```
# Name - Bhagyashri Vinit Dumbre
# Task-1


# Import pandas, matplotlib and seaborn
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns


%matplotlib inline


# import first transaction_data


Transaction_data = pd.read_csv("/QVI_transaction_data.csv")
Transaction_data
```

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	SIZE	PROD_QTY	TOT_SALE
0	10/17/2018	1	1000	1	5	Natural Chip Compny SeaSalt175g	175g	2	6
1	5/14/2019	1	1307	348	66	CCs Nacho Cheese 175g	175g	3	6
2	5/20/2019	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	170g	2	2
3	8/17/2018	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	175g	5	15
4	8/18/2018	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	150g	3	13
...	...	...	...	...	...	...	...	...	...

Kettle Sweet

```
# import 2nd purchase behaviour data


Purchase_behaviour = pd.read_csv("/QVI_purchase_behaviour.csv")
Purchase_behaviour
```

	LYLTY_CARD_NBR	LIFESTAGE	PREMIUM_CUSTOMER
0	1000	YOUNG SINGLES/COUPLES	Premium
1	1002	YOUNG SINGLES/COUPLES	Mainstream
2	1003	YOUNG FAMILIES	Budget
3	1004	OLDER SINGLES/COUPLES	Mainstream
4	1005	MIDAGE SINGLES/COUPLES	Mainstream
...	...	...	...
72632	2370651	MIDAGE SINGLES/COUPLES	Mainstream
72633	2370701	YOUNG FAMILIES	Mainstream
72634	2370751	YOUNG FAMILIES	Premium
72635	2370961	OLDER FAMILIES	Budget
72636	2373711	YOUNG SINGLES/COUPLES	Mainstream

72637 rows × 3 columns

```
Transaction_data .head(10)
```

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	SIZE	PROD_QTY	TOT_SALES
0	10/17/2018	1	1000	1	5	Natural Chip Compny SeaSalt175g	175g	2	6.0
1	5/14/2019	1	1307	348	66	CCs Nacho Cheese 175g	175g	3	6.3
2	5/20/2019	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	170g	2	2.9
3	8/17/2018	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	175g	5	15.0
4	8/18/2018	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	150g	3	13.8

```
Transaction_data .tail(10)
```

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	SIZE	PROD_QTY	TOT_SA
264826	3/25/2019	272	272194	269908	75	Cobs Popd Sea Salt Chips 110g	110g	2	
264827	8/28/2018	272	272197	269911	104	Infuzions Thai SweetChili PotatoMix 110g	110g	2	

```
Purchase_behaviour.head(10)
```

	LYLTY_CARD_NBR	LIFESTAGE	PREMIUM_CUSTOMER
0	1000	YOUNG SINGLES/COUPLES	Premium
1	1002	YOUNG SINGLES/COUPLES	Mainstream
2	1003	YOUNG FAMILIES	Budget
3	1004	OLDER SINGLES/COUPLES	Mainstream
4	1005	MIDAGE SINGLES/COUPLES	Mainstream
5	1007	YOUNG SINGLES/COUPLES	Budget
6	1009	NEW FAMILIES	Premium
7	1010	YOUNG SINGLES/COUPLES	Mainstream
8	1011	OLDER SINGLES/COUPLES	Mainstream
9	1012	OLDER FAMILIES	Mainstream

```
Transaction_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 264836 entries, 0 to 264835
Data columns (total 9 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   DATE            264836 non-null  object 
 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   SIZE            264836 non-null  object 
 7   PROD_QTY        264836 non-null  int64  
 8   TOT_SALES       264836 non-null  float64
dtypes: float64(1), int64(5), object(3)
memory usage: 18.2+ MB
```

```
Purchase_behaviour.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
```

```
Transaction_data.columns
```

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

```
Purchase_behaviour.columns
```

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

```
Transaction_data.shape
```

```
(264836, 9)
```

```
Purchase_behaviour.shape
```

```
(72637, 3)
```

```
# merge both data files in order to do findings.


left_merged = pd.merge(Transaction_data, Purchase_behaviour, how="outer", on=["LYLTY_CARD_NBR"])
left_merged
```

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	SIZE	PROD_QTY	TOT_SALES
0	10/17/2018	1	1000	1	5	Natural Chip Compny SeaSalt175g	175g	2	6.0
1	5/14/2019	1	1307	348	66	CCs Nacho Cheese 175g	175g	3	6.3
2	11/10/2018	1	1307	346	96	WW Original Stacked Chips 160g	160g	2	3.8
3	3/9/2019	1	1307	347	54	CCs Original 175g	175g	1	2.1
4	5/20/2019	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	170g	2	2.9

```
left_merged.shape
```

```
(264836, 11)
```

```
  # to describe data
```

```
left_merged1 = left_merged.iloc[:,6:9]
left_merged1.describe()
```

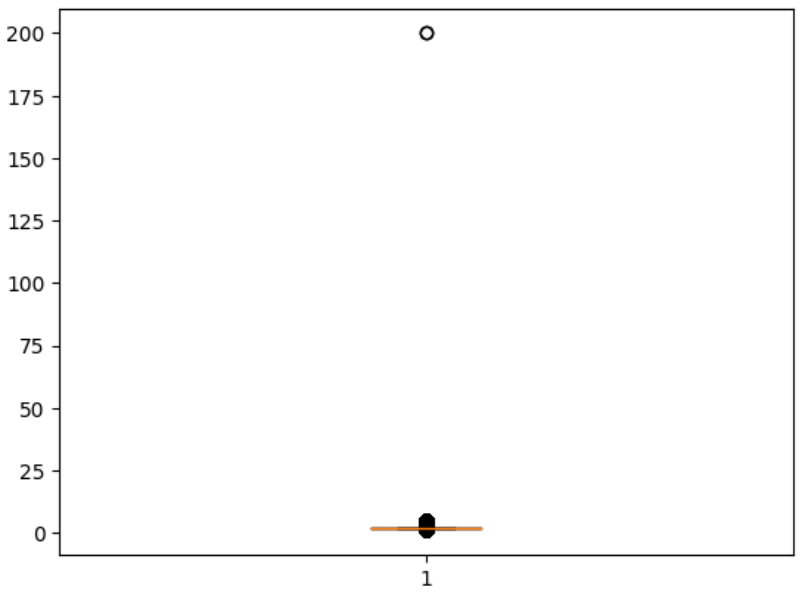
	PROD_QTY	TOT_SALES
count	264836.000000	264836.000000
mean	1.907309	7.304200
std	0.643654	3.083226
min	1.000000	1.500000
25%	2.000000	5.400000
50%	2.000000	7.400000
75%	2.000000	9.200000
max	200.000000	650.000000

```
# type of data
left_merged.dtypes
```

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

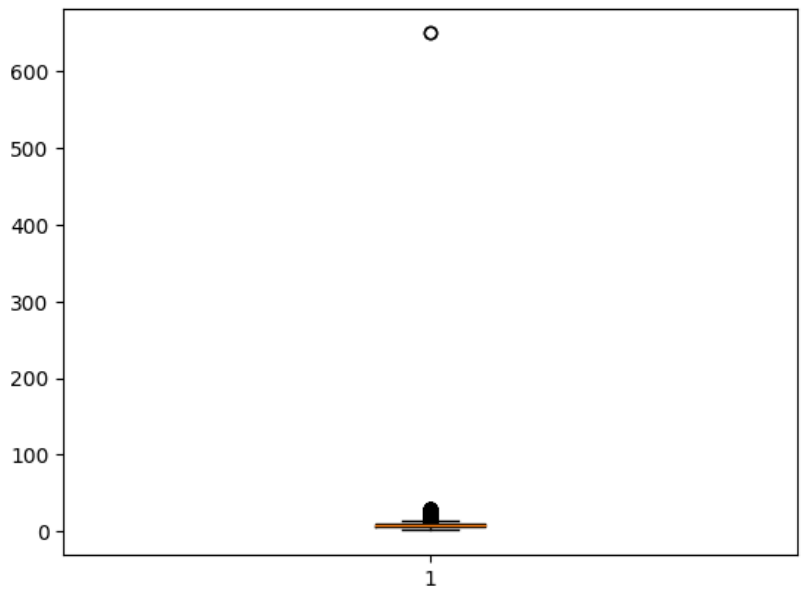
```
# outliers finding in product quantity
plt.boxplot(left_merged1['PROD_QTY'],vert = True)
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x7f266b66fe50>,
  <matplotlib.lines.Line2D at 0x7f266b6b0130>],
 'caps': [<matplotlib.lines.Line2D at 0x7f266b6b03d0>,
  <matplotlib.lines.Line2D at 0x7f266b6b0670>],
 'boxes': [<matplotlib.lines.Line2D at 0x7f266b66fbb0>],
 'medians': [<matplotlib.lines.Line2D at 0x7f266b6b0910>],
 'fliers': [<matplotlib.lines.Line2D at 0x7f266b6b0bb0>],
 'means': []}
```

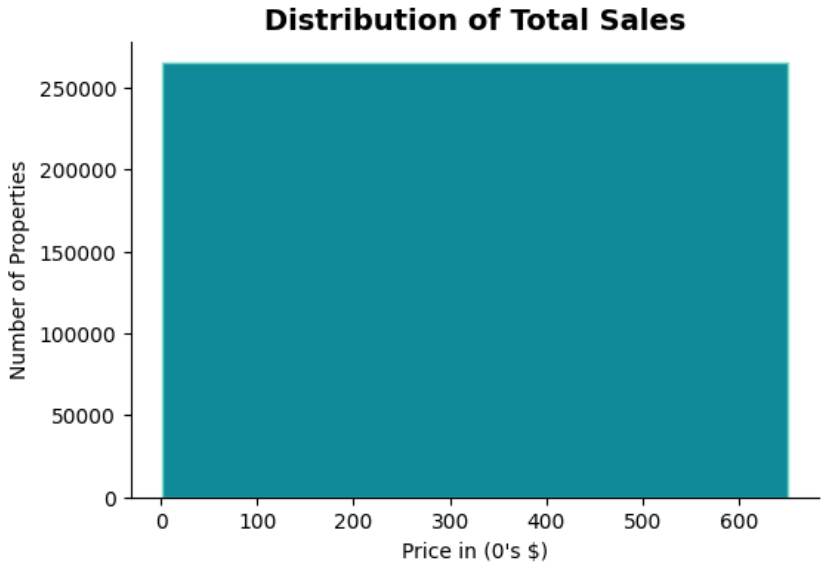


```
# outliers finding in total sales
plt.boxplot(left_merged1['TOT_SALES'],vert = True)
```

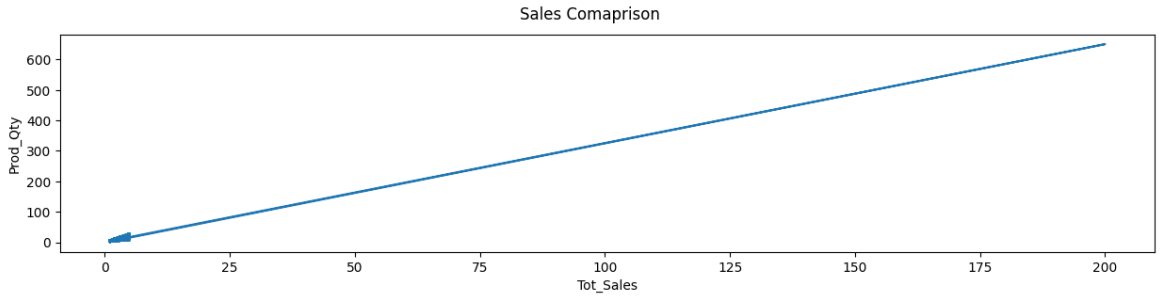
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7f266b59cc40>,
  <matplotlib.lines.Line2D at 0x7f266b59cee0>],
 'caps': [<matplotlib.lines.Line2D at 0x7f266b59d180>,
  <matplotlib.lines.Line2D at 0x7f266b59d420>],
 'boxes': [<matplotlib.lines.Line2D at 0x7f266b59c9a0>],
 'medians': [<matplotlib.lines.Line2D at 0x7f266b59d6c0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7f266b59d960>],
 'means': []}
```




```
plt.figure(figsize = (6, 4)) # determine the size of the figure
plt.hist(left_merged1["TOT_SALES"], # the variable on which to create the histogram
         bins = 1, # create a histogram with 1 bins
         color = "#108A99",
        edgecolor='#76D7C4'
        )
plt.title("Distribution of Total Sales", fontsize = 14, weight = "bold")
plt.xlabel("Price in (0's $)")
plt.ylabel("Number of Properties")
sns.despine() # removes the top and right border of our graph
plt.show()
```



```
fig = plt.figure(figsize=(15,3))
plt.plot(left_merged1.PROD_QTY, left_merged.TOT_SALES)
plt.xlabel('Tot_Sales')
plt.ylabel('Prod_Qty')
plt.suptitle('Sales Comaprison')
plt.show()
```




```
# from above line graph distribution as total sales increases product qty also increased.
```

```
# year wise and month wise total sales
left_merged["DATE_YEAR"]= pd.DatetimeIndex(left_merged['DATE']).year
left_merged["DATE_MONTH"]= pd.DatetimeIndex(left_merged['DATE']).month

pivot_table = left_merged.pivot_table(index=['DATE_YEAR', 'DATE_MONTH'], aggfunc={'TOT_SALES':sum})
pivot_table.reset_index().sort_values(['DATE_YEAR','DATE_MONTH'])
```

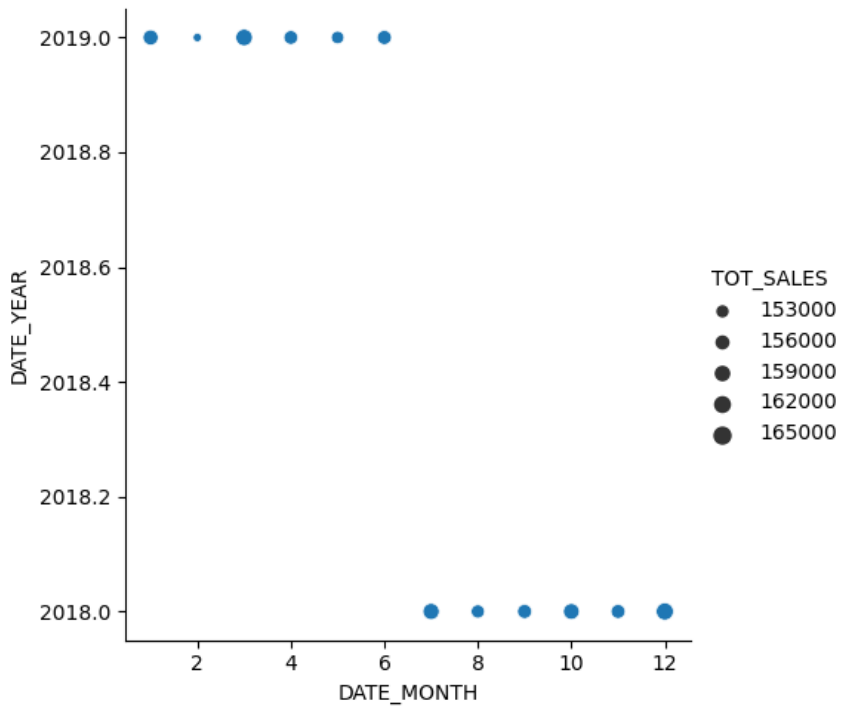
	DATE_YEAR	DATE_MONTH	TOT_SALES
0	2018	7	165275.30
1	2018	8	158731.05
2	2018	9	160522.00
3	2018	10	164415.70
4	2018	11	160233.70
5	2018	12	167913.40
6	2019	1	162642.30
7	2019	2	150665.00
8	2019	3	166265.20
9	2019	4	159845.10
10	2019	5	157367.65
11	2019	6	160538.60

```
# yearly Total Sales
left_merged["DATE_YEAR"]= pd.DatetimeIndex(left_merged['DATE']).year

pivot_table2 = left_merged.pivot_table(index=['DATE_YEAR'], aggfunc={'TOT_SALES':sum})
pivot_table2.reset_index().sort_values(['DATE_YEAR'])
```

	DATE_YEAR	TOT_SALES
0	2018	977091.15
1	2019	957323.85

```
g=sns.relplot(x='DATE_MONTH',
              y='DATE_YEAR',
              data=pivot_table,
              kind = 'scatter',
              size = 'TOT_SALES',
              #style = 'time'
             )
plt.show()
```




```
# from above scatter plot we find out that in 2018 in 12 th month there were highest sales and
# in 2019 in 3rd month there were highest sales
```

```
pivot_table3=left_merged.pivot_table(index=['PROD_NAME','SIZE'], aggfunc={'TOT_SALES':sum})
pivot_table3.reset_index().sort_values(['TOT_SALES'], ascending = False)
```

	PROD_NAME	SIZE	TOT_SALES
11	Dorito Corn Chp Supreme 380g	380g	40352.0
87	Smiths Crnkle Chip Orgnl Big Bag 380g	380g	36367.6
77	Smiths Crinkle Chips Salt & Vinegar 330g	330g	34804.2
33	Kettle Mozzarella Basil & Pesto 175g	175g	34457.4
86	Smiths Crinkle Original 330g	330g	34302.6
...	...	...	...
91	Sunbites Whlegrn Crisps Frch/Onin 90g	90g	4600.2
106	WW Crinkle Cut Original 175g	175g	4532.2
114	Woolworths Mild Salsa 300g	300g	4234.5
113	Woolworths Medium Salsa 300g	300g	4050.0
46	Natural Chip Compny SeaSalt175g	175g	6.0

115 rows × 3 columns

```
# from above pivot table it is find that Dorito Corn Chp Supreme 380g   have highest sales
```

```
pivot_table4=left_merged.pivot_table(index=['PROD_NAME','SIZE'], aggfunc={'PROD_QTY':sum})
pivot_table4.reset_index().sort_values(['PROD_QTY'], ascending = False)
```

	PROD_NAME	SIZE	PROD_QTY
11	Dorito Corn Chp Supreme 380g	380g	6509
33	Kettle Mozzarella Basil & Pesto 175g	175g	6381
42	Kettle Tortilla ChpsHny&Jlpno Chili 150g	150g	6309
8	Cobs Popd Sea Salt Chips 110g	110g	6277
10	Cobs Popd Swt/Chlli &Sr/Cream Chips 110g	110g	6256
...	...	...	...
113	Woolworths Medium Salsa 300g	300g	2700
43	NCC Sour Cream & Garden Chives 175g	175g	2682
106	WW Crinkle Cut Original 175g	175g	2666
21	French Fries Potato Chips 175g	175g	2643

```
# from above pivot table it is find that Dorito Corn Chp Supreme 380g   have highest sales with highest prod qty but Kettle Mozzarella Basil & Pesto 175g   175g having lowest sales
# Smiths Crnkle Chip Orgnl Big Bag 380g and Smiths Crinkle Chips Salt & Vinegar 330g with higher prod qty.
# if we increase the prod qty of Smiths Crnkle Chip Orgnl Big Bag 380g and Smiths Crinkle Chips Salt & Vinegar 330g we will get more sales.
```

```
pivot_merged = pd.merge(pivot_table3, pivot_table4, how="outer", on=["PROD_NAME","SIZE"])
pivot_merged
pivot_merged1 = pivot_merged.reset_index().sort_values(['TOT_SALES'], ascending = False)
pivot_merged1
```

	PROD_NAME	SIZE	TOT_SALES	PROD_QTY
11	Dorito Corn Chp Supreme 380g	380g	40352.0	6509
87	Smiths Crnkle Chip Orgnl Big Bag 380g	380g	36367.6	6164
77	Smiths Crinkle Chips Salt & Vinegar 330g	330g	34804.2	6106
33	Kettle Mozzarella Basil & Pesto 175g	175g	34457.4	6381
86	Smiths Crinkle Original 330g	330g	34302.6	6018
...	...	...	...	...
91	Sunbites Whlegrn Crisps Frch/Onin 90g	90g	4600.2	2706
106	WW Crinkle Cut Original 175g	175g	4532.2	2666
114	Woolworths Mild Salsa 300g	300g	4234.5	2823
113	Woolworths Medium Salsa 300g	300g	4050.0	2700
46	Natural Chip Compny SeaSalt175g	175g	6.0	2

115 rows × 4 columns

```
pivot_merged1.head(20)
```

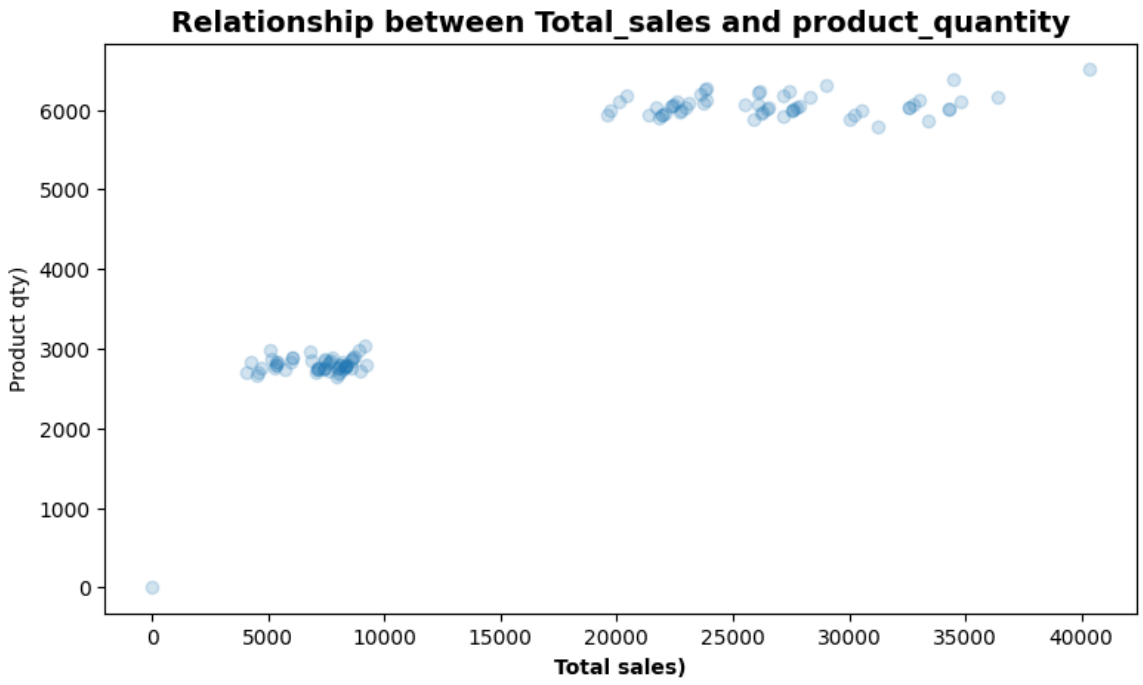
	PROD_NAME	SIZE	TOT_SALES	PROD_QTY
11	Dorito Corn Chp Supreme 380g	380g	40352.0	6509
87	Smiths Crnkle Chip Orgnl Big Bag 380g	380g	36367.6	6164
77	Smiths Crinkle Chips Salt & Vinegar 330g	330g	34804.2	6106
33	Kettle Mozzarella Basil & Pesto 175g	175g	34457.4	6381
86	Smiths Crinkle Original 330g	330g	34302.6	6018
6	Cheezels Cheese 330g	330g	34296.9	6017
12	Doritos Cheese Supreme 330g	330g	33390.6	5858
39	Kettle Sweet Chilli And Sour Cream 175g	175g	33031.8	6120
34	Kettle Original 175g	175g	32740.2	6064
35	Kettle Sea Salt And Vinegar 175g	175g	32589.0	6035
32	Kettle Honey Soy Chicken 175g	175g	32578.2	6033
31	Kettle Chilli 175g	175g	31271.4	5792
49	Old El Paso Salsa Dip Chnky Tom Ht300g	300g	30513.3	5986
50	Old El Paso Salsa Dip Tomato Med 300g	300g	30237.9	5929
51	Old El Paso Salsa Dip Tomato Mild 300g	300g	30033.9	5890
42	Kettle Tortilla ChpsHny&Jlpno Chili 150g	150g	29021.4	6309
37	Kettle Sensations Camembert & Fig 150g	150g	28308.4	6157
102	Twisties Chicken270g	270g	27853.0	6055
40	Kettle Tortilla ChpsBtroot&Ricotta 150g	150g	27770.2	6037
41	Kettle Tortilla ChpsFeta&Garlic 150g	150g	27627.6	6008

```
pivot_merged1.tail(20)
```

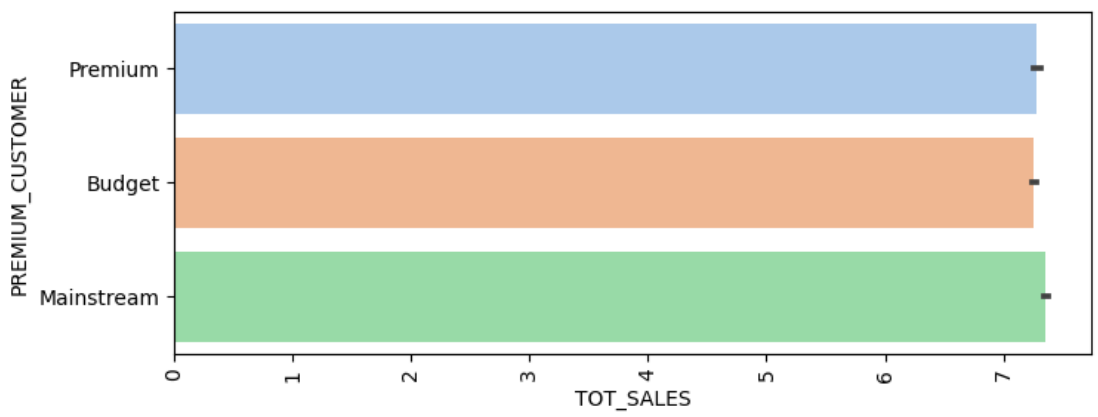
	PROD_NAME	SIZE	TOT_SALES	PROD_QTY
82	Smiths Crinkle Cut French OnionDip 150g	150g	7046.0	2710
26	Infuzions Mango Chutny Papadums 70g	70g	6852.0	2855
0	Burger Rings 220g	220g	6831.0	2970
3	CCs Tasty Cheese 175g	175g	6069.0	2890
2	CCs Original 175g	175g	6048.0	2880
1	CCs Nacho Cheese 175g	175g	5961.9	2839
7	Cheezels Cheese Box 125g	125g	5733.0	2730
111	WW Supreme Cheese Corn Chips 200g	200g	5390.3	2837
108	WW Original Corn Chips 200g	200g	5367.5	2825
109	WW Original Stacked Chips 160g	160g	5323.8	2802
110	WW Sour Cream &OnionStacked Chips 160g	160g	5323.8	2802
107	WW D/Style Chip Sea Salt 200g	200g	5249.7	2763
112	Woolworths Cheese Rings 190g	190g	5169.6	2872
90	Snbts Whlgrn Crisps Cheddr&Mstrd 90g	90g	5076.2	2986
105	WW Crinkle Cut Chicken 175g	175g	4702.2	2766
91	Sunbites Whlegrn Crisps Frch/Onin 90g	90g	4600.2	2706
106	WW Crinkle Cut Original 175g	175g	4532.2	2666
114	Woolworths Mild Salsa 300g	300g	4234.5	2823
113	Woolworths Medium Salsa 300g	300g	4050.0	2700
46	Natural Chip Compny SeaSalt175g	175g	6.0	2

```
plt.figure(figsize = (9, 5))

scatter = plt.scatter(pivot_merged1['TOT_SALES'],
            pivot_merged1['PROD_QTY'],
            alpha = 0.2,
             )
#plt.legend()
plt.title("Relationship between Total_sales and product_quantity",
          fontsize = 14,
          weight = "bold")
plt.xlabel("Total sales)", weight = "bold")
plt.ylabel("Product qty)")
plt.show()
```



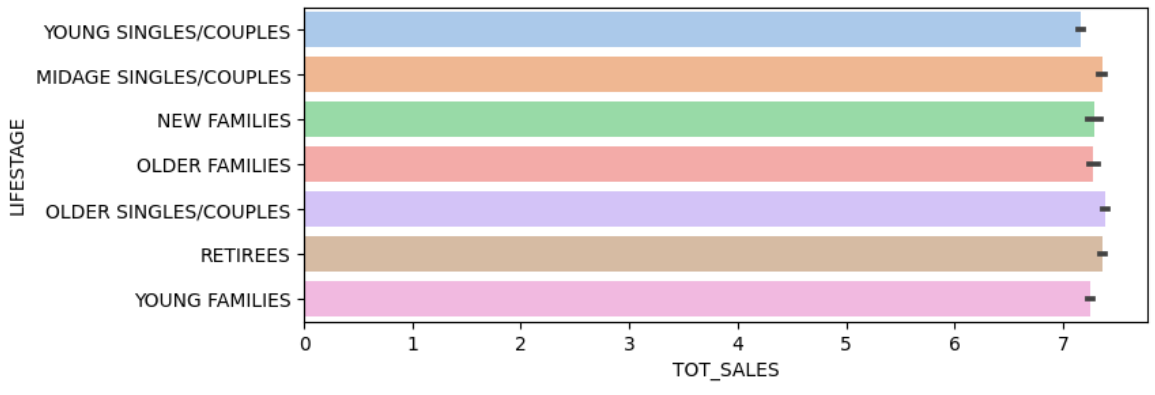

```
plt.figure(figsize = (8,3))
pal = sns.color_palette("pastel")
sns.barplot(x="TOT_SALES",y="PREMIUM_CUSTOMER",data=left_merged, palette=pal)
plt.xticks(rotation = 90, fontsize = 10)
plt.show()
```




```
# from above chart i come to conclusion that Mainstream premium customer gives us more sales.
```

```
```

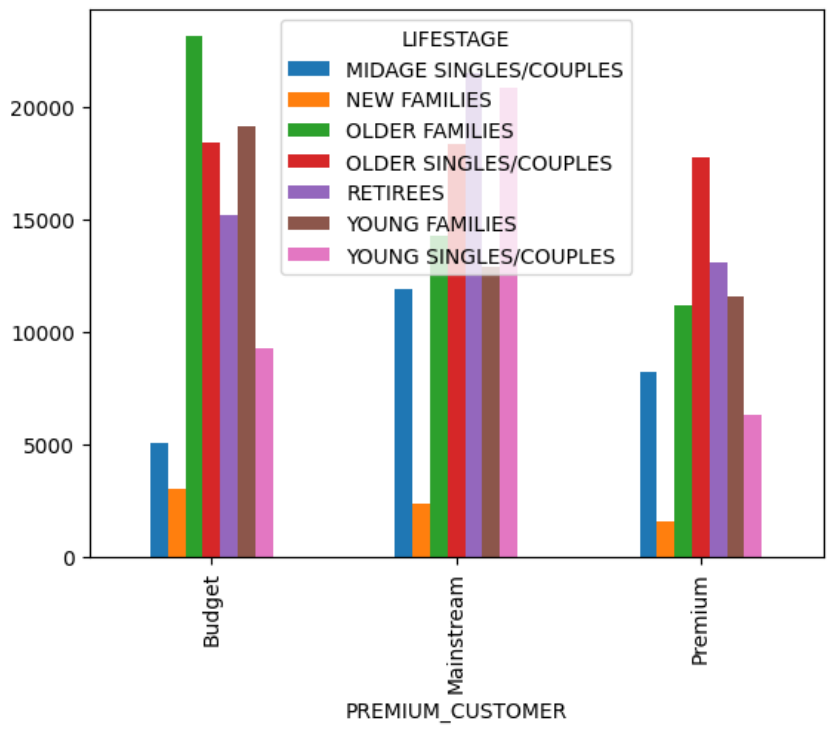
```
plt.figure(figsize = (8,3))
pal = sns.color_palette("pastel")
sns.barplot(x="TOT_SALES",y="LIFESTAGE",data=left_merged, palette=pal)
plt.show()
```




```
# From above chart i came to know about midage/single couples, older/single couples, retirees get more sales.
```

```
pd.crosstab(left_merged.PREMIUM_CUSTOMER,left_merged.LIFESTAGE).plot(kind="bar")
```

```
<Axes: xlabel='PREMIUM_CUSTOMER'>
```




```
# from Budget lifestage Older families and young families give higher sales
# from Mainstream Retirees, young/single couples give more sales.
# from Premium Older/single couples , retirees give more sales.
```

```
left_merged["PREMIUM_CUSTOMER"].value_counts()
left_merged.PREMIUM_CUSTOMER.value_counts().plot(kind="pie")
```

```
<Axes: ylabel='PREMIUM_CUSTOMER'>
```