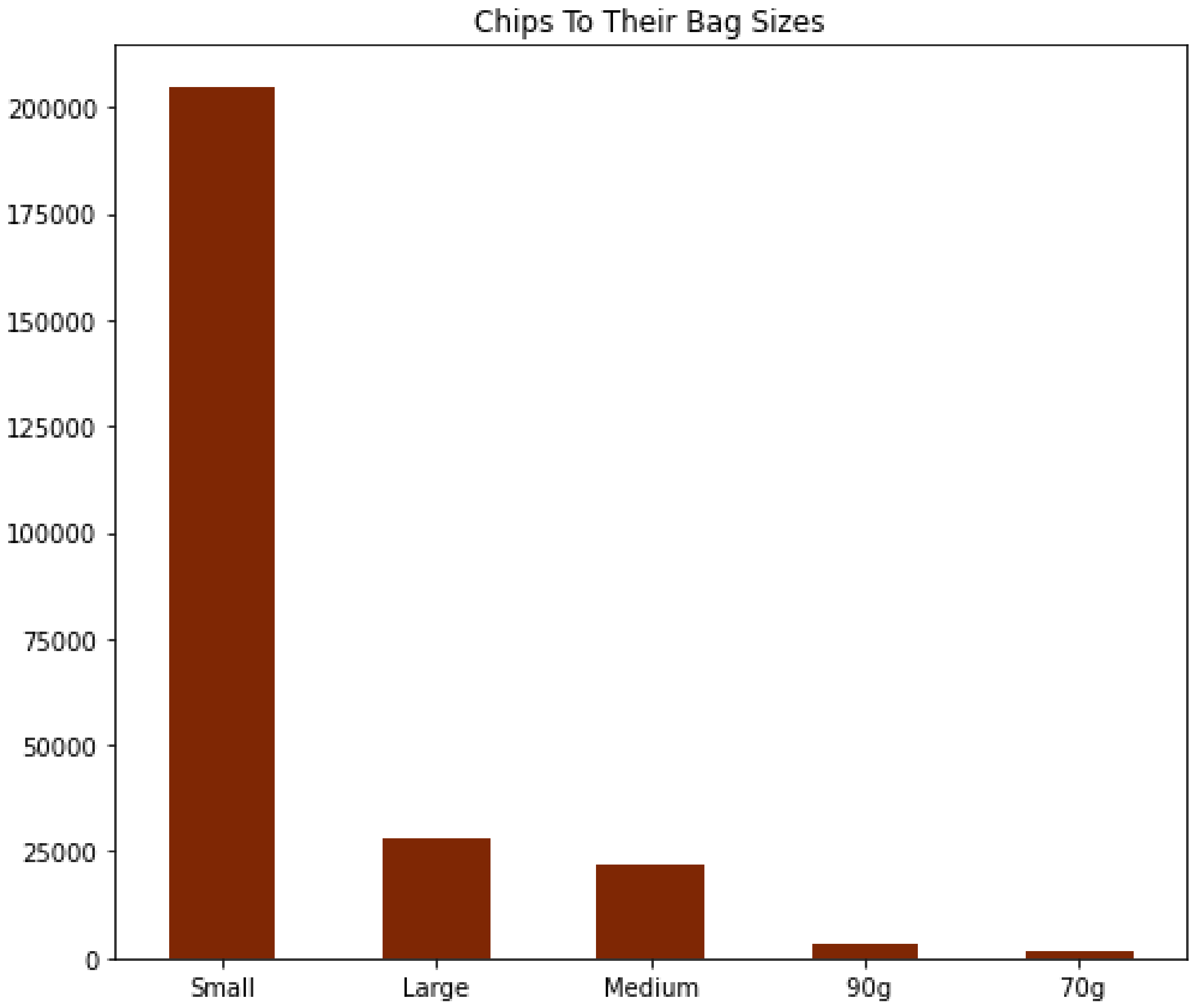
```
   LYLTY_CARD_NBR              LIFESTAGE PREMIUM_CUSTOMER
DATE  \
0            1000  YOUNG SINGLES/COUPLES          Premium 2018-10-17

1            1002  YOUNG SINGLES/COUPLES       Mainstream 2018-09-16
```

```
2              1003         YOUNG FAMILIES           Budget 2019-03-07

3              1003         YOUNG FAMILIES           Budget 2019-03-08

4              1004  OLDER SINGLES/COUPLES       Mainstream 2018-11-02


   STORE_NBR  TXN_ID  PROD_NBR                               PROD_NAME
\
0          1       1         5  Natural Chip        Compny SeaSalt175g

1          1       2        58   Red Rock Deli Chikn&Garlic Aioli 150g

2          1       3        52   Grain Waves Sour    Cream&Chives 210G

3          1       4       106  Natural ChipCo      Hony Soy Chckn175g

4          1       5        96         WW Original Stacked Chips 160g


   PROD_QTY  TOT_SALES WEIGHT    BRAND
0         2        6.0   175g  Natural
1         1        2.7   150g      Red
2         1        3.6   210g    Grain
3         1        3.0   175g  Natural
4         1        1.9   160g       WW
```

```
qvi_merged['WEIGHT'] = qvi_merged['WEIGHT'].astype(str)

# creating custom category acc to chip bag sizes
qvi_merged['BAG_SIZE'] = qvi_merged['WEIGHT'].replace({'70g':'Extra
Small', '90g':'Extra Small',

'110g':'Small' , '125g':'Small' , '134g':'Small',

'135g':'Small' , '150g':'Small' , '160g':'Small',

'165g':'Small' , '170g':'Small' , '175g':'Small',

'180g':'Small','190g':'Small' , '200g':'Medium',

'210g':'Medium','220g':'Medium', '250g':'Medium',
                                                                   '270g':'Medium'
, '300g':'Large' , '330g':'Large',

'380g':'Large'})


qvi_merged['BAG_SIZE'].value_counts()
```

```
Small     204434
Large      27885
Medium     21763
 90g        3008
 70g        1507
Name: BAG_SIZE, dtype: int64
```

```
plt.figure(figsize = [8,7])
qvi_merged['BAG_SIZE'].value_counts().plot.bar(colormap = 'Oranges_r')
plt.xticks(rotation = 360)
plt.title('Chips To Their Bag Sizes')
plt.show()
```



```
# Saving it to a csv file
qvi_merged.to_csv('qvi_merged.csv')
```

# Summary

1. Largest customer type is the Mainstream group
2. Largest membership group is the Older Population
3. Top 10 Members spent over 120$ on chips in an year
4. Top 4 Brands bought by customers are : Doritos, Smiths, Pringles and Kettle
5. Older Individuals purchased the most number of chips which includes single individuals and families. New families purchased the least on chips
6. There doesnt appear to be any statistical difference between purchase prices and customers on their lifestage or membership designation
7. The most purchased bag size chips were the small bags and large bags. The common medium sized bags and extra small bags were sold in the least