notebook-retail-analysis

July 9, 2024

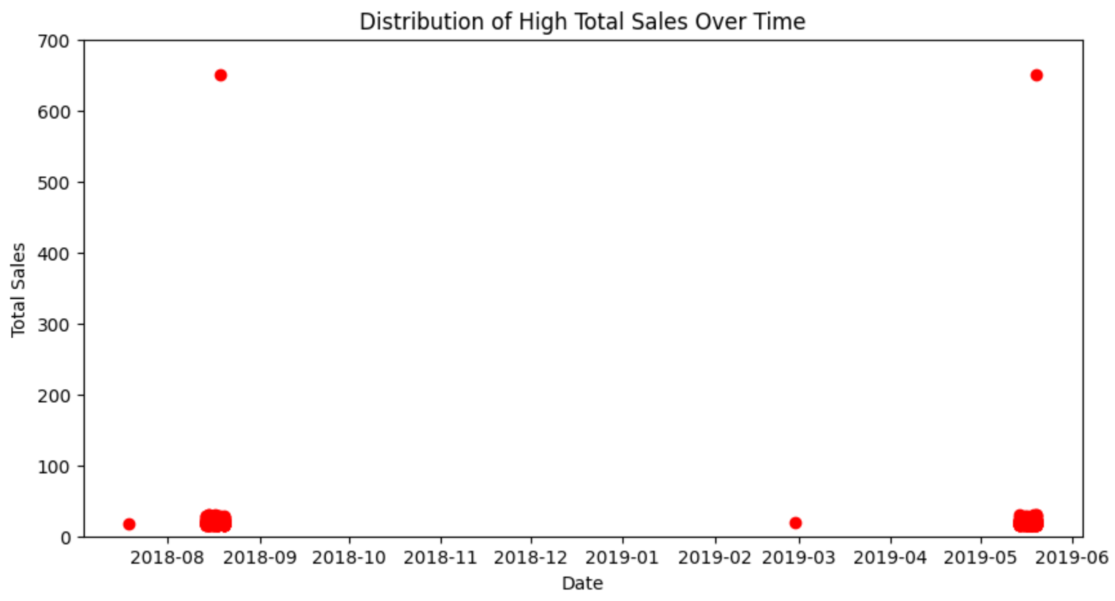
[2]:
```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
import numpy as np
from datetime import datetime, timedelta

# Load data
purchase_behaviour_data = pd.read_csv('QVI_purchase_behaviour.csv')
transaction_data = pd.read_excel('QVI_transaction_data.xlsx', sheet_name='in')

# Function to filter out salsa products
def filter_salsa_products(data):
    salsa_keywords = ['salsa']
    return data[~data['PROD_NAME'].str.contains('|'.join(salsa_keywords),␣
 ↪case=False, na=False)]

# Filter out salsa products
transaction_data = filter_salsa_products(transaction_data)

# Function to identify outliers using Z-score
def z_score_outliers(data, threshold=3):
    z_scores = np.abs(stats.zscore(data.select_dtypes(include=np.number)))
    outliers = (z_scores > threshold).any(axis=1)
    return data[outliers]

# Identify outliers
purchase_behaviour_data_outliers = z_score_outliers(purchase_behaviour_data)
transaction_data_outliers = z_score_outliers(transaction_data)

# Function to detect outliers using IQR
def detect_outliers_iqr(data, feature):
    Q1 = data[feature].quantile(0.25)
    Q3 = data[feature].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = data[(data[feature] < lower_bound) | (data[feature] >␣
↪upper_bound)]
    return outliers

# Apply the function to 'PROD_QTY' and 'TOT_SALES'
prod_qty_outliers = detect_outliers_iqr(transaction_data, 'PROD_QTY')
tot_sales_outliers = detect_outliers_iqr(transaction_data, 'TOT_SALES')

# Convert Excel serial date to datetime
def excel_date_to_datetime(excel_serial_date):
    excel_epoch = datetime(1899, 12, 30)  # Excel's epoch start (not 1900 due␣
↪to a historical error)
    delta = timedelta(days=excel_serial_date)
    return excel_epoch + delta

transaction_data['DATE'] = transaction_data['DATE'].
↪apply(excel_date_to_datetime)

# Plotting the distribution of high total sales over time
plt.figure(figsize=(10, 5))
plt.scatter(transaction_data.loc[transaction_data['TOT_SALES'] > 15, 'DATE'],
            transaction_data.loc[transaction_data['TOT_SALES'] > 15,␣
↪'TOT_SALES'], color='red')
plt.title('Distribution of High Total Sales Over Time')
plt.xlabel('Date')
plt.ylabel('Total Sales')
plt.ylim(0, 700)  # Adjusting the Y-axis limit for clarity
plt.show()

# Display the top 10 highest total sales
top_10_highest_sales = transaction_data.nlargest(10, 'TOT_SALES')
print("Top 10 Highest Total Sales:")
print(top_10_highest_sales)

# Display the top 10 lowest total sales
top_10_lowest_sales = transaction_data.nsmallest(10, 'TOT_SALES')
print("\nTop 10 Lowest Total Sales:")
print(top_10_lowest_sales)

# Feature Engineering
transaction_data['PACK_SIZE'] = transaction_data['PROD_NAME'].str.
↪extract('(\d+)g')
transaction_data['BRAND_NAME'] = transaction_data['PROD_NAME'].str.split().
↪str[0]

# Mapping of variations to standard brand names
```

```python
brand_mapping = {
    'Red': 'RRD',
    'RRD': 'RRD',  # Ensures 'RRD' stays as 'RRD'
    'Snbts': 'Sunbites',
    'Sunbites': 'Sunbites',  # Ensures 'Sunbites' stays as 'Sunbites'
    'Infuzions': 'Infzns',
    'Infzns': 'Infzns',  # Ensures 'Infzns' stays as 'Infzns'
    'WW': 'Woolworths',
    'Woolworths': 'Woolworths',  # Ensures 'Woolworths' stays as 'Woolworths'
    'Smiths': 'Smith',
    'Smith': 'Smith',  # Ensures 'Smith' stays as 'Smith'
    'NCC': 'Natural',
    'Natural': 'Natural',  # Ensures 'Natural' stays as 'Natural'
    'Dorito': 'Doritos',
    'Doritos': 'Doritos',  # Ensures 'Doritos' stays as 'Doritos'
    'Grain': 'GrnWves',
    'GrnWves': 'GrnWves'  # Ensures 'GrnWves' stays as 'GrnWves'
}

# Apply the mapping to standardize brand names
transaction_data['BRAND_NAME'] = transaction_data['BRAND_NAME'].
 ↪replace(brand_mapping)

# Display the unique brands to ensure extraction and standardization worked␣
 ↪correctly
print(transaction_data['BRAND_NAME'].unique())

# Display the first few rows to verify
print(transaction_data[['PACK_SIZE', 'BRAND_NAME']].head())

# Remove Outliers
transaction_data_clean = transaction_data[~transaction_data.index.
 ↪isin(prod_qty_outliers.index)]
transaction_data_clean = transaction_data_clean[~transaction_data_clean.index.
 ↪isin(tot_sales_outliers.index)]

# Re-plotting without extreme outliers
plt.figure(figsize=(10, 5))
plt.scatter(transaction_data_clean['DATE'],
            transaction_data_clean['TOT_SALES'], color='blue')
plt.title('Distribution of Total Sales Over Time (Cleaned)')
plt.xlabel('Date')
plt.ylabel('Total Sales')
plt.show()

# Calculate total spending per customer
```

```
customer_spending = transaction_data_clean.
  ↪groupby('LYLTY_CARD_NBR')['TOT_SALES'].sum().
  ↪reset_index(name='TOTAL_SPENDING')

# Calculate average product quantity per transaction
average_qty = transaction_data_clean.groupby('LYLTY_CARD_NBR')['PROD_QTY'].
  ↪mean().reset_index(name='AVERAGE_QTY')

print(customer_spending.head())
print(average_qty.head())

# Merge customer spending and average quantity data
customer_data = pd.merge(customer_spending, average_qty, on='LYLTY_CARD_NBR')

# Merge with purchase behaviour data
customer_data = pd.merge(customer_data, purchase_behaviour_data,␣
  ↪on='LYLTY_CARD_NBR')

# Analyze total spending by customer segment
segment_spending = customer_data.groupby(['LIFESTAGE',␣
  ↪'PREMIUM_CUSTOMER'])['TOTAL_SPENDING'].sum().reset_index()

# Calculate the number of customers in each segment
customer_segments = customer_data.groupby(['LIFESTAGE',␣
  ↪'PREMIUM_CUSTOMER'])['LYLTY_CARD_NBR'].nunique().
  ↪reset_index(name='CUSTOMER_COUNT')

# Display the number of customers in each segment
print(customer_segments)

print(segment_spending.sort_values(by='TOTAL_SPENDING', ascending=False))

# Calculate total sales per product
product_sales = transaction_data_clean.groupby('PROD_NAME')['TOT_SALES'].sum().
  ↪reset_index(name='TOTAL_SALES')

# Identify top-performing products
top_products = product_sales.sort_values(by='TOTAL_SALES', ascending=False).
  ↪head(10)

print(top_products)

# Resample sales data to monthly frequency and calculate total sales per month
monthly_sales = transaction_data_clean.resample('M', on='DATE')['TOT_SALES'].
  ↪sum().reset_index()
```

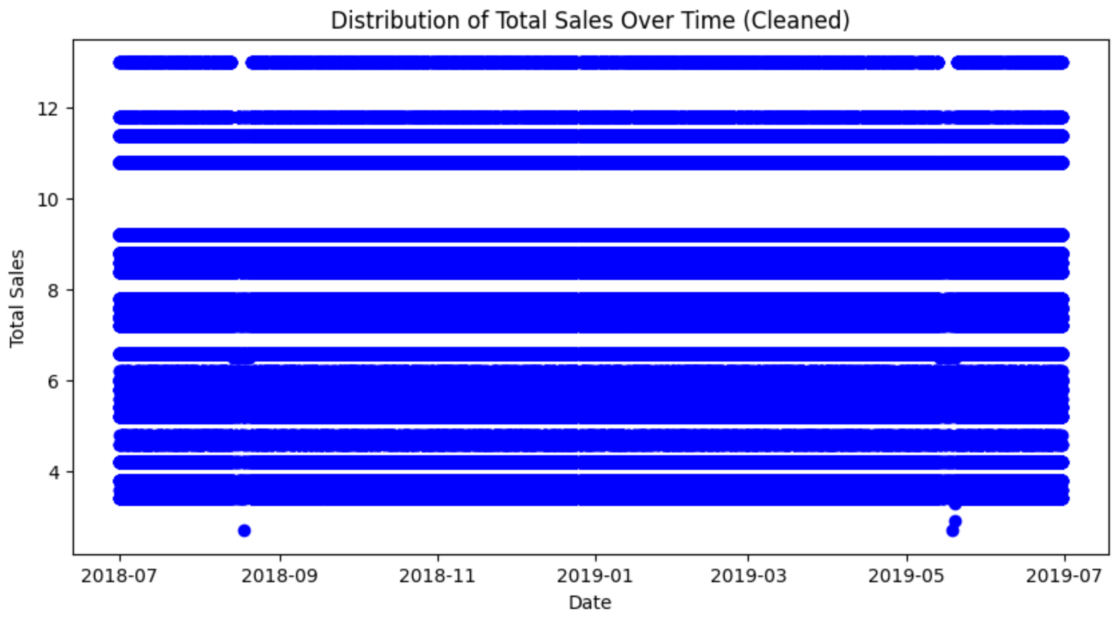
```
# Plot monthly sales trends
plt.figure(figsize=(10, 5))
sns.lineplot(data=monthly_sales, x='DATE', y='TOT_SALES')
plt.title('Monthly Sales Trends')
plt.xlabel('Date')
plt.ylabel('Total Sales')
plt.show()

# Merge transaction data with purchase behaviour data to include LIFESTAGE and␣
 ↪PREMIUM_CUSTOMER
merged_data = pd.merge(transaction_data_clean, purchase_behaviour_data,␣
 ↪on='LYLTY_CARD_NBR')

# Group by LIFESTAGE, PREMIUM_CUSTOMER, and PROD_NAME to calculate total sales␣
 ↪for each product within each segment
segment_product_sales = merged_data.groupby(['LIFESTAGE', 'PREMIUM_CUSTOMER',␣
 ↪'PROD_NAME'])['TOT_SALES'].sum().reset_index()

# Function to get top products for each segment
def get_top_products(segment_data, top_n=3):
    return segment_data.sort_values(by='TOT_SALES', ascending=False).head(top_n)

# Apply the function to get the top 3 products for each segment
top_products_per_segment = (
    segment_product_sales.groupby(['LIFESTAGE', 'PREMIUM_CUSTOMER'],␣
 ↪group_keys=False)
     .apply(get_top_products)
     .reset_index(drop=True)
)

# Set display options to show all rows
pd.set_option('display.max_rows', None)

# Print the top products per segment
print(top_products_per_segment)

# Calculate the number of chips bought per customer by segment
chips_per_customer_segment = (
    merged_data.groupby(['LIFESTAGE', 'PREMIUM_CUSTOMER',␣
 ↪'LYLTY_CARD_NBR'])['PROD_QTY'].sum()
     .groupby(level=[0, 1]).mean().reset_index(name='AVG_QTY_PER_CUSTOMER')
)

# Display the number of chips bought per customer by segment
print(chips_per_customer_segment)
```

```
# Calculate the total amount spent on chips by each customer
customer_total_spent = transaction_data_clean.
 ↪groupby('LYLTY_CARD_NBR')['TOT_SALES'].sum().reset_index(name='TOTAL_SPENT')

# Calculate the total number of chips bought by each customer
customer_total_qty = transaction_data_clean.
 ↪groupby('LYLTY_CARD_NBR')['PROD_QTY'].sum().reset_index(name='TOTAL_QTY')

# Merge the total spending and total quantity data
customer_spending_qty = pd.merge(customer_total_spent, customer_total_qty,␣
 ↪on='LYLTY_CARD_NBR')

# Calculate the average chip price per customer
customer_spending_qty['AVG_CHIP_PRICE'] = customer_spending_qty['TOTAL_SPENT'] /
 ↪ customer_spending_qty['TOTAL_QTY']

# Merge with purchase behaviour data to get customer segments
customer_segment_data = pd.merge(customer_spending_qty,␣
 ↪purchase_behaviour_data, on='LYLTY_CARD_NBR')

# Calculate the average chip price by customer segment
avg_chip_price_segment = customer_segment_data.groupby(['LIFESTAGE',␣
 ↪'PREMIUM_CUSTOMER'])['AVG_CHIP_PRICE'].mean().reset_index()

# Display the average chip price by customer segment
print(avg_chip_price_segment)
```



```
Top 10 Highest Total Sales:
             DATE  STORE_NBR  LYLTY_CARD_NBR  TXN_ID  PROD_NBR  \
69762  2018-08-19        226          226000  226201         4
69763  2019-05-20        226          226000  226210         4
5179   2018-08-15         94           94148   93390        14
55558  2019-05-14        190          190113  190914        14
69496  2018-08-15         49           49303   45789        14
117850 2019-05-19        194          194308  194516        14
150683 2019-05-20        118          118021  120799        14
171815 2018-08-17         24           24095   20797        14
184969 2019-05-20         44           44350   40394        14
72     2018-08-19         96           96203   96025         7

                                     PROD_NAME  PROD_QTY  TOT_SALES
69762         Dorito Corn Chp     Supreme 380g       200      650.0
69763         Dorito Corn Chp     Supreme 380g       200      650.0
5179    Smiths Crnkle Chip  Orgnl Big Bag 380g         5       29.5
55558   Smiths Crnkle Chip  Orgnl Big Bag 380g         5       29.5
69496   Smiths Crnkle Chip  Orgnl Big Bag 380g         5       29.5
117850  Smiths Crnkle Chip  Orgnl Big Bag 380g         5       29.5
150683  Smiths Crnkle Chip  Orgnl Big Bag 380g         5       29.5
171815  Smiths Crnkle Chip  Orgnl Big Bag 380g         5       29.5
184969  Smiths Crnkle Chip  Orgnl Big Bag 380g         5       29.5
72           Smiths Crinkle      Original 330g         5       28.5

Top 10 Lowest Total Sales:
           DATE  STORE_NBR  LYLTY_CARD_NBR  TXN_ID  PROD_NBR  \
13   2018-08-17         13           13213   12447        92
145  2019-05-15        197          197172  197097        72
173  2019-05-19        236          236247  240056        92
215  2018-10-18          1            1411     476        92
224  2018-12-14          2            2256     866        55
525  2018-12-18         20           20311   17286        55
981  2019-03-05         50           50034   46157        95
1694 2019-06-15         90           90016   88697        72
2619 2018-08-21        131          131211  135532        92
2684 2018-11-07        136          136066  138453        92

                                     PROD_NAME  PROD_QTY  TOT_SALES
13            WW Crinkle Cut      Chicken 175g         1        1.7
145          WW Crinkle Cut      Original 175g         1        1.7
173           WW Crinkle Cut      Chicken 175g         1        1.7
215           WW Crinkle Cut      Chicken 175g         1        1.7
224       Snbts Whlgrn Crisps Cheddr&Mstrd 90g         1        1.7
525       Snbts Whlgrn Crisps Cheddr&Mstrd 90g         1        1.7
981   Sunbites Whlegrn    Crisps Frch/Onin 90g         1        1.7
1694         WW Crinkle Cut      Original 175g         1        1.7
2619          WW Crinkle Cut      Chicken 175g         1        1.7
```

```
2684          WW Crinkle Cut      Chicken 175g         1        1.7
['Natural' 'CCs' 'Smith' 'Kettle' 'GrnWves' 'Doritos' 'Twisties'
 'Woolworths' 'Thins' 'Burger' 'Cheezels' 'Infzns' 'RRD' 'Pringles'
 'Tyrrells' 'Cobs' 'French' 'Tostitos' 'Cheetos' 'Sunbites']
  PACK_SIZE BRAND_NAME
0       175    Natural
1       175        CCs
2       170      Smith
3       175      Smith
4       150     Kettle
```

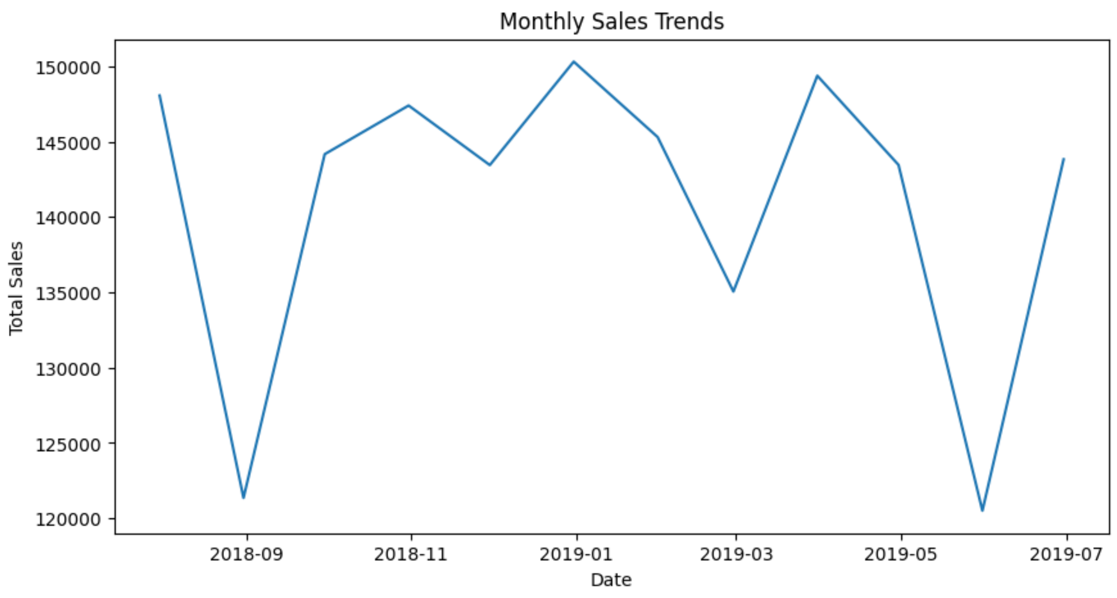


```
   LYLTY_CARD_NBR  TOTAL_SPENDING
0            1000             6.0
1            1010             8.8
2            1011             6.2
3            1013             4.2
4            1025             6.0
   LYLTY_CARD_NBR  AVERAGE_QTY
0            1000          2.0
1            1010          2.0
2            1011          2.0
3            1013          2.0
4            1025          2.0
                LIFESTAGE PREMIUM_CUSTOMER  CUSTOMER_COUNT
0  MIDAGE SINGLES/COUPLES           Budget            1232
1  MIDAGE SINGLES/COUPLES       Mainstream            2918
2  MIDAGE SINGLES/COUPLES          Premium            1997
```

```
3             NEW FAMILIES          Budget              911
4             NEW FAMILIES      Mainstream              701
5             NEW FAMILIES         Premium              495
6           OLDER FAMILIES          Budget             4148
7           OLDER FAMILIES      Mainstream             2511
8           OLDER FAMILIES         Premium             2015
9    OLDER SINGLES/COUPLES          Budget             4325
10   OLDER SINGLES/COUPLES      Mainstream             4271
11   OLDER SINGLES/COUPLES         Premium             4165
12                RETIREES          Budget             3817
13                RETIREES      Mainstream             5475
14                RETIREES         Premium             3372
15          YOUNG FAMILIES          Budget             3545
16          YOUNG FAMILIES      Mainstream             2386
17          YOUNG FAMILIES         Premium             2124
18   YOUNG SINGLES/COUPLES          Budget             2795
19   YOUNG SINGLES/COUPLES      Mainstream             6669
20   YOUNG SINGLES/COUPLES         Premium             1896
                 LIFESTAGE PREMIUM_CUSTOMER  TOTAL_SPENDING
6           OLDER FAMILIES          Budget         150009.9
13                RETIREES      Mainstream         135028.2
19   YOUNG SINGLES/COUPLES      Mainstream         134727.4
15          YOUNG FAMILIES          Budget         123723.0
9    OLDER SINGLES/COUPLES          Budget         120494.4
10   OLDER SINGLES/COUPLES      Mainstream         117243.2
11   OLDER SINGLES/COUPLES         Premium         116249.8
12                RETIREES          Budget          98685.1
7           OLDER FAMILIES      Mainstream          92179.3
14                RETIREES         Premium          85324.0
16          YOUNG FAMILIES      Mainstream          81967.9
1   MIDAGE SINGLES/COUPLES      Mainstream          79566.1
17          YOUNG FAMILIES         Premium          74650.9
8           OLDER FAMILIES         Premium          71766.1
18   YOUNG SINGLES/COUPLES          Budget          50729.4
2   MIDAGE SINGLES/COUPLES         Premium          50664.6
20   YOUNG SINGLES/COUPLES         Premium          34708.6
0   MIDAGE SINGLES/COUPLES          Budget          31061.9
3             NEW FAMILIES          Budget          18899.0
4             NEW FAMILIES      Mainstream          14711.3
5             NEW FAMILIES         Premium           9935.0
                                   PROD_NAME  TOTAL_SALES
11          Dorito Corn Chp     Supreme 380g      37232.0
79    Smiths Crnkle Chip  Orgnl Big Bag 380g      33901.4
71  Smiths Crinkle Chips Salt & Vinegar 330g      32809.2
31    Kettle Mozzarella   Basil & Pesto 175g      32443.2
70         Smiths Crinkle      Original 330g      32398.8
6                        Cheezels Cheese 330g      32045.4
12          Doritos Cheese      Supreme 330g      31441.2
```

```
37   Kettle Sweet Chilli And Sour Cream 175g      31158.0
33      Kettle Sea Salt     And Vinegar 175g      30682.8
30          Kettle Honey Soy    Chicken 175g      30682.8
```

```
/tmp/ipykernel_3111/2346987776.py:153: FutureWarning: 'M' is deprecated and will
be removed in a future version, please use 'ME' instead.
  monthly_sales = transaction_data_clean.resample('M',
on='DATE')['TOT_SALES'].sum().reset_index()
```



```
                 LIFESTAGE PREMIUM_CUSTOMER  \
0   MIDAGE SINGLES/COUPLES           Budget
1   MIDAGE SINGLES/COUPLES           Budget
2   MIDAGE SINGLES/COUPLES           Budget
3   MIDAGE SINGLES/COUPLES       Mainstream
4   MIDAGE SINGLES/COUPLES       Mainstream
5   MIDAGE SINGLES/COUPLES       Mainstream
6   MIDAGE SINGLES/COUPLES          Premium
7   MIDAGE SINGLES/COUPLES          Premium
8   MIDAGE SINGLES/COUPLES          Premium
9             NEW FAMILIES           Budget
10            NEW FAMILIES           Budget
11            NEW FAMILIES           Budget
12            NEW FAMILIES       Mainstream
13            NEW FAMILIES       Mainstream
14            NEW FAMILIES       Mainstream
15            NEW FAMILIES          Premium
16            NEW FAMILIES          Premium
17            NEW FAMILIES          Premium
```

```
18          OLDER FAMILIES          Budget
19          OLDER FAMILIES          Budget
20          OLDER FAMILIES          Budget
21          OLDER FAMILIES      Mainstream
22          OLDER FAMILIES      Mainstream
23          OLDER FAMILIES      Mainstream
24          OLDER FAMILIES         Premium
25          OLDER FAMILIES         Premium
26          OLDER FAMILIES         Premium
27   OLDER SINGLES/COUPLES          Budget
28   OLDER SINGLES/COUPLES          Budget
29   OLDER SINGLES/COUPLES          Budget
30   OLDER SINGLES/COUPLES      Mainstream
31   OLDER SINGLES/COUPLES      Mainstream
32   OLDER SINGLES/COUPLES      Mainstream
33   OLDER SINGLES/COUPLES         Premium
34   OLDER SINGLES/COUPLES         Premium
35   OLDER SINGLES/COUPLES         Premium
36                RETIREES          Budget
37                RETIREES          Budget
38                RETIREES          Budget
39                RETIREES      Mainstream
40                RETIREES      Mainstream
41                RETIREES      Mainstream
42                RETIREES         Premium
43                RETIREES         Premium
44                RETIREES         Premium
45          YOUNG FAMILIES          Budget
46          YOUNG FAMILIES          Budget
47          YOUNG FAMILIES          Budget
48          YOUNG FAMILIES      Mainstream
49          YOUNG FAMILIES      Mainstream
50          YOUNG FAMILIES      Mainstream
51          YOUNG FAMILIES         Premium
52          YOUNG FAMILIES         Premium
53          YOUNG FAMILIES         Premium
54   YOUNG SINGLES/COUPLES          Budget
55   YOUNG SINGLES/COUPLES          Budget
56   YOUNG SINGLES/COUPLES          Budget
57   YOUNG SINGLES/COUPLES      Mainstream
58   YOUNG SINGLES/COUPLES      Mainstream
59   YOUNG SINGLES/COUPLES      Mainstream
60   YOUNG SINGLES/COUPLES         Premium
61   YOUNG SINGLES/COUPLES         Premium
62   YOUNG SINGLES/COUPLES         Premium

                               PROD_NAME  TOT_SALES
0          Dorito Corn Chp     Supreme 380g      715.0
```

```
1     Kettle Mozzarella   Basil & Pesto 175g      637.2
2           Doritos Cheese      Supreme 330g      592.8
3   Smiths Crinkle Chips Salt & Vinegar 330g     1995.0
4                       Cheezels Cheese 330g     1903.8
5     Smiths Crnkle Chip  Orgnl Big Bag 380g     1840.8
6           Dorito Corn Chp     Supreme 380g     1131.0
7     Smiths Crnkle Chip  Orgnl Big Bag 380g     1109.2
8                       Cheezels Cheese 330g     1026.0
9           Doritos Cheese      Supreme 330g      467.4
10          Dorito Corn Chp     Supreme 380g      455.0
11          Kettle Honey Soy    Chicken 175g      453.6
12          Dorito Corn Chp     Supreme 380g      370.5
13    Kettle Mozzarella   Basil & Pesto 175g      345.6
14                        Kettle Chilli 175g      324.0
15    Smiths Crnkle Chip  Orgnl Big Bag 380g      259.6
16   Doritos Corn Chips  Cheese Supreme 170g      228.8
17         Smiths Crinkle      Original 330g      228.0
18          Dorito Corn Chp     Supreme 380g     3048.5
19  Smiths Crinkle Chips Salt & Vinegar 330g     3043.8
20                      Cheezels Cheese 330g     2918.4
21          Dorito Corn Chp     Supreme 380g     1943.5
22  Smiths Crinkle Chips Salt & Vinegar 330g     1869.6
23    Smiths Crnkle Chip  Orgnl Big Bag 380g     1746.4
24         Smiths Crinkle      Original 330g     1504.8
25          Dorito Corn Chp     Supreme 380g     1501.5
26  Smiths Crinkle Chips Salt & Vinegar 330g     1436.4
27          Dorito Corn Chp     Supreme 380g     2600.0
28    Smiths Crnkle Chip  Orgnl Big Bag 380g     2489.8
29      Kettle Sea Salt     And Vinegar 175g     2440.8
30          Dorito Corn Chp     Supreme 380g     2730.0
31    Smiths Crnkle Chip  Orgnl Big Bag 380g     2336.4
32          Doritos Cheese      Supreme 330g     2245.8
33          Dorito Corn Chp     Supreme 380g     2730.0
34                      Cheezels Cheese 330g     2485.2
35         Smiths Crinkle      Original 330g     2416.8
36                        Kettle Chilli 175g     2311.2
37    Smiths Crnkle Chip  Orgnl Big Bag 380g     2183.0
38    Kettle Mozzarella   Basil & Pesto 175g     2138.4
39          Dorito Corn Chp     Supreme 380g     2886.0
40         Smiths Crinkle      Original 330g     2724.6
41          Kettle Honey Soy    Chicken 175g     2700.0
42          Dorito Corn Chp     Supreme 380g     2145.0
43    Kettle Mozzarella   Basil & Pesto 175g     1879.2
44    Smiths Crnkle Chip  Orgnl Big Bag 380g     1817.2
45          Dorito Corn Chp     Supreme 380g     2574.0
46         Smiths Crinkle      Original 330g     2428.2
47                      Kettle Original 175g     2408.4
48    Smiths Crnkle Chip  Orgnl Big Bag 380g     1663.8
```

```
49         Dorito Corn Chp     Supreme 380g     1612.0
50                     Cheezels Cheese 330g     1527.6
51         Dorito Corn Chp     Supreme 380g     1592.5
52                     Cheezels Cheese 330g     1539.0
53                     Kettle Original 175g     1468.8
54         Dorito Corn Chp     Supreme 380g     1079.0
55         Doritos Cheese      Supreme 330g     1014.6
56     Kettle Sea Salt     And Vinegar 175g     1004.4
57         Dorito Corn Chp     Supreme 380g     3445.0
58   Smiths Crnkle Chip  Orgnl Big Bag 380g     3174.2
59   Kettle Mozzarella   Basil & Pesto 175g     3142.8
60         Dorito Corn Chp     Supreme 380g      754.0
61   Smiths Crnkle Chip  Orgnl Big Bag 380g      731.6
62   Kettle Mozzarella   Basil & Pesto 175g      712.8
                 LIFESTAGE PREMIUM_CUSTOMER  AVG_QTY_PER_CUSTOMER
0   MIDAGE SINGLES/COUPLES           Budget              6.689935
1   MIDAGE SINGLES/COUPLES       Mainstream              6.828650
2   MIDAGE SINGLES/COUPLES          Premium              6.685028
3             NEW FAMILIES           Budget              5.251372
4             NEW FAMILIES       Mainstream              5.298146
5             NEW FAMILIES          Premium              5.147475
6           OLDER FAMILIES           Budget              9.641755
7           OLDER FAMILIES       Mainstream              9.824771
8           OLDER FAMILIES          Premium              9.568238
9    OLDER SINGLES/COUPLES           Budget              7.161618
10   OLDER SINGLES/COUPLES       Mainstream              7.168345
11   OLDER SINGLES/COUPLES          Premium              7.149580
12                RETIREES           Budget              6.561174
13                RETIREES       Mainstream              6.378082
14                RETIREES          Premium              6.440688
15          YOUNG FAMILIES           Budget              9.266573
16          YOUNG FAMILIES       Mainstream              9.217100
17          YOUNG FAMILIES          Premium              9.347458
18   YOUNG SINGLES/COUPLES           Budget              4.868694
19   YOUNG SINGLES/COUPLES       Mainstream              4.946169
20   YOUNG SINGLES/COUPLES          Premium              4.900844
                 LIFESTAGE PREMIUM_CUSTOMER  AVG_CHIP_PRICE
0   MIDAGE SINGLES/COUPLES           Budget        3.820015
1   MIDAGE SINGLES/COUPLES       Mainstream        4.069664
2   MIDAGE SINGLES/COUPLES          Premium        3.837673
3             NEW FAMILIES           Budget        3.949506
4             NEW FAMILIES       Mainstream        3.964496
5             NEW FAMILIES          Premium        3.902643
6           OLDER FAMILIES           Budget        3.847186
7           OLDER FAMILIES       Mainstream        3.831940
8           OLDER FAMILIES          Premium        3.812338
9    OLDER SINGLES/COUPLES           Budget        3.938227
10   OLDER SINGLES/COUPLES       Mainstream        3.868939
```

```
11   OLDER SINGLES/COUPLES          Premium        3.969827
12                RETIREES           Budget        3.998533
13                RETIREES       Mainstream        3.895328
14                RETIREES          Premium        3.983547
15          YOUNG FAMILIES           Budget        3.867016
16          YOUNG FAMILIES       Mainstream        3.826046
17          YOUNG FAMILIES          Premium        3.849675
18   YOUNG SINGLES/COUPLES           Budget        3.730527
19   YOUNG SINGLES/COUPLES       Mainstream        4.155596
20   YOUNG SINGLES/COUPLES          Premium        3.732626
```

```
/tmp/ipykernel_3111/2346987776.py:176: DeprecationWarning:
DataFrameGroupBy.apply operated on the grouping columns. This behavior is
deprecated, and in a future version of pandas the grouping columns will be
excluded from the operation. Either pass `include_groups=False` to exclude the
groupings or explicitly select the grouping columns after groupby to silence
this warning.
  .apply(get_top_products)
```