In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
df= pd.read_csv(r'C:\Users\Ceejay\Downloads\QVI_data.csv')
df.head(12)
```

Out[1]:

	LYLTY_CARD_NBR	DATE	STORE_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY
0	1000	2018-10-17	1	1	5	Natural Chip Compny SeaSalt175g	2
1	1002	2018-09-16	1	2	58	Red Rock Deli Chikn&Garlic Aioli 150g	1
2	1003	2019-03-07	1	3	52	Grain Waves Sour Cream&Chives 210G	1
3	1003	2019-03-08	1	4	106	Natural ChipCo Hony Soy Chckn175g	1
4	1004	2018-11-02	1	5	96	WW Original Stacked Chips 160g	1
5	1005	2018-12-28	1	6	86	Cheetos Puffs 165g	1
6	1007	2018-12-04	1	7	49	Infuzions SourCream&Herbs Veg Strws 110g	1
7	1007	2018-12-05	1	8	10	RRD SR Slow Rst Pork Belly 150g	1
8	1009	2018-11-20	1	9	20	Doritos Cheese Supreme 330g	1
9	1010	2018-09-09	1	10	51	Doritos Mexicana 170g	2
10	1010	2018-12-14	1	11	59	Old El Paso Salsa Dip Tomato Med 300g	1
11	1011	2018-07-29	1	12	84	GrnWves Plus Btroot & Chilli Jam 180g	2

In [2]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 264834 entries, 0 to 264833
Data columns (total 12 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   LYLTY_CARD_NBR    264834 non-null  int64  
 1   DATE              264834 non-null  object 
 2   STORE_NBR         264834 non-null  int64  
 3   TXN_ID            264834 non-null  int64  
 4   PROD_NBR          264834 non-null  int64  
 5   PROD_NAME         264834 non-null  object 
 6   PROD_QTY          264834 non-null  int64  
 7   TOT_SALES         264834 non-null  float64
 8   PACK_SIZE         264834 non-null  int64  
 9   BRAND             264834 non-null  object 
 10  LIFESTAGE         264834 non-null  object 
 11  PREMIUM_CUSTOMER  264834 non-null  object 
dtypes: float64(1), int64(6), object(5)
memory usage: 24.2+ MB
```

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

A column containing the year and the month of the transaction will be created. This would then be used to group the data.

In [4]:
```
df['YEARMO']= df.DATE.dt.strftime('%y-%m')
```

In [5]:
```
df.head()
```

Out[5]:

	LYLTY_CARD_NBR	DATE	STORE_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT
0	1000	2018-10-17	1	1	5	Natural Chip Compny SeaSalt175g	2	
1	1002	2018-09-16	1	2	58	Red Rock Deli Chikn&Garlic Aioli 150g	1	
2	1003	2019-03-07	1	3	52	Grain Waves Sour Cream&Chives 210G	1	
3	1003	2019-03-08	1	4	106	Natural ChipCo Hony Soy Chckn175g	1	
4	1004	2018-11-02	1	5	96	WW Original Stacked Chips 160g	1	

We can find the total sales each month, for each of the stores.

In [6]:
```
sales= df.groupby(['STORE_NBR', 'YEARMO'])['TOT_SALES'].sum()
sales= sales.to_frame()
```

Likewise, we can also find the total number of unique customers that visited each month for each of the stores

In [7]:
```
customers= df.groupby(['STORE_NBR', 'YEARMO'])['LYLTY_CARD_NBR'].nunique()
customers= customers.to_frame()
```

We can also get the number of transactions that occured each month. This will be used to derive the average transaction per customer metric after concatenating the various 'mini datasets'. After that this column will be removed from the new dataset.

In [8]:
```
trans_per_cust= df.groupby(['STORE_NBR', 'YEARMO'])['TXN_ID'].nunique()
trans_per_cust= trans_per_cust.to_frame()
```

In [9]:
```
data= pd.concat([sales, customers, trans_per_cust], axis= 1)
data['TRANSACT_PER_CUST']= data['TXN_ID']/data['LYLTY_CARD_NBR']
data.drop('TXN_ID', axis= 1, inplace= True)
```

We will now rename the columns for easier understanding of the variables...

In [10]:
```
data.columns= ['tot_sales', 'tot_customers', 'txn_per_customer']
```

In [11]:
```
data
```

Out[11]:

		tot_sales	tot_customers	txn_per_customer
STORE_NBR	**YEARMO**			
1	**18-07**	206.9	49	1.061224
	18-08	176.1	42	1.023810
	18-09	278.8	59	1.050847
	18-10	188.1	44	1.022727
	18-11	192.6	46	1.021739
...	**...**	...	...	...
272	**19-02**	395.5	45	1.066667
	19-03	442.3	50	1.060000
	19-04	445.1	54	1.018519
	19-05	314.6	34	1.176471
	19-06	312.1	34	1.088235

3169 rows × 3 columns

Now, we remove stores where the monthly records were not recorded....

In [12]:
```
no_record= pd.pivot_table(df, index= 'STORE_NBR', columns= 'YEARMO', values= '
```

In [13]:
```
no_record
```

Out[13]:

YEARMO	18-07	18-08	18-09	18-10	18-11	18-12	19-01	19-02	19-03	19-04	19-05	19-06
STORE_NBR												
1	52.0	43.0	62.0	45.0	47.0	47.0	36.0	55.0	49.0	43.0	51.0	43.0
2	41.0	43.0	37.0	43.0	40.0	38.0	45.0	32.0	46.0	49.0	50.0	42.0
3	138.0	134.0	119.0	119.0	118.0	129.0	121.0	139.0	130.0	110.0	123.0	122.0
4	160.0	151.0	138.0	155.0	139.0	133.0	168.0	102.0	135.0	137.0	126.0	134.0
5	120.0	112.0	125.0	107.0	111.0	125.0	118.0	106.0	97.0	109.0	104.0	127.0
...	...	...	...	...	...	...	...	...	...	...	...	...
268	52.0	54.0	34.0	48.0	51.0	43.0	38.0	37.0	47.0	50.0	52.0	40.0
269	139.0	132.0	124.0	148.0	136.0	133.0	144.0	133.0	122.0	139.0	130.0	127.0
270	139.0	154.0	126.0	119.0	133.0	149.0	155.0	125.0	143.0	132.0	128.0	127.0
271	129.0	101.0	114.0	114.0	122.0	117.0	120.0	102.0	101.0	109.0	127.0	129.0
272	52.0	48.0	36.0	51.0	45.0	47.0	50.0	48.0	53.0	56.0	40.0	37.0

272 rows × 12 columns

In [14]:
```
no_record.isnull().sum()
```

Out[14]:
```
YEARMO
18-07    6
18-08    9
18-09    8
18-10    7
18-11    8
18-12    9
19-01    9
19-02    8
19-03    7
19-04    7
19-05    9
19-06    8
dtype: int64
```

In [15]:
```
arr= [ ]
for i in no_record.index:
    if no_record.loc[i].isnull().any():
        arr.append(i)
arr
```

Out[15]:
```
[11, 31, 44, 76, 85, 92, 117, 193, 206, 211, 218, 252]
```

In [16]:
```
data.drop(arr, inplace= True)
```

Two dataframes will be created to hold the metrics for the pre -trial and trial periods

In [20]:
```
trial= data[data.index.get_level_values('YEARMO').isin(['19-03','19-04'])]
pre_trial= data[~data.index.get_level_values('YEARMO').isin(['19-02', '19-03',
```

Now, we will create a function that will calculate the stores that is correlated to the trial stores. The stores will be correlated on the 'tot_sales' and 'tot_customers' columns.

In [21]:
```
def calculate_corr(store):
    corr= [ ]
    data1= pre_trial[['tot_sales', 'tot_customers']]
    for x in data1.index:
        table= data1.loc[store].corrwith(data1.loc[x[0]])
        corr.append(table)
    df1= pd.DataFrame(corr)
    df1.index= data1.index
    df1.drop_duplicates(inplace= True)
    df1.index= [s[0] for s in df1.index]
    df1.index.name= 'store_number'
    df1= df1.abs()
    df1['avg_score']= df1.mean(axis= 1) # to hold the mean correlation score
    return df1
```

Finding stores correlated to store 77

In [22]:
```
corr77= calculate_corr(77)
```

In [23]:
```
corr77
```

Out[23]:

store_number	tot_sales	tot_customers	avg_score
1	0.138045	0.292531	0.215288
2	0.092418	0.226371	0.159395
3	0.383045	0.722707	0.552876
4	0.529948	0.526731	0.528340
5	0.253424	0.172905	0.213164
...	...	...	...
268	0.375997	0.452769	0.414383
269	0.410675	0.406491	0.408583
270	0.110342	0.250728	0.180535
271	0.214199	0.023417	0.118808
272	0.247381	0.105710	0.176545

260 rows × 3 columns

In [24]:
```
corr_table= corr77.sort_values(by= 'avg_score', ascending= False).head(10)
corr_table
```

Out[24]:

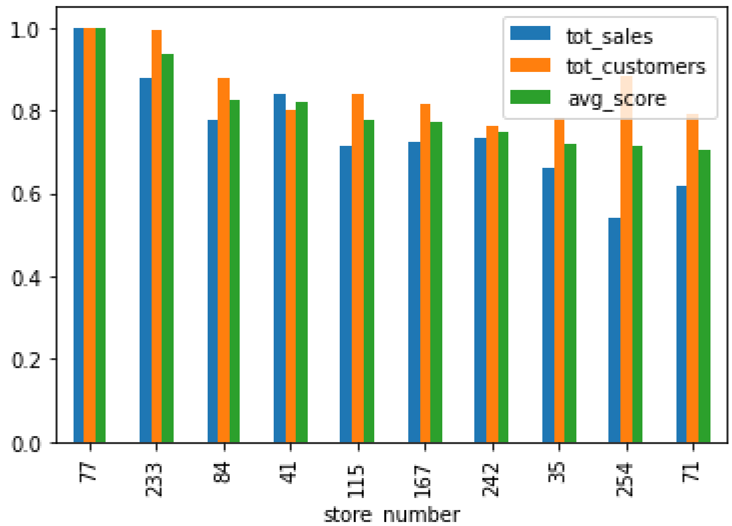
store_number	tot_sales	tot_customers	avg_score
77	1.000000	1.000000	1.000000
233	0.880004	0.994132	0.937068
84	0.777507	0.876067	0.826787
41	0.836940	0.799126	0.818033
115	0.715956	0.839229	0.777592
167	0.725318	0.814820	0.770069
242	0.735269	0.760095	0.747682
35	0.658044	0.778959	0.718502
254	0.542027	0.881578	0.711802
71	0.615592	0.790893	0.703242

From the table above, it is clearly shown that store 233 correlates most with the trial store 77

Visualising this....

In [25]:
```
corr_table.plot(kind= 'bar')
```

Out[25]: <AxesSubplot:xlabel='store_number'>



Checking to see if there is any significant difference between the stores before the trial period...

In [26]:
```
from scipy.stats import ttest_ind
def pre_significance_test(sig_df1, sig_df2):
    #Conducts a ttest for the pre trial period
    ''' Null hypothesis: There is no significant difference between the two sto
        Alternate hypothesis: There is a significant difference between the two
        Alpha= 0.05'''

    arr= [ ]
    for i in pre_trial.columns:
        tab= ttest_ind(sig_df1[i], sig_df2[i])
        arr.append(tab)
    temp_df= pd.DataFrame(arr)
    temp_df.index= pre_trial.columns
    return temp_df
```

In [27]:
```
pre_significance_test(pre_trial.loc[77], pre_trial.loc[233])
```

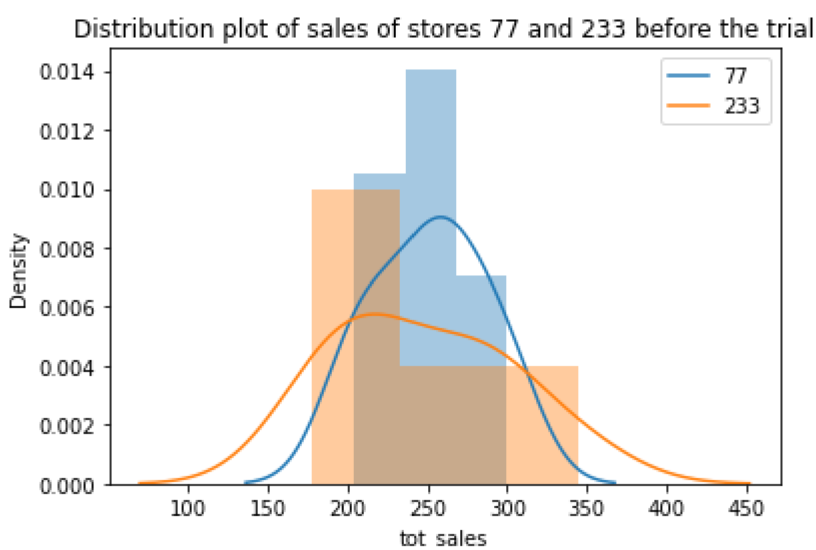
Out[27]:

	statistic	pvalue
tot_sales	0.191110	0.850844
tot_customers	-0.034001	0.973297
txn_per_customer	0.523638	0.607708

The pvalue is greater than 0.05 so therefore, there is no significant difference between stores 77 and 233 for the pre-trial period. This confirms that stores 77 and 233 are identical.

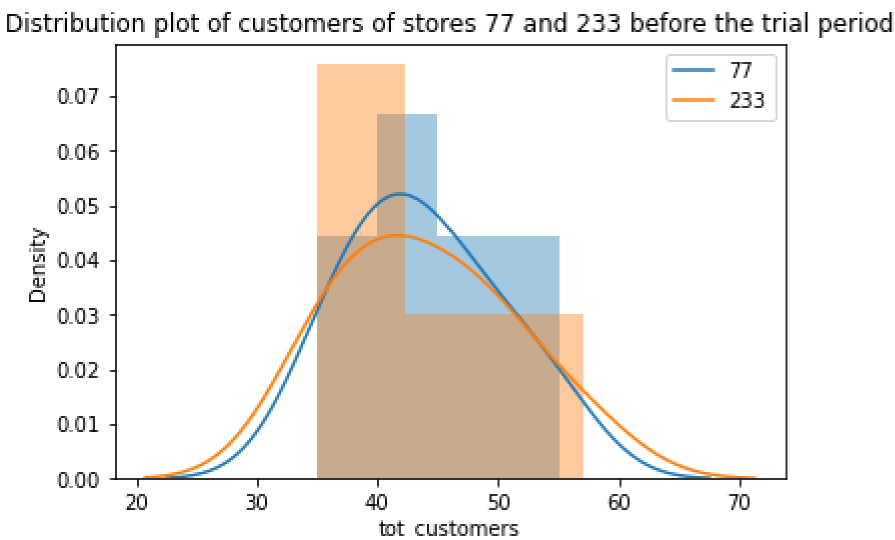
In [28]:
```
sns.distplot(pre_trial.loc[77]['tot_sales'])
sns.distplot(pre_trial.loc[233]['tot_sales'])
plt.legend(['77', '233'])
plt.title('Distribution plot of sales of stores 77 and 233 before the trial')
plt.show()
```

```
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



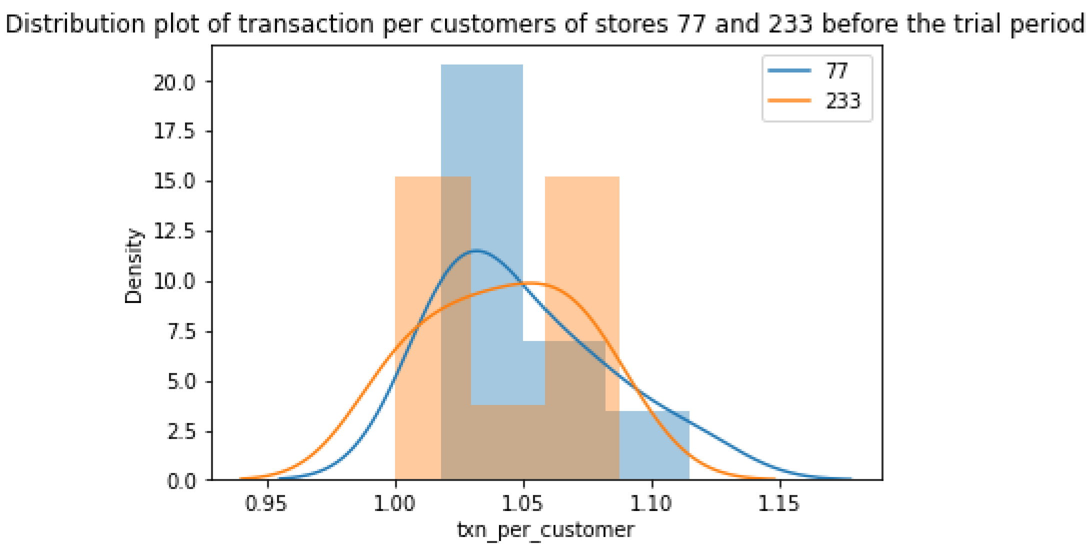
In [29]:
```
sns.distplot(pre_trial.loc[77]['tot_customers'])
sns.distplot(pre_trial.loc[233]['tot_customers'])
plt.legend(['77', '233'])
plt.title('Distribution plot of customers of stores 77 and 233 before the trial
plt.show()
```

```
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
```



In [30]:
```
sns.distplot(pre_trial.loc[77]['txn_per_customer'])
sns.distplot(pre_trial.loc[233]['txn_per_customer'])
plt.legend(['77', '233'])
plt.title('Distribution plot of transaction per customers of stores 77 and 233
plt.show()
```

```
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
```



Checking to see if there is any significant difference between both stores during the trial period..........

In [31]:
```
def trial_significance_test(sig_df3, sig_df4):
    #Conducts a ttest for the last two months of the trial period i.e (19-03 ar
    '''Null hypothesis: There is no significant difference between the two stor
    Alternate hypothesis: There is a significant difference between the two sto
    Alpha= 0.05 '''

    arr2= [ ]
    for i in trial.columns:
        tab= ttest_ind(sig_df3[i], sig_df4[i], alternative= 'greater')
        arr2.append(tab)
    temp_df1= pd.DataFrame(arr2)
    temp_df1.index= trial.columns
    return temp_df1
```

In [33]:
```
trial_significance_test(post_trial.loc[77], post_trial.loc[233])
```

Out[33]:

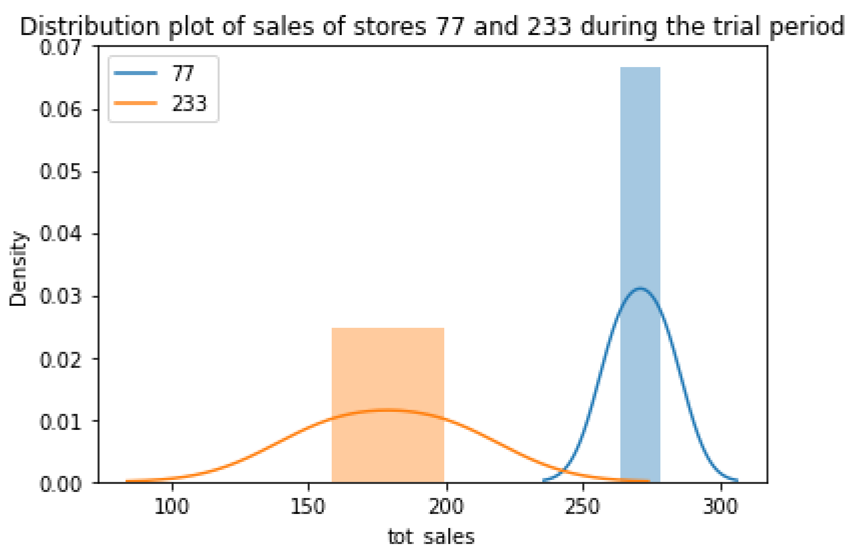
	statistic	pvalue
tot_sales	4.267336	0.025384
tot_customers	2.586131	0.061309
txn_per_customer	0.332434	0.385585

The pvalue for the total sales is less than 0.05 so, the alternate hypothesis that the total sales for the trial store is greater than that of the control store is acceped. For the other metrics; total customers and transaction per customer, the null hypothesis is accepted that the mean values for the total customers and transaction per customer is identical.........

Visualising the difference.....

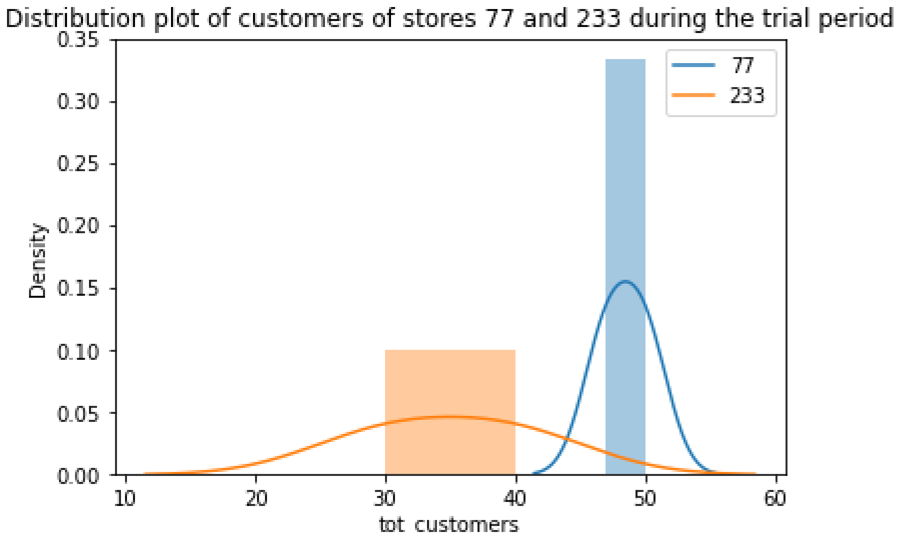
In [34]:
```
sns.distplot(trial.loc[77]['tot_sales'])
sns.distplot(trial.loc[233]['tot_sales'])
plt.legend(['77', '233'])
plt.title('Distribution plot of sales of stores 77 and 233 during the trial per
plt.show()
```

```
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
```



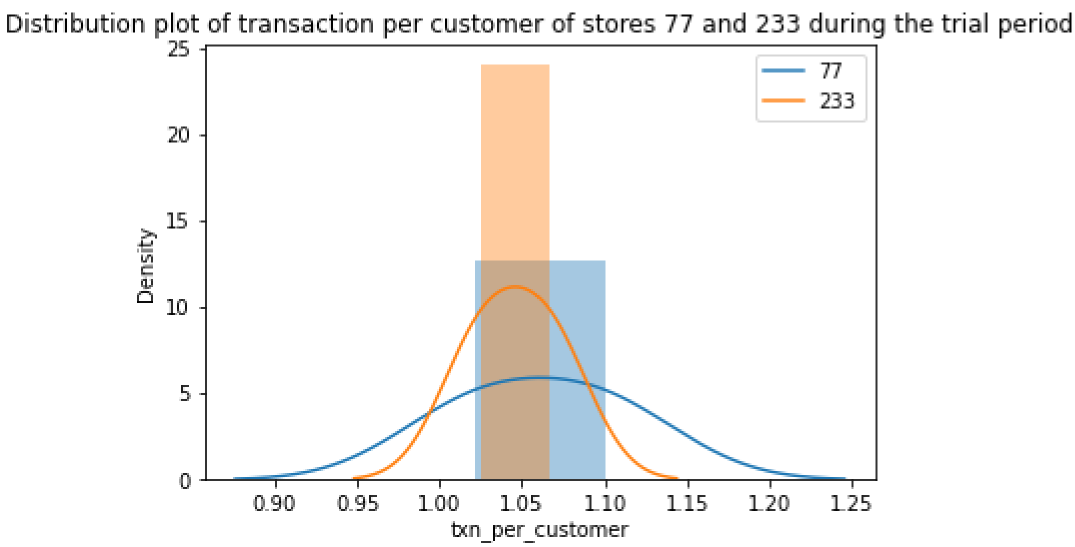
In [35]:
```
sns.distplot(trial.loc[77]['tot_customers'])
sns.distplot(trial.loc[233]['tot_customers'])
plt.legend(['77', '233'])
plt.title('Distribution plot of customers of stores 77 and 233 during the trial
plt.show()
```

```
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
```



In [36]:
```
sns.distplot(trial.loc[77]['txn_per_customer'])
sns.distplot(trial.loc[233]['txn_per_customer'])
plt.legend(['77', '233'])
plt.title('Distribution plot of transaction per customer of stores 77 and 233 d
plt.show()
```

```
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
```



The visualization confirms the hypothesis test conducted. There is a significant difference between the total sales for stores 77 and 233. The difference of the total customers between the two stores are somewhat significant and there is no significant difference between the transaction per customers between both stores

Store 77 only perform well in terms of total sales but not in terms of total customers and transaction per customers.

Finding stores correlated to store 86

In [37]:
```
corr86= calculate_corr(86)
```

In [38]:
```
corr86
```

Out[38]:

	tot_sales	tot_customers	avg_score
store_number			
1	0.483857	0.486520	0.485188
2	0.237462	0.188549	0.213006
3	0.232796	0.273946	0.253371
4	0.048187	0.294060	0.171123
5	0.001803	0.387089	0.194446
...	...	...	...
268	0.417161	0.082707	0.249934
269	0.672849	0.098401	0.385625
270	0.668908	0.774747	0.721827
271	0.330326	0.099780	0.215053
272	0.061332	0.337759	0.199545

260 rows × 3 columns

In [39]:
```
corr_table1= corr86.sort_values(by= 'avg_score', ascending= False).head(10)
corr_table1
```

Out[39]:

	tot_sales	tot_customers	avg_score
store_number			
86	1.000000	1.000000	1.000000
155	0.885236	0.930845	0.908040
23	0.707142	0.885187	0.796165
260	0.668425	0.833410	0.750918
27	0.753940	0.724051	0.738995
214	0.799645	0.668273	0.733959
270	0.668908	0.774747	0.721827
240	0.814900	0.618404	0.716652
120	0.821346	0.608472	0.714909
6	0.734499	0.597523	0.666011

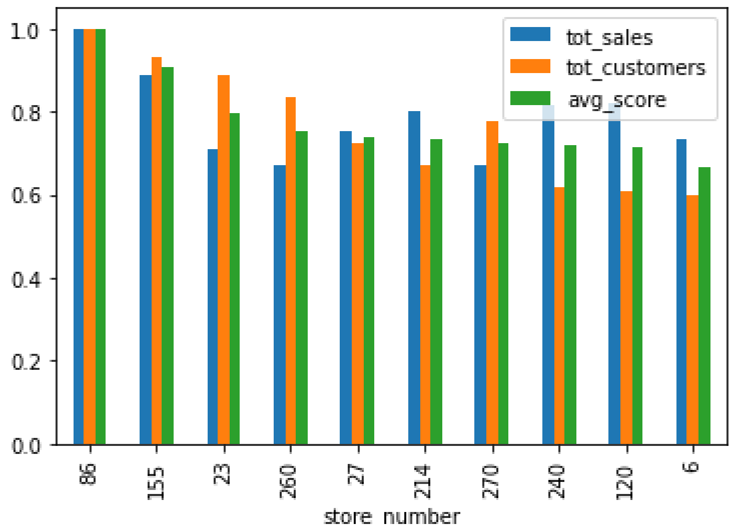
From the table above, it is clearly shown that store 155

correlates most with the trial store 86

Visualising this....

In [40]:
```
corr_table1.plot(kind= 'bar')
```

Out[40]: <AxesSubplot:xlabel='store_number'>



Checking to see if there is any significant difference between the stores before the trial period...

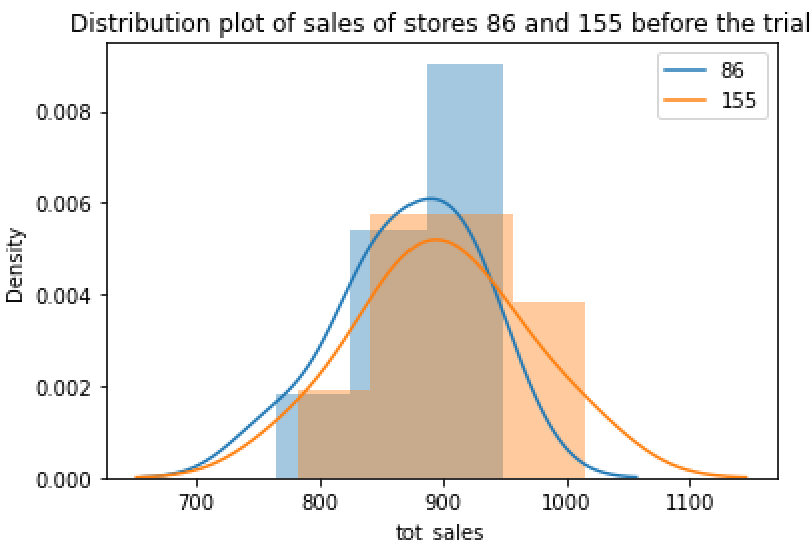
In [41]:
```
pre_significance_test(pre_trial.loc[86], pre_trial.loc[155])
```

Out[41]:

	statistic	pvalue
tot_sales	-0.913340	0.374622
tot_customers	0.045549	0.964234
txn_per_customer	-1.569054	0.136198

In [42]:
```
sns.distplot(pre_trial.loc[86]['tot_sales'])
sns.distplot(pre_trial.loc[155]['tot_sales'])
plt.legend(['86', '155'])
plt.title('Distribution plot of sales of stores 86 and 155 before the trial')
plt.show()
```

```
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
```

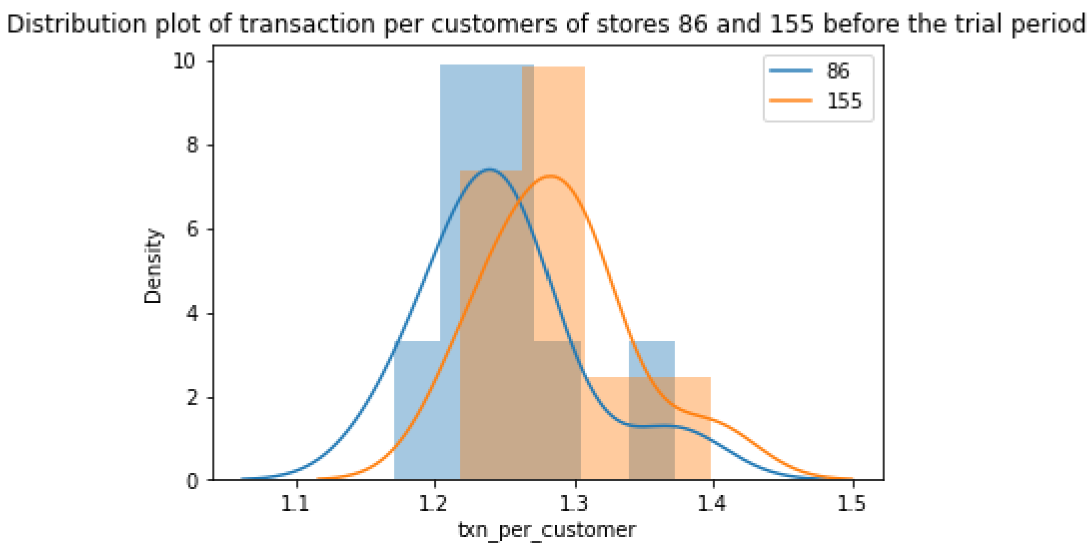


In [43]:
```
sns.distplot(pre_trial.loc[86]['tot_customers'])
sns.distplot(pre_trial.loc[155]['tot_customers'])
plt.legend(['86', '155'])
plt.title('Distribution plot of customers of stores 86 and 155 before the trial
plt.show()
```

```
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
```

In [44]:
```
sns.distplot(pre_trial.loc[86]['txn_per_customer'])
sns.distplot(pre_trial.loc[155]['txn_per_customer'])
plt.legend(['86', '155'])
plt.title('Distribution plot of transaction per customers of stores 86 and 155
plt.show()
```

```
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
```



Lets check if there is any significant difference between both stores for the period during the trial....

In [45]:
```
trial_significance_test(trial.loc[86], trial.loc[155])
```

Out[45]:

	statistic	pvalue
tot_sales	1.234512	0.171189
tot_customers	2.414953	0.068538
txn_per_customer	-1.074767	0.802535

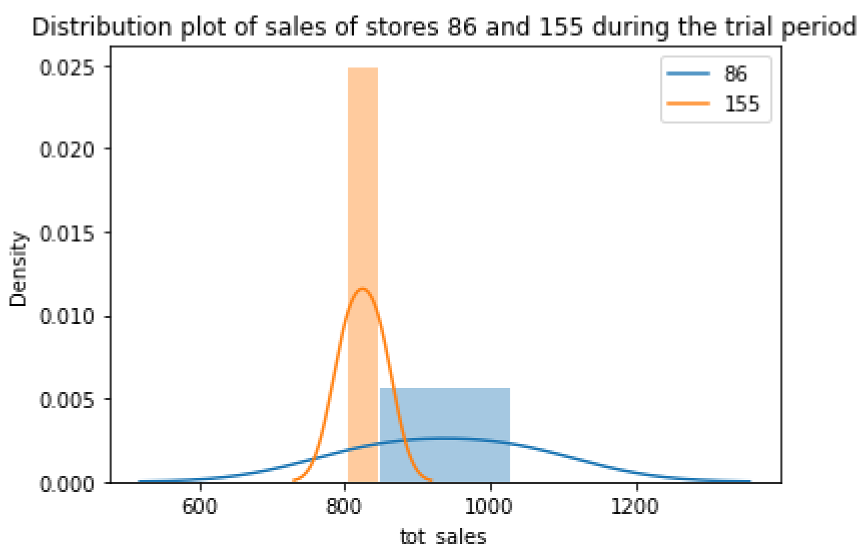
The pvalue for the total sales, total customers and transaction per customer is greater than 0.05 so the null hypothesis is accepted that the mean values for the total sales, total customers and transaction

per customer is identical for both stores.........

Visualising the difference.....

In [46]:
```
sns.distplot(trial.loc[86]['tot_sales'])
sns.distplot(trial.loc[155]['tot_sales'])
plt.legend(['86', '155'])
plt.title('Distribution plot of sales of stores 86 and 155 during the trial per
plt.show()
```

```
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
```

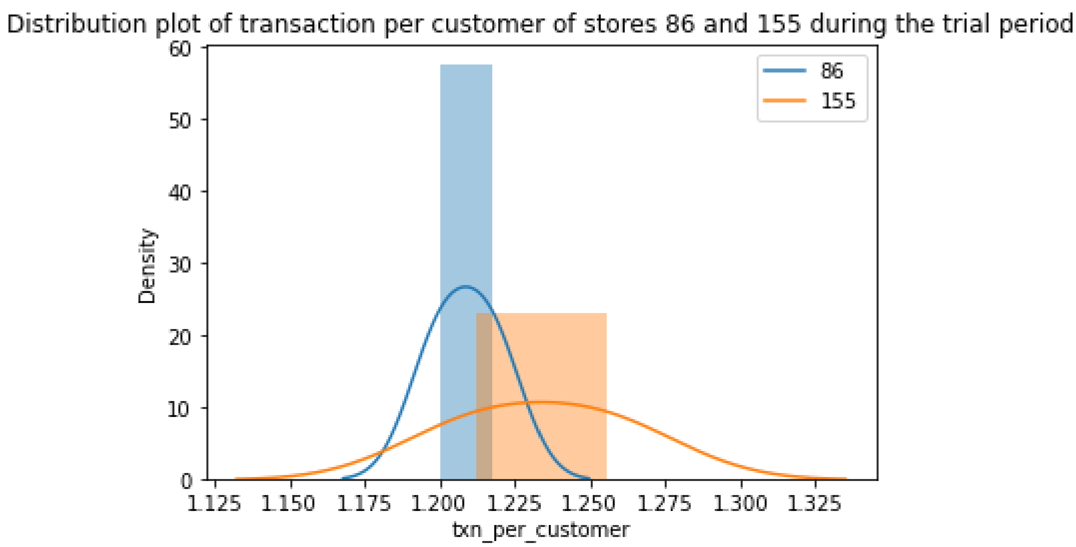


In [47]:
```
sns.distplot(trial.loc[86]['tot_customers'])
sns.distplot(trial.loc[155]['tot_customers'])
plt.legend(['86', '155'])
plt.title('Distribution plot of customers of stores 86 and 155 during the trial
plt.show()
```

```
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
```

In [48]:
```
sns.distplot(trial.loc[86]['txn_per_customer'])
sns.distplot(trial.loc[155]['txn_per_customer'])
plt.legend(['86', '155'])
plt.title('Distribution plot of transaction per customer of stores 86 and 155 d
plt.show()
```

```
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
```



The visualization confirms the hypothesis test conducted. The difference between the total customers for stores 86 and 155 is somewhat significant. There is no significant difference between the transaction per customers and total sales for both stores.

Store 86 only perform well in terms of total customers but not in terms of total sales and transaction per customers.

Finding stores correlated to store 88...

In [49]:
```
corr88= calculate_corr(88)
```

In [50]:
```
corr88
```

Out[50]:

store_number	tot_sales	tot_customers	avg_score
1	0.657299	0.343999	0.500649
2	0.202082	0.127006	0.164544
3	0.371783	0.533943	0.452863
4	0.407844	0.351343	0.379593
5	0.310585	0.280154	0.295370
...	...	...	...
268	0.060247	0.717665	0.388956
269	0.145038	0.060771	0.102905
270	0.636188	0.131879	0.384034
271	0.017985	0.255827	0.136906
272	0.505246	0.094311	0.299778

260 rows × 3 columns

In [51]:
```
corr_table2= corr88.sort_values(by= 'avg_score', ascending= False).head(10)
corr_table2
```

Out[51]:

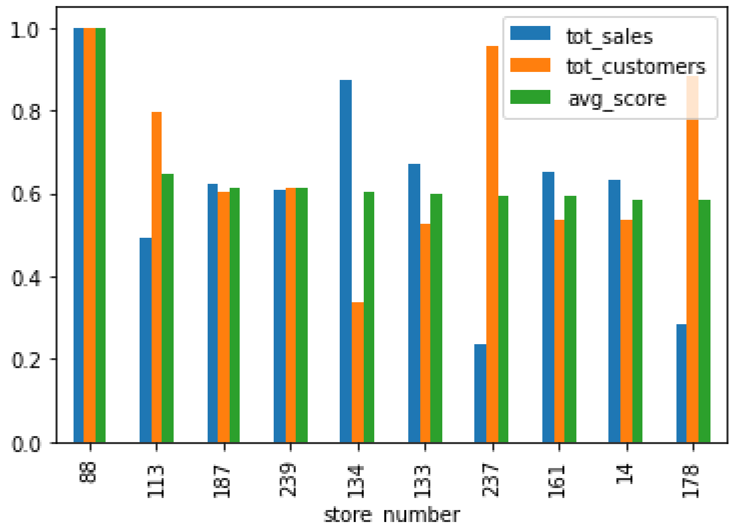
store_number	tot_sales	tot_customers	avg_score
88	1.000000	1.000000	1.000000
113	0.489780	0.797347	0.643564
187	0.619833	0.604224	0.612029
239	0.608617	0.613465	0.611041
134	0.870708	0.338051	0.604379
133	0.669638	0.524265	0.596951
237	0.236508	0.953220	0.594864
161	0.653020	0.534989	0.594004
14	0.632443	0.533647	0.583045
178	0.284718	0.880290	0.582504

From the table above, it is clearly shown that stores 113, 134, 237, 178 correlates most with the trial store 88.

Visualising this......

In [52]:
```
corr_table2.plot(kind= 'bar')
```

Out[52]: <AxesSubplot:xlabel='store_number'>



A statistical test(t_test) will be conducted to show which of these stores is most identical to the trial store 88 or has the same distribution with the trial store.

The null hypothesis is that the stores are identical while the null hypothesis is that the stores are different.

Conducting the t-test for stores 88, 113........

In [53]:
```
pre_significance_test(pre_trial.loc[88], pre_trial.loc[113])
```

Out[53]:

	statistic	pvalue
tot_sales	13.941365	2.277913e-10
tot_customers	11.014370	7.047736e-09
txn_per_customer	-3.890683	1.299023e-03

The pvalue for all metrics for stores 88 and 113 is lesser than 0.05, so the null hypothesis is rejected and stores 88 and 113 are not identical.

Conducting the ttest for stores 88 and 134....

In [54]:
```
pre_significance_test(pre_trial.loc[88], pre_trial.loc[134])
```

Out[54]:

	statistic	pvalue
tot_sales	48.786741	7.784755e-19
tot_customers	39.250045	2.458734e-17
txn_per_customer	9.468581	5.839998e-08

The pvalue for all metrics for stores 88 and 134 is lesser than 0.05, so the null hypothesis is rejected and stores 88 and 134 are not identical.

Conducting the ttest for stores 88 and 237....

In [55]:
```
pre_significance_test(pre_trial.loc[88], pre_trial.loc[237])
```

Out[55]:

	statistic	pvalue
tot_sales	0.992560	0.335699
tot_customers	-0.179154	0.860066
txn_per_customer	1.532020	0.145050

The pvalue for all metrics for stores 88 and 237 is greaterer than 0.05, so we fail to reject the null hypothesis. This confirms that stores 88 and 237 are identical.

Conducting the ttest for stores 88 and 178....

In [56]:
```
pre_significance_test(pre_trial.loc[88], pre_trial.loc[178])
```

Out[56]:

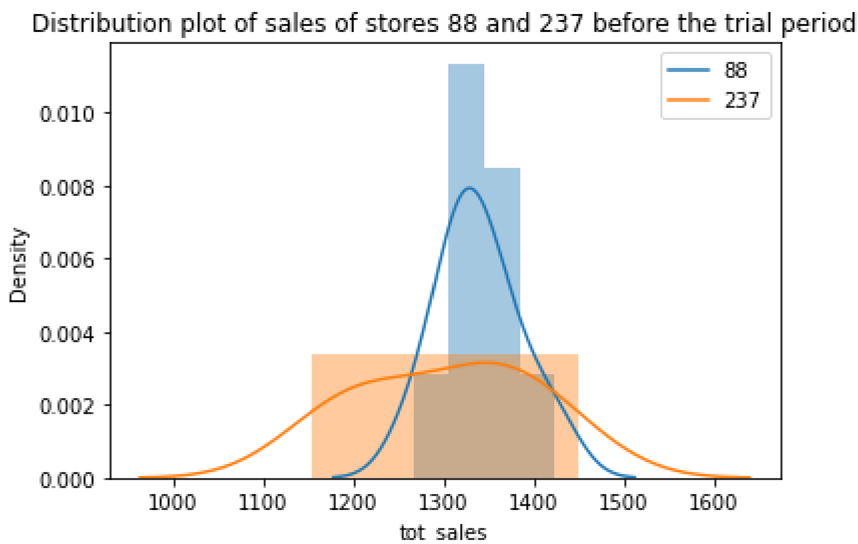
	statistic	pvalue
tot_sales	13.880579	2.430580e-10
tot_customers	9.200267	8.646888e-08
txn_per_customer	-2.061957	5.584234e-02

The pvalue for all metrics for stores 88 and 237 is greaterer than 0.05, so we fail to reject the null hypothesis. This confirms that stores 88 and 237 are identical.

Of all the stores correlated to store 88, store 237 correlates the most, so we will use this in our analysis.

In [57]:
```
sns.distplot(pre_trial.loc[88]['tot_sales'])
sns.distplot(pre_trial.loc[237]['tot_sales'])
plt.legend(['88', '237'])
plt.title('Distribution plot of sales of stores 88 and 237 before the trial pe
plt.show()
```

```
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
```

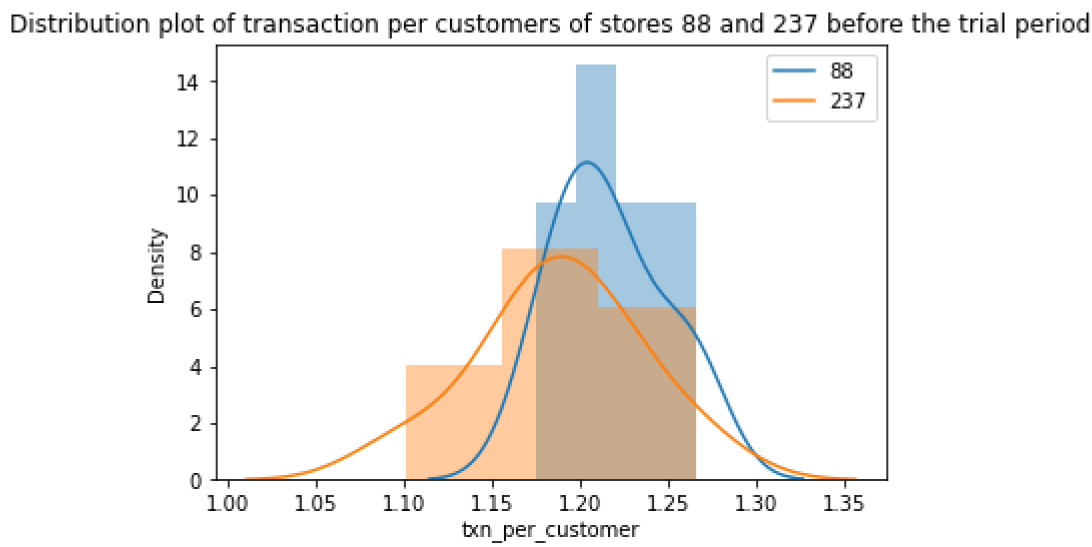


In [58]:
```
sns.distplot(pre_trial.loc[88]['tot_customers'])
sns.distplot(pre_trial.loc[237]['tot_customers'])
plt.legend(['88', '237'])
plt.title('Distribution plot of customers of stores 88 and 237 before the trial
plt.show()
```

```
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
```

In [59]:
```
sns.distplot(pre_trial.loc[88]['txn_per_customer'])
sns.distplot(pre_trial.loc[237]['txn_per_customer'])
plt.legend(['88', '237'])
plt.title('Distribution plot of transaction per customers of stores 88 and 237
plt.show()
```

```
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
```



Lets check if there is any significant difference between both stores for the period during the trial....

In [60]:
```
trial_significance_test(trial.loc[88], trial.loc[237])
```

Out[60]:

	statistic	pvalue
tot_sales	13.268006	0.002816
tot_customers	3.781177	0.031684
txn_per_customer	60.558224	0.000136

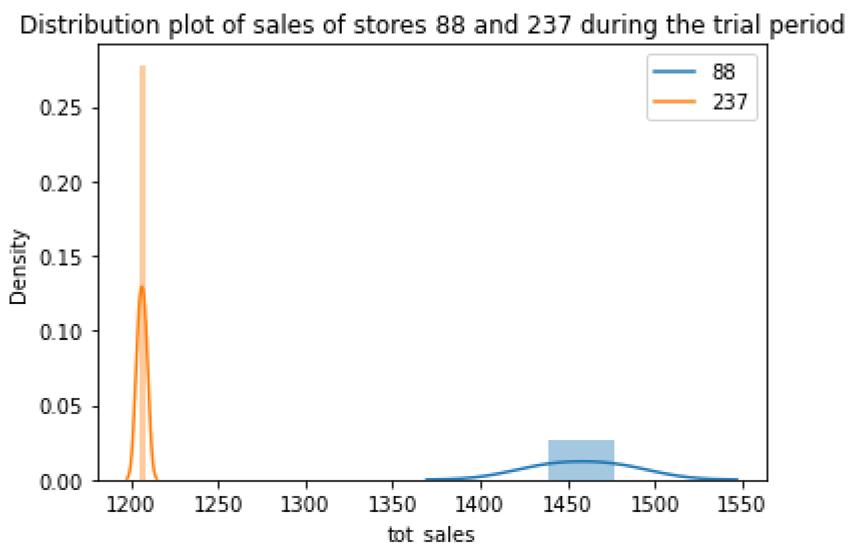
The pvalue for the total sales, total customers and transaction per customer is lesser than 0.05 so the null hypothesis is rejected that the mean values for the total

sales, total customers and transaction per customer is identical for both stores. This shows that the metrics for the trial store is greater than that of the control store.........

Visualising the difference.....

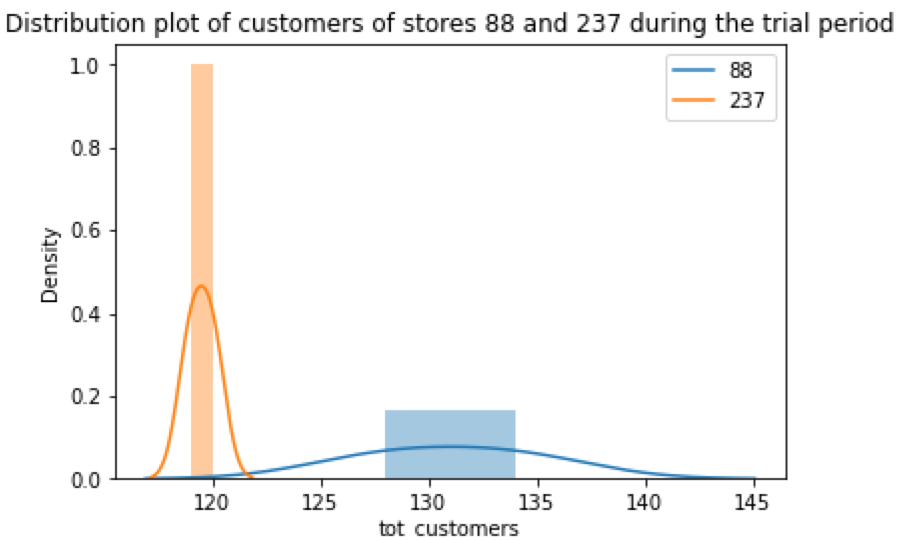
In [61]:
```
sns.distplot(trial.loc[88]['tot_sales'])
sns.distplot(trial.loc[237]['tot_sales'])
plt.legend(['88', '237'])
plt.title('Distribution plot of sales of stores 88 and 237 during the trial per
plt.show()
```

```
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
```



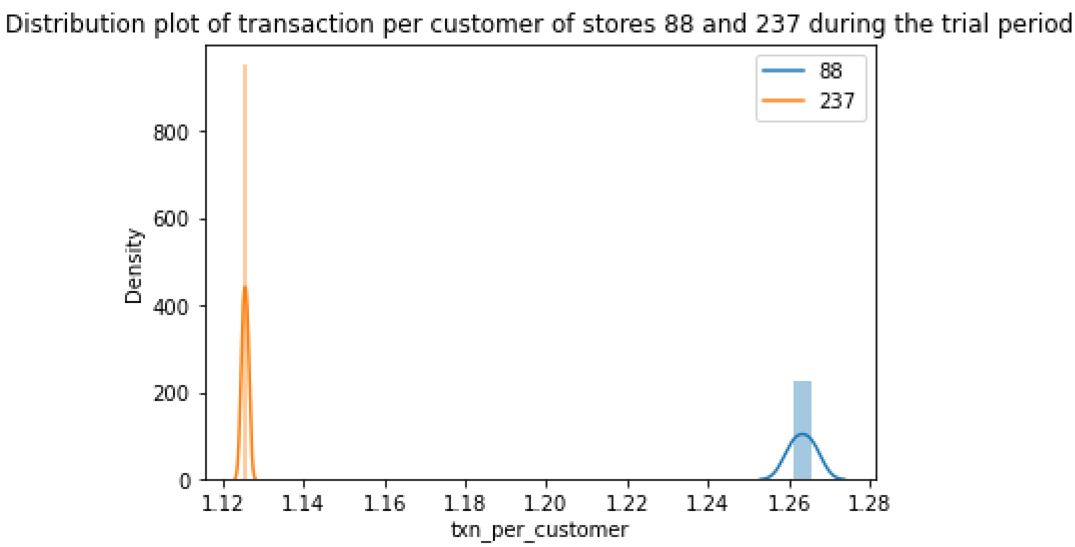
In [62]:
```
sns.distplot(trial.loc[88]['tot_customers'])
sns.distplot(trial.loc[237]['tot_customers'])
plt.legend(['88', '237'])
plt.title('Distribution plot of customers of stores 88 and 237 during the trial
plt.show()
```

```
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
```



In [63]:
```
sns.distplot(trial.loc[88]['txn_per_customer'])
sns.distplot(trial.loc[237]['txn_per_customer'])
plt.legend(['88', '237'])
plt.title('Distribution plot of transaction per customer of stores 88 and 237 (
plt.show()
```

```
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Ceejay\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
```



The visualization confirms the hypothesis test conducted. The difference between the total customers for stores 86 and 155 is greatly significant.

Store 88 does better than its relative control store 237 for all metrics.

Conclusion

- The best performing trial store is store 88 since it does better in sales, customers and transaction per customer than its related control store.
- The worst performing trial store is store 86, as it does worse in sales and transaction per customer than its related control store.
- The trial carried out in store 88 should be emulated by other trial stores as it is the most effective.