

```
In [1]: 1 import pandas as pd
        2 import numpy as np
        3 import matplotlib.pyplot as plt
        4 import seaborn as sns
        5 %matplotlib inline
```

```
In [2]: 1 store = pd.read_csv("QVI_data.csv")
        2 store.head()
```

```
Out[2]:
```

	LYLTY_CARD_NBR	DATE	STORE_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	PACK_SIZE	BRAND	LIFE
0	1000	2018-10-17	1	1	5	Natural Chip Compny SeaSalt175g	2	6.0	175	NATURAL	SINGLES/C
1	1002	2018-09-16	1	2	58	Red Rock Deli Chikn&Garlic Aioli 150g	1	2.7	150	RRD	SINGLES/C
2	1003	2019-03-07	1	3	52	Grain Waves Sour Cream&Chives 210G	1	3.6	210	GRNWVES	YOUNG F
3	1003	2019-03-08	1	4	106	Natural ChipCo Hony Soy Chckn175g	1	3.0	175	NATURAL	YOUNG F
4	1004	2018-11-02	1	5	96	WW Original Stacked Chips 160g	1	1.9	160	WOOLWORTHS	SINGLES/C

```
In [3]: 1 store['DATE'] = pd.to_datetime(store['DATE'])
```

```
In [4]: 1 df = store[store['STORE_NBR'].isin([77, 86, 88])].reset_index(drop=True)
        2 df.head()
```

```
Out[4]:
```

	LYLTY_CARD_NBR	DATE	STORE_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	PACK_SIZE	BRAND	LIFE
0	77000	2019-03-28	77	74911	18	Cheetos Chs & Bacon Balls 190g	1	3.3	190	CHEETOS	SINGLES/CO M
1	77000	2019-04-13	77	74912	69	Smiths Chip Thinly S/Cream&Onion 175g	1	3.0	175	SMITHS	SINGLES/CO M
2	77000	2018-09-26	77	74910	36	Kettle Chilli 175g	2	10.8	175	KETTLE	SINGLES/CO M
3	77001	2019-02-27	77	74913	7	Smiths Crinkle Original 330g	2	11.4	330	SMITHS	YOUNG FA
4	77001	2019-01-21	77	74914	9	Kettle Tortilla ChpsBtroot&Ricotta 150g	2	9.2	150	KETTLE	YOUNG FA

```
In [5]: 1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3974 entries, 0 to 3973
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  -
0   LYLTY_CARD_NBR        3974 non-null  int64
1   DATE                  3974 non-null  datetime64[ns]
2   STORE_NBR             3974 non-null  int64
3   TXN_ID                3974 non-null  int64
4   PROD_NBR              3974 non-null  int64
5   PROD_NAME             3974 non-null  object
6   PROD_QTY              3974 non-null  int64
7   TOT_SALES             3974 non-null  float64
8   PACK_SIZE             3974 non-null  int64
9   BRAND                 3974 non-null  object
10  LIFESTAGE             3974 non-null  object
11  PREMIUM_CUSTOMER      3974 non-null  object
dtypes: datetime64[ns](1), float64(1), int64(6), object(4)
memory usage: 372.7+ KB
```

In [ ]:

1

In [6]:

1 df.duplicated().sum()

Out[6]: 0

In [7]:

1 df.isnull().sum()

Out[7]:

LYLTY_CARD_NBR	0
DATE	0
STORE_NBR	0
TXN_ID	0
PROD_NBR	0
PROD_NAME	0
PROD_QTY	0
TOT_SALES	0
PACK_SIZE	0
BRAND	0
LIFESTAGE	0
PREMIUM_CUSTOMER	0

dtype: int64

## Data Cleaning Completed

## Exploratory Data Analysis

### Key Metrics

```
In [8]: 1 def calculate_metrics(df):
2         total_sales = df['TOT_SALES'].sum()
3         total_customer = df['LYLTY_CARD_NBR'].nunique()
4         Average_per_transaction = df["TXN_ID"].nunique()/total_customer
5
6         return total_sales, total_customer, Average_per_transaction
```

```
In [9]: 1 metrics_dict = {}
2
3         for store_num in df['STORE_NBR']:
4             single_store = df[df['STORE_NBR']==store_num]
5             metrics_dict[store_num] = calculate_metrics(single_store)
6
7         print(metrics_dict)
```

```
{77: (3040.0, 356, 1.5786516853932584), 86: (10635.35, 273, 5.589743589743589), 88: (16333.25, 388, 4.786082474226804)}
```

```
In [10]: 1 for store_nbr, metrics in metrics_dict.items():
2         print(f"Store {store_nbr} Metrics:")
3         print(f"Total Sales: {metrics[0]}")
4         print(f"Total Customers: {metrics[1]}")
5         print(f"Avg Transactions per Customer: {metrics[2]}\n")
6
```

Store 77 Metrics:

Total Sales: 3040.0

Total Customers: 356

Avg Transactions per Customer: 1.5786516853932584

Store 86 Metrics:

Total Sales: 10635.35

Total Customers: 273

Avg Transactions per Customer: 5.589743589743589

Store 88 Metrics:

Total Sales: 16333.25

Total Customers: 388

Avg Transactions per Customer: 4.786082474226804

```
In [11]: 1 store_no = df.groupby('STORE_NBR')['TOT_SALES'].sum()  
2 store_no.plot(kind="bar", color = 'green')  
3 plt.title("Distribution of Total Sales in each store ")  
4 plt.ylabel("Total Sales")  
5 plt.xlabel("Store Number")  
6 plt.show()
```

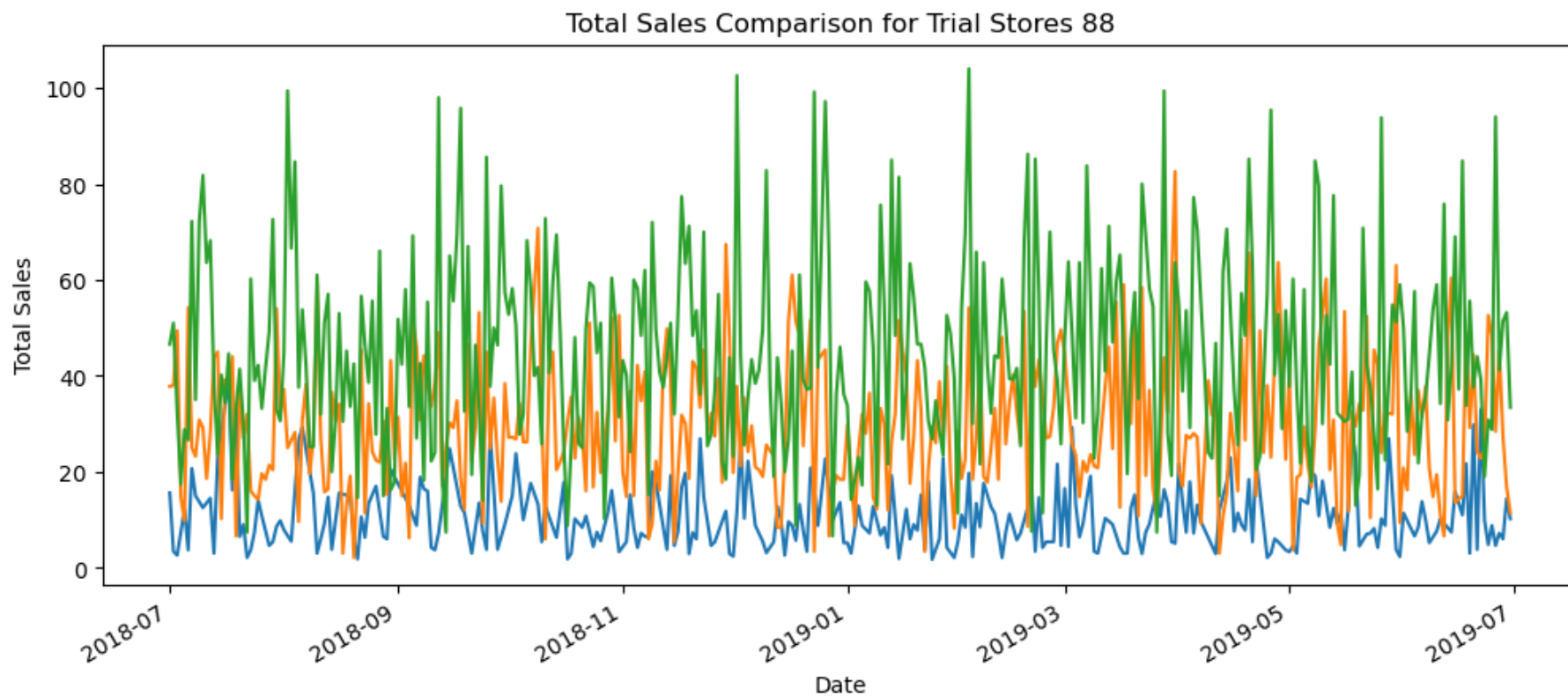


1 Store number 88 generates highest total sales as compared to others and 77 generates the lowest sales

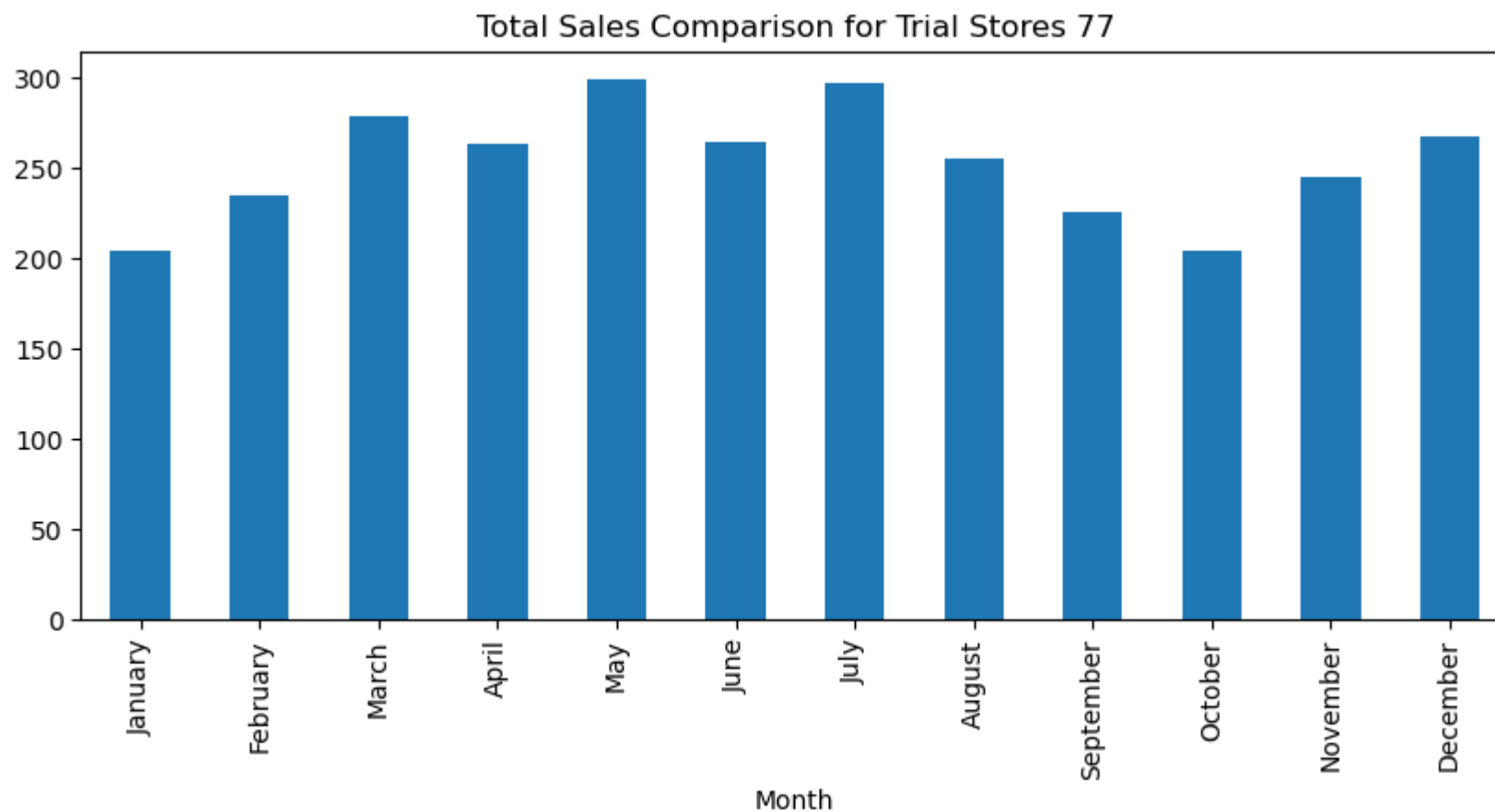
```
In [12]: 1 df['Month'] = df['DATE'].dt.month_name()
2 month_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September',
3               'October', 'November', 'December']
4 df['Month'] = pd.Categorical(df['Month'], categories=month_order, ordered=True)
```

```
In [13]: 1 store['Month'] = store['DATE'].dt.month_name()
2 month_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September',
3               'October', 'November', 'December']
4 store['Month'] = pd.Categorical(store['Month'], categories=month_order, ordered=True)
```

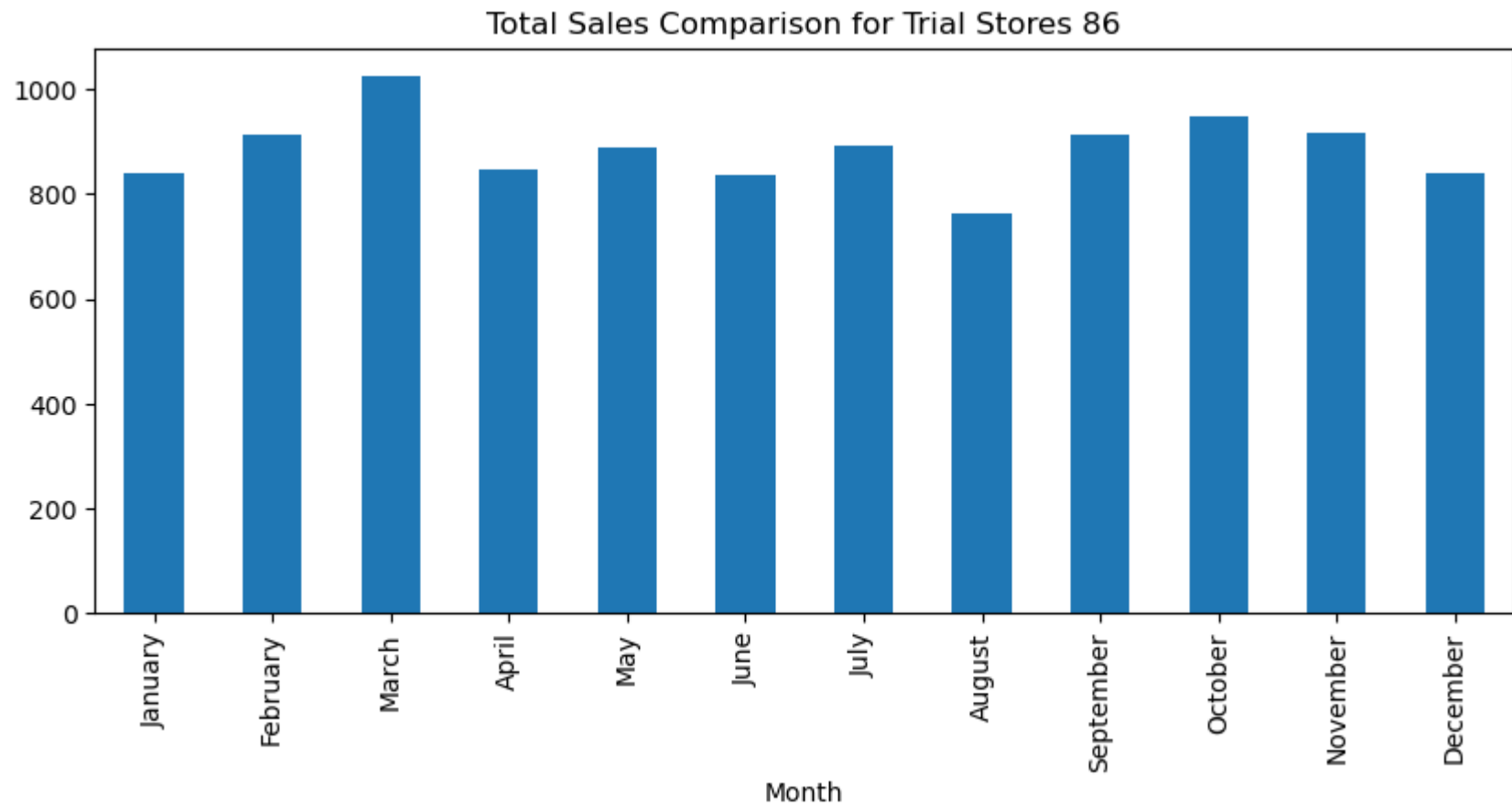
```
In [14]: 1 plt.figure(figsize=(12,5))
2 for store_num in df['STORE_NBR'].unique():
3     single_store = df[df['STORE_NBR']==store_num]
4     total_sale_overtime = single_store.groupby('DATE')['TOT_SALES'].sum()
5     total_sale_overtime.plot(label = f"Store {store_num}")
6     plt.title(f'Total Sales Comparison for Trial Stores {store_num}')
7 plt.xlabel('Date')
8 plt.ylabel('Total Sales')
9 plt.show()
```

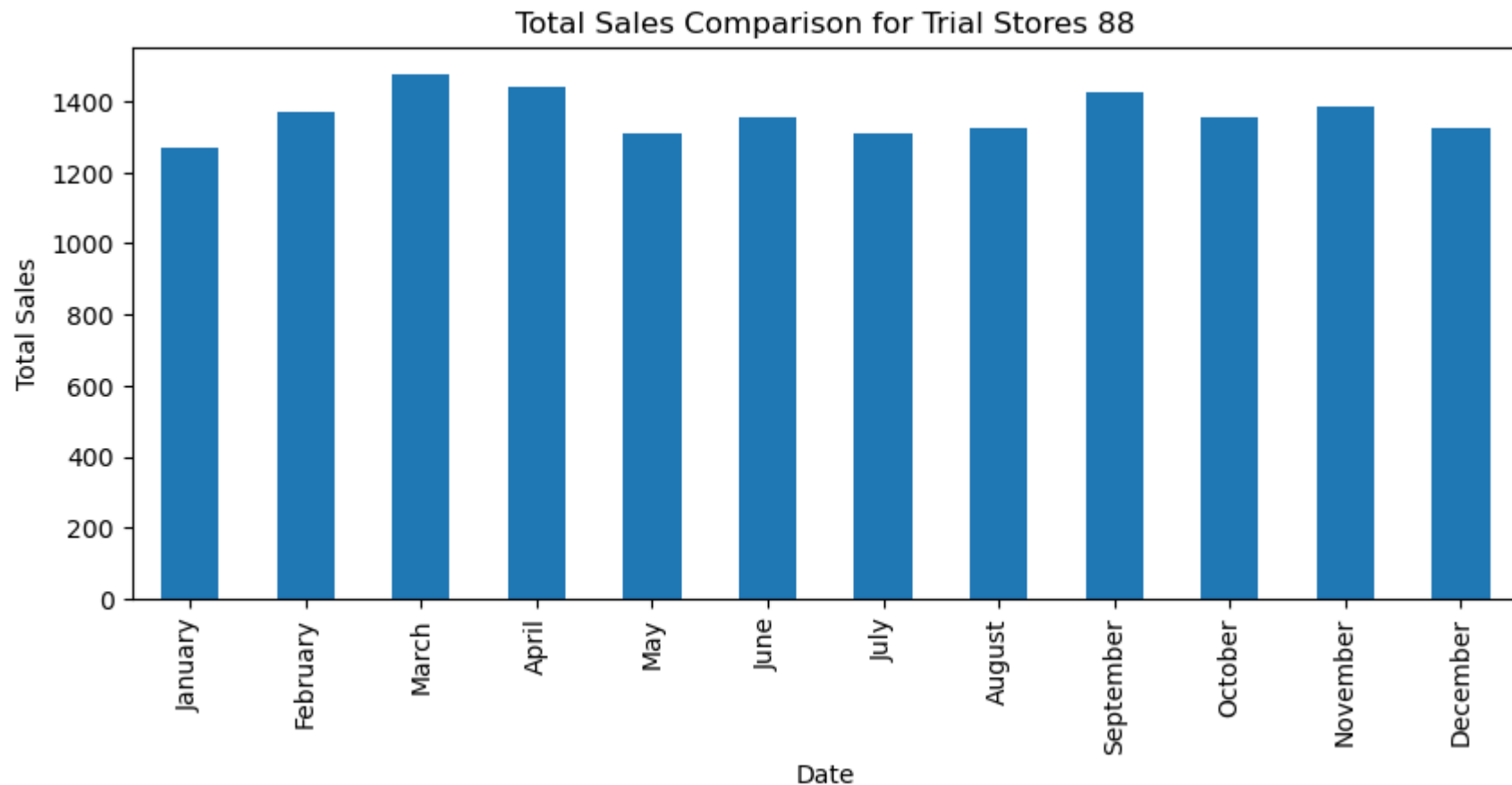


```
In [15]: 1
2 for store_num in df['STORE_NBR'].unique():
3     plt.figure(figsize=(10,4))
4     single_store = df[df['STORE_NBR']==store_num]
5     total_sale_overtime = single_store.groupby('Month')['TOT_SALES'].sum()
6     total_sale_overtime.plot(kind = 'bar',label = f"Store {store_num}")
7     plt.title(f'Total Sales Comparison for Trial Stores {store_num}')
8     plt.xlabel('Date')
9     plt.ylabel('Total Sales')
10    plt.show()
```









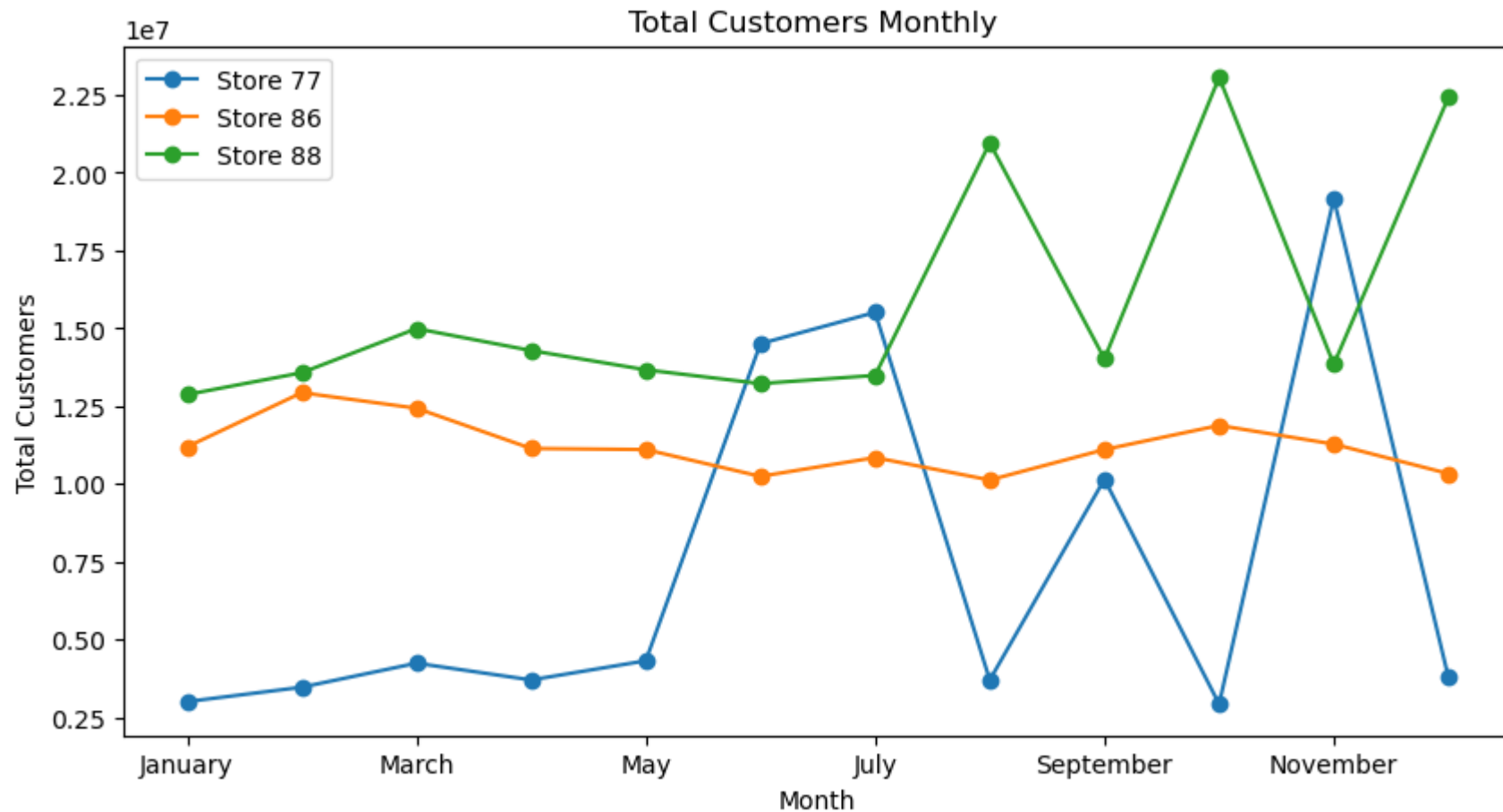
- ```
1 1. Store 77:
2   - Highest Sales Months: May and July
3   - Lowest Sales Months: October and January
4
5   Insight: Store 77 experiences peak sales during the summer months (May and July), potentially due to increased
6   customer activity during warmer weather. On the other hand, sales dip in the colder months (October and
7   January),
8   suggesting a seasonal pattern.
9
10 2. Store 86:
11   - Highest Sales Months: March and October
12   - Lowest Sales Month: August
```

```
13     Insight: Store 86 sees elevated sales in March and October, possibly indicating specific promotions or events
14     during these periods. August registers the lowest sales, which could be influenced by factors like
15     holidays or
16     reduced customer engagement.
17 3. Store 88:
18     - Highest Sales Months: March and April
19     - Lowest Sales Months: July and January
20
21     Insight: Store 88 performs exceptionally well in March and April, potentially benefiting from seasonal trends
22     or
23     promotional activities. July and January mark the lowest sales, suggesting a need for targeted strategies
24     during these
25     months to boost sales.
```

In [ ]:

1

```
In [32]: 1 plt.figure(figsize=(10,5))
2 for i in df['STORE_NBR'].unique():
3     single_store = df[df['STORE_NBR']==i]
4     total_customer = single_store.groupby('Month')['LYLTY_CARD_NBR'].sum()
5     total_customer.plot(label = f"Store {i}", marker = "o")
6
7
8
9
10
11 plt.title('Total Customers Monthly ')
12 plt.xlabel('Month')
13
14 plt.ylabel('Total Customers')
15 plt.legend()
16 plt.show()
```



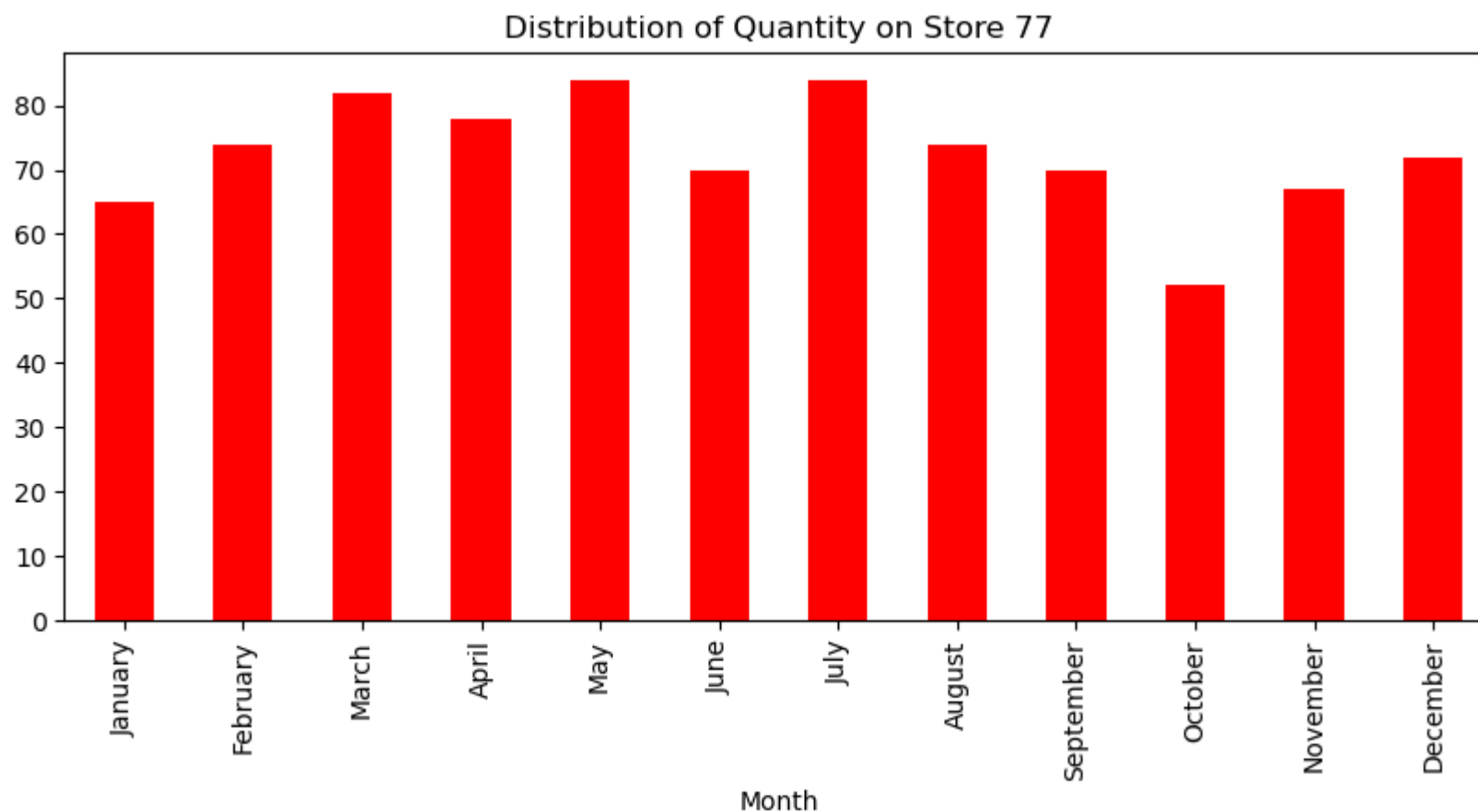
- ```
1 1. Store 77:
2   - Customer Count Peaks: May to July, October
3   - Customer Count Lows: August, December
4
5   Insight:Store 77 experiences peaks in the number of customers during the mid-year months (May to July) and
6   later in
7   October. Customer count declines in August and December, suggesting a seasonal pattern with two
8   significant peaks.
9
10 2. Store 86:
11   - Customer Count Consistency: Throughout the Year
```

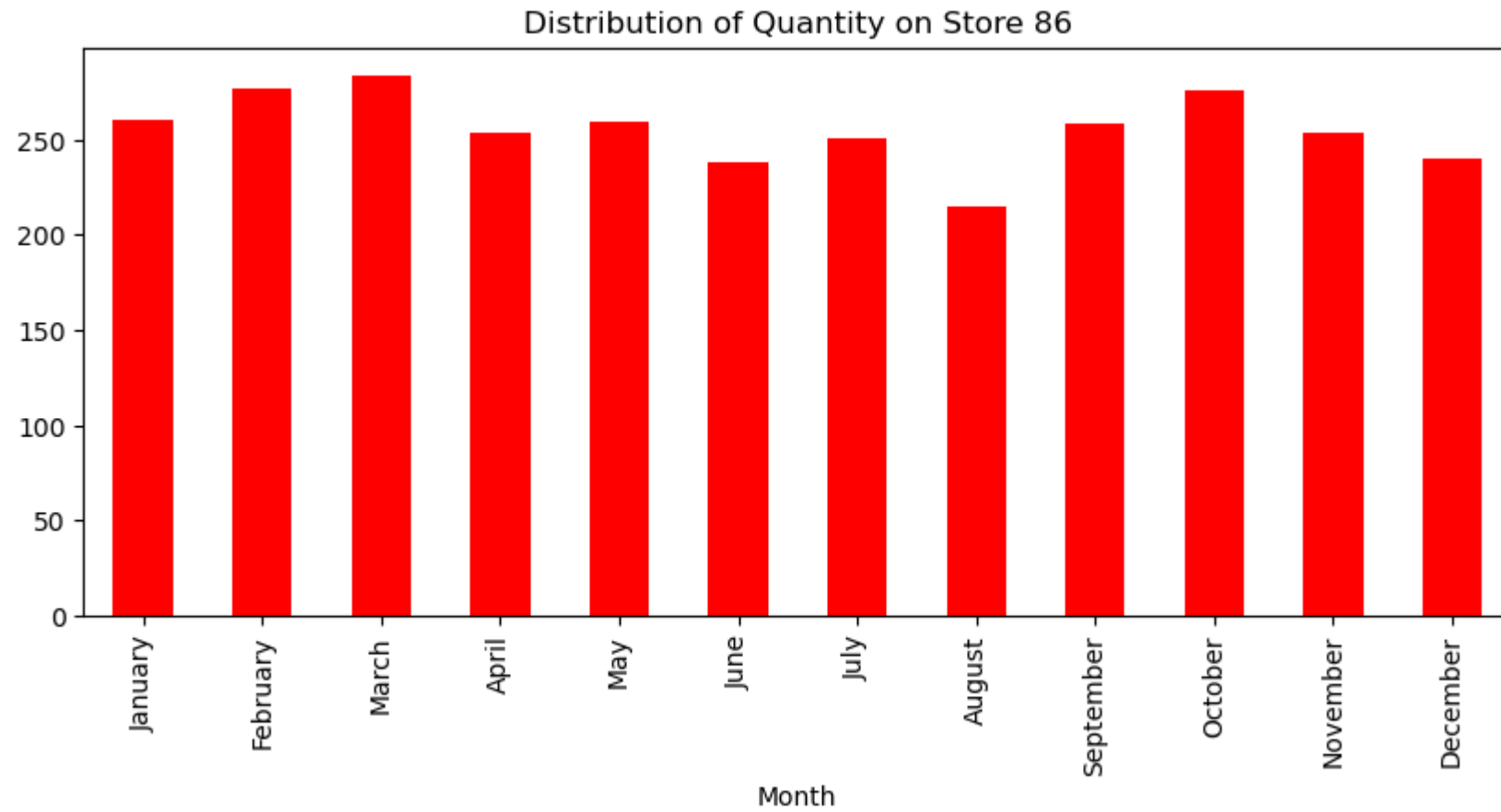
```
11     Insight: Store 86 maintains a consistent number of customers throughout the year, indicating a stable customer
12     base
13     without significant seasonal variations.
14 3. Store 88:
15     - Customer Count Peaks: July, October, December
16     - Customer Count Lows: September, November
17
18     Insight:Store 88 exhibits peaks in customer count during July, October, and December. The customer count
19     decreases in
20     September and November, with subsequent peaks and declines in the following months.
```

```
In [33]: 1 df.columns
```

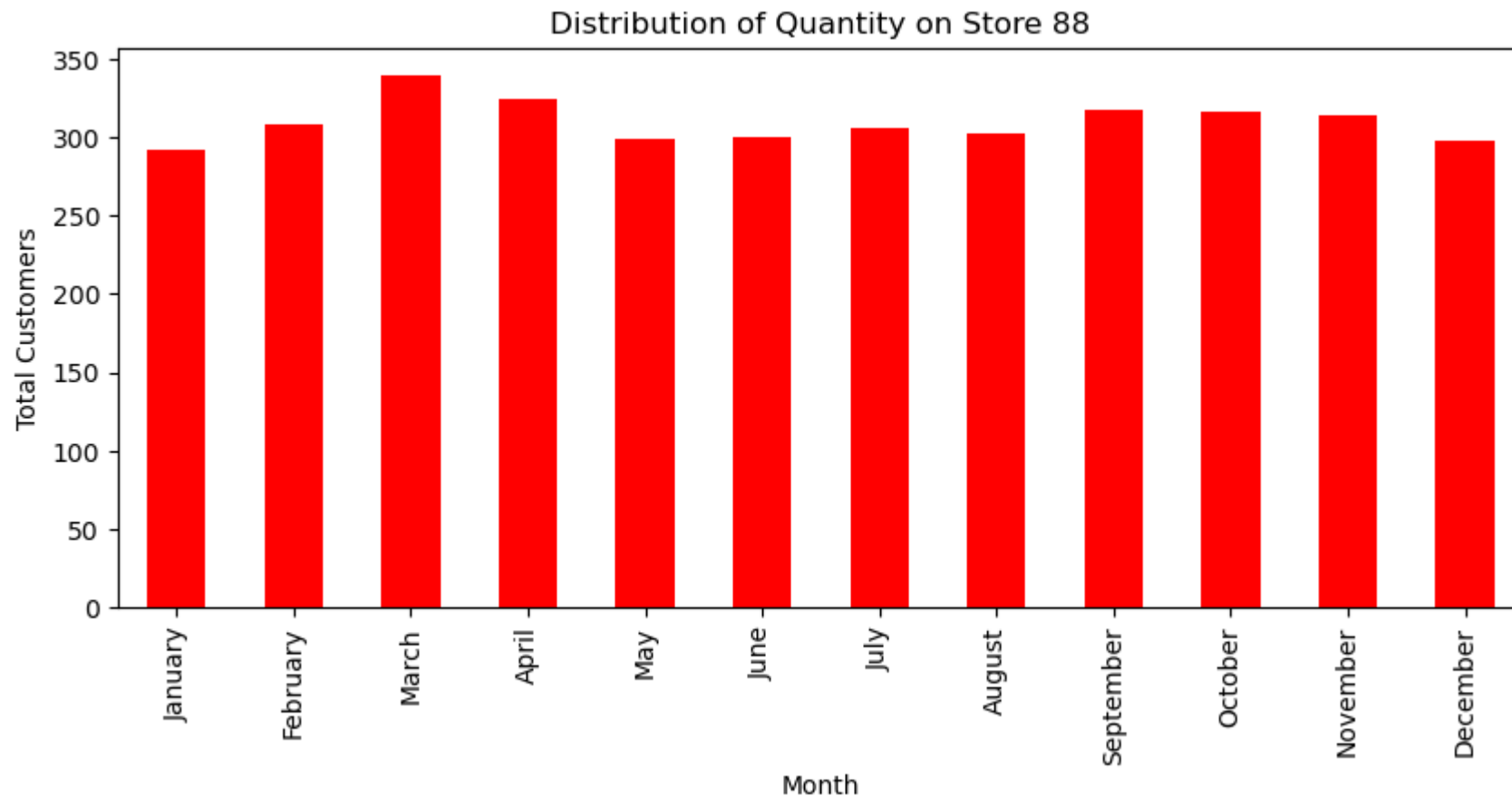
```
Out[33]: Index(['LYLTY_CARD_NBR', 'DATE', 'STORE_NBR', 'TXN_ID', 'PROD_NBR',
               'PROD_NAME', 'PROD_QTY', 'TOT_SALES', 'PACK_SIZE', 'BRAND', 'LIFESTAGE',
               'PREMIUM_CUSTOMER', 'Month'],
              dtype='object')
```

```
In [43]: 1
2 for i in df['STORE_NBR'].unique():
3     plt.figure(figsize=(10,4))
4     single_store = df[df['STORE_NBR']==i]
5     total_customer = single_store.groupby('Month')['PROD_QTY'].sum()
6     total_customer.plot(kind='bar',label = f"Store {i}", color="r")
7     plt.title(f'Distribution of Quantity on Store {i}')
8
9 plt.xlabel('Month')
10
11 plt.ylabel('Total Customers')
12
13 plt.show()
```









```
1 1. Store 77:
2   - Highest Quantity Sold Months: May and July
3   - Lowest Quantity Sold Month: October
4
5   Insight:Store 77 experiences the highest quantity of products sold during May and July, while October records
6   the
7       lowest quantity sold.
8
9 2. Store 86:
10  - Highest Quantity Sold Months: March and October
11  - Lowest Quantity Sold Month: August
```

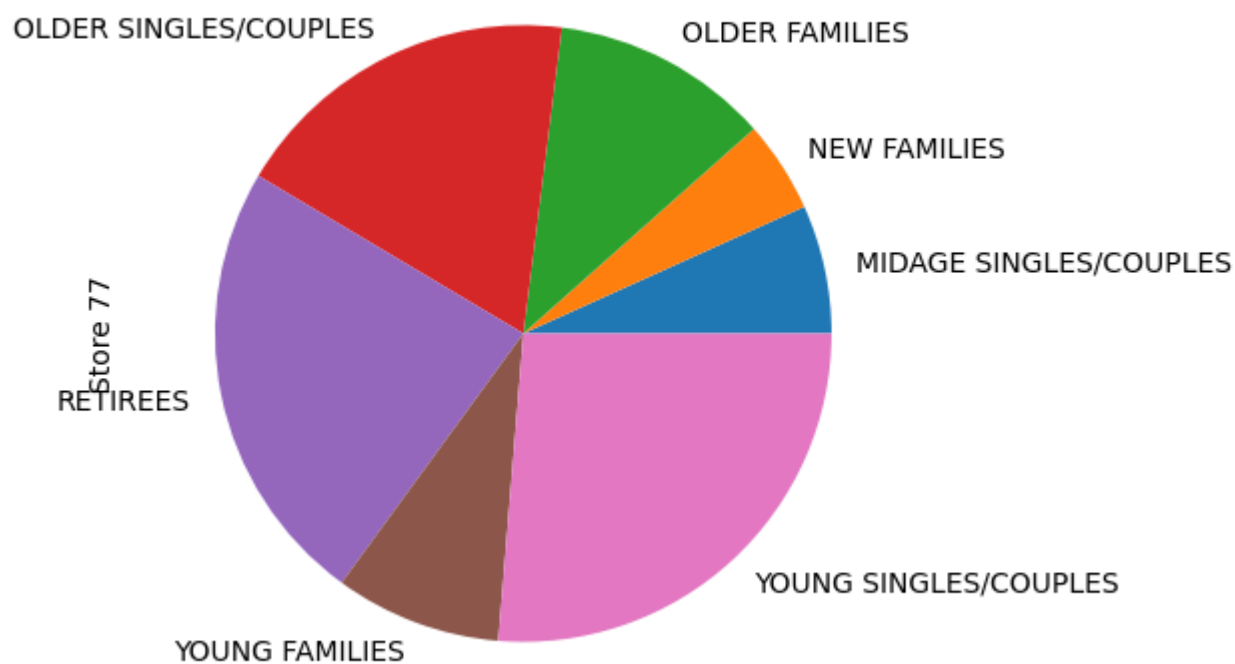
```
12     Insight:Store 86 achieves the highest quantity of products sold in March and October, with the lowest
13     quantity sold
14     in August.
15 3. Store 88:
16     - Highest Quantity Sold Months: March and April
17     - Lowest Quantity Sold Month: May
18
19     Insight: Store 88 sees the highest quantity of products sold in March and April, and the lowest quantity sold
    in May.
```

In [ ]:

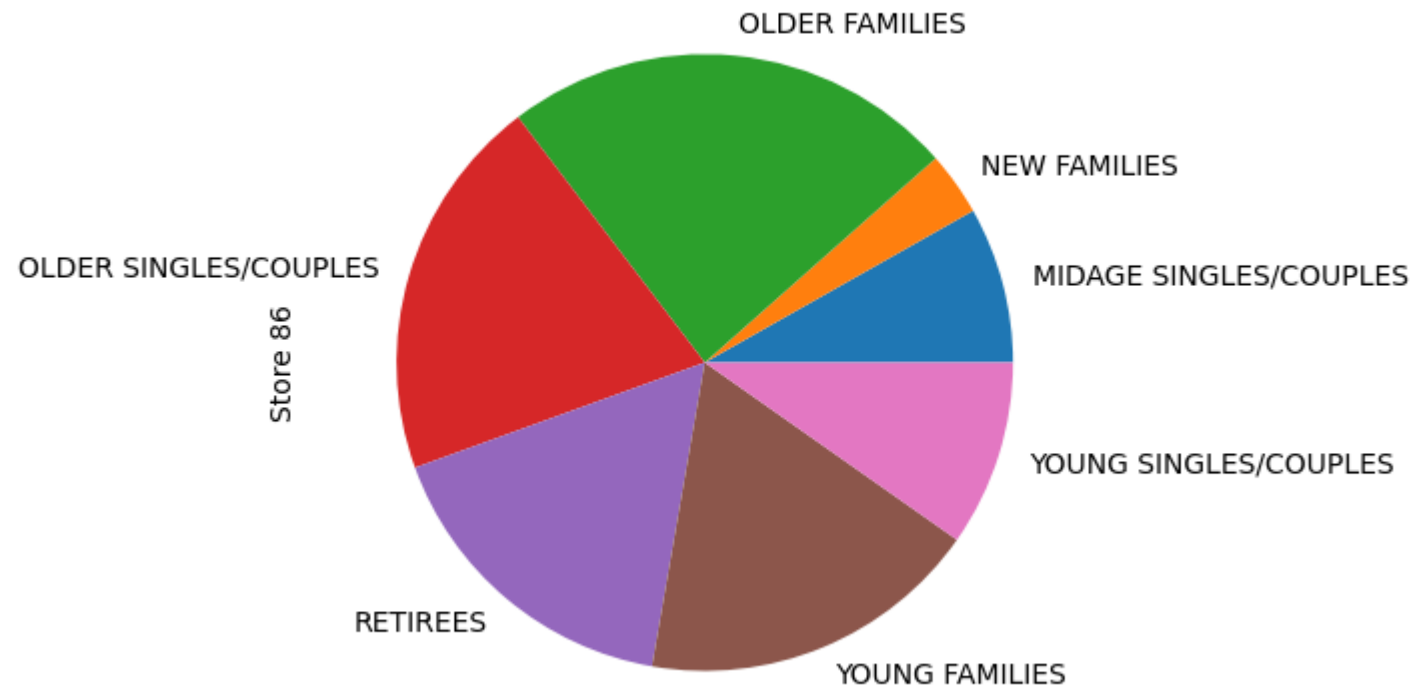
1

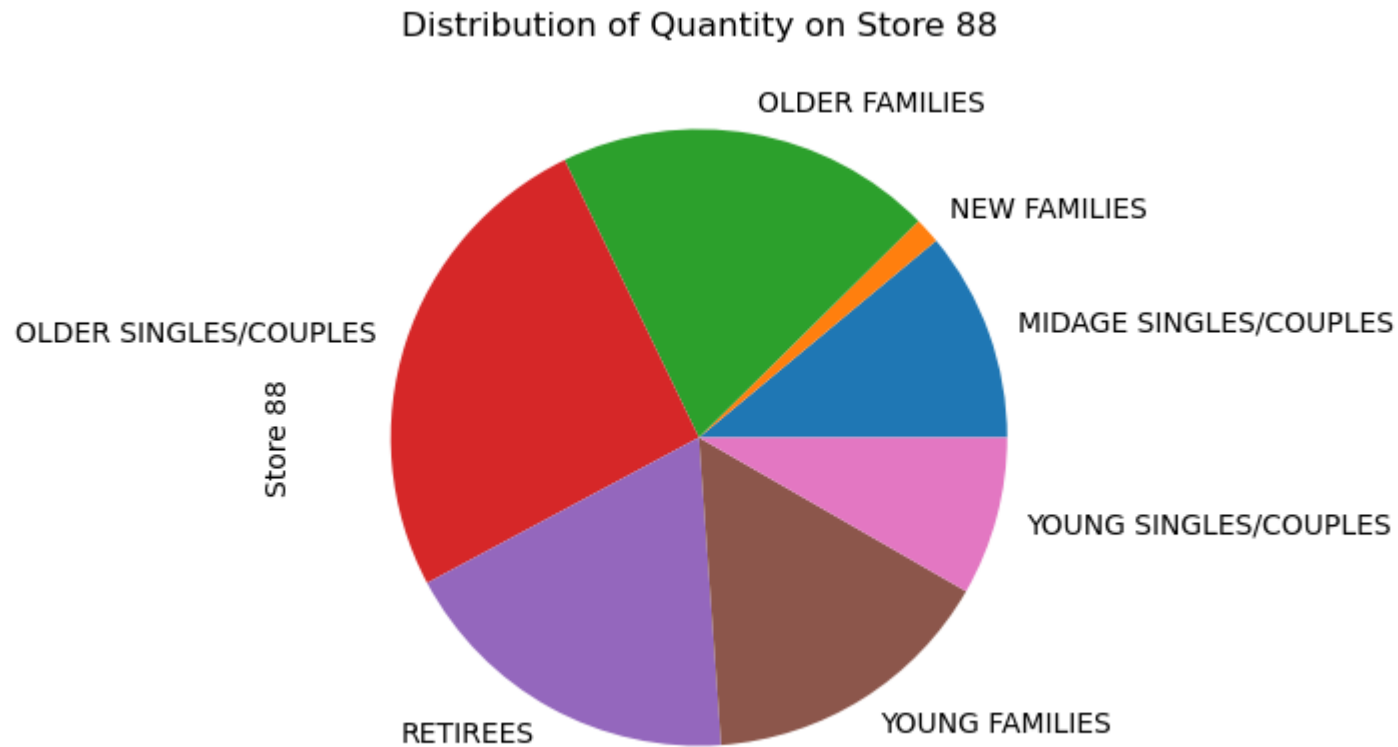
```
In [66]: 1 for i in df['STORE_NBR'].unique():
2         plt.figure(figsize=(10,5))
3         single_store = df[df['STORE_NBR']==i]
4         total = single_store.groupby('LIFESTAGE')['TOT_SALES'].sum()
5         total.plot(kind='pie',label = f"Store {i}")
6         plt.title(f'Distribution of Quantity on Store {i}')
7
8         plt.show()
```

Distribution of Quantity on Store 77



### Distribution of Quantity on Store 86



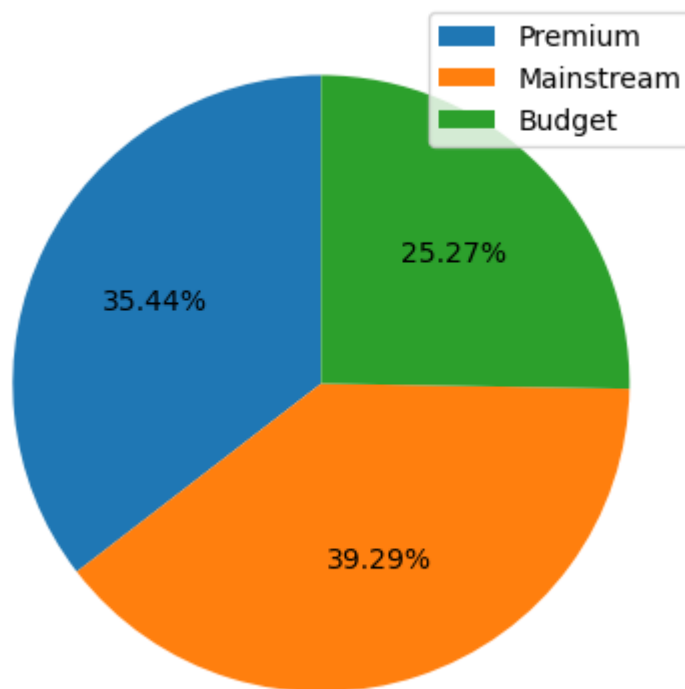


```
In [97]: 1 df['LIFESTAGE'].unique()
```

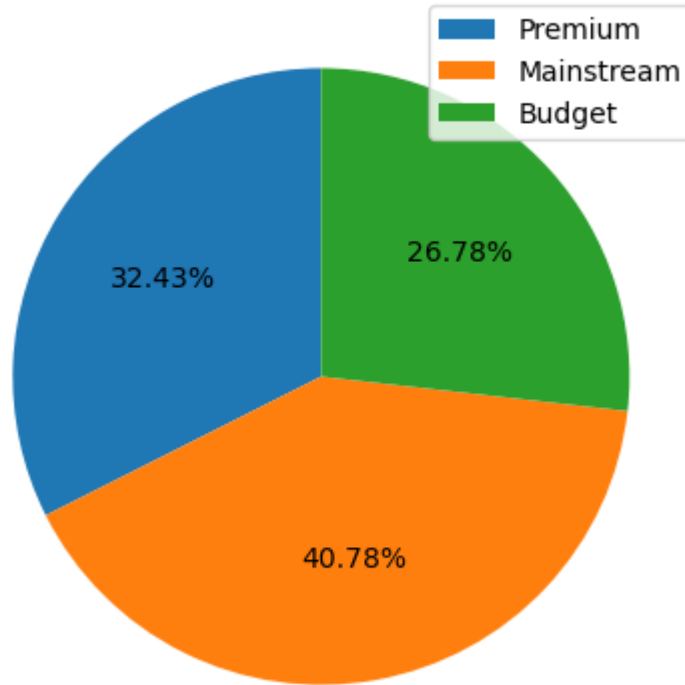
```
Out[97]: array(['MIDAGE SINGLES/COUPLES', 'YOUNG FAMILIES', 'RETIREEES',  
                'OLDER FAMILIES', 'YOUNG SINGLES/COUPLES', 'OLDER SINGLES/COUPLES',  
                'NEW FAMILIES'], dtype=object)
```

```
In [96]: 1 for i in df['STORE_NBR'].unique():
2         plt.figure(figsize=(10,5))
3         single_store = df[df['STORE_NBR']==i]
4         total = single_store.groupby('PREMIUM_CUSTOMER')['TOT_SALES'].sum()
5         plt.pie(total, autopct='%0.2f%%', startangle=90)
6         plt.legend(['Premium', 'Mainstream', 'Budget'])
7
8         plt.title(f'Distribution of Premium Customer on Store {i}')
9
10        plt.show()
```

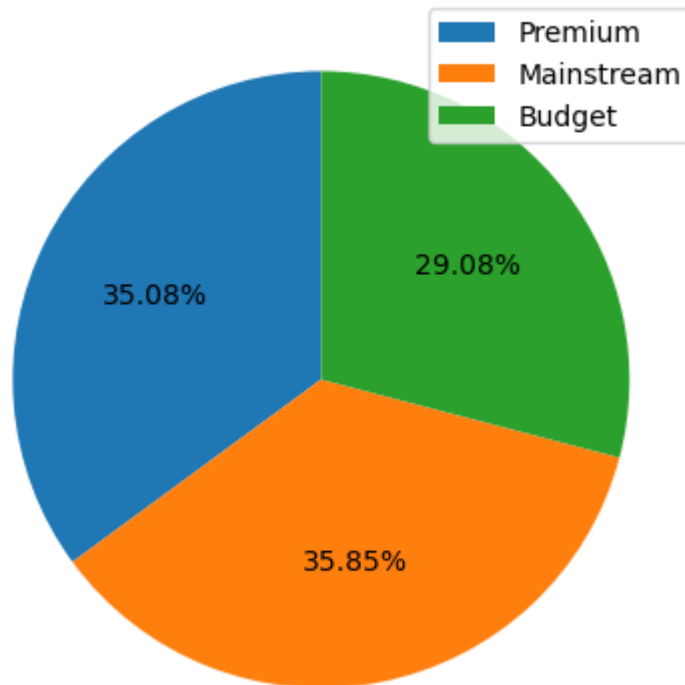
Distribution of Premium Customer on Store 77



Distribution of Premium Customer on Store 86



## Distribution of Premium Customer on Store 88



In [ ]:

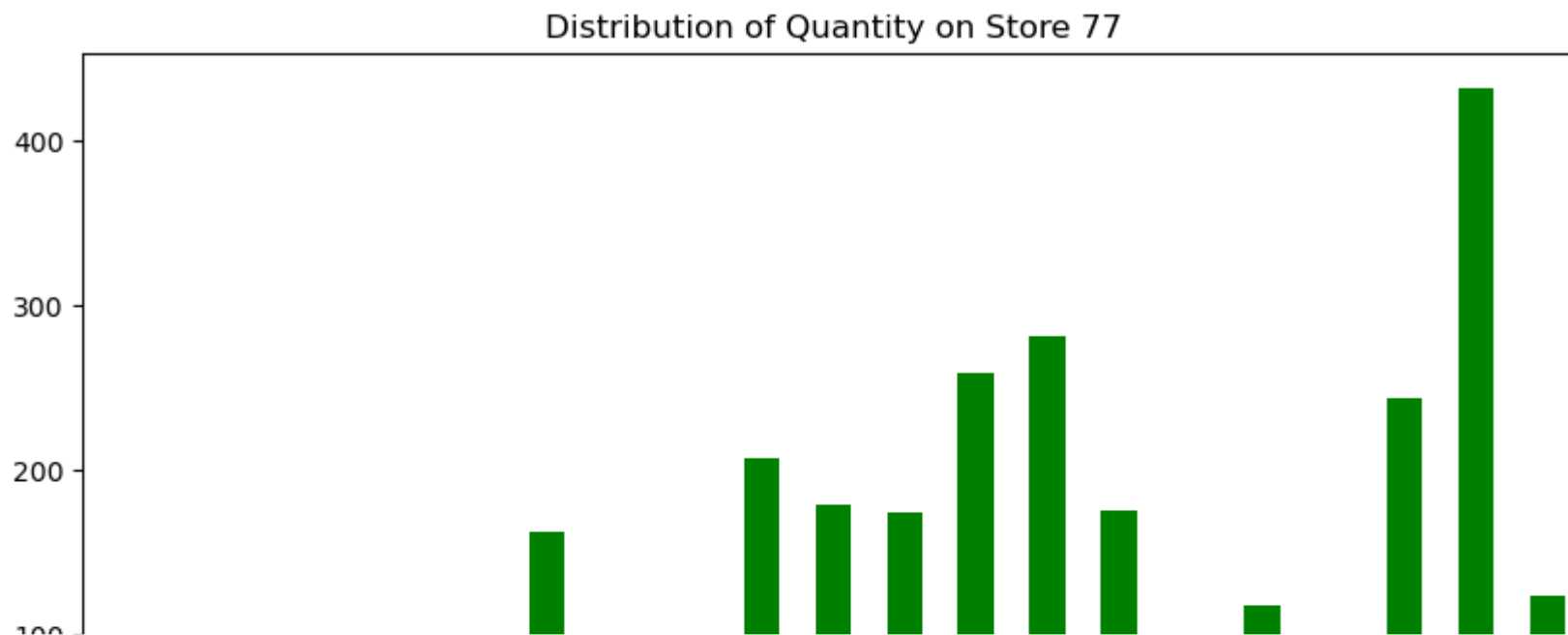
1

In [45]:

```
1 df['Segment'] = df['LIFESTAGE'] + "-" + df['PREMIUM_CUSTOMER']
```



```
In [98]: 1
2 for i in df['STORE_NBR'].unique():
3     plt.figure(figsize=(10,5))
4     single_store = df[df['STORE_NBR']==i]
5     total_customer = single_store.groupby('Segment')['TOT_SALES'].sum()
6     total_customer.plot(kind='bar',label = f"Store {i}", color="g")
7     plt.title(f'Distribution of Quantity on Store {i}')
8
9
10 plt.ylabel('Total Sales')
11
12 plt.show()
```

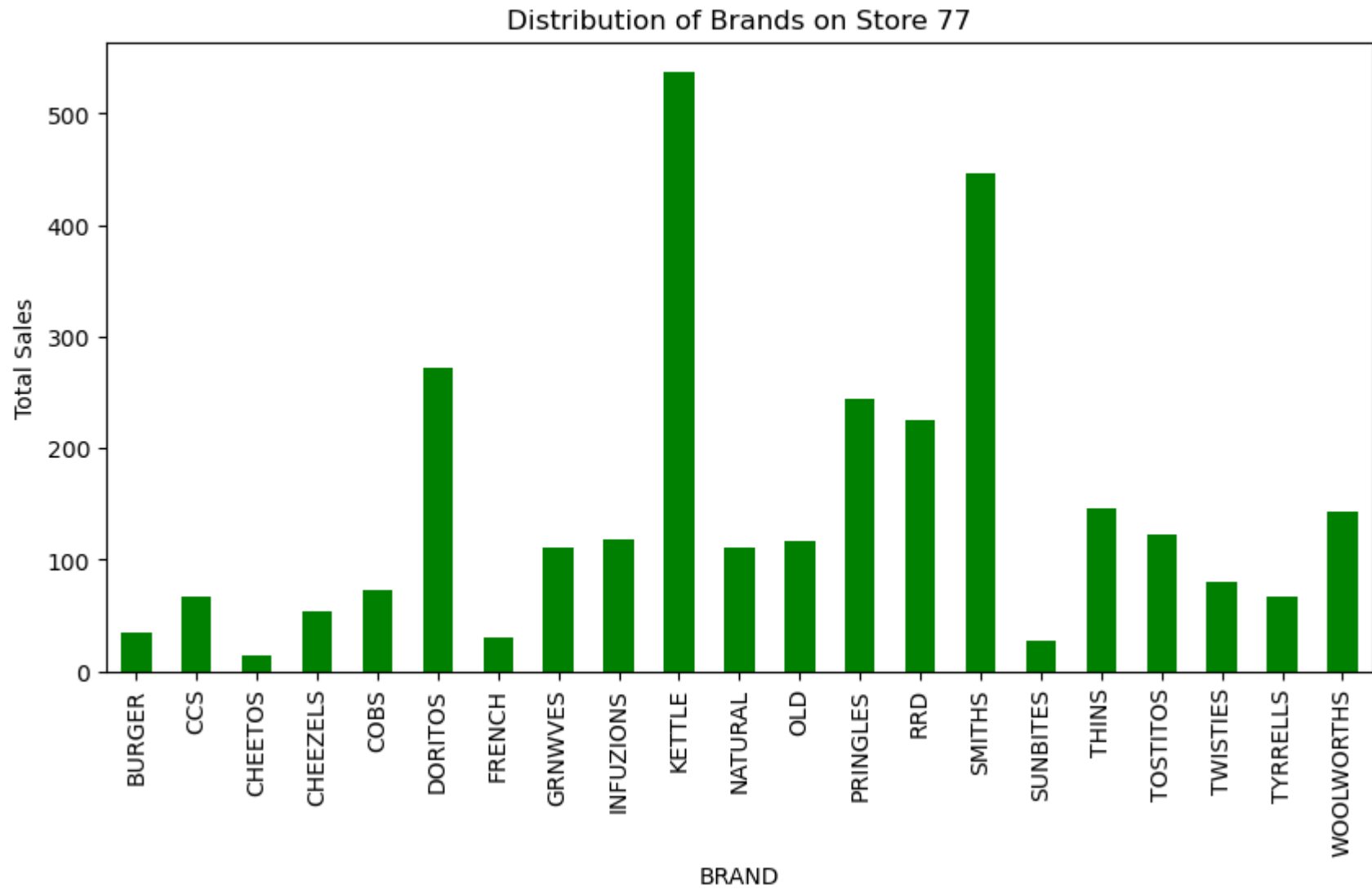


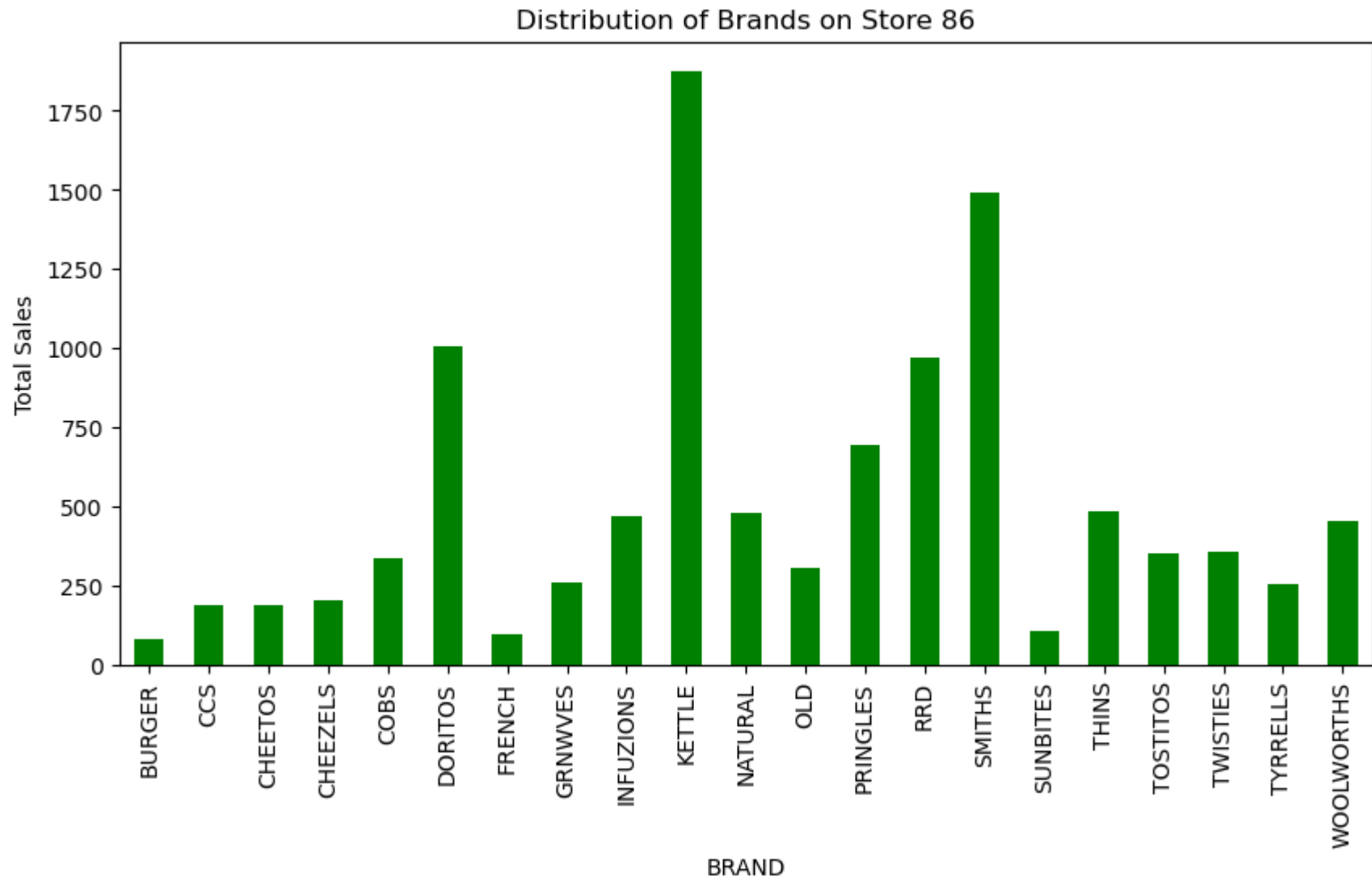
```
1 In store 77 Young Single/Couples- Mainstream genrates highest total sales .
2 In store 86 Older Families-Budget generates highest total sales followed by Retirees-Mainstream.
3 In store 88 Older Families-Budget generates highest total sales followed by Older single/couples-Mainstream.
```

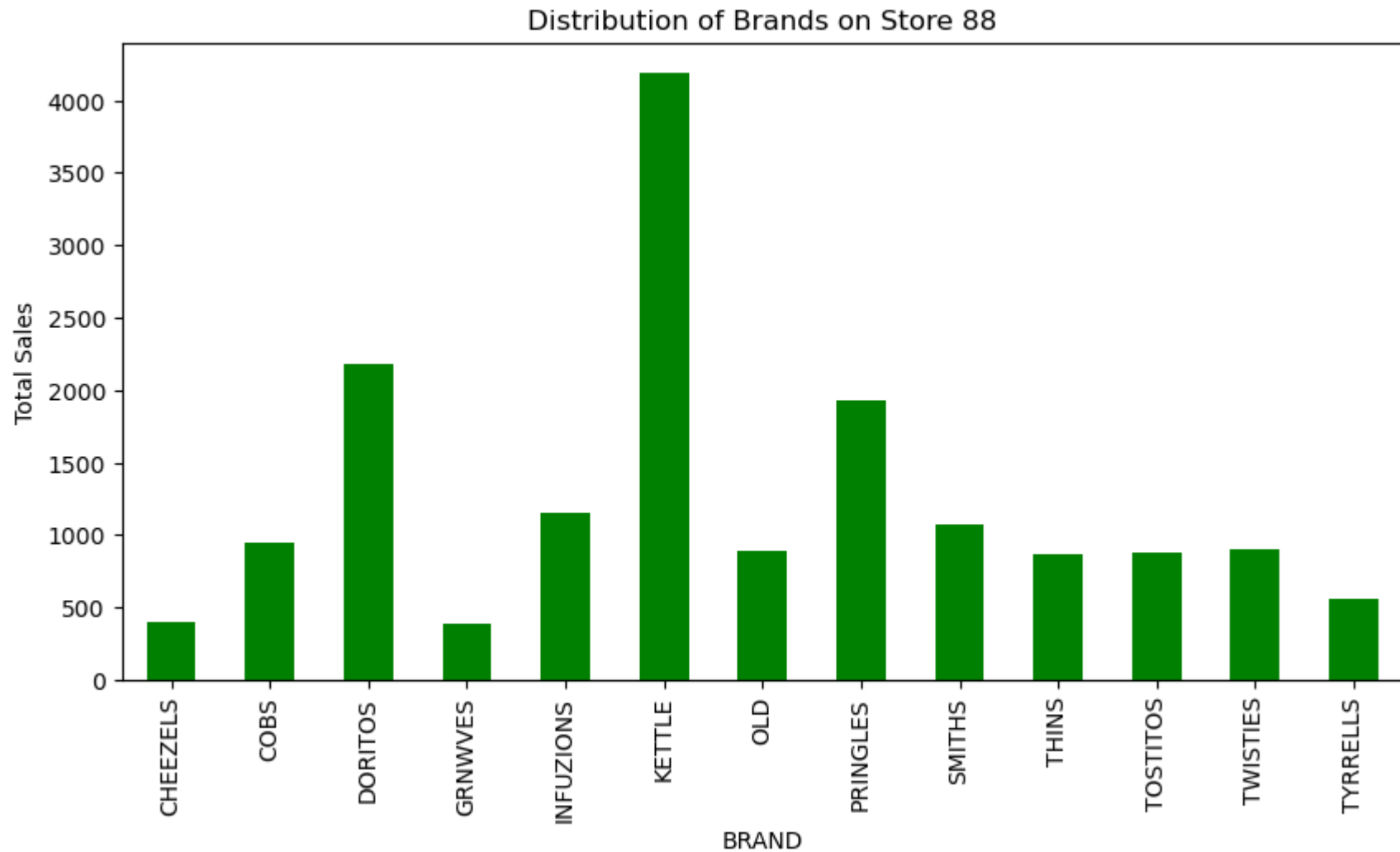
```
In [ ]:
```

```
1
```

```
In [104]: 1 for i in df['STORE_NBR'].unique():
2         plt.figure(figsize=(10,5))
3         single_store = df[df['STORE_NBR']==i]
4         brand = single_store.groupby('BRAND')['TOT_SALES'].sum()
5         brand.plot(kind='bar',label = f"Store {i}", color="g")
6         plt.title(f'Distribution of Brands on Store {i}')
7
8
9         plt.ylabel('Total Sales')
10
11 plt.show()
```





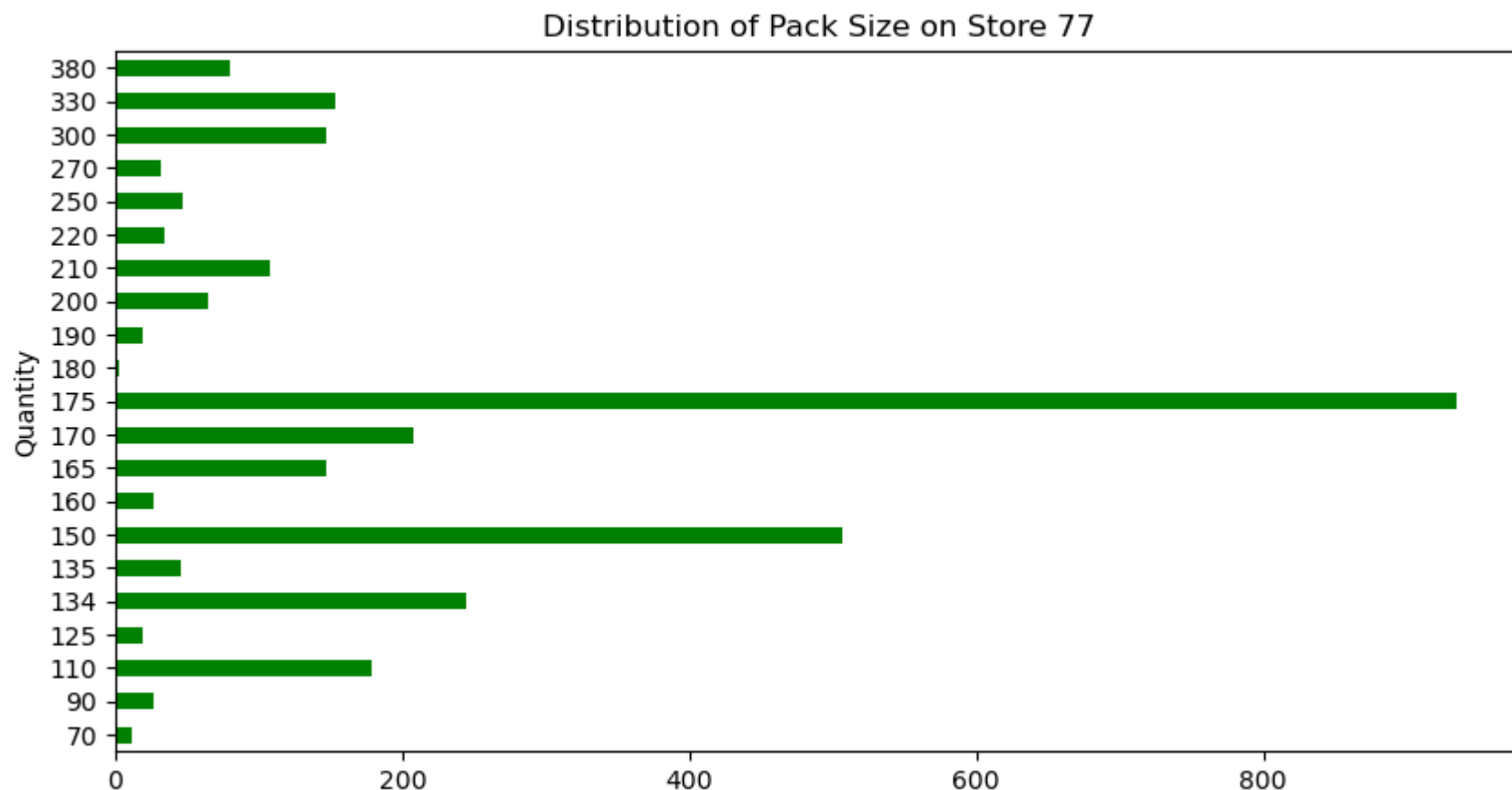


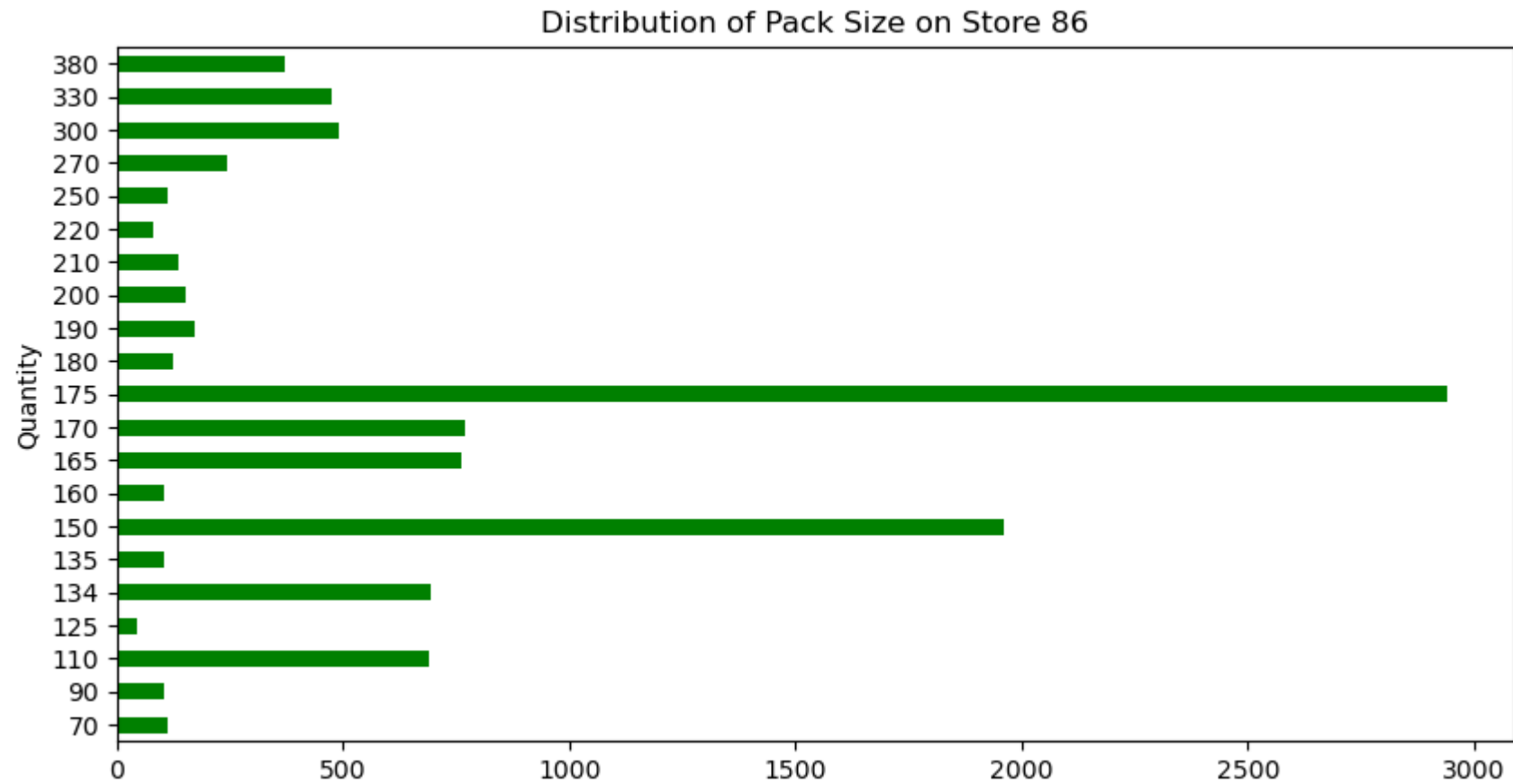
- 1 In store 77 and 86 Kettle and Smiths are two brand which generate highest total sales and GRNWVES and Sunbites
- 2 generates lowest sales respectively.
- 3 In store 88 Kettle and Pringles are two brand which generate highest total and GRNWVES
- 4 generates lowest sales .

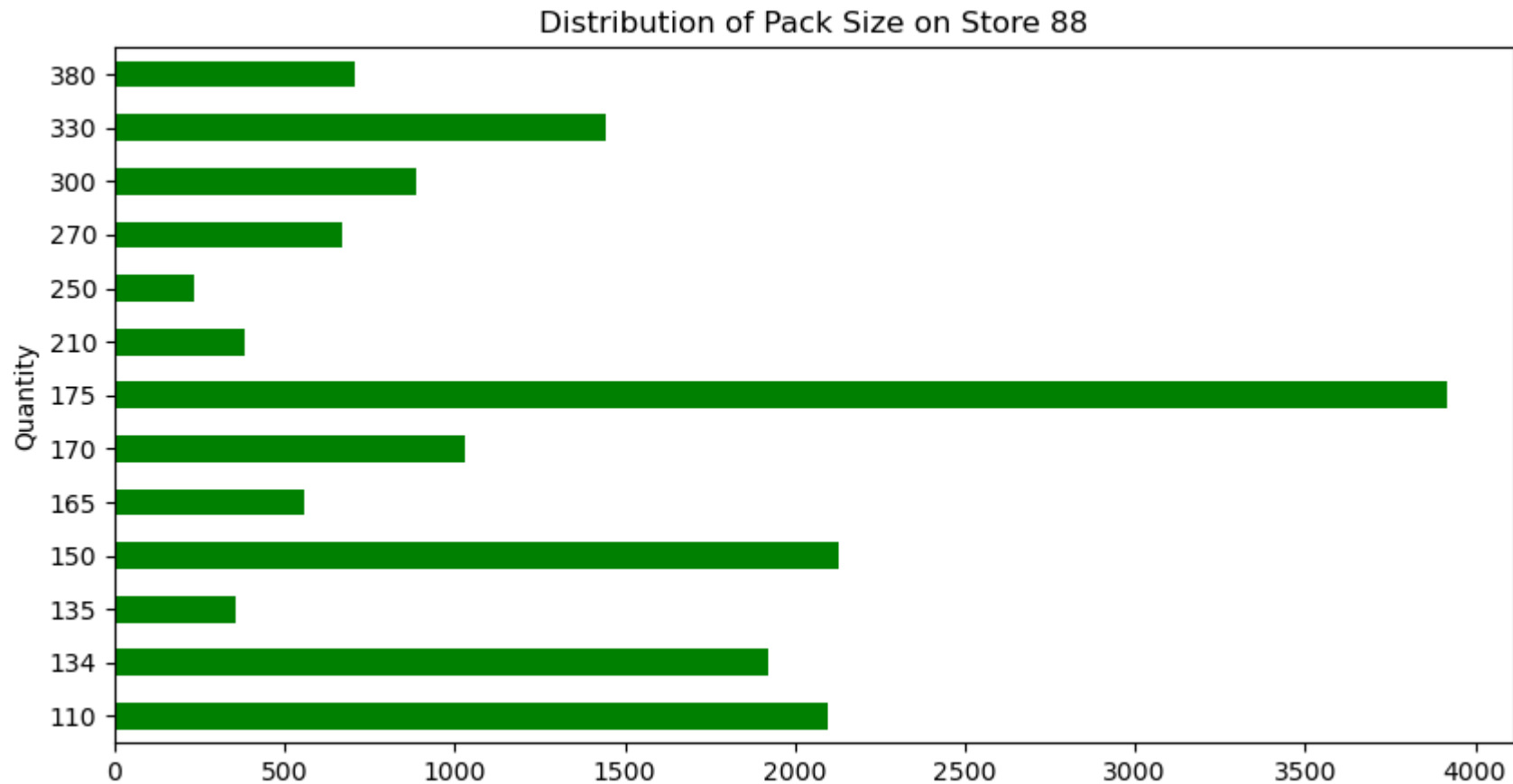
In [ ]:

1

```
In [106]: 1 for i in df['STORE_NBR'].unique():
2         plt.figure(figsize=(10,5))
3         single_store = df[df['STORE_NBR']==i]
4         brand = single_store.groupby('PACK_SIZE')['TOT_SALES'].sum()
5         brand.plot(kind='barh',label = f"Store {i}", color="g")
6         plt.title(f'Distribution of Pack Size on Store {i}')
7         plt.ylabel('Quantity')
8         plt.show()
```







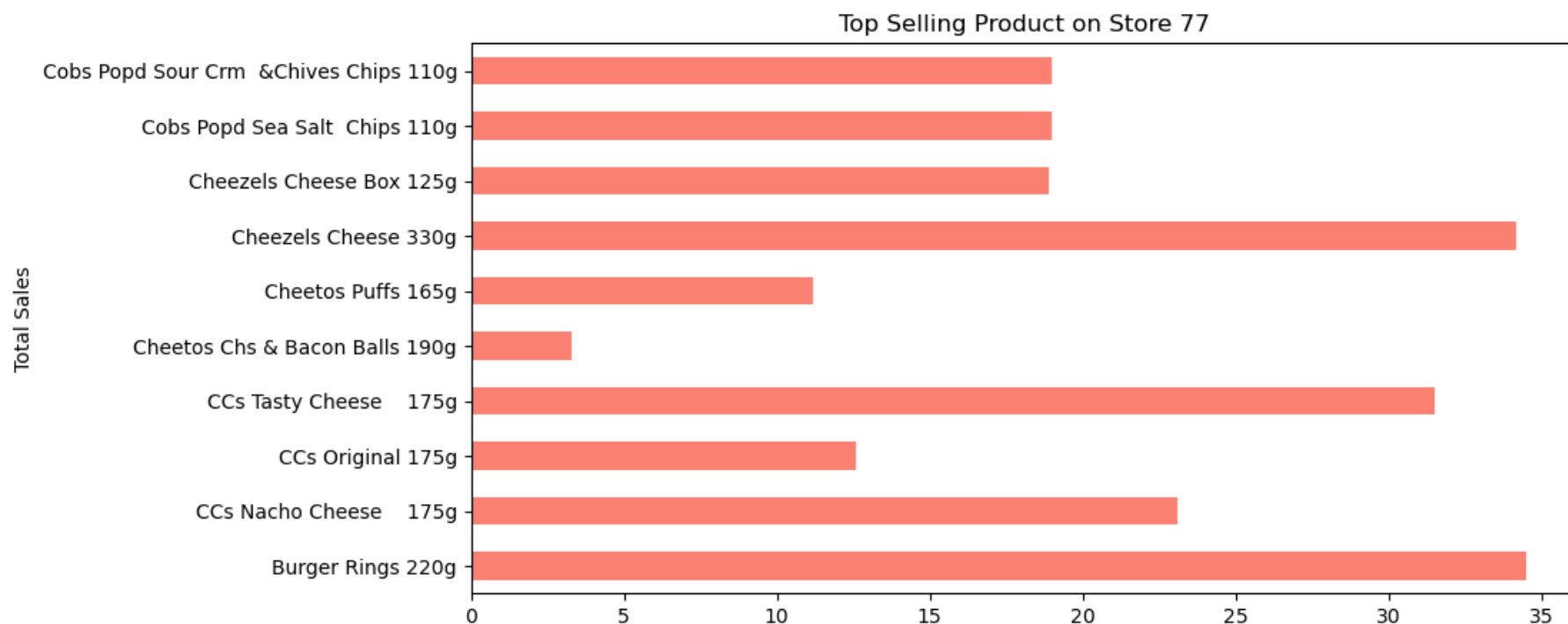
1 In all the Stores pack size 175g is being preferred more than others followed by 150g.

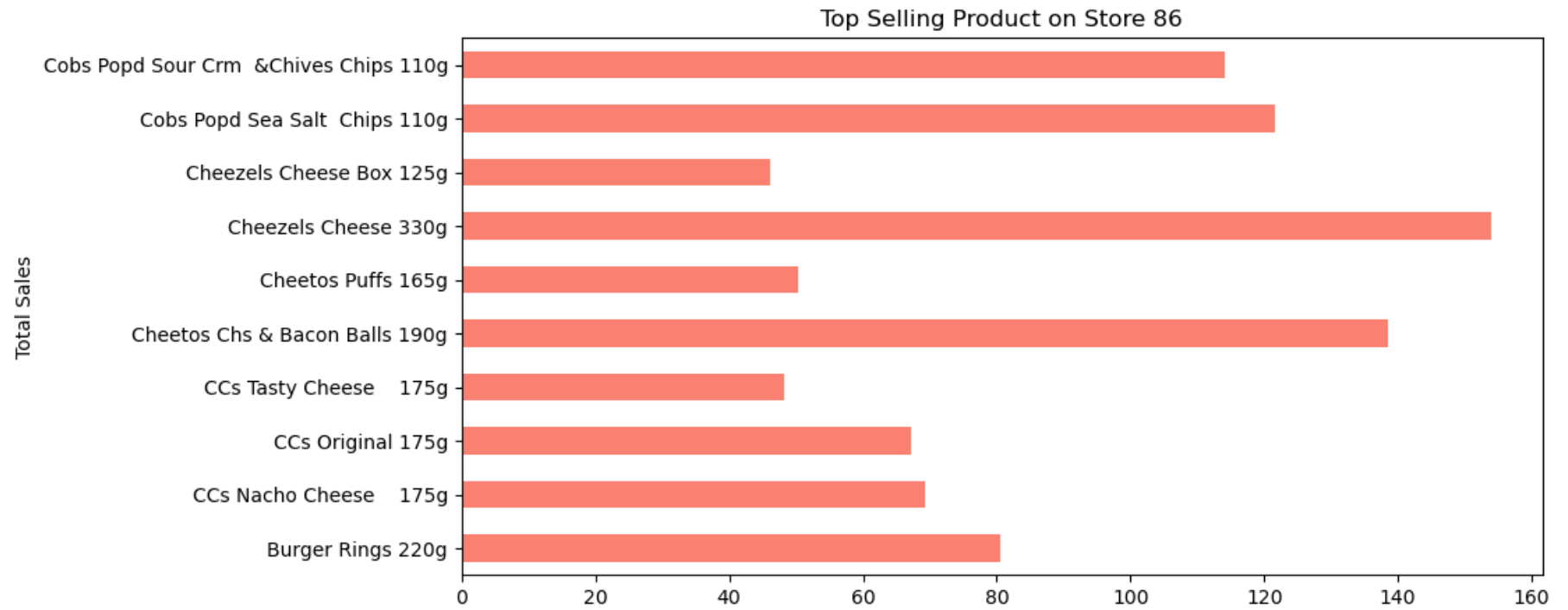
In [ ]:

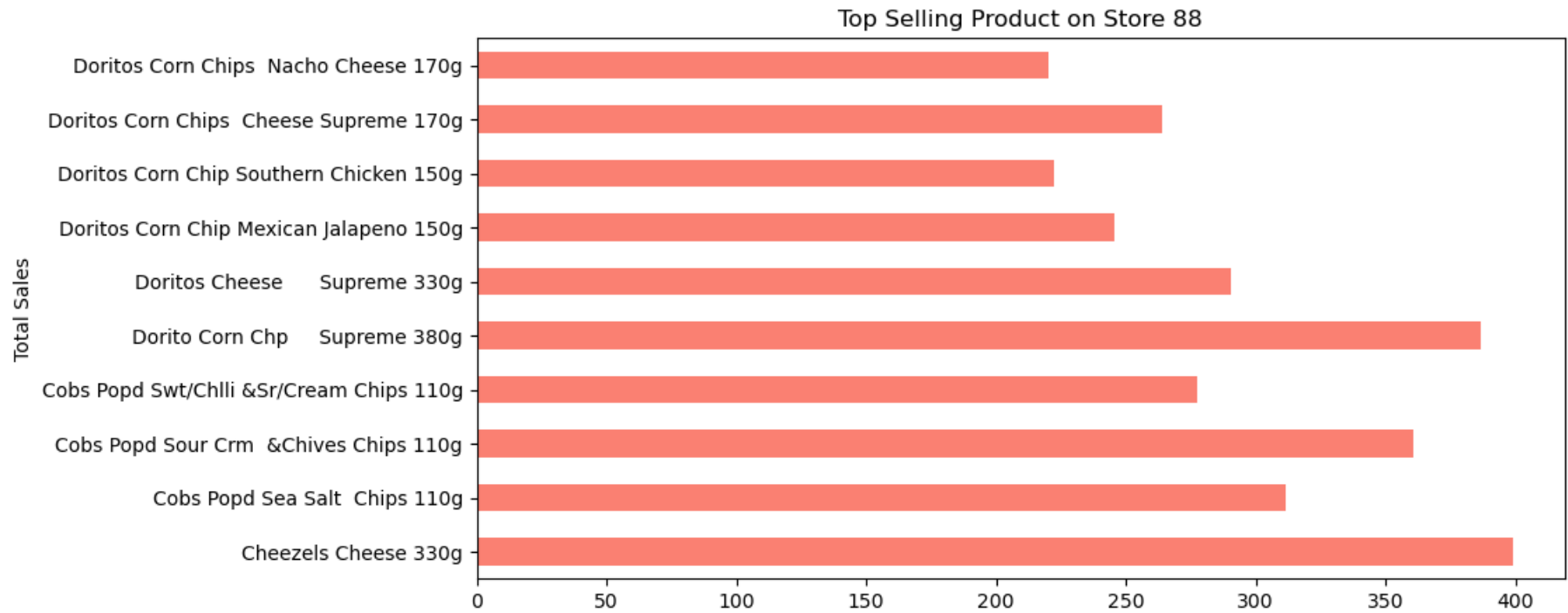
1



```
In [116]: 1 for i in df['STORE_NBR'].unique():
2         plt.figure(figsize=(10,5))
3         single_store = df[df['STORE_NBR']==i]
4         brand = single_store.groupby('PROD_NAME')['TOT_SALES'].sum().head(10)
5         brand.plot(kind='barh',label = f"Store {i}", color="salmon")
6         plt.title(f'Top Selling Product on Store {i}')
7         plt.ylabel('Total Sales')
8         plt.show()
```







- 1 In store 77 Chezzel Cheese 330g and Burger Rings 220g generate highest total sales and Cheetos cheese and Bacon Balls are
- 2 lowest selling product.
- 3 In store 86 Chezzel Cheese 330g and Cheetos cheese & Bacon Balls generate highest total sales and Cheetos Cheese Box are
- 4 lowest selling product.
- 5 In store 88 Chezzel Cheese 330g and Dorito Corn Chips Supreme generate highest total sales and Dorito Corn Chips Southern
- 6 Chicken are lowest selling product.

In [99]: 1 df.columns

Out[99]: Index(['LYLTY\_CARD\_NBR', 'DATE', 'STORE\_NBR', 'TXN\_ID', 'PROD\_NBR',  
 'PROD\_NAME', 'PROD\_QTY', 'TOT\_SALES', 'PACK\_SIZE', 'BRAND', 'LIFESTAGE',  
 'PREMIUM\_CUSTOMER', 'Month', 'Segment'],  
 dtype='object')

In [ ]:

1