In [2]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
```

In [3]:
```
transaction_data = pd.read_excel("QVI_transaction_data.xlsx")
```

In [4]:
```
transaction_data.head()
```

Out[4]:

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

In [5]:
```
transaction_data.describe()
```

Out[5]:

	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PRODUCT_NBR	PRODUCT_QTY	TOTAL_SAL
count	264836.00000	2.648360e+05	2.648360e+05	264836.000000	264836.000000	264836.0000
mean	135.08011	1.355495e+05	1.351583e+05	56.583157	1.907309	7.3042
std	76.78418	8.057998e+04	7.813303e+04	32.826638	0.643654	3.0832
min	1.00000	1.000000e+03	1.000000e+00	1.000000	1.000000	1.5000
25%	70.00000	7.002100e+04	6.760150e+04	28.000000	2.000000	5.4000
50%	130.00000	1.303575e+05	1.351375e+05	56.000000	2.000000	7.4000
75%	203.00000	2.030942e+05	2.027012e+05	85.000000	2.000000	9.2000
max	272.00000	2.373711e+06	2.415841e+06	114.000000	200.000000	650.0000

In [6]:
```
purchase_behaviour = pd.read_csv("QVI_purchase_behaviour.csv")
```

In [7]:
```
purchase_behaviour.head()
```

Out[7]:

	LYLTY_CARD_NBR	LIFESTAGE	PREMIUM_CUSTOMER	Unnamed: 3	Unnamed: 4	Unnamed: 5
0	1000	YOUNG SINGLES/COUPLES	Premium	NaN	NaN	NaN
1	1002	YOUNG SINGLES/COUPLES	Mainstream	NaN	NaN	NaN
2	1003	YOUNG FAMILIES	Budget	NaN	NaN	NaN
3	1004	OLDER SINGLES/COUPLES	Mainstream	NaN	NaN	NaN
4	1005	MIDAGE SINGLES/COUPLES	Mainstream	NaN	NaN	NaN

In [8]:
```
purchase_behaviour.describe()
```

Out[8]:

	LYLTY_CARD_NBR	Unnamed: 3	Unnamed: 4	Unnamed: 5
count	7.263700e+04	0.0	0.0	0.0
mean	1.361859e+05	NaN	NaN	NaN
std	8.989293e+04	NaN	NaN	NaN
min	1.000000e+03	NaN	NaN	NaN
25%	6.620200e+04	NaN	NaN	NaN
50%	1.340400e+05	NaN	NaN	NaN
75%	2.033750e+05	NaN	NaN	NaN
max	2.373711e+06	NaN	NaN	NaN

In [9]:
```
transaction_data.isnull().sum()
```

Out[9]:
```
DATE              0
STORE_NBR         0
LYLTY_CARD_NBR    0
TXN_ID            0
PRODUCT_NBR       0
PRODUCT_NAME      0
PRODUCT_QTY       0
TOTAL_SALES       0
dtype: int64
```

In [10]:
```
purchase_behaviour.isnull().sum()
```

Out[10]:
```
LYLTY_CARD_NBR          0
LIFESTAGE               0
PREMIUM_CUSTOMER        0
Unnamed: 3          72637
Unnamed: 4          72637
Unnamed: 5          72637
Unnamed: 6          72636
dtype: int64
```

CLEAN THE DATA THAT IS UNNAMED 3,4,5,6 BECAUSE THERE IS NOT VALUES IN

COLUMN

In [11]:
```
purchase_behaviour = purchase_behaviour.drop(columns=['Unnamed: 3', 'Unnamed: 4', '
```

In [12]:
```
print(purchase_behaviour[['LYLTY_CARD_NBR', 'LIFESTAGE', 'PREMIUM_CUSTOMER']].head(
```

```
   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 [13]:
```
purchase_behaviour.isnull().sum()
```

Out[13]:
```
LYLTY_CARD_NBR      0
LIFESTAGE           0
PREMIUM_CUSTOMER    0
dtype: int64
```

In [14]:
```
merged_data = pd.merge(purchase_behaviour, transaction_data, on = 'LYLTY_CARD_NBR',
merged_data.head()
```

Out[14]:

	LYLTY_CARD_NBR	LIFESTAGE	PREMIUM_CUSTOMER	DATE	STORE_NBR	TXN_ID	PRODU
0	1000	YOUNG SINGLES/COUPLES	Premium	2018-10-17	1	1	
1	1307	MIDAGE SINGLES/COUPLES	Budget	2019-05-14	1	348	
2	1343	MIDAGE SINGLES/COUPLES	Budget	2019-05-20	1	383	
3	2373	MIDAGE SINGLES/COUPLES	Budget	2018-08-17	2	974	
4	2426	MIDAGE SINGLES/COUPLES	Budget	2018-08-18	2	1038	

In [15]:
```
print(len(merged_data))
print(len(transaction_data))
```

```
264836
264836
```

In [16]:
```
merged_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 264836 entries, 0 to 264835
Data columns (total 10 columns):
 #   Column            Non-Null Count   Dtype         
---  ------            --------------   -----         
 0   LYLTY_CARD_NBR    264836 non-null  int64         
 1   LIFESTAGE         264836 non-null  object        
 2   PREMIUM_CUSTOMER  264836 non-null  object        
 3   DATE              264836 non-null  datetime64[ns]
 4   STORE_NBR         264836 non-null  int64         
 5   TXN_ID            264836 non-null  int64         
 6   PRODUCT_NBR       264836 non-null  int64         
 7   PRODUCT_NAME      264836 non-null  object        
 8   PRODUCT_QTY       264836 non-null  int64         
 9   TOTAL_SALES       264836 non-null  float64       
dtypes: datetime64[ns](1), float64(1), int64(5), object(3)
memory usage: 22.2+ MB
```

Checking the product name column

In [17]:
```
merged_data["PRODUCT_NAME"].unique()
```

Out[17]:
```
array(['Natural Chip        Compny SeaSalt175g',
       'CCs Nacho Cheese    175g',
       'Smiths Crinkle Cut  Chips Chicken 170g',
       'Smiths Chip Thinly  S/Cream&Onion 175g',
       'Kettle Tortilla ChpsHny&Jlpno Chili 150g',
       'Old El Paso Salsa   Dip Tomato Mild 300g',
       'Smiths Crinkle Chips Salt & Vinegar 330g',
       'Grain Waves         Sweet Chilli 210g',
       'Doritos Corn Chip Mexican Jalapeno 150g',
       'Grain Waves Sour    Cream&Chives 210G',
       'Kettle Sensations   Siracha Lime 150g',
       'Twisties Cheese     270g', 'WW Crinkle Cut      Chicken 175g',
       'Thins Chips Light&  Tangy 175g', 'CCs Original 175g',
       'Burger Rings 220g', 'NCC Sour Cream &    Garden Chives 175g',
       'Doritos Corn Chip Southern Chicken 150g',
       'Cheezels Cheese Box 125g', 'Smiths Crinkle      Original 330g',
       'Infzns Crn Crnchers Tangy Gcamole 110g',
       'Kettle Sea Salt     And Vinegar 175g',
       'Smiths Chip Thinly  Cut Original 175g', 'Kettle Original 175g',
       'Red Rock Deli Thai  Chilli&Lime 150g',
       'Pringles Sthrn FriedChicken 134g', 'Pringles Sweet&Spcy BBQ 134g',
       'Red Rock Deli SR    Salsa & Mzzrlla 150g',
       'Thins Chips         Originl saltd 175g',
       'Red Rock Deli Sp    Salt & Truffle 150G',
       'Smiths Thinly       Swt Chli&S/Cream175G', 'Kettle Chilli 175g',
       'Doritos Mexicana    170g',
       'Smiths Crinkle Cut  French OnionDip 150g',
       'Natural ChipCo      Hony Soy Chckn175g',
       'Dorito Corn Chp     Supreme 380g', 'Twisties Chicken270g',
       'Smiths Thinly Cut   Roast Chicken 175g',
       'Smiths Crinkle Cut  Tomato Salsa 150g',
       'Kettle Mozzarella   Basil & Pesto 175g',
       'Infuzions Thai SweetChili PotatoMix 110g',
       'Kettle Sensations   Camembert & Fig 150g',
       'Smith Crinkle Cut   Mac N Cheese 150g',
       'Kettle Honey Soy    Chicken 175g',
       'Thins Chips Seasonedchicken 175g',
       'Smiths Crinkle Cut  Salt & Vinegar 170g',
       'Infuzions BBQ Rib   Prawn Crackers 110g',
       'GrnWves Plus Btroot & Chilli Jam 180g',
       'Tyrrells Crisps     Lightly Salted 165g',
       'Kettle Sweet Chilli And Sour Cream 175g',
       'Doritos Salsa       Medium 300g', 'Kettle 135g Swt Pot Sea Salt',
       'Pringles SourCream  Onion 134g',
       'Doritos Corn Chips  Original 170g',
       'Twisties Cheese     Burger 250g',
       'Old El Paso Salsa   Dip Chnky Tom Ht300g',
       'Cobs Popd Swt/Chlli &Sr/Cream Chips 110g',
       'Woolworths Mild     Salsa 300g',
       'Natural Chip Co     Tmato Hrb&Spce 175g',
       'Smiths Crinkle Cut  Chips Original 170g',
       'Cobs Popd Sea Salt  Chips 110g',
       'Smiths Crinkle Cut  Chips Chs&Onion170g',
       'French Fries Potato Chips 175g',
       'Old El Paso Salsa   Dip Tomato Med 300g',
       'Doritos Corn Chips  Cheese Supreme 170g',
       'Pringles Original   Crisps 134g',
       'RRD Chilli&         Coconut 150g',
       'WW Original Corn    Chips 200g',
       'Thins Potato Chips  Hot & Spicy 175g',
       'Cobs Popd Sour Crm  &Chives Chips 110g',
       'Smiths Crnkle Chip  Orgnl Big Bag 380g',
       'Doritos Corn Chips  Nacho Cheese 170g',
       'Kettle Sensations   BBQ&Maple 150g',
```

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

In [18]:
```
split_prods = merged_data["PRODUCT_NAME"].str.replace(r'([0-9]+[gG])','').str.repla
```

```
C:\Users\Ashish\AppData\Local\Temp\ipykernel_16548\2093013135.py:1: FutureWarning:
The default value of regex will change from True to False in a future version.
  split_prods = merged_data["PRODUCT_NAME"].str.replace(r'([0-9]+[gG])','').str.re
place(r'[^\w]',' ').str.split()
```

In [19]:
```
word_counts = {}
def count_words(line):
  for word in line:
    if word not in word_counts:
      word_counts[word] = 1
    else:
      word_counts[word] += 1
split_prods.apply(lambda line: count_words(line))
print(pd.Series(word_counts).sort_values(ascending = False))
```

```
Chips       49770
Kettle      41288
Smiths      28860
Salt        27976
Cheese      27890
            ...
Sunbites     1432
Pc           1431
Garden       1419
NCC          1419
Fries        1418
Length: 198, dtype: int64
```

In [20]:
```
print(merged_data.describe(),'\n')
```

```
       LYLTY_CARD_NBR     STORE_NBR        TXN_ID    PRODUCT_NBR  \
count    2.648360e+05  264836.00000  2.648360e+05  264836.000000
mean     1.355495e+05     135.08011  1.351583e+05      56.583157
std      8.057998e+04      76.78418  7.813303e+04      32.826638
min      1.000000e+03       1.00000  1.000000e+00       1.000000
25%      7.002100e+04      70.00000  6.760150e+04      28.000000
50%      1.303575e+05     130.00000  1.351375e+05      56.000000
75%      2.030942e+05     203.00000  2.027012e+05      85.000000
max      2.373711e+06     272.00000  2.415841e+06     114.000000

         PRODUCT_QTY    TOTAL_SALES
count  264836.000000  264836.000000
mean        1.907309       7.304200
std         0.643654       3.083226
min         1.000000       1.500000
25%         2.000000       5.400000
50%         2.000000       7.400000
75%         2.000000       9.200000
max       200.000000     650.000000
```

In [21]:
```
print(merged_data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 264836 entries, 0 to 264835
Data columns (total 10 columns):
 #   Column            Non-Null Count   Dtype         
---  ------            --------------   -----         
 0   LYLTY_CARD_NBR    264836 non-null  int64         
 1   LIFESTAGE         264836 non-null  object        
 2   PREMIUM_CUSTOMER  264836 non-null  object        
 3   DATE              264836 non-null  datetime64[ns]
 4   STORE_NBR         264836 non-null  int64         
 5   TXN_ID            264836 non-null  int64         
 6   PRODUCT_NBR       264836 non-null  int64         
 7   PRODUCT_NAME      264836 non-null  object        
 8   PRODUCT_QTY       264836 non-null  int64         
 9   TOTAL_SALES       264836 non-null  float64       
dtypes: datetime64[ns](1), float64(1), int64(5), object(3)
memory usage: 22.2+ MB
None
```

In [22]:
```
merged_data["PRODUCT_QTY"].value_counts(bins=4).sort_index()
```

Out[22]:
```
(0.8, 50.75]       264834
(50.75, 100.5]          0
(100.5, 150.25]         0
(150.25, 200.0]         2
Name: PRODUCT_QTY, dtype: int64
```

In [23]:
```
merged_data.sort_values(by="PRODUCT_QTY", ascending=False).head()
```

Out[23]:

	LYLTY_CARD_NBR	LIFESTAGE	PREMIUM_CUSTOMER	DATE	STORE_NBR	TXN_ID
69762	226000	OLDER FAMILIES	Premium	2018-08-19	226	226201
69763	226000	OLDER FAMILIES	Premium	2019-05-20	226	226210
217237	201060	YOUNG FAMILIES	Premium	2019-05-18	201	200202
238333	219004	YOUNG SINGLES/COUPLES	Mainstream	2018-08-14	219	218018
238471	261331	YOUNG SINGLES/COUPLES	Mainstream	2019-05-19	261	261111

In [24]:
```
merged_data = merged_data[merged_data["PRODUCT_QTY"] < 6]
```

In [25]:
```
len(merged_data[merged_data["LYLTY_CARD_NBR"]==226000])
```

Out[25]:
```
0
```

In [26]:
```
merged_data["DATE"].describe()
```

```
C:\Users\Ashish\AppData\Local\Temp\ipykernel_16548\550868082.py:1: FutureWarning:
Treating datetime data as categorical rather than numeric in `.describe` is deprec
ated and will be removed in a future version of pandas. Specify `datetime_is_numer
ic=True` to silence this warning and adopt the future behavior now.
  merged_data["DATE"].describe()
```

Out[26]:
```
count                  264834
unique                    364
top       2018-12-24 00:00:00
freq                      939
first     2018-07-01 00:00:00
last      2019-06-30 00:00:00
Name: DATE, dtype: object
```

There are 365 days in a year but in the DATE column there are only 364 unique values so one is missing

In [27]:
```
pd.date_range(start=merged_data["DATE"].min(), end=merged_data["DATE"].max()).diffe
```

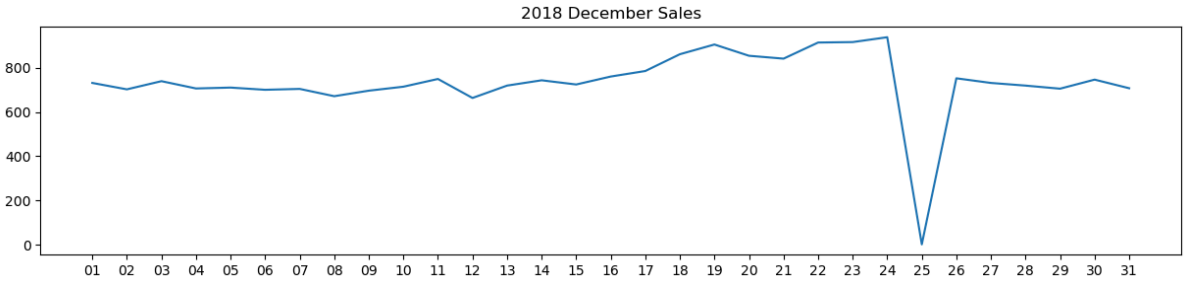
Out[27]:
```
DatetimeIndex(['2018-12-25'], dtype='datetime64[ns]', freq=None)
```

In [28]:
```
check_null_date = pd.merge(pd.Series(pd.date_range(start=merged_data["DATE"].min(),
```

In [29]:
```
trans_by_date = check_null_date["DATE"].value_counts()
dec = trans_by_date[(trans_by_date.index >= pd.datetime(2018,12,1)) & (trans_by_dat
dec.index = dec.index.strftime('%d')
ax = dec.plot(figsize=(15,3))
ax.set_xticks(np.arange(len(dec)))
ax.set_xticklabels(dec.index)
plt.title("2018 December Sales")
plt.savefig("2018 December Sales.png", bbox_inches="tight")
```

```
plt.show()
```

```
C:\Users\Ashish\AppData\Local\Temp\ipykernel_16548\4065098678.py:2: FutureWarning:
The pandas.datetime class is deprecated and will be removed from pandas in a futur
e version. Import from datetime module instead.
  dec = trans_by_date[(trans_by_date.index >= pd.datetime(2018,12,1)) & (trans_by_
date.index < pd.datetime(2019,1,1))].sort_index()
```



In [30]:
```
check_null_date["DATE"].value_counts().sort_values().head()
```

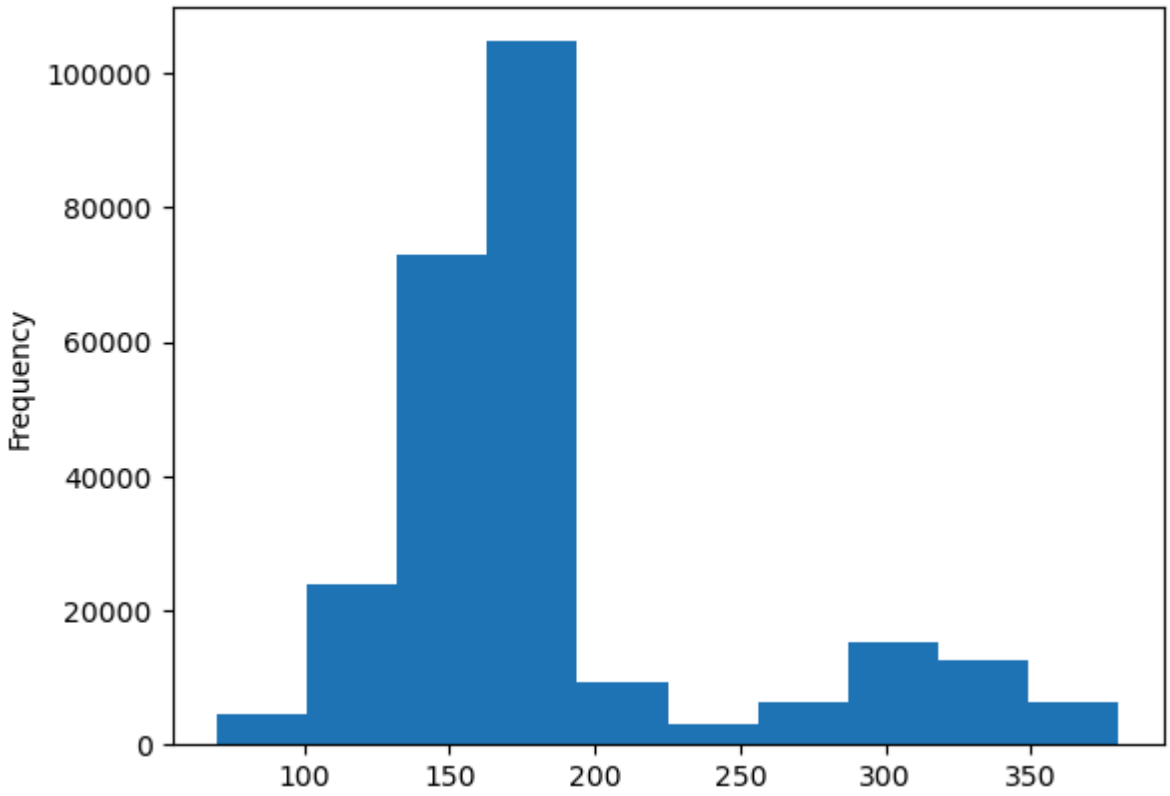
Out[30]:
```
2018-12-25      1
2018-11-25    648
2018-10-18    658
2019-06-13    659
2019-06-24    662
Name: DATE, dtype: int64
```

Explore Packet sizes

In [31]:
```
merged_data["PRODUCT_NAME"] = merged_data["PRODUCT_NAME"].str.replace(r'[0-9]+(G)',
pack_sizes = merged_data["PRODUCT_NAME"].str.extract(r'([0-9]+[gG])')[0].str.replac
print(pack_sizes.describe())
pack_sizes.plot.hist()
```

```
C:\Users\Ashish\AppData\Local\Temp\ipykernel_16548\356870788.py:1: FutureWarning:
The default value of regex will change from True to False in a future version.
  merged_data["PRODUCT_NAME"] = merged_data["PRODUCT_NAME"].str.replace(r'[0-9]+
(G)','g')
count    258770.000000
mean        182.324276
std          64.955035
min          70.000000
25%         150.000000
50%         170.000000
75%         175.000000
max         380.000000
Name: 0, dtype: float64
```

Out[31]:
```
<Axes: ylabel='Frequency'>
```



In [32]:
```
merged_data["PRODUCT_NAME"].str.split().str[0].value_counts().sort_index()
```

Out[32]:
```
Burger         1564
CCs            4551
Cheetos        2927
Cheezels       4603
Cobs           9693
Dorito         3183
Doritos       24962
French         1418
Grain          6272
GrnWves        1468
Infuzions     11057
Infzns         3144
Kettle        41288
NCC            1419
Natural        6050
Old            9324
Pringles      25102
RRD           11894
Red            5885
Smith          2963
Smiths        28860
Snbts          1576
Sunbites       1432
Thins         14075
Tostitos       9471
Twisties       9454
Tyrrells       6442
WW            10320
Woolworths     4437
Name: PRODUCT_NAME, dtype: int64
```

In [33]:
```
merged_data["PRODUCT_NAME"].str.split()[merged_data["PRODUCT_NAME"].str.split().str
```

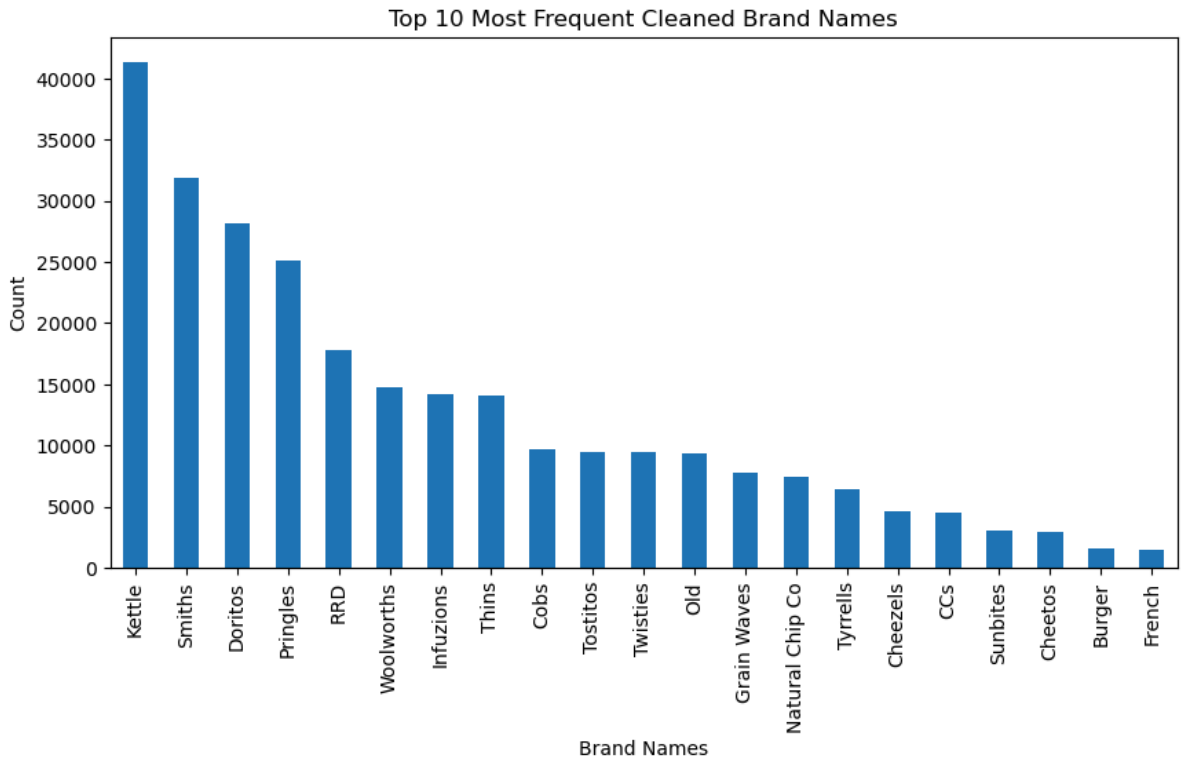
Out[33]:
```
[Red, Rock, Deli, Sp, Salt, &, Truffle, g]        1498
[Red, Rock, Deli, Thai, Chilli&Lime, 150g]        1495
[Red, Rock, Deli, SR, Salsa, &, Mzzrlla, 150g]    1458
[Red, Rock, Deli, Chikn&Garlic, Aioli, 150g]      1434
Name: PRODUCT_NAME, dtype: int64
```

In [34]:
```
merged_data["Cleaned_Brand_Names"] = merged_data["PRODUCT_NAME"].str.split().str[0]
```

In [35]:
```
def clean_brand_names(line):
    brand = line["Cleaned_Brand_Names"]
    if brand == "Dorito":
        return "Doritos"
    elif brand == "GrnWves" or brand == "Grain":
        return "Grain Waves"
    elif brand == "Infzns":
        return "Infuzions"
    elif brand == "Natural" or brand == "NCC":
        return "Natural Chip Co"
    elif brand == "Red":
        return "RRD"
    elif brand == "Smith":
        return "Smiths"
    elif brand == "Snbts":
        return "Sunbites"
    elif brand == "WW":
        return "Woolworths"
    else:
        return brand
```

In [36]:
```
merged_data["Cleaned_Brand_Names"] = merged_data.apply(lambda line: clean_brand_nam
```

In [37]:
```
merged_data["Cleaned_Brand_Names"].value_counts().nlargest(22).plot.bar(figsize=(10

# Add labels and title
plt.title("Top 10 Most Frequent Cleaned Brand Names")
plt.xlabel("Brand Names")
plt.ylabel("Count")

# Display the plot
plt.show()
```



In [38]:
```
merged_data.isnull().sum()
```

Out[38]:
```
LYLTY_CARD_NBR          0
LIFESTAGE               0
PREMIUM_CUSTOMER        0
DATE                    0
STORE_NBR               0
TXN_ID                  0
PRODUCT_NBR             0
PRODUCT_NAME            0
PRODUCT_QTY             0
TOTAL_SALES             0
Cleaned_Brand_Names     0
dtype: int64
```

Who spends the most on chips (total sales), describing customers by lifestage and how premium their general purchasing behaviour is

In [47]:
```
grouped_sales = pd.DataFrame(merged_data.groupby(["LIFESTAGE", "PREMIUM_CUSTOMER"])
grouped_sales.sort_values(ascending=False, by="sum")
```

Out[47]:

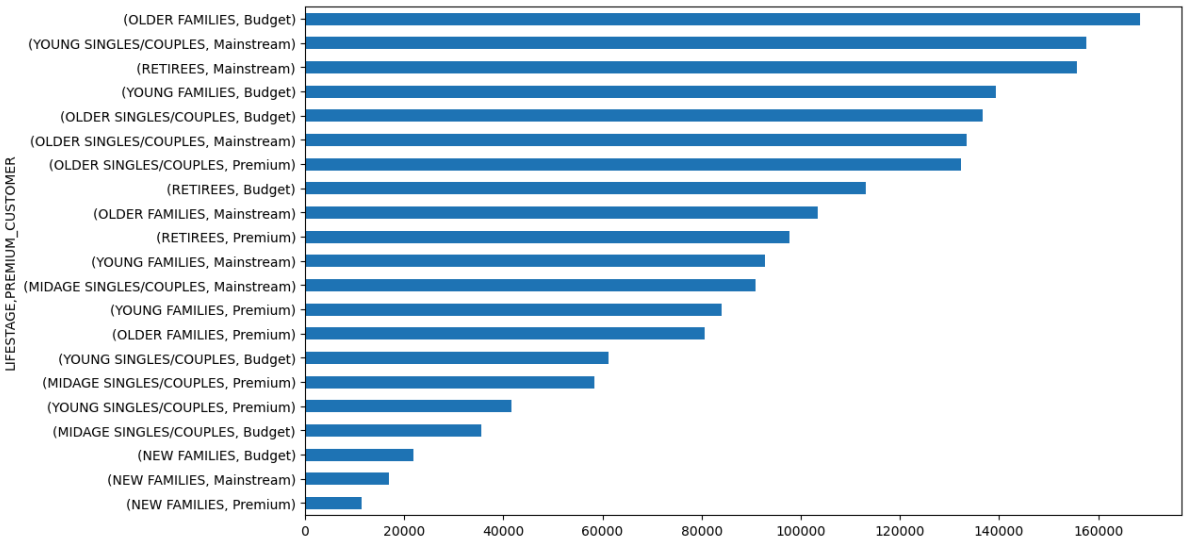
LIFESTAGE	PREMIUM_CUSTOMER	sum	mean
OLDER FAMILIES	Budget	168363.25	7.269570
YOUNG SINGLES/COUPLES	Mainstream	157621.60	7.558339
RETIREES	Mainstream	155677.05	7.252262
YOUNG FAMILIES	Budget	139345.85	7.287201
OLDER SINGLES/COUPLES	Budget	136769.80	7.430315
	Mainstream	133393.80	7.282116
	Premium	132263.15	7.449766
RETIREES	Budget	113147.80	7.443445
OLDER FAMILIES	Mainstream	103445.55	7.262395
RETIREES	Premium	97646.05	7.456174
YOUNG FAMILIES	Mainstream	92788.75	7.189025
MIDAGE SINGLES/COUPLES	Mainstream	90803.85	7.647284
YOUNG FAMILIES	Premium	84025.50	7.266756
OLDER FAMILIES	Premium	80658.40	7.208079
YOUNG SINGLES/COUPLES	Budget	61141.60	6.615624
MIDAGE SINGLES/COUPLES	Premium	58432.65	7.112056
YOUNG SINGLES/COUPLES	Premium	41642.10	6.629852
MIDAGE SINGLES/COUPLES	Budget	35514.80	7.074661
NEW FAMILIES	Budget	21928.45	7.297321
	Mainstream	17013.90	7.317806
	Premium	11491.10	7.231655

In [48]:
```
grouped_sales["sum"].sum()
```

Out[48]:
```
1933115.0000000002
```

In [50]:
```
grouped_sales["sum"].sort_values().plot.barh(figsize=(12,7))
```

Out[50]:
```
<Axes: ylabel='LIFESTAGE,PREMIUM_CUSTOMER'>
```

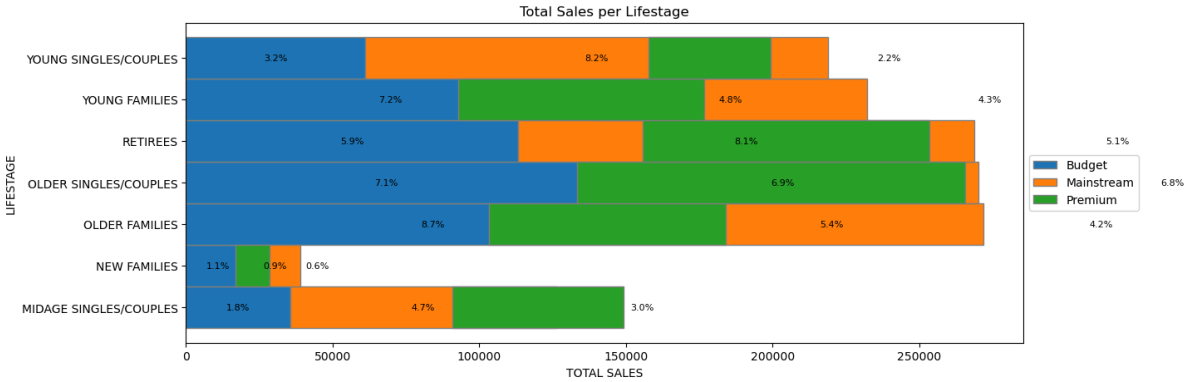


How many chips are bought per customer by segment

In [51]:

```
# Values of each group
bars1 = grouped_sales[grouped_sales.index.get_level_values("PREMIUM_CUSTOMER") == '
bars2 = grouped_sales[grouped_sales.index.get_level_values("PREMIUM_CUSTOMER") == '
bars3 = grouped_sales[grouped_sales.index.get_level_values("PREMIUM_CUSTOMER") == '

bars1_text = (bars1 / sum(grouped_sales["sum"])).apply("{:.1%}".format)
bars2_text = (bars2 / sum(grouped_sales["sum"])).apply("{:.1%}".format)
bars3_text = (bars3 / sum(grouped_sales["sum"])).apply("{:.1%}".format)

# Names of group and bar width
names = grouped_sales.index.get_level_values("LIFESTAGE").unique()

# The position of the bars on the x-axis
r = np.arange(len(names))

plt.figure(figsize=(13,5))

# Create brown bars
budget_bar = plt.barh(r, bars1, edgecolor='grey', height=1, label="Budget")
# Create green bars (middle), on top of the firs ones
mains_bar = plt.barh(r, bars2, left=bars1, edgecolor='grey', height=1, label="Mains
# Create green bars (top)
tmp_bar = np.add(bars1, bars2)
prem_bar = plt.barh(r, bars3, left=bars2, edgecolor='grey', height=1, label="Premiu

for i in range(7):
    budget_width = budget_bar[i].get_width()
    budget_main_width = budget_width + mains_bar[i].get_width()
    plt.text(budget_width/2, i, bars1_text[i], va='center', ha='center', size=8)
    plt.text(budget_width + mains_bar[i].get_width()/2, i, bars2_text[i], va='cente
    plt.text(budget_main_width + prem_bar[i].get_width()/2, i, bars3_text[i], va='c

# Custom X axis
plt.yticks(r, names)
plt.ylabel("LIFESTAGE")
plt.xlabel("TOTAL SALES")
plt.legend(loc='center left', bbox_to_anchor=(1.0, 0.5))

plt.title("Total Sales per Lifestage")

plt.savefig("lifestage_sales.png", bbox_inches="tight")
```

```
# Show graphic
plt.show()
```



In [52]:
```
stage_agg_prem = merged_data.groupby("LIFESTAGE")["PREMIUM_CUSTOMER"].agg(pd.Series
print("Top contributor per LIFESTAGE by PREMIUM category")
print(stage_agg_prem)
```

```
Top contributor per LIFESTAGE by PREMIUM category
LIFESTAGE
NEW FAMILIES                 Budget
OLDER FAMILIES               Budget
OLDER SINGLES/COUPLES        Budget
YOUNG FAMILIES               Budget
MIDAGE SINGLES/COUPLES    Mainstream
RETIREES                  Mainstream
YOUNG SINGLES/COUPLES     Mainstream
Name: PREMIUM_CUSTOMER, dtype: object
```

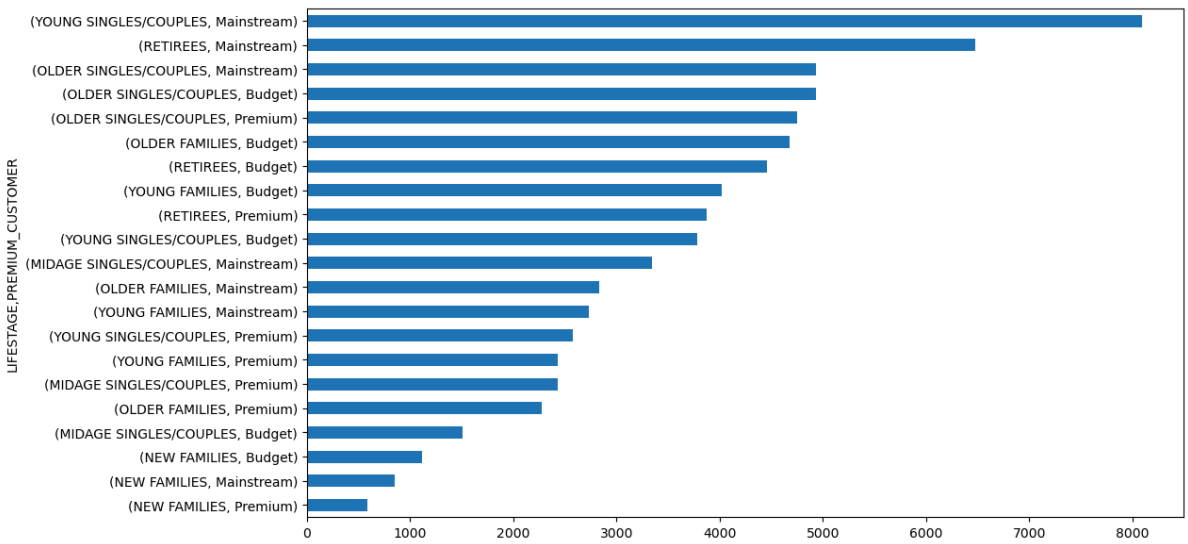
In [53]:
```
unique_cust = merged_data.groupby(["LIFESTAGE", "PREMIUM_CUSTOMER"])["LYLTY_CARD_NE
pd.DataFrame(unique_cust)
```

Out[53]:

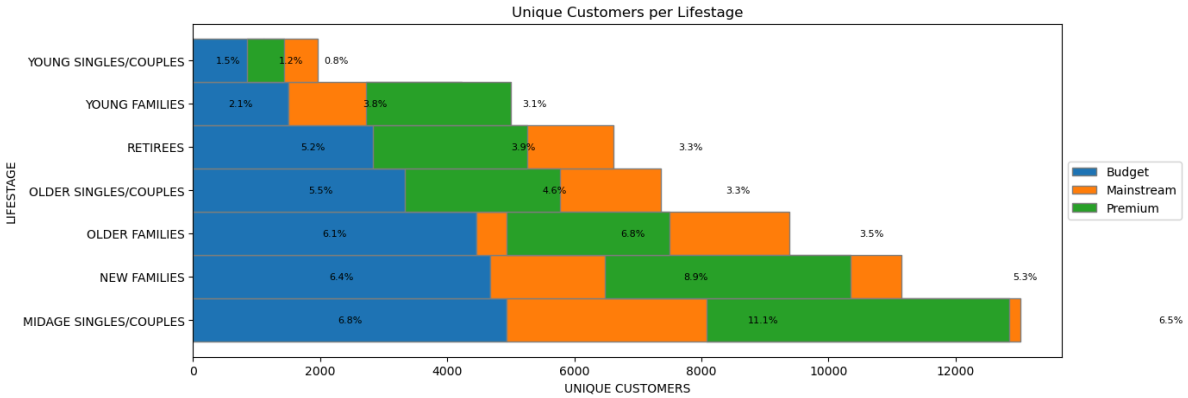
LIFESTAGE	PREMIUM_CUSTOMER	LYLTY_CARD_NBR
YOUNG SINGLES/COUPLES	Mainstream	8088
RETIREES	Mainstream	6479
OLDER SINGLES/COUPLES	Mainstream	4930
	Budget	4929
	Premium	4750
OLDER FAMILIES	Budget	4675
RETIREES	Budget	4454
YOUNG FAMILIES	Budget	4017
RETIREES	Premium	3872
YOUNG SINGLES/COUPLES	Budget	3779
MIDAGE SINGLES/COUPLES	Mainstream	3340
OLDER FAMILIES	Mainstream	2831
YOUNG FAMILIES	Mainstream	2728
YOUNG SINGLES/COUPLES	Premium	2574
YOUNG FAMILIES	Premium	2433
MIDAGE SINGLES/COUPLES	Premium	2431
OLDER FAMILIES	Premium	2273
MIDAGE SINGLES/COUPLES	Budget	1504
NEW FAMILIES	Budget	1112
	Mainstream	849
	Premium	588

In [54]:
```
unique_cust.sort_values().plot.barh(figsize=(12,7))
```

Out[54]:
```
<Axes: ylabel='LIFESTAGE,PREMIUM_CUSTOMER'>
```



In [55]:
```
# Values of each group
ncust_bars1 = unique_cust[unique_cust.index.get_level_values("PREMIUM_CUSTOMER") ==
ncust_bars2 = unique_cust[unique_cust.index.get_level_values("PREMIUM_CUSTOMER") ==
ncust_bars3 = unique_cust[unique_cust.index.get_level_values("PREMIUM_CUSTOMER") ==

ncust_bars1_text = (ncust_bars1 / sum(unique_cust)).apply("{:.1%}".format)
ncust_bars2_text = (ncust_bars2 / sum(unique_cust)).apply("{:.1%}".format)
ncust_bars3_text = (ncust_bars3 / sum(unique_cust)).apply("{:.1%}".format)

# # Names of group and bar width
#names = unique_cust.index.get_level_values("LIFESTAGE").unique()

# # The position of the bars on the x-axis
#r = np.arange(len(names))

plt.figure(figsize=(13,5))

# # Create brown bars
budget_bar = plt.barh(r, ncust_bars1, edgecolor='grey', height=1, label="Budget")
# # Create green bars (middle), on top of the firs ones
mains_bar = plt.barh(r, ncust_bars2, left=ncust_bars1, edgecolor='grey', height=1,
# # Create green bars (top)
prem_bar = plt.barh(r, ncust_bars3, left=ncust_bars2, edgecolor='grey', height=1, l

for i in range(7):
    budget_width = budget_bar[i].get_width()
    budget_main_width = budget_width + mains_bar[i].get_width()
    plt.text(budget_width/2, i, ncust_bars1_text[i], va='center', ha='center', size
    plt.text(budget_width + mains_bar[i].get_width()/2, i, ncust_bars2_text[i], va=
    plt.text(budget_main_width + prem_bar[i].get_width()/2, i, ncust_bars3_text[i],

# Custom X axis
plt.yticks(r, names)
plt.ylabel("LIFESTAGE")
plt.xlabel("UNIQUE CUSTOMERS")
plt.legend(loc='center left', bbox_to_anchor=(1.0, 0.5))

plt.title("Unique Customers per Lifestage")

plt.savefig("lifestage_customers.png", bbox_inches="tight")

# # Show graphic
plt.show()
```



In [56]:
```
freq_per_cust = merged_data.groupby(["LYLTY_CARD_NBR", "LIFESTAGE", "PREMIUM_CUSTOM
freq_per_cust.groupby(["LIFESTAGE", "PREMIUM_CUSTOMER"]).agg(["mean", "count"]).sor
```

Out[56]:

LIFESTAGE	PREMIUM_CUSTOMER	mean	count
OLDER FAMILIES	Mainstream	5.031438	2831
	Budget	4.954011	4675
	Premium	4.923009	2273
YOUNG FAMILIES	Budget	4.760269	4017
	Premium	4.752569	2433
	Mainstream	4.731305	2728
OLDER SINGLES/COUPLES	Premium	3.737684	4750
	Budget	3.734429	4929
	Mainstream	3.715619	4930
MIDAGE SINGLES/COUPLES	Mainstream	3.555090	3340
RETIREES	Budget	3.412887	4454
	Premium	3.382231	3872
MIDAGE SINGLES/COUPLES	Premium	3.379679	2431
	Budget	3.337766	1504
RETIREES	Mainstream	3.313166	6479
NEW FAMILIES	Mainstream	2.738516	849
	Premium	2.702381	588
	Budget	2.702338	1112
YOUNG SINGLES/COUPLES	Mainstream	2.578388	8088
	Budget	2.445621	3779
	Premium	2.440171	2574

In [57]:
```
grouped_sales.sort_values(ascending=False, by="mean")
```

Out[57]:

		sum	mean
LIFESTAGE	**PREMIUM_CUSTOMER**		
MIDAGE SINGLES/COUPLES	**Mainstream**	90803.85	7.647284
YOUNG SINGLES/COUPLES	**Mainstream**	157621.60	7.558339
RETIREES	**Premium**	97646.05	7.456174
OLDER SINGLES/COUPLES	**Premium**	132263.15	7.449766
RETIREES	**Budget**	113147.80	7.443445
OLDER SINGLES/COUPLES	**Budget**	136769.80	7.430315
NEW FAMILIES	**Mainstream**	17013.90	7.317806
	Budget	21928.45	7.297321
YOUNG FAMILIES	**Budget**	139345.85	7.287201
OLDER SINGLES/COUPLES	**Mainstream**	133393.80	7.282116
OLDER FAMILIES	**Budget**	168363.25	7.269570
YOUNG FAMILIES	**Premium**	84025.50	7.266756
OLDER FAMILIES	**Mainstream**	103445.55	7.262395
RETIREES	**Mainstream**	155677.05	7.252262
NEW FAMILIES	**Premium**	11491.10	7.231655
OLDER FAMILIES	**Premium**	80658.40	7.208079
YOUNG FAMILIES	**Mainstream**	92788.75	7.189025
MIDAGE SINGLES/COUPLES	**Premium**	58432.65	7.112056
	Budget	35514.80	7.074661
YOUNG SINGLES/COUPLES	**Premium**	41642.10	6.629852
	Budget	61141.60	6.615624

In [66]:
```
from scipy.stats import ttest_ind
mainstream = merged_data["PREMIUM_CUSTOMER"] == "Mainstream"
young_midage = (merged_data["LIFESTAGE"] == "MIDAGE SINGLES/COUPLES") | (merged_dat

budget_premium = (merged_data["PREMIUM_CUSTOMER"] == "Budget") | (merged_data["PREM

a = merged_data[young_midage & mainstream]["TOTAL_SALES"]
b = merged_data[young_midage & budget_premium]["TOTAL_SALES"]
stat, pval = ttest_ind(a.values, b.values, equal_var=False)

print(pval)
pval < 0.0000001
```

```
1.8542040107536954e-281
```

Out[66]:
```
True
```

In [67]:
```
merged_data.groupby(["LIFESTAGE", "PREMIUM_CUSTOMER"])["Cleaned_Brand_Names"].agg(p
```

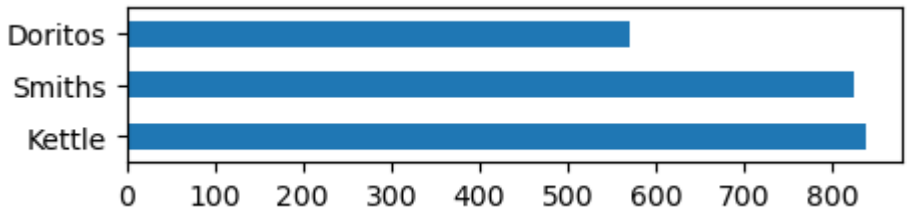
Out[67]:

LIFESTAGE	PREMIUM_CUSTOMER	
MIDAGE SINGLES/COUPLES	Budget	Kettle
YOUNG FAMILIES	Premium	Kettle
	Mainstream	Kettle
	Budget	Kettle
RETIREES	Premium	Kettle
	Mainstream	Kettle
	Budget	Kettle
OLDER SINGLES/COUPLES	Premium	Kettle
YOUNG SINGLES/COUPLES	Mainstream	Kettle
OLDER SINGLES/COUPLES	Mainstream	Kettle
OLDER FAMILIES	Mainstream	Kettle
	Budget	Kettle
NEW FAMILIES	Premium	Kettle
	Mainstream	Kettle
	Budget	Kettle
MIDAGE SINGLES/COUPLES	Premium	Kettle
	Mainstream	Kettle
OLDER SINGLES/COUPLES	Budget	Kettle
YOUNG SINGLES/COUPLES	Premium	Kettle
OLDER FAMILIES	Premium	Smiths
YOUNG SINGLES/COUPLES	Budget	Smiths

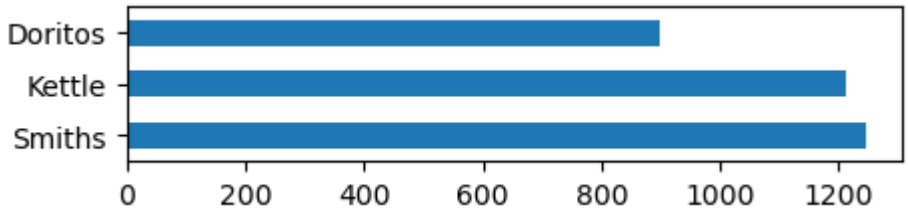
Name: Cleaned_Brand_Names, dtype: object

In [68]:
```
for stage in merged_data["LIFESTAGE"].unique():
    for prem in merged_data["PREMIUM_CUSTOMER"].unique():
        print('==========',stage, '-', prem,'==========')
        summary = merged_data[(merged_data["LIFESTAGE"] == stage) & (merged_data["P
        print(summary)
        plt.figure()
        summary.plot.barh(figsize=(5,1))
        plt.show()
```

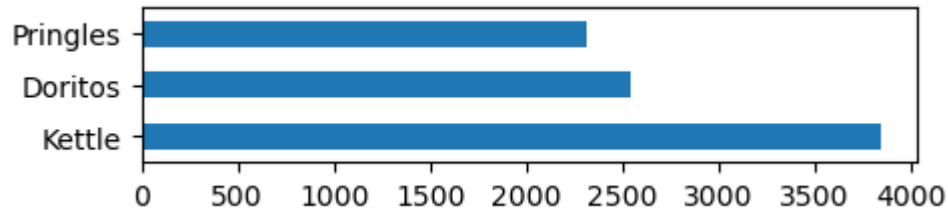
```
========== YOUNG SINGLES/COUPLES - Premium ==========
Kettle     838
Smiths     826
Doritos    570
Name: Cleaned_Brand_Names, dtype: int64
```



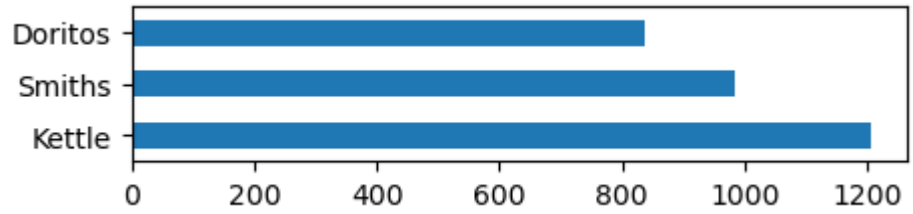

```
========== YOUNG SINGLES/COUPLES - Budget ==========
Smiths     1245
Kettle     1211
Doritos     899
Name: Cleaned_Brand_Names, dtype: int64
```



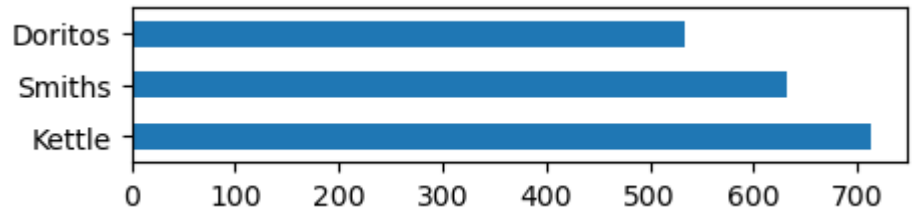

```
========== YOUNG SINGLES/COUPLES - Mainstream ==========
Kettle      3844
Doritos     2541
Pringles    2315
Name: Cleaned_Brand_Names, dtype: int64
```



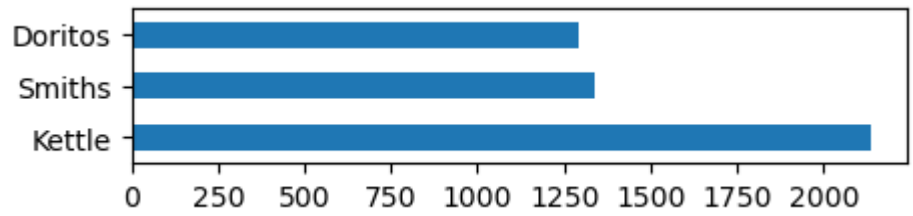

```
========== MIDAGE SINGLES/COUPLES - Premium ==========
Kettle     1206
Smiths      986
Doritos     837
Name: Cleaned_Brand_Names, dtype: int64
```



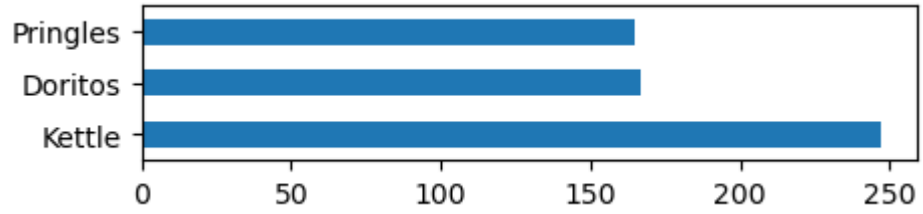

```
========== MIDAGE SINGLES/COUPLES - Budget ==========
Kettle     713
Smiths     633
Doritos    533
Name: Cleaned_Brand_Names, dtype: int64
```



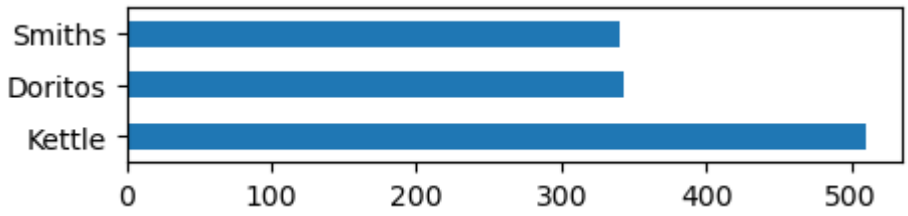

```
========== MIDAGE SINGLES/COUPLES - Mainstream ==========
Kettle     2136
Smiths     1337
Doritos    1291
Name: Cleaned_Brand_Names, dtype: int64
```



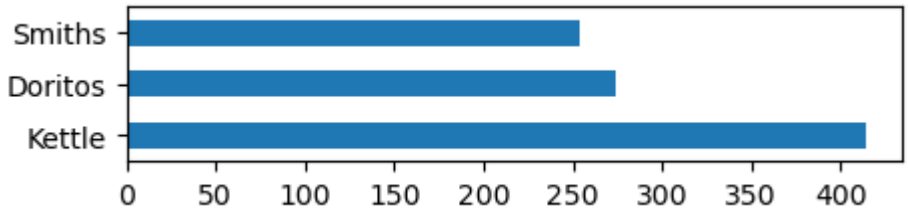

```
========== NEW FAMILIES - Premium ==========
Kettle      247
Doritos     167
Pringles    165
Name: Cleaned_Brand_Names, dtype: int64
```



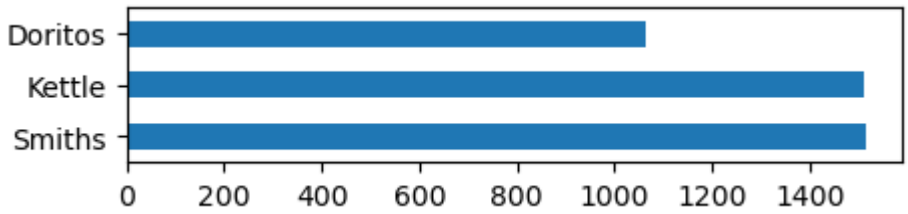

```
========== NEW FAMILIES - Budget ==========
Kettle     510
Doritos    343
Smiths     341
Name: Cleaned_Brand_Names, dtype: int64
```



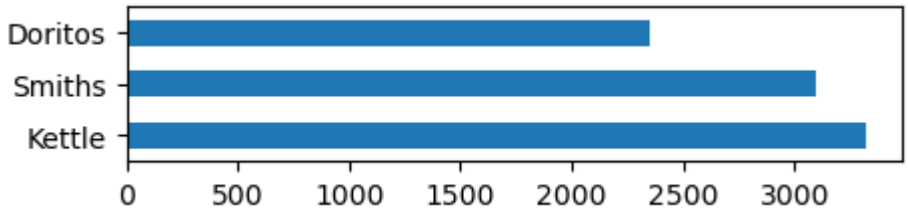

```
========== NEW FAMILIES - Mainstream ==========
Kettle     414
Doritos    274
Smiths     254
Name: Cleaned_Brand_Names, dtype: int64
```



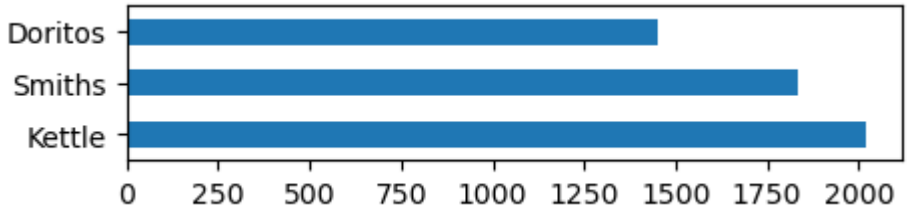

```
========== OLDER FAMILIES - Premium ==========
Smiths     1515
Kettle     1512
Doritos    1065
Name: Cleaned_Brand_Names, dtype: int64
```



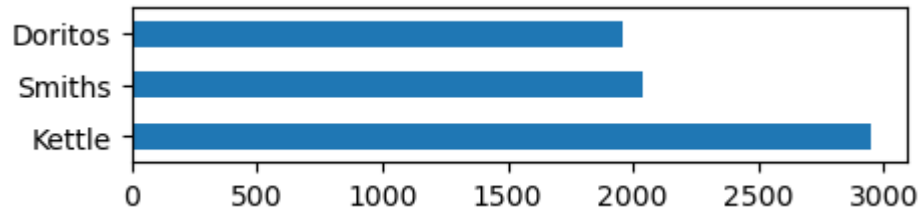

```
========== OLDER FAMILIES - Budget ==========
Kettle     3320
Smiths     3093
Doritos    2351
Name: Cleaned_Brand_Names, dtype: int64
```



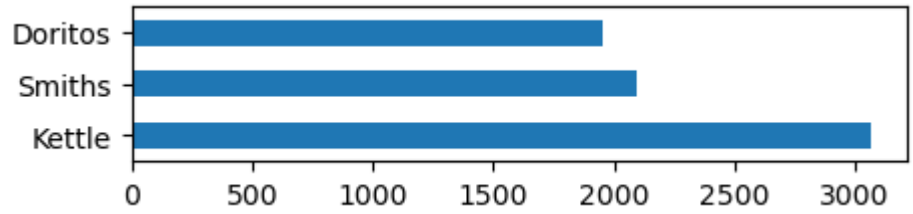

```
========== OLDER FAMILIES - Mainstream ==========
Kettle     2019
Smiths     1835
Doritos    1449
Name: Cleaned_Brand_Names, dtype: int64
```



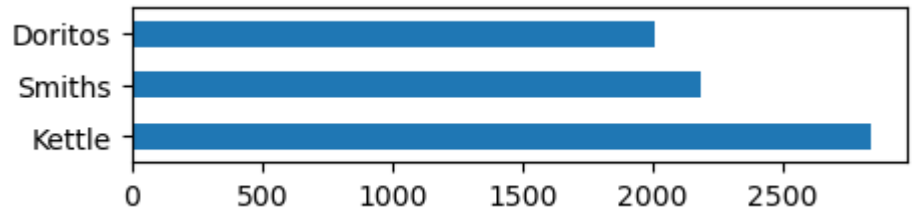

```
========== OLDER SINGLES/COUPLES - Premium ==========
Kettle     2947
Smiths     2042
Doritos    1958
Name: Cleaned_Brand_Names, dtype: int64
```



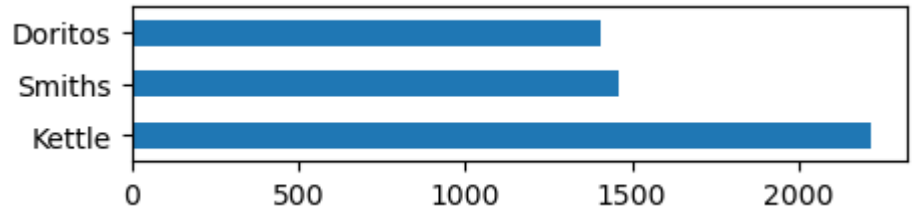

```
========== OLDER SINGLES/COUPLES - Budget ==========
Kettle     3065
Smiths     2098
Doritos    1954
Name: Cleaned_Brand_Names, dtype: int64
```



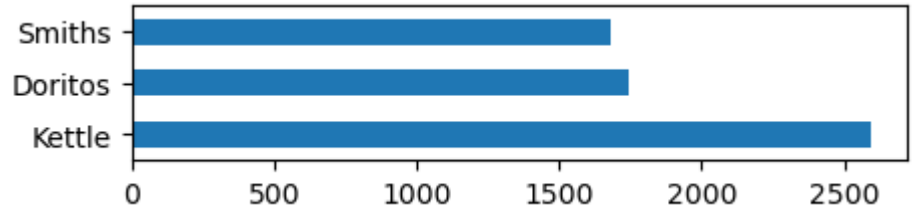

```
========== OLDER SINGLES/COUPLES - Mainstream ==========
Kettle     2835
Smiths     2180
Doritos    2008
Name: Cleaned_Brand_Names, dtype: int64
```



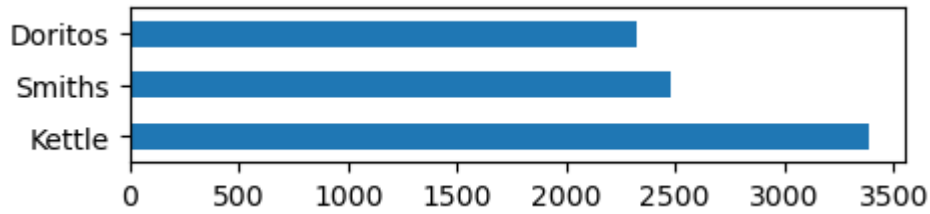

```
========== RETIREES - Premium ==========
Kettle     2216
Smiths     1458
Doritos    1409
Name: Cleaned_Brand_Names, dtype: int64
```



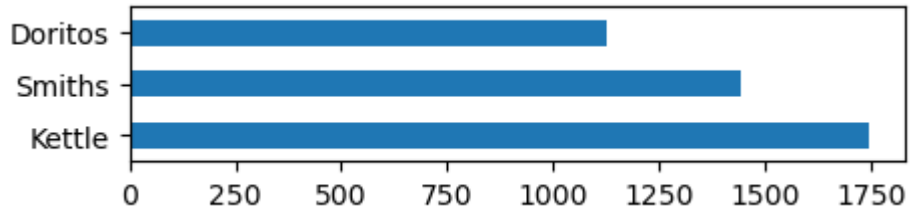

```
========== RETIREES - Budget ==========
Kettle     2592
Doritos    1742
Smiths     1679
Name: Cleaned_Brand_Names, dtype: int64
```



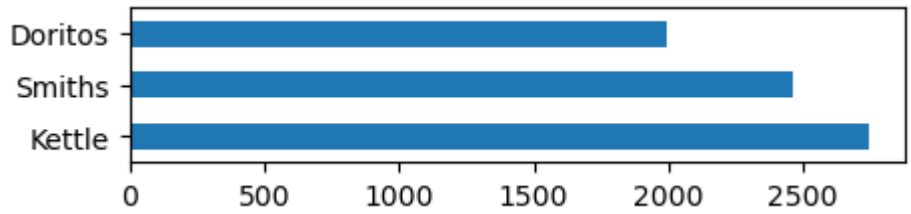

```
========== RETIREES - Mainstream ==========
Kettle     3386
Smiths     2476
Doritos    2320
Name: Cleaned_Brand_Names, dtype: int64
```



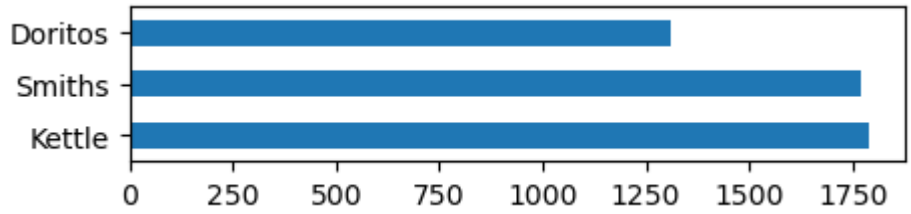

```
========== YOUNG FAMILIES - Premium ==========
Kettle     1745
Smiths     1442
Doritos    1129
Name: Cleaned_Brand_Names, dtype: int64
```




```
========== YOUNG FAMILIES - Budget ==========
Kettle     2743
Smiths     2459
Doritos    1996
Name: Cleaned_Brand_Names, dtype: int64
```




```
========== YOUNG FAMILIES - Mainstream ==========
Kettle     1789
Smiths     1772
Doritos    1309
Name: Cleaned_Brand_Names, dtype: int64
```



In [73]:
```
pip install mlxtend
```

```
Collecting mlxtend
  Downloading mlxtend-0.23.1-py3-none-any.whl (1.4 MB)
     ---------------------------------------- 0.0/1.4 MB ? eta -:--:--
     ----- ---------------------------------- 0.2/1.4 MB 6.1 MB/s eta 0:00:01
     ------------------------------ -------- 1.1/1.4 MB 14.3 MB/s eta 0:00:01
     ---------------------------------------- 1.4/1.4 MB 13.1 MB/s eta 0:00:00
Requirement already satisfied: scipy>=1.2.1 in c:\users\ashish\anaconda3\lib\site-
packages (from mlxtend) (1.10.1)
Requirement already satisfied: numpy>=1.16.2 in c:\users\ashish\anaconda3\lib\site
-packages (from mlxtend) (1.24.3)
Requirement already satisfied: pandas>=0.24.2 in c:\users\ashish\anaconda3\lib\sit
e-packages (from mlxtend) (1.5.3)
Requirement already satisfied: scikit-learn>=1.0.2 in c:\users\ashish\anaconda3\li
b\site-packages (from mlxtend) (1.3.0)
Requirement already satisfied: matplotlib>=3.0.0 in c:\users\ashish\anaconda3\lib
\site-packages (from mlxtend) (3.7.1)
Requirement already satisfied: joblib>=0.13.2 in c:\users\ashish\anaconda3\lib\sit
e-packages (from mlxtend) (1.2.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\ashish\anaconda3\lib\s
ite-packages (from matplotlib>=3.0.0->mlxtend) (1.0.5)
Requirement already satisfied: cycler>=0.10 in c:\users\ashish\anaconda3\lib\site-
packages (from matplotlib>=3.0.0->mlxtend) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\ashish\anaconda3\lib
\site-packages (from matplotlib>=3.0.0->mlxtend) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\ashish\anaconda3\lib
\site-packages (from matplotlib>=3.0.0->mlxtend) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\ashish\appdata\roaming
\python\python311\site-packages (from matplotlib>=3.0.0->mlxtend) (23.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\ashish\appdata\roaming\py
thon\python311\site-packages (from matplotlib>=3.0.0->mlxtend) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\ashish\anaconda3\lib\s
ite-packages (from matplotlib>=3.0.0->mlxtend) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\ashish\appdata\roa
ming\python\python311\site-packages (from matplotlib>=3.0.0->mlxtend) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\ashish\anaconda3\lib\site-
packages (from pandas>=0.24.2->mlxtend) (2022.7)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\ashish\anaconda3\l
ib\site-packages (from scikit-learn>=1.0.2->mlxtend) (2.2.0)
Requirement already satisfied: six>=1.5 in c:\users\ashish\appdata\roaming\python
\python311\site-packages (from python-dateutil>=2.7->matplotlib>=3.0.0->mlxtend)
(1.16.0)
Installing collected packages: mlxtend
Successfully installed mlxtend-0.23.1
Note: you may need to restart the kernel to use updated packages.
[notice] A new release of pip is available: 23.0.1 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [74]:

```
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

temp = merged_data.reset_index().rename(columns = {"index": "transaction"})
temp["Segment"] = temp["LIFESTAGE"] + ' - ' + temp['PREMIUM_CUSTOMER']
segment_brand_encode = pd.concat([pd.get_dummies(temp["Segment"]), pd.get_dummies(t

frequent_sets = apriori(segment_brand_encode, min_support=0.01, use_colnames=True)
rules = association_rules(frequent_sets, metric="lift", min_threshold=1)

set_temp = temp["Segment"].unique()
rules[rules["antecedents"].apply(lambda x: list(x)).apply(lambda x: x in set_temp)]
```

```
C:\Users\Ashish\anaconda3\Lib\site-packages\mlxtend\frequent_patterns\fpcommon.py:
109: DeprecationWarning: DataFrames with non-bool types result in worse computatio
nalperformance and their support might be discontinued in the future.Please use a
DataFrame with bool type
  warnings.warn(
```

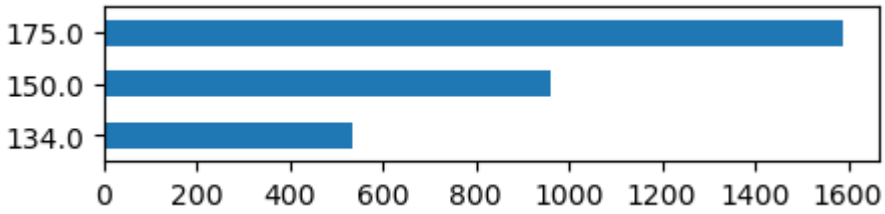
Out[74]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	levera
1	(OLDER FAMILIES - Budget)	(Smiths)	0.087451	0.120162	0.011679	0.133549	1.111409	0.0011
3	(OLDER SINGLES/COUPLES - Budget)	(Kettle)	0.069504	0.155901	0.011573	0.166513	1.068064	0.0007
4	(OLDER SINGLES/COUPLES - Premium)	(Kettle)	0.067038	0.155901	0.011128	0.165991	1.064716	0.0006
6	(RETIREES - Mainstream)	(Kettle)	0.081055	0.155901	0.012785	0.157738	1.011779	0.0001
8	(YOUNG SINGLES/COUPLES - Mainstream)	(Kettle)	0.078744	0.155901	0.014515	0.184329	1.182344	0.0022

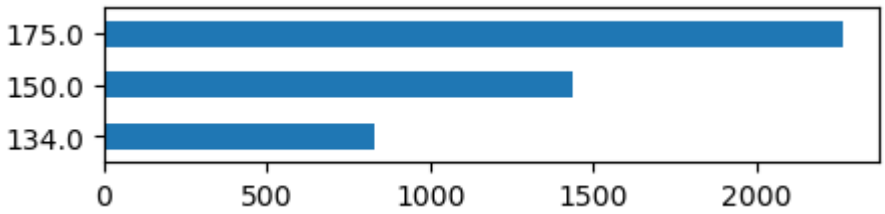
In [77]:

```
merged_pack = pd.concat([merged_data, pack_sizes.rename("Pack_Size")], axis=1)

for stage in merged_data["LIFESTAGE"].unique():
    for prem in merged_data["PREMIUM_CUSTOMER"].unique():
        print('==========',stage, '-', prem,'==========')
        summary = merged_pack[(merged_pack["LIFESTAGE"] == stage) & (merged_pack["P
        print(summary)
        plt.figure()
        summary.plot.barh(figsize=(5,1))
        plt.show()
```

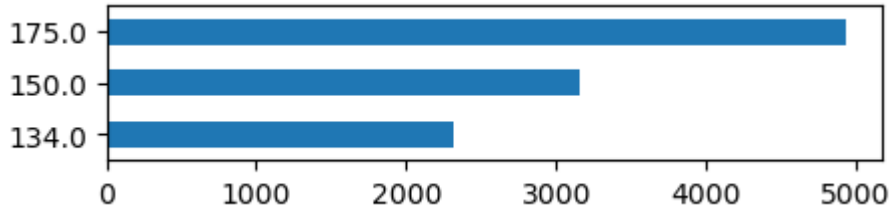
```
========== YOUNG SINGLES/COUPLES - Premium ==========
134.0     537
150.0     961
175.0    1587
Name: Pack_Size, dtype: int64
```



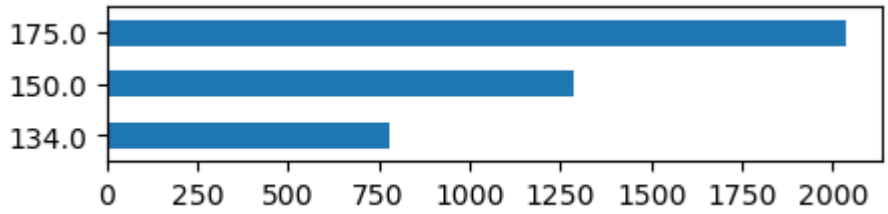

```
========== YOUNG SINGLES/COUPLES - Budget ==========
134.0     832
150.0    1439
175.0    2262
Name: Pack_Size, dtype: int64
```



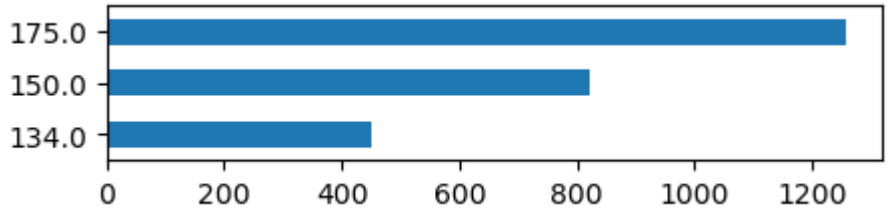
```
========== YOUNG SINGLES/COUPLES - Mainstream ==========
134.0    2315
150.0    3159
175.0    4928
Name: Pack_Size, dtype: int64
```



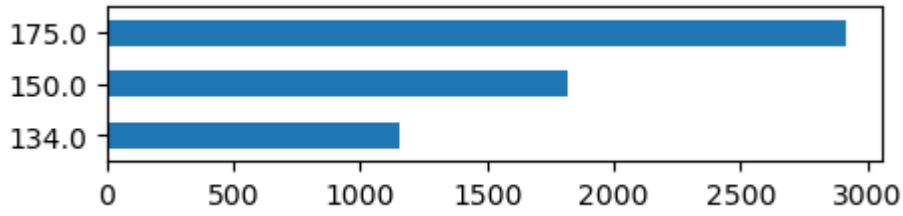

```
========== MIDAGE SINGLES/COUPLES - Premium ==========
134.0     781
150.0    1285
175.0    2034
Name: Pack_Size, dtype: int64
```



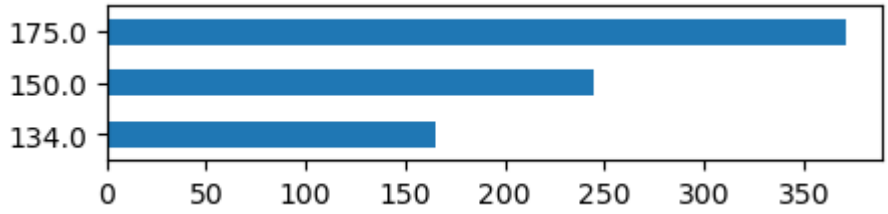

```
========== MIDAGE SINGLES/COUPLES - Budget ==========
134.0     449
150.0     821
175.0    1256
Name: Pack_Size, dtype: int64
```



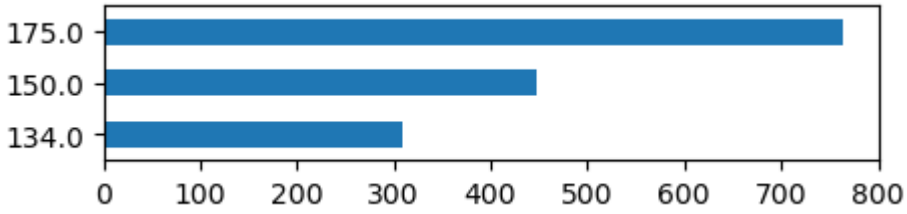

```
========== MIDAGE SINGLES/COUPLES - Mainstream ==========
134.0    1159
150.0    1819
175.0    2912
Name: Pack_Size, dtype: int64
```



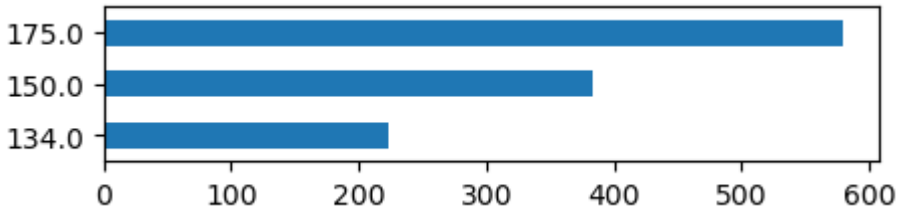

```
========== NEW FAMILIES - Premium ==========
134.0    165
150.0    245
175.0    371
Name: Pack_Size, dtype: int64
```



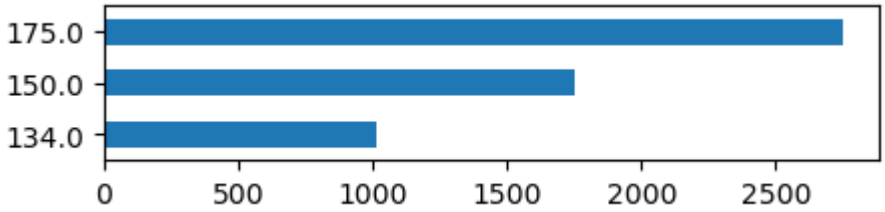
```
========== NEW FAMILIES - Budget ==========
134.0    309
150.0    448
175.0    763
Name: Pack_Size, dtype: int64
```



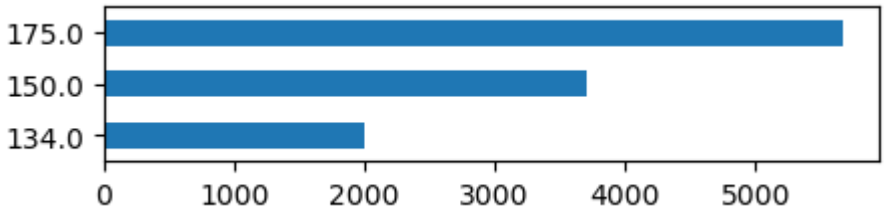

```
========== NEW FAMILIES - Mainstream ==========
134.0    224
150.0    384
175.0    579
Name: Pack_Size, dtype: int64
```



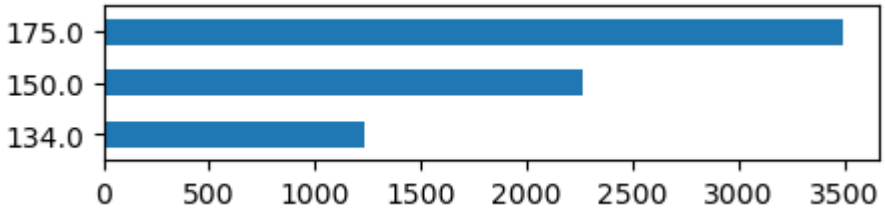

```
========== OLDER FAMILIES - Premium ==========
134.0    1014
150.0    1750
175.0    2747
Name: Pack_Size, dtype: int64
```



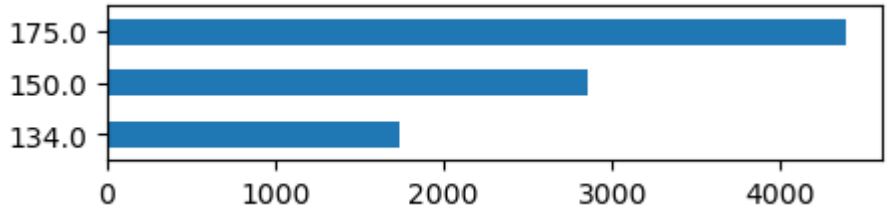

```
========== OLDER FAMILIES - Budget ==========
134.0    1996
150.0    3708
175.0    5662
Name: Pack_Size, dtype: int64
```



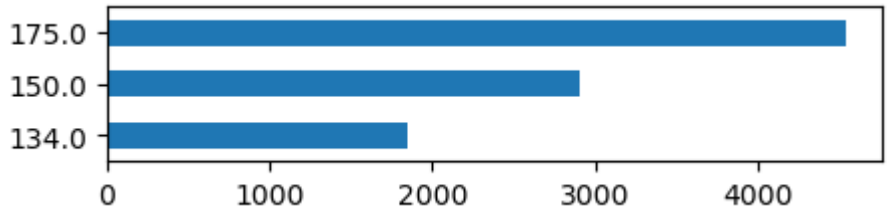

```
========== OLDER FAMILIES - Mainstream ==========
134.0    1234
150.0    2261
175.0    3489
Name: Pack_Size, dtype: int64
```



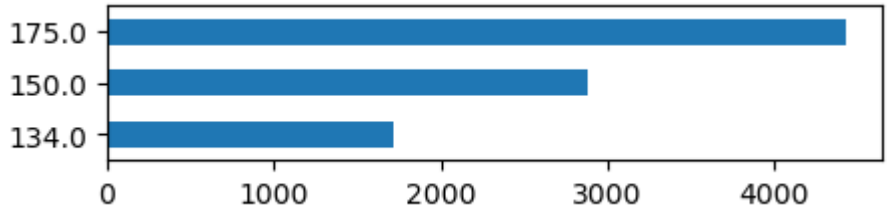
```
========== OLDER SINGLES/COUPLES - Premium ==========
134.0    1744
150.0    2854
175.0    4382
Name: Pack_Size, dtype: int64
```



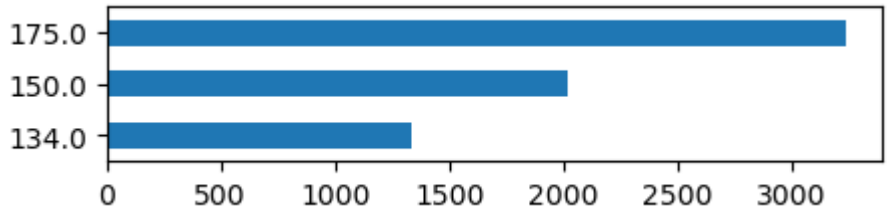

```
========== OLDER SINGLES/COUPLES - Budget ==========
134.0    1843
150.0    2899
175.0    4535
Name: Pack_Size, dtype: int64
```



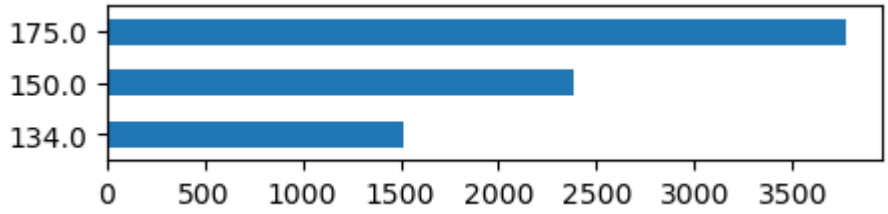

```
========== OLDER SINGLES/COUPLES - Mainstream ==========
134.0    1720
150.0    2875
175.0    4422
Name: Pack_Size, dtype: int64
```



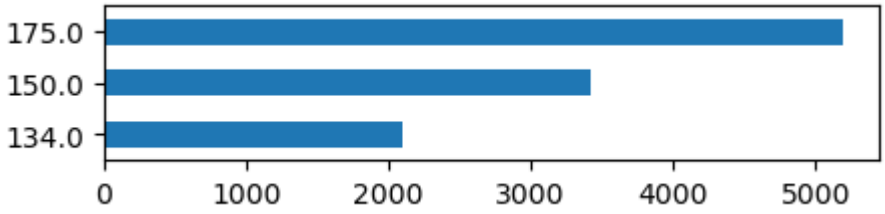

```
========== RETIREES - Premium ==========
134.0    1331
150.0    2015
175.0    3232
Name: Pack_Size, dtype: int64
```



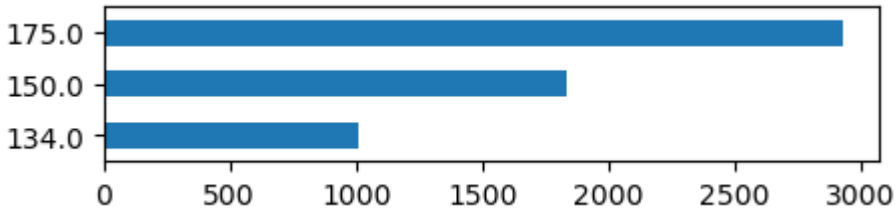

```
========== RETIREES - Budget ==========
134.0    1517
150.0    2381
175.0    3768
Name: Pack_Size, dtype: int64
```



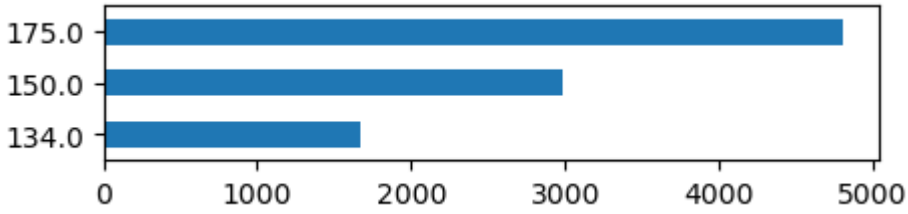
```
========== RETIREES - Mainstream ==========
134.0    2103
150.0    3415
175.0    5187
Name: Pack_Size, dtype: int64
```



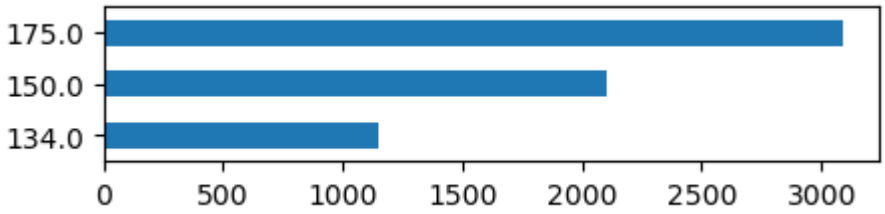

```
========== YOUNG FAMILIES - Premium ==========
134.0    1007
150.0    1832
175.0    2926
Name: Pack_Size, dtype: int64
```




```
========== YOUNG FAMILIES - Budget ==========
134.0    1674
150.0    2981
175.0    4800
Name: Pack_Size, dtype: int64
```




```
========== YOUNG FAMILIES - Mainstream ==========
134.0    1148
150.0    2101
175.0    3087
Name: Pack_Size, dtype: int64
```



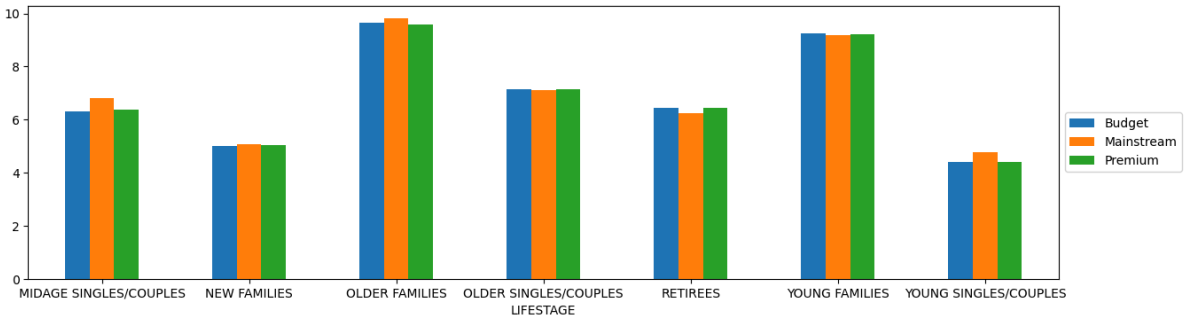
In [78]:
```
(temp.groupby(["LIFESTAGE", "PREMIUM_CUSTOMER"])["PRODUCT_QTY"].sum() / temp.groupb
```

Out[78]:

LIFESTAGE	PREMIUM_CUSTOMER	
OLDER FAMILIES	Mainstream	9.804309
	Budget	9.639572
	Premium	9.578091
YOUNG FAMILIES	Budget	9.238486
	Premium	9.209207
	Mainstream	9.180352
OLDER SINGLES/COUPLES	Premium	7.154947
	Budget	7.145466
	Mainstream	7.098783
MIDAGE SINGLES/COUPLES	Mainstream	6.796108
RETIREES	Budget	6.458015
	Premium	6.426653
MIDAGE SINGLES/COUPLES	Premium	6.386672
	Budget	6.313830
RETIREES	Mainstream	6.253743
NEW FAMILIES	Mainstream	5.087161
	Premium	5.028912
	Budget	5.009892
YOUNG SINGLES/COUPLES	Mainstream	4.776459
	Budget	4.411485
	Premium	4.402098

dtype: float64

In [79]:
```
(temp.groupby(["LIFESTAGE", "PREMIUM_CUSTOMER"])["PRODUCT_QTY"].sum() / temp.groupb
plt.legend(loc="center left", bbox_to_anchor=(1.0, 0.5))
plt.savefig("Average purchase quantity per segment.png", bbox_inches="tight")
```



In [80]:
```
#Average chips price per transaction by segments
temp["Unit_Price"] = temp["TOTAL_SALES"] / temp["PRODUCT_QTY"]
temp.groupby(["Segment"]).mean()["Unit_Price"].sort_values(ascending=False)
```

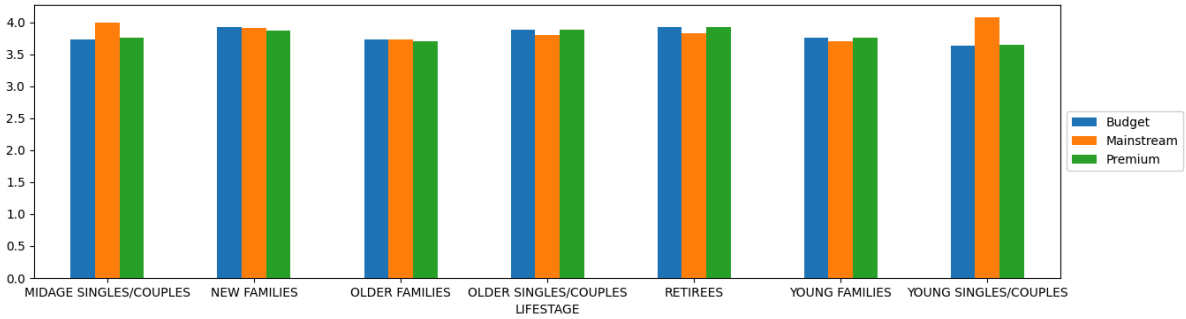
```
C:\Users\Ashish\AppData\Local\Temp\ipykernel_16548\2659809162.py:3: FutureWarning:
The default value of numeric_only in DataFrameGroupBy.mean is deprecated. In a fut
ure version, numeric_only will default to False. Either specify numeric_only or se
lect only columns which should be valid for the function.
  temp.groupby(["Segment"]).mean()["Unit_Price"].sort_values(ascending=False)
```

Out[80]:
```
Segment
YOUNG SINGLES/COUPLES - Mainstream     4.071485
MIDAGE SINGLES/COUPLES - Mainstream    4.000101
RETIREES - Budget                      3.924883
RETIREES - Premium                     3.921323
NEW FAMILIES - Budget                  3.919251
NEW FAMILIES - Mainstream              3.916581
OLDER SINGLES/COUPLES - Premium        3.887220
OLDER SINGLES/COUPLES - Budget         3.877022
NEW FAMILIES - Premium                 3.871743
RETIREES - Mainstream                  3.833343
OLDER SINGLES/COUPLES - Mainstream     3.803800
YOUNG FAMILIES - Budget                3.753659
MIDAGE SINGLES/COUPLES - Premium       3.752915
YOUNG FAMILIES - Premium               3.752402
OLDER FAMILIES - Budget                3.733344
MIDAGE SINGLES/COUPLES - Budget        3.728496
OLDER FAMILIES - Mainstream            3.727383
YOUNG FAMILIES - Mainstream            3.707097
OLDER FAMILIES - Premium               3.704625
YOUNG SINGLES/COUPLES - Premium        3.645518
YOUNG SINGLES/COUPLES - Budget         3.637681
Name: Unit_Price, dtype: float64
```

In [81]:
```
temp.groupby(["LIFESTAGE", "PREMIUM_CUSTOMER"]).mean()["Unit_Price"].unstack().plot
plt.legend(loc="center left", bbox_to_anchor=(1,0.5))
```

```
C:\Users\Ashish\AppData\Local\Temp\ipykernel_16548\1605091624.py:1: FutureWarning:
The default value of numeric_only in DataFrameGroupBy.mean is deprecated. In a fut
ure version, numeric_only will default to False. Either specify numeric_only or se
lect only columns which should be valid for the function.
  temp.groupby(["LIFESTAGE", "PREMIUM_CUSTOMER"]).mean()["Unit_Price"].unstack().p
lot.bar(figsize=(15,4), rot=0)
```

Out[81]:
```
<matplotlib.legend.Legend at 0x23ecdd4d890>
```



In [82]:
```
z = temp.groupby(["Segment", "Cleaned_Brand_Names"]).sum()["TOTAL_SALES"].sort_valu
z[z["Segment"] == "YOUNG SINGLES/COUPLES - Mainstream"]
```

```
C:\Users\Ashish\AppData\Local\Temp\ipykernel_16548\1025979435.py:1: FutureWarning:
The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a futu
re version, numeric_only will default to False. Either specify numeric_only or sel
ect only columns which should be valid for the function.
  z = temp.groupby(["Segment", "Cleaned_Brand_Names"]).sum()["TOTAL_SALES"].sort_v
alues(ascending=False).reset_index()
```

Out[82]:

	Segment	Cleaned_Brand_Names	TOTAL_SALES
0	YOUNG SINGLES/COUPLES - Mainstream	Kettle	35423.6
8	YOUNG SINGLES/COUPLES - Mainstream	Doritos	21705.9
23	YOUNG SINGLES/COUPLES - Mainstream	Pringles	16006.2
24	YOUNG SINGLES/COUPLES - Mainstream	Smiths	15265.7
55	YOUNG SINGLES/COUPLES - Mainstream	Infuzions	8749.4
59	YOUNG SINGLES/COUPLES - Mainstream	Old	8180.4
65	YOUNG SINGLES/COUPLES - Mainstream	Twisties	7539.8
73	YOUNG SINGLES/COUPLES - Mainstream	Tostitos	7238.0
74	YOUNG SINGLES/COUPLES - Mainstream	Thins	7217.1
92	YOUNG SINGLES/COUPLES - Mainstream	Cobs	6144.6
124	YOUNG SINGLES/COUPLES - Mainstream	RRD	4958.1
129	YOUNG SINGLES/COUPLES - Mainstream	Tyrrells	4800.6
148	YOUNG SINGLES/COUPLES - Mainstream	Grain Waves	4201.0
189	YOUNG SINGLES/COUPLES - Mainstream	Cheezels	3318.3
246	YOUNG SINGLES/COUPLES - Mainstream	Natural Chip Co	2130.0
258	YOUNG SINGLES/COUPLES - Mainstream	Woolworths	1929.8
318	YOUNG SINGLES/COUPLES - Mainstream	Cheetos	898.8
327	YOUNG SINGLES/COUPLES - Mainstream	CCs	850.5
383	YOUNG SINGLES/COUPLES - Mainstream	French	429.0
393	YOUNG SINGLES/COUPLES - Mainstream	Sunbites	391.0
415	YOUNG SINGLES/COUPLES - Mainstream	Burger	243.8

In [ ]: