In [1]:
```python
# Dependencies
import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime, timedelta
import collections
import re

import warnings
warnings.filterwarnings("ignore")
mpl.rcParams['agg.path.chunksize'] = 10000
```

In [2]:
```python
# Reading the Dataset
df_purchase = pd.read_csv(r'QVI_purchase_behaviour.csv')
df_transaction = pd.read_excel(r'QVI_transaction_data.xlsx')
```

In [3]:
```python
df_purchase.head()
```

Out[3]:

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

In [4]:
```python
df_transaction.head()
```

Out[4]:

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

Exploratory Data Analysis - EDA

In [5]:
```python
# Checking the dimension of the dataset
df_purchase.shape
```

Out[5]:
```
(72637, 3)
```

In [6]:
```python
# Checking for null values in the dataset
df_purchase.isnull().sum()
```

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

In [7]:
```python
df_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
```

In [8]:
```python
df_purchase['PREMIUM_CUSTOMER'].value_counts()
```

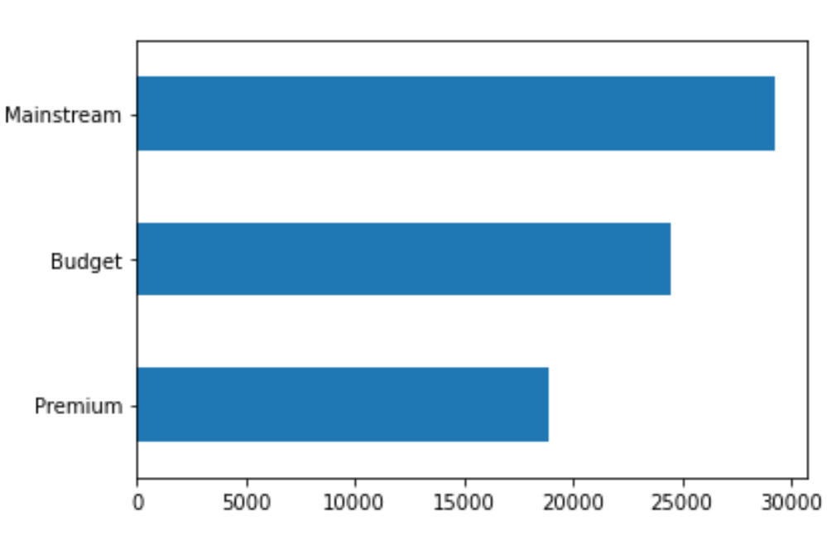
Out[8]:
```
Mainstream    29245
Budget        24470
Premium       18922
Name: PREMIUM_CUSTOMER, dtype: int64
```

In [9]:
```python
df_purchase['LIFESTAGE'].value_counts()
```

Out[9]:
```
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
```

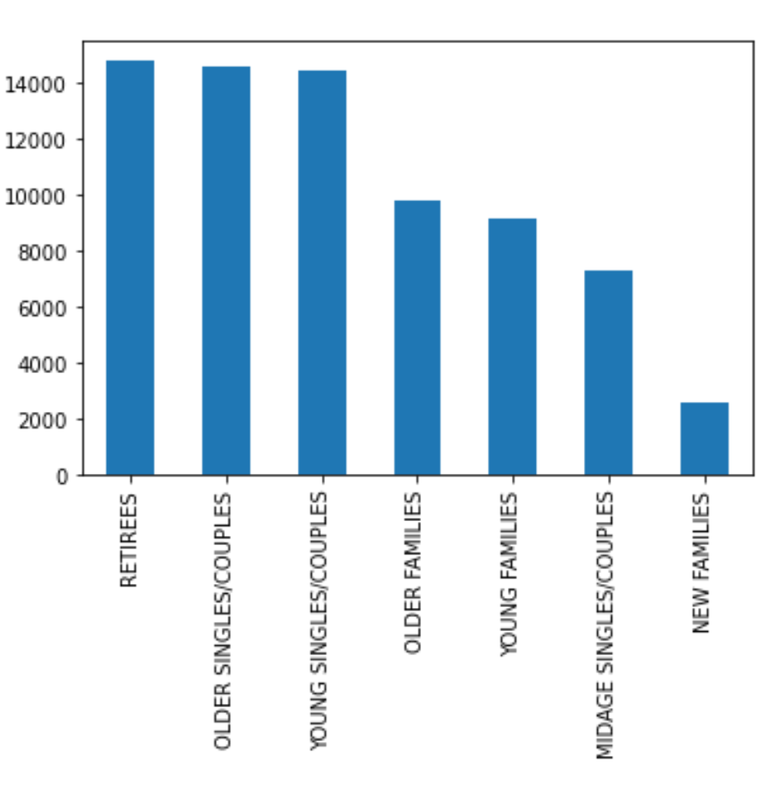
In [10]:
```python
df_purchase['PREMIUM_CUSTOMER'].value_counts().sort_values().plot(kind='barh')
```

Out[10]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1b6b87fd2e0>
```



In [11]:
```python
df_purchase['LIFESTAGE'].value_counts().sort_values(ascending = False).plot(kind='bar')
```

Out[11]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1b6b63e0bb0>
```



In [12]:
```python
df_transaction.shape
```

Out[12]:
```
(264836, 8)
```

In [13]:
```python
df_transaction.isnull().sum()
```

Out[13]:
```
DATE              0
STORE_NBR         0
LYLTY_CARD_NBR    0
TXN_ID            0
PROD_NBR          0
PROD_NAME         0
PROD_QTY          0
TOT_SALES         0
dtype: int64
```

In [14]:
```python
df_transaction.head()
```

Out[14]:

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

In [15]:
```python
df_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
```

In [16]:
```python
# Converting Date from int to date format
df_transaction['DATE'] = pd.to_datetime(df_transaction['DATE'], unit = 'D', origin = '1899-1
2-30')
```

Out[16]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES
0	2018-10-17	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0
1	2019-05-14	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3
2	2019-05-20	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9
	2018-08-					Smiths Chip Thinly		

In [17]:
```python
# Extracting the Brand name from Prod_Name
df_transaction['BRAND_NAME'] = df_transaction['PROD_NAME'].str.extract('([A-Za-z]+)')
```

In [18]:
```python
# Extracting Pack Size from Prod_name
df_transaction['PACK_SIZE'] = df_transaction['PROD_NAME'].str.extract('([0-9]+)').astype('in
t')
```

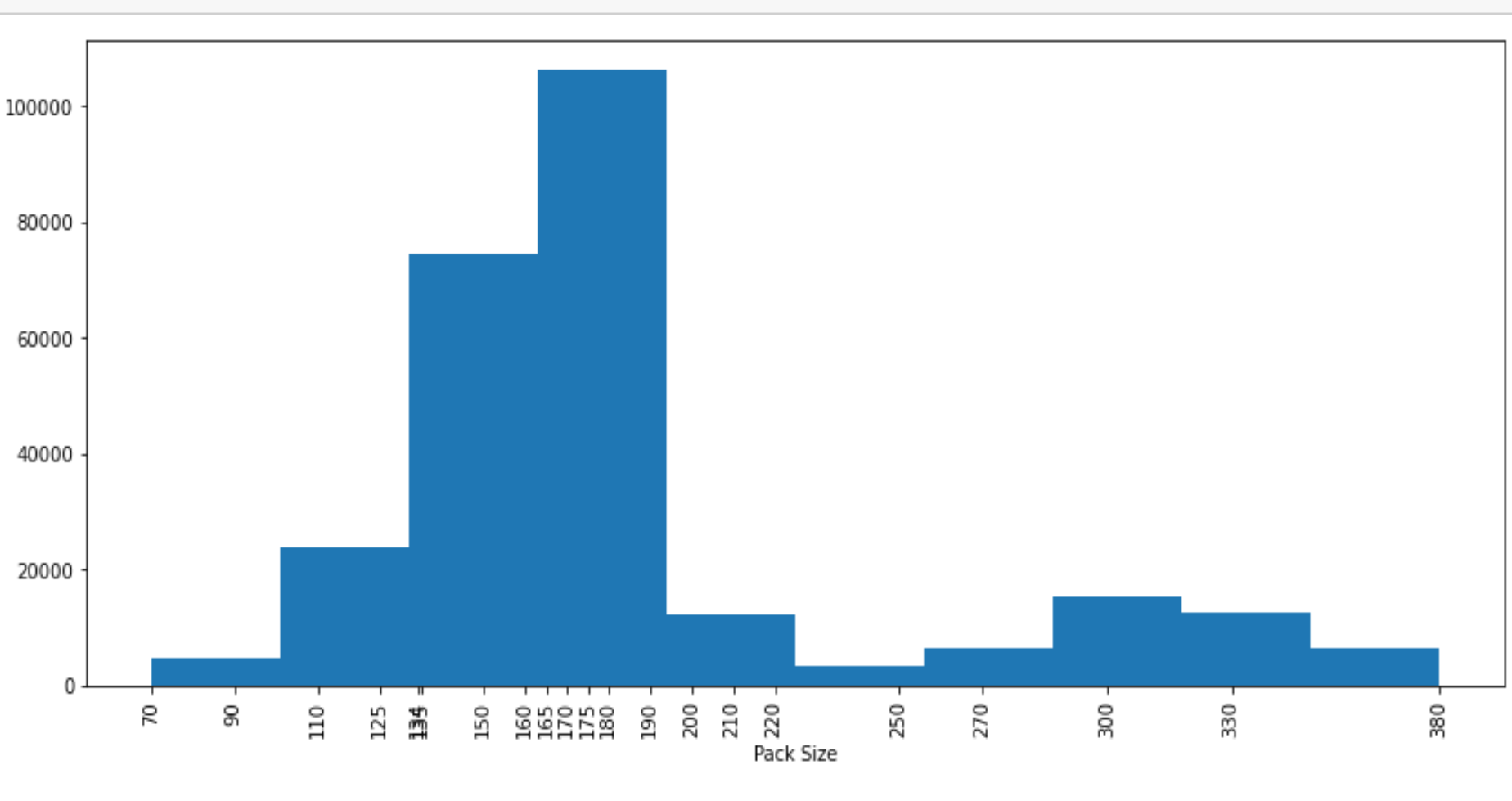
Out[18]:

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

In [19]:
```python
df_transaction.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 264836 entries, 0 to 264835
Data columns (total 10 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       
 8   BRAND_NAME      264836 non-null  object        
 9   PACK_SIZE       264836 non-null  int32         
dtypes: datetime64[ns](1), float64(1), int32(1), int64(5), object(2)
memory usage: 19.2+ MB
```

In [20]:
```python
fig, ax = plt.subplots(figsize=(14,6))
plt.hist(df_transaction['PACK_SIZE'])
plt.xticks(df_transaction['PACK_SIZE'].unique(), rotation = 90)
ax.set_xlabel('Pack Size')
ax.set_ylabel('Frequency')
plt.show()
```



In [21]:
```python
df_transaction['BRAND_NAME'].unique()
```

Out[21]:
```
array(['Natural', 'CCs', 'Smiths', 'Kettle', 'Old', 'Grain', 'Doritos',
       'Twisties', 'WW', 'Thins', 'Burger', 'NCC', 'Cheezels', 'Infzns',
       'Red', 'Pringles', 'Dorito', 'Infuzions', 'Smith', 'GrnWves',
       'Tyrrells', 'Cobs', 'Woolworths', 'French', 'RRD', 'Tostitos',
       'Cheetos', 'Snbts', 'Sunbites'], dtype=object)
```

In [22]:
```python
# Changing brand name Burger to Smiths, since Burger rings are Smiths Brand Chips
df_transaction['BRAND_NAME'][(df_transaction['BRAND_NAME']== 'Burger')] = 'Smiths'
```

In [23]:
```python
# Changing NCC to Natural, since NCC is Natual Chips company
df_transaction['BRAND_NAME'][(df_transaction['BRAND_NAME']== 'NCC')] = 'Natural'
```

In [24]:
```python
# Changing Infzns to Infuzions
df_transaction['BRAND_NAME'][(df_transaction['BRAND_NAME']== 'Infzns')] = 'Infuzions'
```

In [25]:
```python
# Dorito to Doritos
df_transaction['BRAND_NAME'][(df_transaction['BRAND_NAME']== 'Dorito')] = 'Doritos'
```

In [26]:
```python
# Smith to Smiths
df_transaction['BRAND_NAME'][(df_transaction['BRAND_NAME']== 'Smith')] = 'Smiths'
```

In [27]:
```python
# Snbts to Sunbites
df_transaction['BRAND_NAME'][(df_transaction['BRAND_NAME']== 'Snbts')] = 'Sunbites'
```

In [28]:
```python
# Red to RRD
df_transaction['BRAND_NAME'][(df_transaction['BRAND_NAME']== 'Red')] = 'RRD'
```

In [29]:
```python
# Grain and GrnWves to Sunbites
df_transaction['BRAND_NAME'][(df_transaction['BRAND_NAME']== 'Grain')] = 'Sunbites'
df_transaction['BRAND_NAME'][(df_transaction['BRAND_NAME']== 'GrnWves')] = 'Sunbites'
```

In [30]:
```python
# French and WW to Woolworths
df_transaction['BRAND_NAME'][(df_transaction['BRAND_NAME']== 'French')] = 'Woolworths'
df_transaction['BRAND_NAME'][(df_transaction['BRAND_NAME']== 'WW')] = 'Woolworths'
```

In [31]:
```python
df_transaction['BRAND_NAME'].unique()
```

Out[31]:
```
array(['Natural', 'CCs', 'Smiths', 'Kettle', 'Old', 'Sunbites', 'Doritos',
       'Twisties', 'Woolworths', 'Thins', 'Cheezels', 'Infuzions', 'RRD',
       'Pringles', 'Tyrrells', 'Cobs', 'Tostitos', 'Cheetos'],
      dtype=object)
```

In [32]:
```python
df_transaction['PROD_NAME'].str.split()[0]
```

Out[32]:
```
['Natural', 'Chip', 'Compny', 'SeaSalt175g']
```

In [33]:
```python
prod_ = np.array_split(df_transaction['PROD_NAME'].unique(), len(df_transaction['PROD_NAME']
```

In [34]:
```python
prod_list = [prod.tolist() for prod in prod_]
```

In [35]:
```python
prod_split = [prod[0].split() for prod in prod_list]
prod_words = sum(prod_split,[])
```

In [36]:
```python
prod_words = [word for word in prod_words if re.search('(^[A-Za-z])', word) and not re.searc
```

In [37]:
```python
counter = collections.Counter(prod_words)
pd.DataFrame.from_dict(data = counter, orient='index').reset_index().rename(columns={'index'
:'word',0:'count'}).sort_values(by='count', ascending=False)
```

Out[37]:

	word	count
9	Chips	21
6	Smiths	16
7	Crinkle	14
8	Cut	14
13	Kettle	13
...	...	...
88	Basil	1
87	Mozzarella	1
86	Roast	1
84	Chp	1
185	Bolognese	1

186 rows × 2 columns

In [38]:
```python
# Dropping Salsa Products
df_transaction.drop(index = df_transaction[df_transaction['PROD_NAME'].str.find('Salsa') !=
-1].index, inplace = True)
```

In [39]:
```python
# Summary of dataset
df_transaction.describe()
```

Out[39]:

	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_QTY	TOT_SALES	PACK_SIZE
count	246742.000000	2.467420e+05	2.467420e+05	246742.000000	246742.000000	246742.000000	246742.000000
mean	135.051098	1.355310e+05	1.351311e+05	56.351789	1.908062	7.321322	175.585178
std	76.787096	8.071528e+04	7.814772e+04	33.695428	0.659831	3.077828	59.434727
min	1.000000	1.000000e+03	1.000000e+00	1.000000	1.000000	1.700000	70.000000
25%	70.000000	7.001500e+04	6.756925e+04	26.000000	2.000000	5.800000	150.000000
50%	130.000000	1.303670e+05	1.351830e+05	53.000000	2.000000	7.400000	170.000000
75%	203.000000	2.030840e+05	2.026538e+05	87.000000	2.000000	8.800000	175.000000
max	272.000000	2.373711e+06	2.415841e+06	114.000000	200.000000	650.000000	380.000000

In [40]:
```python
# There seems to be outlier where 200 chips packets are bought at once
df_transaction[df_transaction['PROD_QTY'] == 200]
```

Out[40]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	BRAND_NAME
69762	2018-08-19	226	226000	226201	4	Dorito Corn Chp Supreme 380g	200	650.0	Doritos
69763	2019-	226	226000	226210	4	Dorito Corn Chp Supreme	200	650.0	Doritos

In [41]:
```python
df_transaction[df_transaction['LYLTY_CARD_NBR'] == 226000]
```

Out[41]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	BRAND_NAME
69762	2018-08-19	226	226000	226201	4	Dorito Corn Chp Supreme 380g	200	650.0	Doritos
69763	2019-	226	226000	226210	4	Dorito Corn Chp Supreme	200	650.0	Doritos

In [42]:
```python
# Let's remove this outlier based upon the loyalty card number
df_transaction.drop(index = df_transaction[df_transaction['LYLTY_CARD_NBR'] == 226000].index
```

In [43]:
```python
# Let's drop loyalty card number from df_purchase as well, as we don't have any other purcha
ses from this customer
```

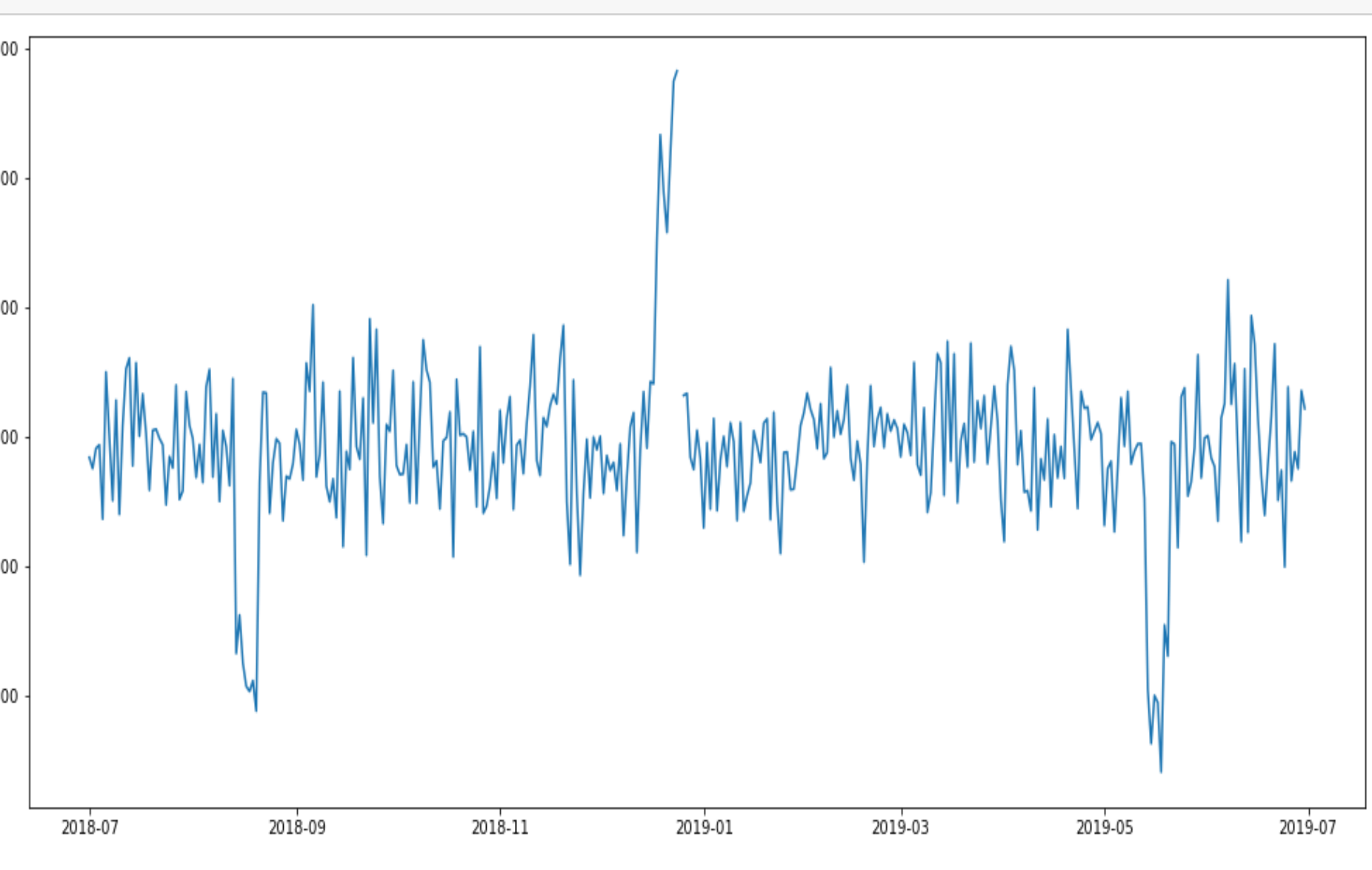
In [44]:
```python
df_transaction.describe()
```

Out[44]:

	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_QTY	TOT_SALES	PACK_SIZE
count	246740.000000	2.467400e+05	2.467400e+05	246740.000000	246740.000000	246740.000000	246740.000000
mean	135.050361	1.355303e+05	1.351304e+05	56.352213	1.906456	7.316113	175.583521
std	76.786971	8.071520e+04	7.814760e+04	33.695235	0.342499	2.474897	59.432118
min	1.000000	1.000000e+03	1.000000e+00	1.000000	1.000000	1.700000	70.000000
25%	70.000000	7.001500e+04	6.756875e+04	26.000000	2.000000	5.800000	150.000000
50%	130.000000	1.303670e+05	1.351815e+05	53.000000	2.000000	7.400000	170.000000
75%	203.000000	2.030832e+05	2.026522e+05	87.000000	2.000000	8.800000	175.000000
max	272.000000	2.373711e+06	2.415841e+06	114.000000	5.000000	29.500000	380.000000

In [45]:
```python
# Total Sales by Day
df_SalesbyDate = df_transaction[['DATE','TOT_SALES']].groupby(['DATE']).sum().reset_index()
Date = pd.DataFrame(pd.date_range(start='2018-07-01', end='2019-06-30')).rename({0:'DATE'},
axis = 1)
```

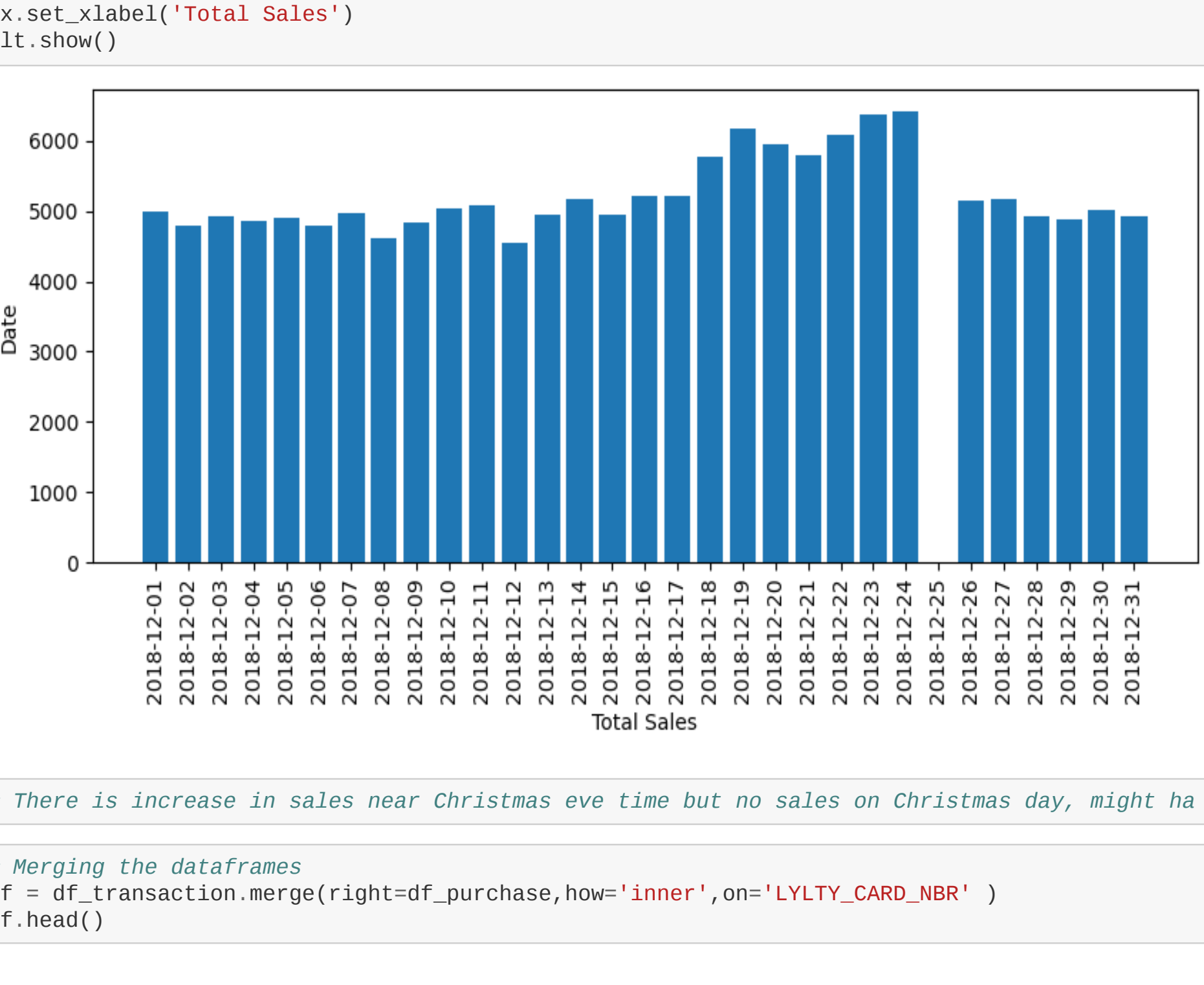
In [46]:
```python
plt.figure(figsize=(17,8))
plt.plot(df_SalesbyDate['DATE'], df_SalesbyDate['TOT_SALES'])
plt.show()
```



In [47]:
```python
# There is missing sales value for a date in Decemeber, let's zoom in December
December = df_SalesbyDate[(df_SalesbyDate['DATE'] >= '2018-12-01') & (df_SalesbyDate['DATE']
<= '2018-12-31')]
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 31 entries, 153 to 183
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype         
---  ------     --------------  -----         
 0   DATE       31 non-null     datetime64[ns]
 1   TOT_SALES  30 non-null     float64       
dtypes: datetime64[ns](1), float64(1)
memory usage: 744.0 bytes
```

In [48]:
```python
fig, ax = plt.subplots(figsize=(10,4),dpi=100)
ax.bar(December['DATE'],December['TOT_SALES'])
plt.xticks(December['DATE'], rotation=90)
ax.set_ylabel('Date')
ax.set_xlabel('Total Sales')
plt.show()
```



In [49]:
```python
# There is increase in sales near Christmas eve time but no sales on Christmas day, might ha
```

In [50]:
```python
# Merging the dataframes
df = df_transaction.merge(right=df_purchase,how='inner',on='LYLTY_CARD_NBR' )
df.head()
```

Out[50]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	BRAND_NAME	PACI
0	2018-10-17	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0	Natural	
1	2019-05-14	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3	CCs	
2	2018-11-10	1	1307	346	96	WW Original Stacked Chips 160g	2	3.8	Woolworths	
3	2019-03-09	1	1307	347	54	CCs Original 175g	1	2.1	CCs	
4	2019-05-20	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9	Smiths	

In [51]:
```python
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   BRAND_NAME        246740 non-null  object        
 9   PACK_SIZE         246740 non-null  int32         
 10  LIFESTAGE         246740 non-null  object        
 11  PREMIUM_CUSTOMER  246740 non-null  object        
dtypes: datetime64[ns](1), float64(1), int32(1), int64(5), object(4)
memory usage: 23.5+ MB
```

In [52]:
```python
# Converting the Clean dataframe to csv file for Analysis
df.to_csv('Clean_QV_Data.csv', index=False)
```

This notebook contains EDA only, for Data Analysis proceed to Analysis notebook.