In [2]:

```
import numpy as np
import pandas as pd
import seaborn as sns; sns.set()
import plotly.graph_objects as go
import plotly.express as px
import plotly
import matplotlib.pyplot as plt
import re
from scipy import stats

%notebook matplotlib
```

In [3]:

```
df_purchase = pd.read_csv("./QVI_purchase_behaviour.csv")
df_purchase.head()
```

Out[3]:

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

In [4]:

```
df_purchase.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 72637 entries, 0 to 72636
Data columns (total 3 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   LYLTY_CARD_NBR    72637 non-null  int64 
 1   LIFESTAGE         72637 non-null  object
 2   PREMIUM_CUSTOMER  72637 non-null  object
dtypes: int64(1), object(2)
memory usage: 1.1+ MB
```

In [5]:

```
df_purchase.shape
```

Out[5]:

```
(72637, 3)
```

In [6]:

```
print(df_purchase["PREMIUM_CUSTOMER"].nunique())
print(df_purchase["LIFESTAGE"].nunique())
```

```
3
7
```

In [7]:
```
df_transaction = pd.read_excel("QVI_transaction_data.xlsx")
df_transaction.head()
```

Out[7]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT
0	43390	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	
1	43599	1	1307	348	66	CCs Nacho Cheese 175g	3	
2	43605	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	
3	43329	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	
4	43330	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	

In [8]:
```
df_transaction.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 264836 entries, 0 to 264835
Data columns (total 8 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   DATE            264836 non-null  int64  
 1   STORE_NBR       264836 non-null  int64  
 2   LYLTY_CARD_NBR  264836 non-null  int64  
 3   TXN_ID          264836 non-null  int64  
 4   PROD_NBR        264836 non-null  int64  
 5   PROD_NAME       264836 non-null  object 
 6   PROD_QTY        264836 non-null  int64  
 7   TOT_SALES       264836 non-null  float64
dtypes: float64(1), int64(6), object(1)
memory usage: 15.2+ MB
```

In [9]:
```
df_transaction.shape
```

Out[9]:
```
(264836, 8)
```

In [10]:
```
df_analyze = df_transaction.merge(df_purchase,how="left",on="LYLTY_CARD_NBR").dr
```

In [11]:
```
df_analyze.shape
```

Out[11]:
```
(264836, 10)
```

In [12]:
```
df_analyze.head()
```

Out[12]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	T
0	43390	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	
1	43599	1	1307	348	66	CCs Nacho Cheese 175g	3	
2	43605	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	
3	43329	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	
4	43330	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	

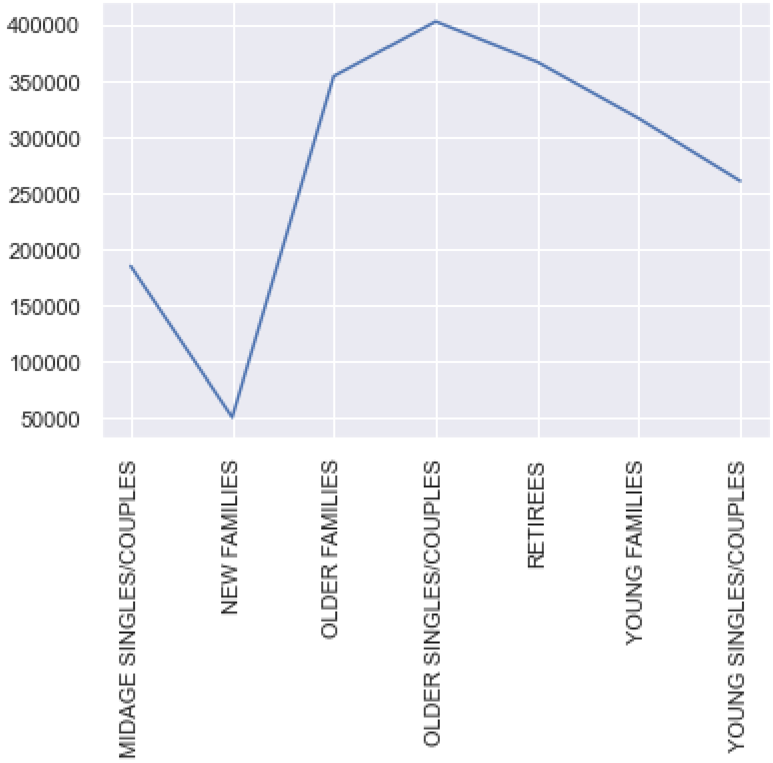
In [13]:
```
df_Life_sales = df_analyze[["TOT_SALES","LIFESTAGE"]].groupby("LIFESTAGE").agg(np
df_Life_sales
```

Out[13]:

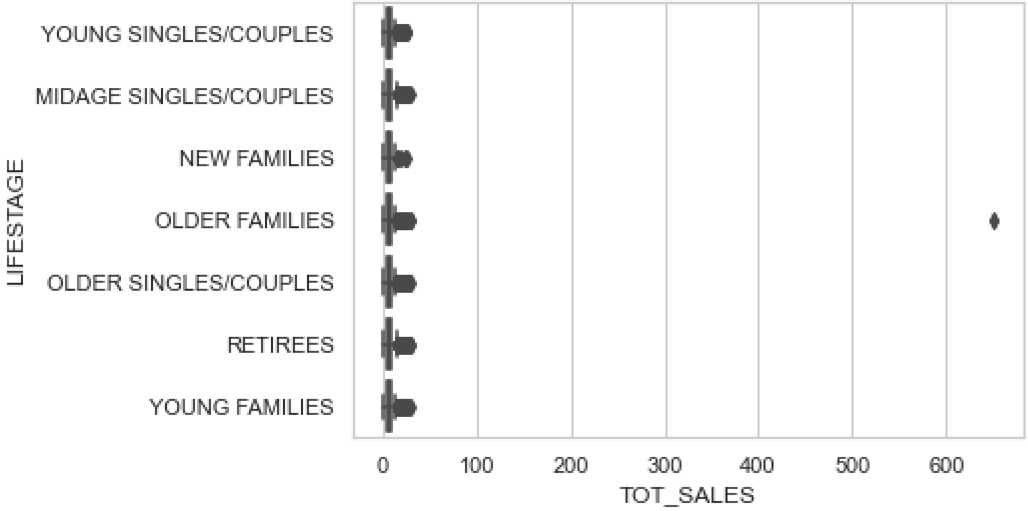
	TOT_SALES
LIFESTAGE	
MIDAGE SINGLES/COUPLES	184751.30
NEW FAMILIES	50433.45
OLDER FAMILIES	353767.20
OLDER SINGLES/COUPLES	402426.75
RETIREES	366470.90
YOUNG FAMILIES	316160.10
YOUNG SINGLES/COUPLES	260405.30

In [14]:

```
# Before detecting anomalies
plt.plot(df_Life_sales)

plt.tick_params(axis="x", labelrotation=90)
plt.tick_params(axis="y", labelrotation=0)

plt.figure;
```



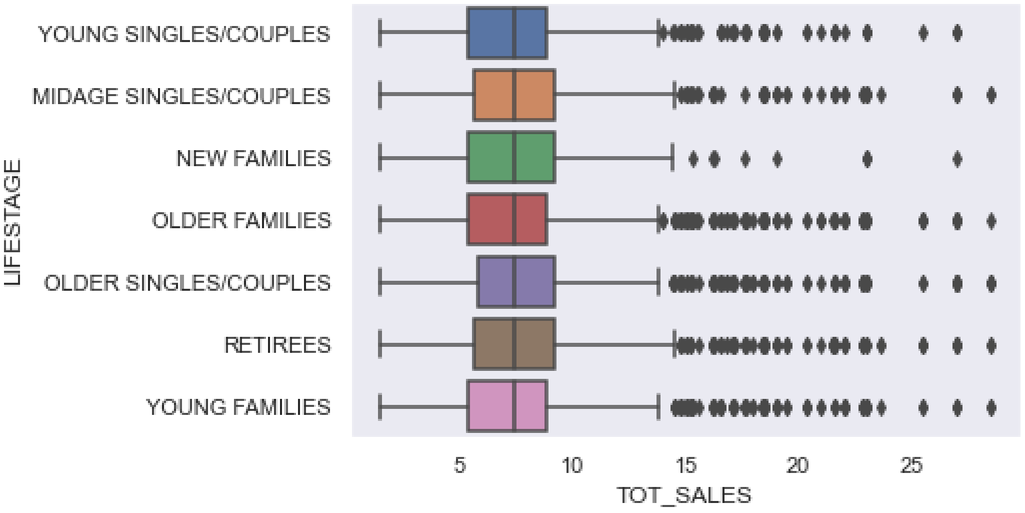
In [15]:
```
sns.set_theme(style="whitegrid")
ax = sns.boxplot(x=df_analyze["TOT_SALES"], y=df_analyze["LIFESTAGE"])
```



In [16]:
```
# Remove outliers
def reject_outliers(sr, iq_range=0.995):
    pcnt = (1 - iq_range) / 2
    qlow, median, qhigh = sr.quantile([pcnt, 0.50, 1-pcnt])
    iqr = qhigh - qlow
    return sr[ (sr - median).abs() <= iqr]

df_analyze["TOT_SALES"]  = reject_outliers(df_analyze["TOT_SALES"], 0.999)
df_analyze["TOT_SALES"]
```

Out[16]:
```
0          6.0
1          6.3
2          2.9
3         15.0
4         13.8
          ...
264831    10.8
264832     4.4
264833     8.8
264834     7.8
264835     8.8
Name: TOT_SALES, Length: 264836, dtype: float64
```

In [17]:
```
sns.set_theme(style="dark")
ax = sns.boxplot(x=df_analyze["TOT_SALES"], y=df_analyze["LIFESTAGE"])
```

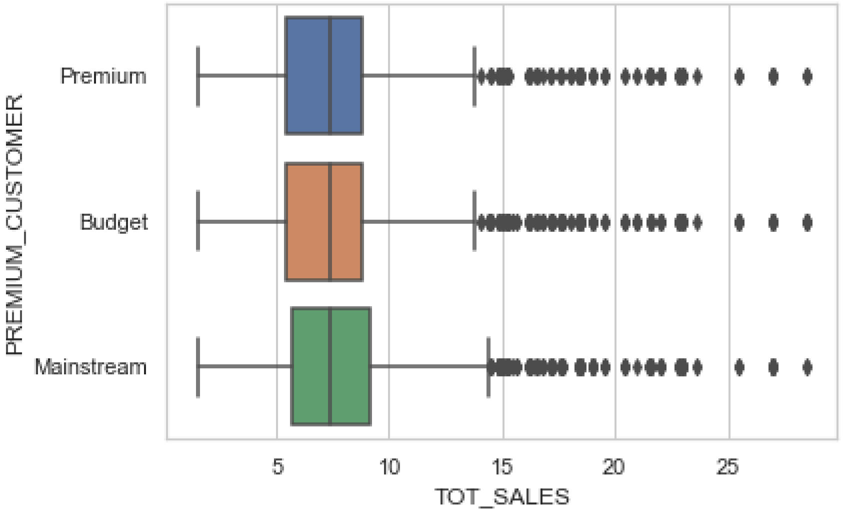


In [18]:
```
df_status_sales = df_analyze[["TOT_SALES","PREMIUM_CUSTOMER"]].groupby("PREMIUM_C
df_status_sales
```

Out[18]:

	TOT_SALES
PREMIUM_CUSTOMER	
Budget	676182.05
Mainstream	750656.00
Premium	506070.45

In [19]:
```
sns.set_theme(style="whitegrid")
ax = sns.boxplot(x=df_analyze["TOT_SALES"], y=df_analyze["PREMIUM_CUSTOMER"])
```



In [20]:

```
df_life_status_sales = df_analyze[["TOT_SALES","PREMIUM_CUSTOMER","LIFESTAGE"]].g
df_life_status_sales
```

Out[20]:

```
                                                TOT_SALES
PREMIUM_CUSTOMER                LIFESTAGE
          Budget   MIDAGE SINGLES/COUPLES        35514.80
                             NEW FAMILIES        21928.45
                           OLDER FAMILIES       168363.25
                   OLDER SINGLES/COUPLES        136769.80
                                 RETIREES       113147.80
                           YOUNG FAMILIES       139316.35
                   YOUNG SINGLES/COUPLES         61141.60
      Mainstream   MIDAGE SINGLES/COUPLES        90774.35
                             NEW FAMILIES        17013.90
                           OLDER FAMILIES       103416.05
                   OLDER SINGLES/COUPLES        133393.80
                                 RETIREES       155647.55
                           YOUNG FAMILIES        92788.75
                   YOUNG SINGLES/COUPLES        157621.60
         Premium   MIDAGE SINGLES/COUPLES        58432.65
                             NEW FAMILIES        11491.10
                           OLDER FAMILIES        80628.90
                   OLDER SINGLES/COUPLES        132233.65
                                 RETIREES        97616.55
                           YOUNG FAMILIES        84025.50
                   YOUNG SINGLES/COUPLES         41642.10
```

In [21]:

```
df_life_status_sales.index
```

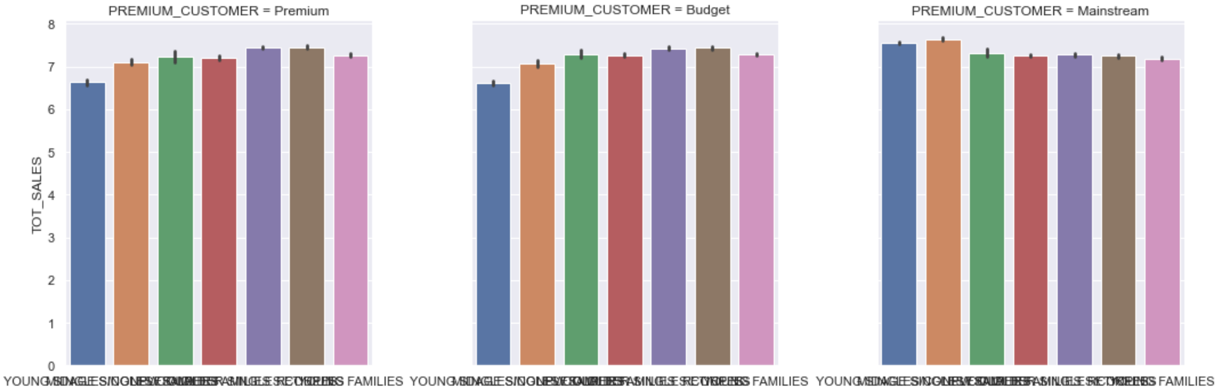
Out[21]:

```
MultiIndex([(     'Budget', 'MIDAGE SINGLES/COUPLES'),
            (     'Budget',           'NEW FAMILIES'),
            (     'Budget',         'OLDER FAMILIES'),
            (     'Budget',  'OLDER SINGLES/COUPLES'),
            (     'Budget',               'RETIREES'),
            (     'Budget',         'YOUNG FAMILIES'),
            (     'Budget',  'YOUNG SINGLES/COUPLES'),
            ('Mainstream', 'MIDAGE SINGLES/COUPLES'),
            ('Mainstream',           'NEW FAMILIES'),
            ('Mainstream',         'OLDER FAMILIES'),
            ('Mainstream',  'OLDER SINGLES/COUPLES'),
            ('Mainstream',               'RETIREES'),
            ('Mainstream',         'YOUNG FAMILIES'),
            ('Mainstream',  'YOUNG SINGLES/COUPLES'),
            (    'Premium', 'MIDAGE SINGLES/COUPLES'),
            (    'Premium',           'NEW FAMILIES'),
            (    'Premium',         'OLDER FAMILIES'),
            (    'Premium',  'OLDER SINGLES/COUPLES'),
            (    'Premium',               'RETIREES'),
            (    'Premium',         'YOUNG FAMILIES'),
            (    'Premium',  'YOUNG SINGLES/COUPLES')],
           names=['PREMIUM_CUSTOMER', 'LIFESTAGE'])
```

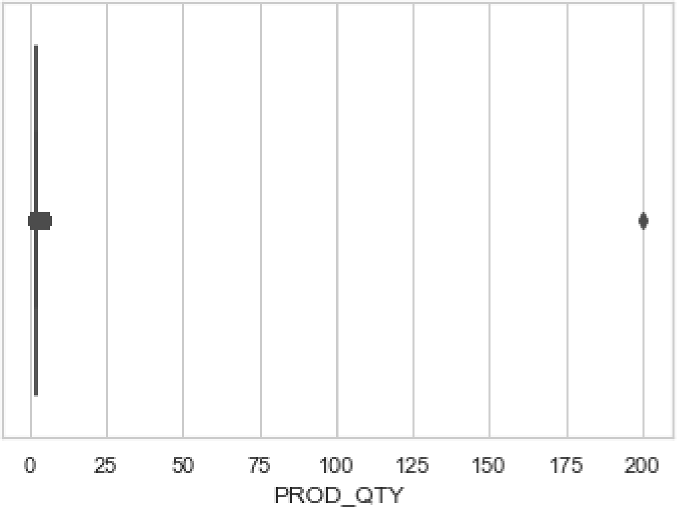
In [22]:

```

sns.set(style = "darkgrid")

fig = sns.factorplot(x="LIFESTAGE", y='TOT_SALES',data= df_analyze[["TOT_SALES",'
                     kind='bar', col="PREMIUM_CUSTOMER")
fig.set_xlabels('');
```

```
c:\users\user\appdata\local\programs\python\python38-32\lib\site-packages\seabo
rn\categorical.py:3704: UserWarning: The `factorplot` function has been renamed
to `catplot`. The original name will be removed in a future release. Please upd
ate your code. Note that the default `kind` in `factorplot` (`'point'`) has cha
nged `'strip'` in `catplot`.
  warnings.warn(msg)
```



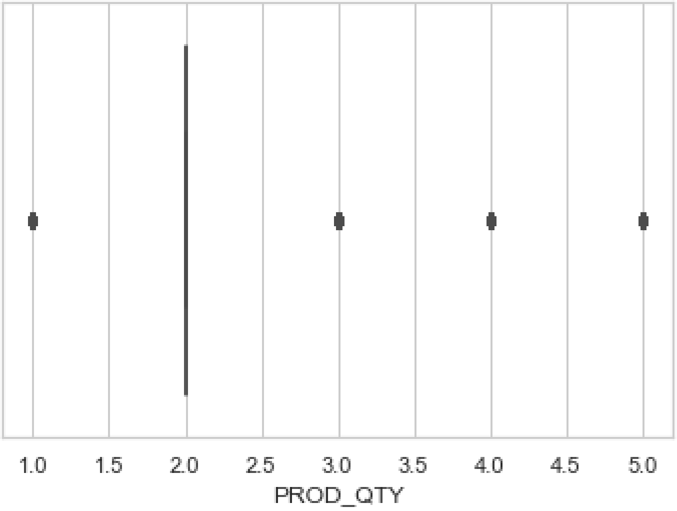
In [23]:
```
sns.set_theme(style="whitegrid")
ax = sns.boxplot( x=df_analyze["PROD_QTY"])
```



In [24]:
```
df_analyze["PROD_QTY"]  = reject_outliers(df_analyze["PROD_QTY"], 0.999)
df_analyze["PROD_QTY"]
```

Out[24]:
```
0         2.0
1         3.0
2         2.0
3         5.0
4         3.0
         ...
264831    2.0
264832    1.0
264833    2.0
264834    2.0
264835    2.0
Name: PROD_QTY, Length: 264836, dtype: float64
```

In [25]:
```
sns.set_theme(style="whitegrid")
ax = sns.boxplot(x=df_analyze["PROD_QTY"])
```



In [26]:
```
df_analyze.shape
```

Out[26]:
```
(264836, 10)
```

In [27]:
```
df_analyze = df_analyze.dropna()
```

In [28]:
```
df_analyze.shape
```

Out[28]:
```
(264827, 10)
```

In [29]:
```
df_analyze["PROD_NAME"]
```

Out[29]:
```
0           Natural Chip        Compny SeaSalt175g
1                     CCs Nacho Cheese    175g
2           Smiths Crinkle Cut  Chips Chicken 170g
3           Smiths Chip Thinly  S/Cream&Onion 175g
4         Kettle Tortilla ChpsHny&Jlpno Chili 150g
                            ...
264831     Kettle Sweet Chilli And Sour Cream 175g
264832               Tostitos Splash Of  Lime 175g
264833                    Doritos Mexicana    170g
264834     Doritos Corn Chip Mexican Jalapeno 150g
264835               Tostitos Splash Of  Lime 175g
Name: PROD_NAME, Length: 264827, dtype: object
```

In [30]:
```
pd.to_numeric(df_analyze['DATE']);
```

In [31]:
```
pd.to_datetime(df_analyze['DATE'])
df_analyze.head()
```

Out[31]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT
0	43390	1	1000	1	5	Natural Chip Compny SeaSalt175g	2.0	
1	43599	1	1307	348	66	CCs Nacho Cheese 175g	3.0	
2	43605	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2.0	
3	43329	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5.0	
4	43330	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3.0	

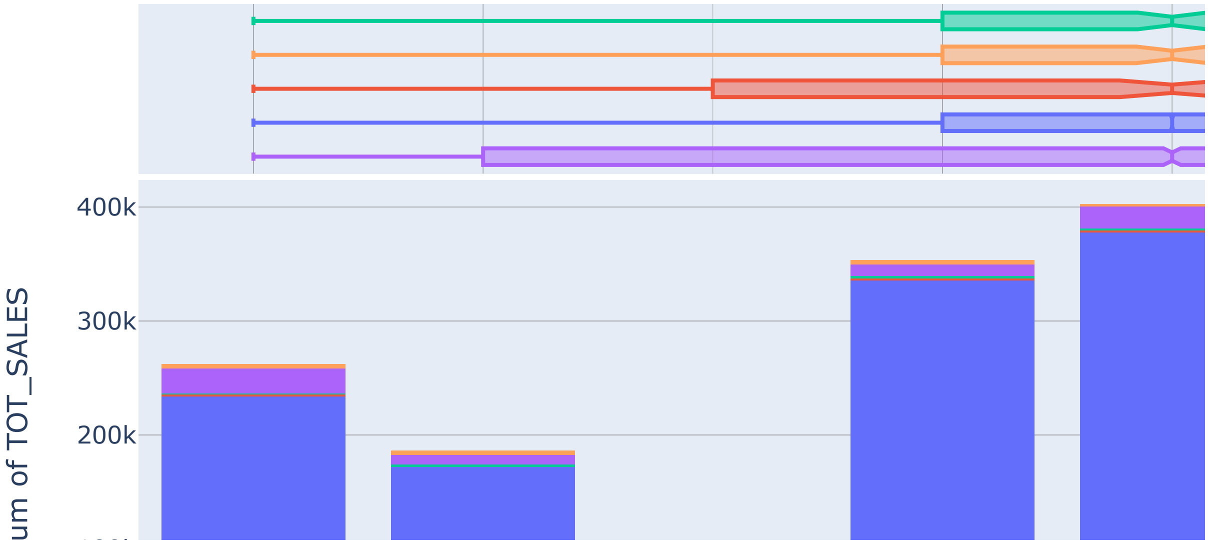
In [32]:
```
df_analyze = df_analyze.dropna()
```

In [33]:
```
# df_analyze["DATE"] = pd.to_datetime(df_analyze["DATE"], origin = '1899-12-30').
df_analyze["DATE"]
```

Out[33]:
```
0         43390
1         43599
2         43605
3         43329
4         43330
          ...
264831    43533
264832    43325
264833    43410
264834    43461
264835    43365
Name: DATE, Length: 264827, dtype: int64
```

In [34]:
```
df_analyze['LYLTY_CARD_NBR'] = df_analyze['LYLTY_CARD_NBR'].astype('str')
df_analyze['TXN_ID'] = df_analyze['TXN_ID'].astype('str')
df_analyze['STORE_NBR'] = df_analyze['STORE_NBR'].astype('str')
df_analyze['PROD_NBR'] = df_analyze['PROD_NBR'].astype('str')
```

In [35]:
```
fig = px.histogram(df_analyze, x="LIFESTAGE", y="TOT_SALES", color="PROD_QTY",
                   marginal="box",
                   hover_data=df_analyze.columns)
fig.show()
```



In [36]:
```
df_analyze["DATE"] = pd.to_datetime(df_analyze['DATE'], origin = '1899-12-30', ur
```

In [37]:

```
df_analyze.head()
```

Out[37]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_
0	2018-10-17	1	1000	1	5	Natural Chip Compny SeaSalt175g	2.0	
1	2019-05-14	1	1307	348	66	CCs Nacho Cheese 175g	3.0	
2	2019-05-20	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2.0	
3	2018-08-17	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5.0	
4	2018-08-18	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3.0	

In [38]:

```
df_total_sales_day = df_analyze[["DATE","TOT_SALES"]].groupby(["DATE"]).agg(np.su
df_total_sales_day
```

Out[38]:

	TOT_SALES
DATE	
2018-07-01	5372.2
2018-07-02	5315.4
2018-07-03	5321.8
2018-07-04	5309.9
2018-07-05	5080.9
...	...
2019-06-26	5305.0
2019-06-27	5202.8
2019-06-28	5299.6
2019-06-29	5497.6
2019-06-30	5423.4

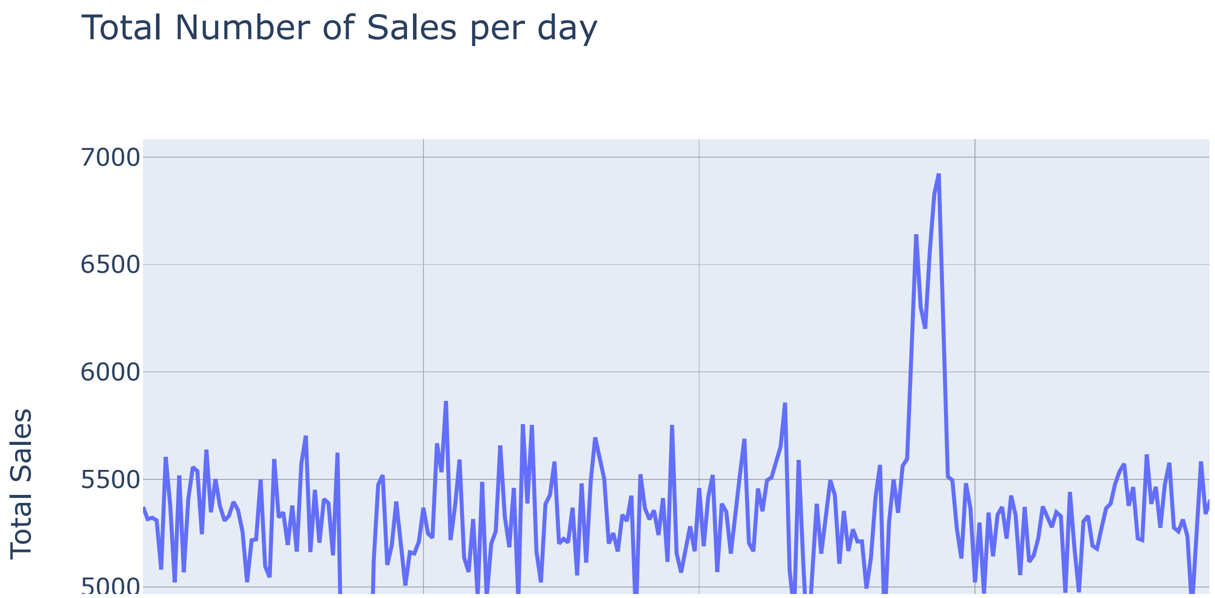
364 rows × 1 columns

In [47]:

```
df_total_sales_day.index = pd.to_datetime(df_total_sales_day.index).strftime('%Y-
```

In [92]:

```
df_total_sales_day.loc['2018-12-25'] = np.nan
plt = px.line(x=df_total_sales_day.index, y=df_total_sales_day["TOT_SALES"], data
                labels={
                     "TOT_SALES": 'Total Sales',
                     "DATE": "Date"
                 },
             title = "Total Number of Sales per day");
plt.show()
```


Total Number of Sales per day

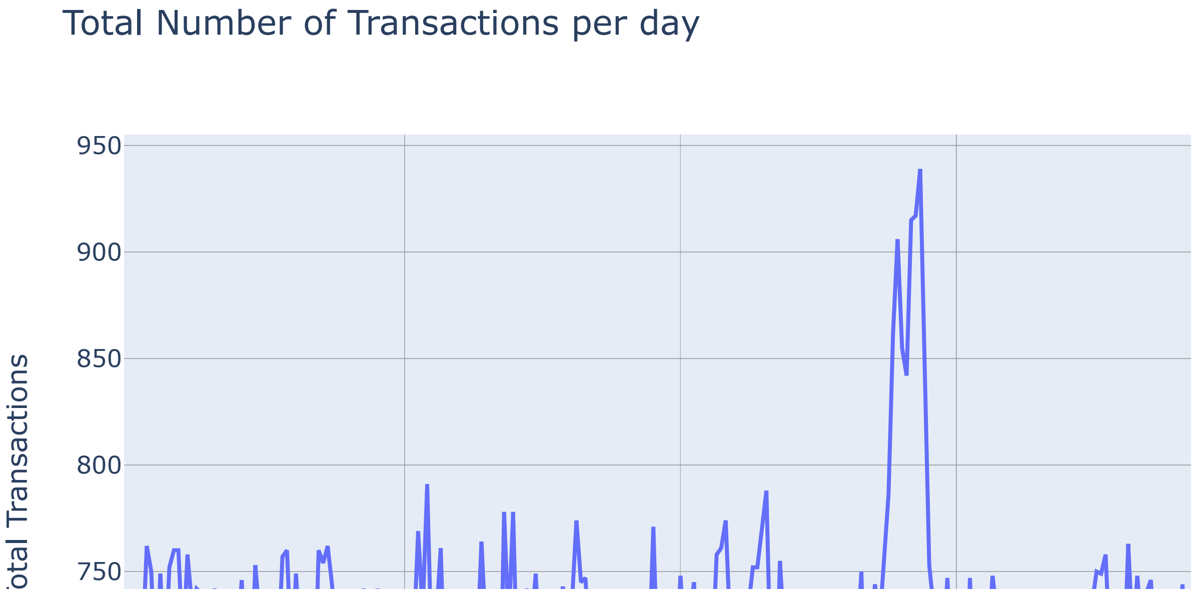
In [88]:

```
df_total_transactions = df_analyze[["DATE","TOT_SALES"]].groupby(["DATE"]).agg(le
df_total_transactions.loc['2018-12-25'] = np.nan
df_total_transactions = df_total_transactions.rename(columns={'TOT_SALES':"TOT_Tr
df_total_transactions
```

Out[88]:

```
                     TOT_Transactions
               DATE
2018-07-01 00:00:00             724.0
2018-07-02 00:00:00             711.0
2018-07-03 00:00:00             722.0
2018-07-04 00:00:00             714.0
2018-07-05 00:00:00             712.0
                ...               ...
2019-06-27 00:00:00             709.0
2019-06-28 00:00:00             730.0
2019-06-29 00:00:00             745.0
2019-06-30 00:00:00             744.0
         2018-12-25               NaN

365 rows × 1 columns
```

```
In [91]: df_total_transactions.loc['2018-12-25'] = np.nan
         sns.set_style("darkgrid")
         plt = px.line(x=df_total_transactions.index, y=df_total_transactions["TOT_Transac
                       labels={
                             "TOT_Transactions": 'Total Transactions',
                             "DATE": "Date"
                         },
                     title = "Total Number of Transactions per day");

         plt.show()
```



In [93]:
```
df_products = df_analyze.copy()
df_products.head()
```

Out[93]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_
0	2018-10-17	1	1000	1	5	Natural Chip Compny SeaSalt175g	2.0	
1	2019-05-14	1	1307	348	66	CCs Nacho Cheese 175g	3.0	
2	2019-05-20	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2.0	
3	2018-08-17	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5.0	
4	2018-08-18	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3.0	

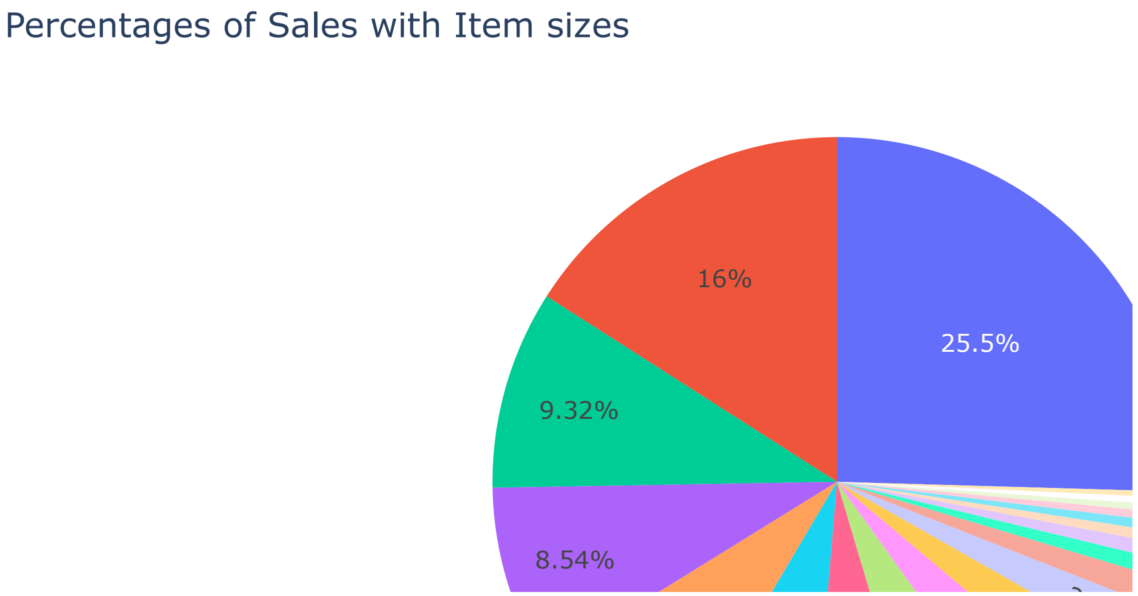
In [123]:
```
df_products["Brands"] = df_products["PROD_NAME"].apply(lambda x: x.strip().split(
df_products["Size (g)"] = df_products["PROD_NAME"].apply(lambda x: x.strip().spli
df_products = df_products[df_products["Size (g)"] != "salt" ]
df_products["Brands"]
df_products["Size (g)"]
```

Out[123]:
```
0         175g
1         175g
2         170g
3         175g
4         150g
          ... 
264831    175g
264832    175g
264833    170g
264834    150g
264835    175g
Name: Size (g), Length: 261570, dtype: object
```

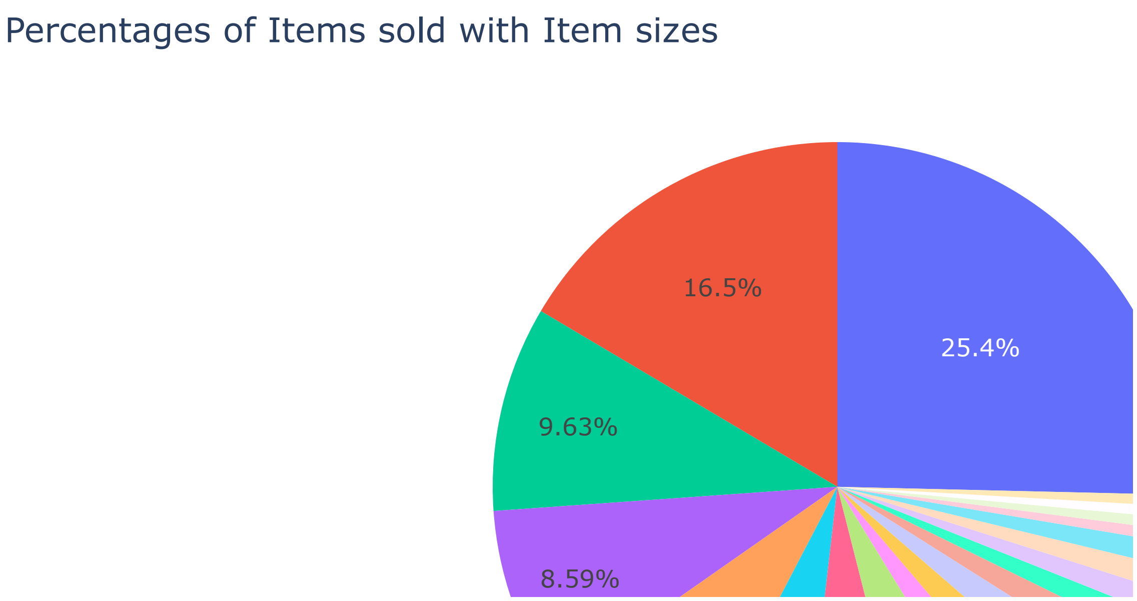
In [128]:
```
df_grouped_product = df_products.groupby(["Size (g)"]).agg(np.sum)
```

In [130]:

```
fig = px.pie(df_grouped_product , values="TOT_SALES", names=df_grouped_product.ir
fig.show()
```



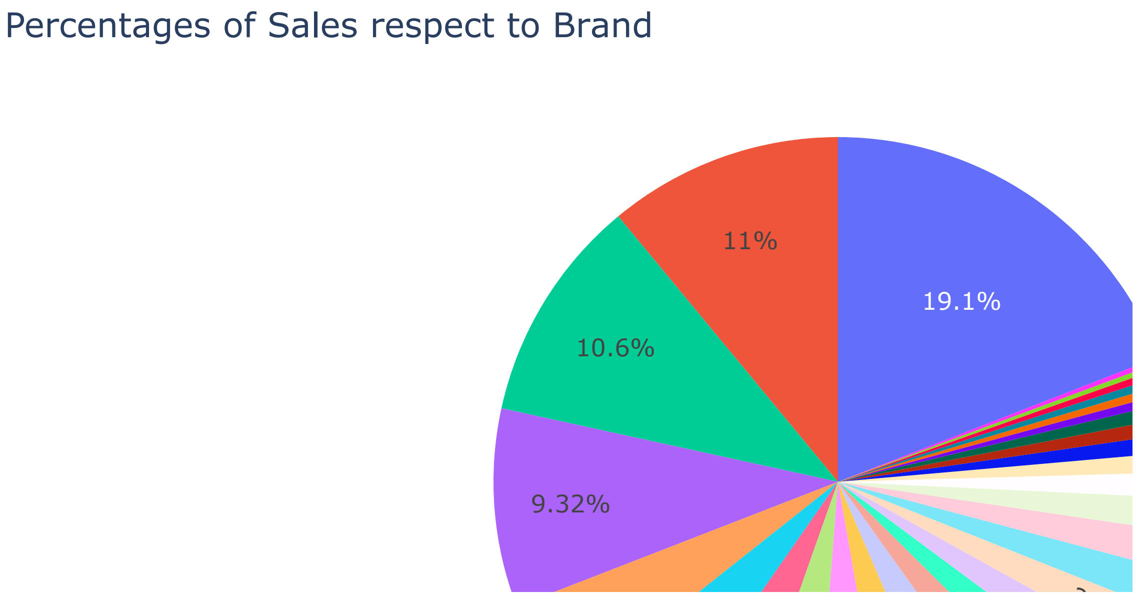
In [131]:
```
fig = px.pie(df_grouped_product , values="PROD_QTY", names=df_grouped_product.ind
fig.show()
```



In [142]:
```
df_grouped_product = df_products.groupby("Brands").agg(np.sum)
```

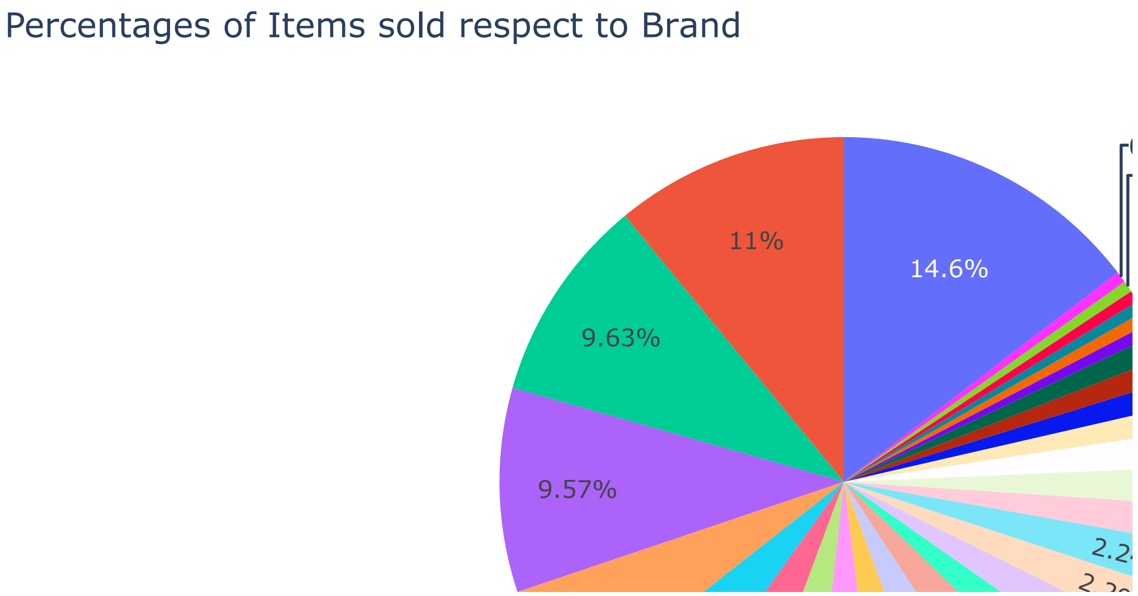
In [143]:

```
= px.pie(df_grouped_product , values="TOT_SALES", names=df_grouped_product.index
.show()
```


Percentages of Sales respect to Brand

In [144]:

```
fig = px.pie(df_grouped_product , values="PROD_QTY", names=df_grouped_product.inc
fig.show()
```



Percentages of Items sold respect to Brand

In [148]:

```
df_store = df_analyze.copy()
```

In [203]:
```
df_store_grouped =df_store.groupby("STORE_NBR").agg(np.sum)
df_store_grouped["STORE"] = df_store_grouped.index.copy()
df_store_grouped["STORE"] = df_store_grouped["STORE"].apply(lambda x: "Store " +
df_store_grouped
```

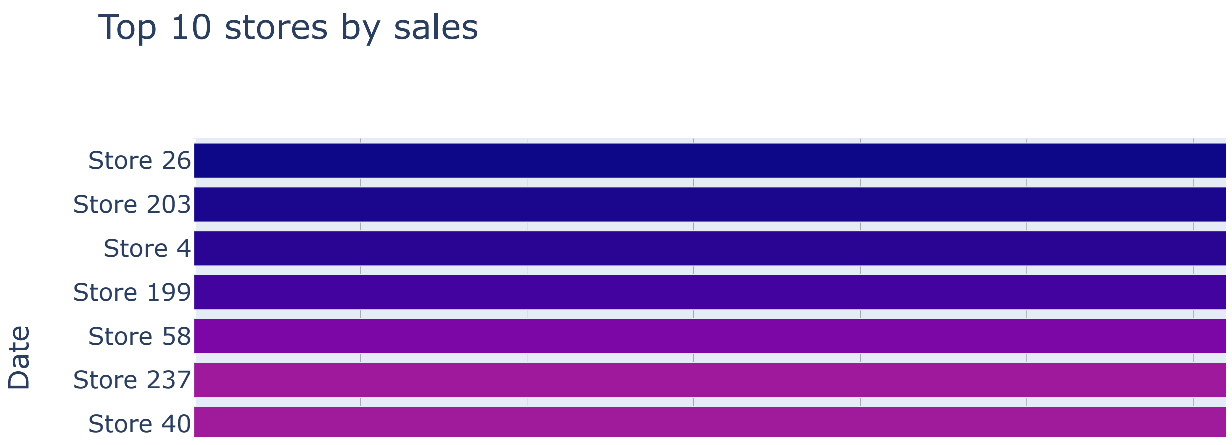
Out[203]:

	PROD_QTY	TOT_SALES	STORE
STORE_NBR			
226	4001.0	17605.45	Store 226
88	3718.0	16333.25	Store 88
165	3602.0	15973.75	Store 165
40	3499.0	15559.50	Store 40
237	3515.0	15539.50	Store 237
...	...	...	...
206	2.0	7.60	Store 206
252	2.0	7.40	Store 252
11	2.0	6.70	Store 11
76	2.0	6.00	Store 76
211	2.0	5.20	Store 211

272 rows × 3 columns

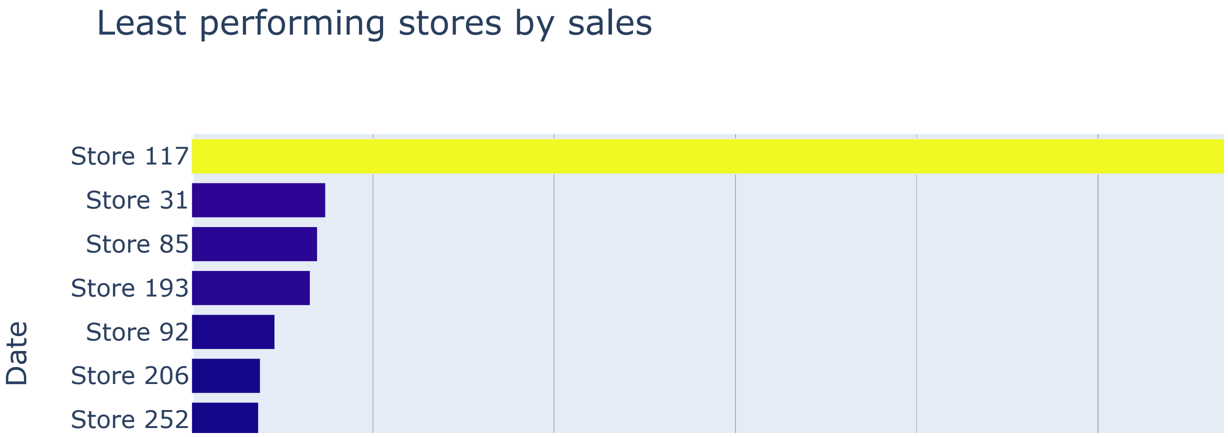
Top 10 Stores

In [219]:
```
fig = px.bar(df_store_grouped[:10], x="TOT_SALES", y=df_store_grouped[:10]["STORE
             hover_data=["PROD_QTY", "TOT_SALES"],
             height=400,
             labels={
                     "TOT_SALES": 'Total Sales',
                     "y": "Date"
                 },
             title='Top 10 stores by sales')
fig.show()
```



In [218]:

```
df_store_grouped_des = df_store_grouped.sort_values("TOT_SALES")
fig = px.bar(df_store_grouped_des[:10], x="TOT_SALES", y=df_store_grouped_des[:10
             hover_data=["PROD_QTY", "TOT_SALES"],
             height=400,
             labels={
                     "TOT_SALES": 'Total Sales',
                     "y": "Date"
                 },
             title='Least performing stores by sales')
fig.show()
```



Least performing stores by sales

In [221]:

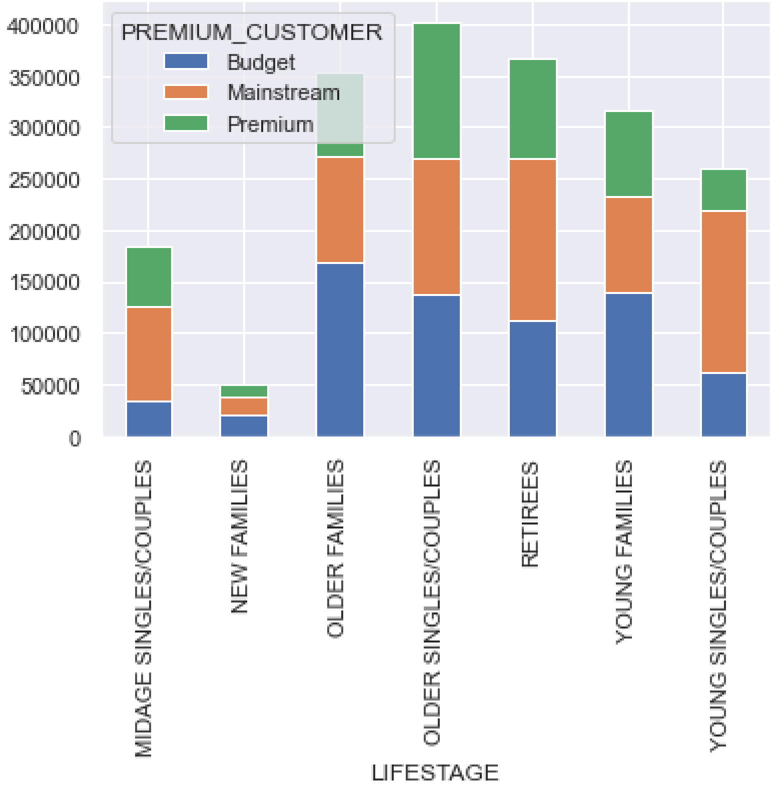
```
df_customer_sal = df_analyze.copy()
df_customer_sal.head()
```

Out[221]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_
0	2018-10-17	1	1000	1	5	Natural Chip Compny SeaSalt175g	2.0	
1	2019-05-14	1	1307	348	66	CCs Nacho Cheese 175g	3.0	
2	2019-05-20	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2.0	
3	2018-08-17	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5.0	
4	2018-08-18	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3.0	

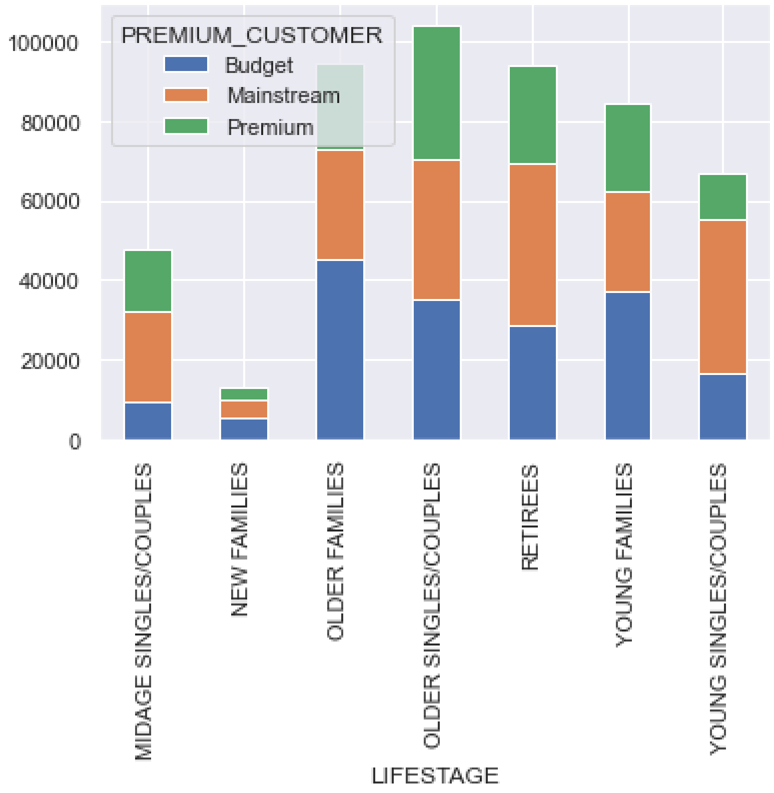
In [228]:

```
df_customer_sal_g = df_customer_sal.groupby(["LIFESTAGE", "PREMIUM_CUSTOMER"]).ag
# df_customer_sal_d = df_customer_sal.set_index(['PROD_QTY','TOT_SALES']).value

df_customer_sal_g["TOT_SALES"].unstack().plot(kind='bar', stacked=True);
# df_customer_sal_g
```



In [229]:

```
df_customer_sal_g["PROD_QTY"].unstack().plot(kind='bar', stacked=True);
```



Summary:

> Overall Sale Trends
>
> - The highest sales happened a day before christmas (24 December 2018): 6923 Sales
>   Also this trend start increase drasctically 1 week before christmas day and the transactions in this week also have uptrend - no doubt

- The lowest sales happened on 18 May 2019: 4036.5 Sales
  Also this extreme downtrend start 1 week before, and this is also happened on August last year(2018) almost in the same date but for the transactions still look the same as the another days/weeks, so maybe there are some big discount in these two weeks maybe

Focused on these Customer Segments:

- Retirees
- Older Singles/Couples
- Older Families
- Young Families
- Young Singles/Couples

With top 5 brands

- Kettle
- Smiths
- Doritos
- Pringles
- Old

  Top 5 sizes
- 175g
- 150g
- 134g
- 110g
- 170g

Focused on Stores with respect to Sales

- Leader was Store 226

Conclusion

It is better to focus on the products in the mid range sales area as they are mostly used by the segments where sales are not generated enough

```
In [ ]:
```