In [ ]:
```
import pandas as pd
import numpy as npy
import matplotlib.pyplot as plt
import seaborn as sns
import re
```

In [ ]:
```
df = pd.read_csv(r'A:\Data Analytics\(Forage)Quantium\Quantium_data_model.csv')
```

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

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

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

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 264834 entries, 0 to 264833
Data columns (total 12 columns):
 #   Column            Non-Null Count   Dtype         
---  ------            --------------   -----         
 0   DATE              264834 non-null  datetime64[ns]
 1   STORE_NBR         264834 non-null  int64         
 2   LYLTY_CARD_NBR    264834 non-null  int64         
 3   TXN_ID            264834 non-null  int64         
 4   PROD_NBR          264834 non-null  int64         
 5   PROD_NAME         264833 non-null  object        
 6   PROD_QTY          264834 non-null  int64         
 7   TOT_SALES         264833 non-null  float64       
 8   PACK_SIZE         264833 non-null  float64       
 9   BRAND             264833 non-null  object        
 10  LIFESTAGE         264833 non-null  object        
 11  PREMIUM_CUSTOMER  264833 non-null  object        
dtypes: datetime64[ns](1), float64(2), int64(5), object(4)
memory usage: 24.2+ MB
```

In [ ]:
```
df['month_ID']= df['DATE'].dt.strftime('%Y%m')
```

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

Out[ ]:

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

In [ ]:
```
df_agg_tsales = df.groupby(['STORE_NBR','month_ID'])['TOT_SALES'].agg(['sum']).rena
```

In [ ]:
```
df_agg_tqty = df.groupby(['STORE_NBR','month_ID'])['PROD_QTY'].agg(['sum']).rename(
```

In [ ]:
```
df_agg_tcustomers = df.groupby(['STORE_NBR','month_ID'])['LYLTY_CARD_NBR'].nunique(
```

In [ ]:
```
df_agg_ttransactions = df.groupby(['STORE_NBR','month_ID'])['TXN_ID'].nunique().res
```

In [ ]:
```
df_agg = df_agg_tsales.merge(df_agg_tqty, on=['STORE_NBR', 'month_ID'], how='inner'
```

In [ ]:
```
df_agg = df_agg.merge(df_agg_tcustomers, on=['STORE_NBR', 'month_ID'], how='inner')
```

In [ ]:
```
df_agg = df_agg.merge(df_agg_ttransactions, on=['STORE_NBR', 'month_ID'], how='inne
```

In [ ]:
```
df_agg
```

Out[ ]:

	STORE_NBR	month_ID	TOT_SALES	TOT_QTY	TOT_Customers	TOT_Transactions
0	0	201812	0.0	0	1	1
1	1	201807	206.9	62	49	52
2	1	201808	176.1	54	42	43
3	1	201809	278.8	75	59	62
4	1	201810	188.1	58	44	45
...	...	...	...	...	...	...
3165	272	201902	395.5	91	45	48
3166	272	201903	442.3	101	50	53
3167	272	201904	445.1	105	54	55
3168	272	201905	314.6	71	34	40
3169	272	201906	312.1	70	34	37

3170 rows × 6 columns

In [ ]:
```
df_agg['TRN_P_Customer'] = df_agg['TOT_Transactions']/df_agg['TOT_Customers']
```

In [ ]:
```
df_agg
```

Out[ ]:

	STORE_NBR	month_ID	TOT_SALES	TOT_QTY	TOT_Customers	TOT_Transactions	TRI
0	0	201812	0.0	0	1	1	
1	1	201807	206.9	62	49	52	
2	1	201808	176.1	54	42	43	
3	1	201809	278.8	75	59	62	
4	1	201810	188.1	58	44	45	
...	...	...	...	...	...	...	
3165	272	201902	395.5	91	45	48	
3166	272	201903	442.3	101	50	53	
3167	272	201904	445.1	105	54	55	
3168	272	201905	314.6	71	34	40	
3169	272	201906	312.1	70	34	37	

3170 rows × 7 columns

In [ ]:
```
df_agg['Chips_per_customer'] = df_agg['TOT_QTY'] / df_agg['TOT_Customers']
```

In [ ]:
```
df_agg['avgPrice_per_unit'] = df_agg['TOT_SALES'] / df_agg['TOT_QTY']
```

In [ ]:
```
df_agg
```

Out[ ]:

	STORE_NBR	month_ID	TOT_SALES	TOT_QTY	TOT_Customers	TOT_Transactions	TR
0	0	201812	0.0	0	1	1	
1	1	201807	206.9	62	49	52	
2	1	201808	176.1	54	42	43	
3	1	201809	278.8	75	59	62	
4	1	201810	188.1	58	44	45	
...	...	...	...	...	...	...	
3165	272	201902	395.5	91	45	48	
3166	272	201903	442.3	101	50	53	
3167	272	201904	445.1	105	54	55	
3168	272	201905	314.6	71	34	40	
3169	272	201906	312.1	70	34	37	

3170 rows × 9 columns

In [ ]:
```
#Pre-trial Observaltions for all the stores

df_agg_po = df_agg.query('month_ID < "201903"')
df_agg_po
```

Out[ ]:

	STORE_NBR	month_ID	TOT_SALES	TOT_QTY	TOT_Customers	TOT_Transactions	TRI
0	0	201812	0.0	0	1	1	
1	1	201807	206.9	62	49	52	
2	1	201808	176.1	54	42	43	
3	1	201809	278.8	75	59	62	
4	1	201810	188.1	58	44	45	
...	...	...	...	...	...	...	
3161	272	201810	430.6	99	44	50	
3162	272	201811	376.2	87	41	45	
3163	272	201812	403.9	89	47	47	
3164	272	201901	423.0	96	46	50	
3165	272	201902	395.5	91	45	48	

2113 rows × 9 columns

In [ ]:
```
df_agg_po.query('month_ID > "201903"')
```

Out[ ]:

	STORE_NBR	month_ID	TOT_SALES	TOT_QTY	TOT_Customers	TOT_Transactions	TRN_P_(

Data analysis for Trial Store 77

In [ ]:
```
df_agg_po.query('STORE_NBR == 77')
```

Out[ ]:

	STORE_NBR	month_ID	TOT_SALES	TOT_QTY	TOT_Customers	TOT_Transactions	TRN
881	77	201807	296.8	84	51	55	
882	77	201808	255.5	74	47	48	
883	77	201809	225.2	70	42	44	
884	77	201810	204.5	52	37	38	
885	77	201811	245.3	67	41	44	
886	77	201812	267.3	72	46	48	
887	77	201901	204.4	65	35	39	
888	77	201902	235.0	74	45	45	

In [ ]:
```
df_agg_po_str1 = df_agg_po.query('STORE_NBR == 77')[['month_ID','TOT_SALES','TOT_Cu
```

In [ ]:
```
# str1_sales_mean = df_agg_po_str1[['TOT_SALES']].mean()
```

In [ ]:
```
df_agg_po_str1
```

Out[ ]:

	month_ID	TOT_SALES	TOT_Customers
0	201807	296.8	51
1	201808	255.5	47
2	201809	225.2	42
3	201810	204.5	37
4	201811	245.3	41
5	201812	267.3	46
6	201901	204.4	35
7	201902	235.0	45

In [ ]:
```
def tpl(n):
    np =df_agg_po.query('STORE_NBR == @n')[['TOT_SALES']].reset_index(drop=True)
    r = df_agg_po_str1['TOT_SALES'].corr(np['TOT_SALES'])
    return r

sales_list = []
for i in range(1,273):
        coe = tpl(i)  # Call the custom function
        sales_list.append(coe)
```

```
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
```

```
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
```

In [ ]:
```
#sales_list
```

In [ ]:
```
sales_list = pd.DataFrame(sales_list).rename(columns={0:'Sales_coe'})
sales_list
```

Out[ ]:

	Sales_coe
0	0.050163
1	-0.200796
2	0.616574
3	-0.094125
4	-0.045765
...	...
267	0.350807
268	-0.315842
269	0.324937
270	0.362891
271	0.114645

272 rows × 1 columns

In [ ]:
```
sales_list.query('Sales_coe >= 0.8')
```

Out[ ]:

	Sales_coe
16	0.843806
76	1.000000
118	0.831832
232	0.894375

Customer Data Analysis

In [ ]:
```
def tpl(n):
    np = df_agg_po.query('STORE_NBR == @n')[['TOT_Customers']].reset_index(drop=Tru
    r = df_agg_po_str1['TOT_Customers'].corr(np['TOT_Customers'])
    return r

customer_list = []
for i in range(1,273):
        cmr_coe = tpl(i)  # Call the custom function
        customer_list.append(cmr_coe)
```

```
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2922: RuntimeWarning: invalid value encountered in divid
e
  c /= stddev[:, None]
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2923: RuntimeWarning: invalid value encountered in divid
e
  c /= stddev[None, :]
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2922: RuntimeWarning: invalid value encountered in divid
e
  c /= stddev[:, None]
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2923: RuntimeWarning: invalid value encountered in divid
e
  c /= stddev[None, :]
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2922: RuntimeWarning: invalid value encountered in divid
e
  c /= stddev[:, None]
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2923: RuntimeWarning: invalid value encountered in divid
e
  c /= stddev[None, :]
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
```

```
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2922: RuntimeWarning: invalid value encountered in divid
e
  c /= stddev[:, None]
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2923: RuntimeWarning: invalid value encountered in divid
e
  c /= stddev[None, :]
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
```

```
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
```

In [ ]:
```
#customer_list
```

In [ ]:
```
customer_list = pd.DataFrame(customer_list).rename(columns={0:'Customer_coe'})
```

In [ ]:
```
customer_list
```

Out[ ]:

	Customer_coe
0	0.350572
1	-0.455226
2	0.756913
3	-0.302046
4	0.277619
...	...
267	0.283287
268	-0.492708
269	-0.176932
270	-0.056788
271	0.238448

272 rows × 1 columns

In [ ]:
```
customer_list.query('Customer_coe >= 0.8')
```

Out[ ]:

	Customer_coe
40	0.811844
76	1.000000
118	0.977709
232	0.990542

In [ ]:
```
sales_list.query('Sales_coe >= 0.8')
```

Out[ ]:

	Sales_coe
16	0.843806
76	1.000000
118	0.831832
232	0.894375

Index 232 (Store 233) has strogest positive relation with Store 77's total sales and total customer data thus we will select #Store233 as the control store for trial store 77

In [ ]:
```
df_po_st1 = df_agg_po.query('STORE_NBR == 77')[['month_ID','TOT_Customers', 'TOT_SA
```

In [ ]:
```
df_agg_po.query('STORE_NBR == 233')[['month_ID','TOT_Customers', 'TOT_SALES']]
```

Out[ ]:

	month_ID	TOT_Customers	TOT_SALES
2700	201807	51	290.7
2701	201808	48	285.9
2702	201809	42	228.6
2703	201810	35	185.7
2704	201811	40	211.6
2705	201812	47	279.8
2706	201901	35	177.5
2707	201902	45	244.0

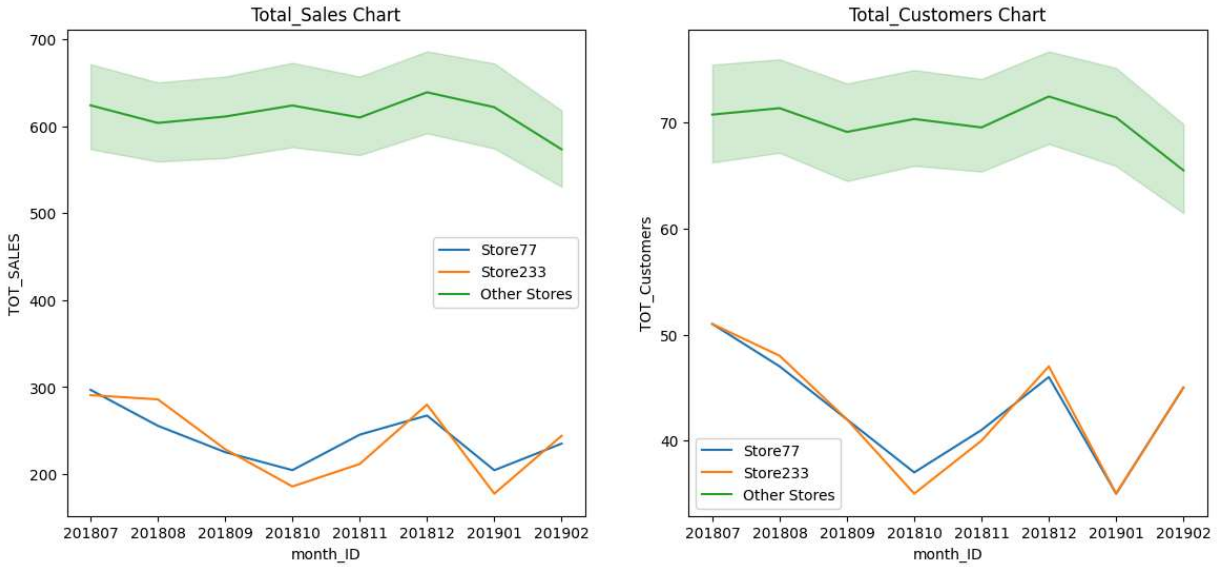
In [ ]:
```
df_po_st1_c = df_agg_po.query('STORE_NBR == 233')[['month_ID','TOT_Customers', 'TOT
```

We are selecting Store 233 to be the control store for trial store 77

In [ ]:
```
plt.figure(figsize=(14,6))
plt.subplot(1,2,1)
sns.lineplot(data=df_po_st1, x='month_ID', y='TOT_SALES', label='Store77')
sns.lineplot(data=df_po_st1_c, x='month_ID', y='TOT_SALES', label='Store233')
sns.lineplot(data=df_agg_po.query('STORE_NBR != [77,233]'), x='month_ID', y='TOT_SA
```

```
plt.title('Total_Sales Chart')
plt.subplot(1,2,2)
sns.lineplot(data=df_po_st1, x='month_ID', y='TOT_Customers', label='Store77')
sns.lineplot(data=df_po_st1_c, x='month_ID', y='TOT_Customers', label='Store233')
sns.lineplot(data=df_agg_po.query('STORE_NBR != [77,233]'), x='month_ID', y='TOT_Cu
plt.title('Total_Customers Chart')
```

Out[ ]: Text(0.5, 1.0, 'Total_Customers Chart')



Store77 Statistical analysis

In [ ]:
```
str1c_factor = df_po_st1['TOT_SALES'].sum()/df_po_st1_c['TOT_SALES'].sum()
```

In [ ]:
```
str1_sales = df_agg.query('STORE_NBR == 77')[['month_ID','TOT_SALES']]
```

In [ ]:
```
str1_c_sales = df_agg.query('STORE_NBR == 233')[['month_ID','TOT_SALES']]
```

In [ ]:
```
str1_c_sales['TOT_SALES'] * str1c_factor
```

Out[ ]:
```
2700    295.311377
2701    290.435235
2702    232.226284
2703    188.645761
2704    214.956613
2705    284.238470
2706    180.315684
2707    247.870575
2708    202.258325
2709    161.115874
2710    349.863221
2711    224.505725
Name: TOT_SALES, dtype: float64
```

In [ ]:
```
str1_c_ScaledSales = str1_c_sales['TOT_SALES'] * str1c_factor
```

In [ ]:
```
str1_c_ScaledSales
```

Out[ ]:
```
2700    295.311377
2701    290.435235
2702    232.226284
2703    188.645761
2704    214.956613
2705    284.238470
2706    180.315684
2707    247.870575
2708    202.258325
2709    161.115874
2710    349.863221
2711    224.505725
Name: TOT_SALES, dtype: float64
```

In [ ]:
```
str1_c_sales['Scaled_Sales'] = str1_c_ScaledSales
```

In [ ]:
```
str1_c_sales
```

Out[ ]:

	month_ID	TOT_SALES	Scaled_Sales
2700	201807	290.7	295.311377
2701	201808	285.9	290.435235
2702	201809	228.6	232.226284
2703	201810	185.7	188.645761
2704	201811	211.6	214.956613
2705	201812	279.8	284.238470
2706	201901	177.5	180.315684
2707	201902	244.0	247.870575
2708	201903	199.1	202.258325
2709	201904	158.6	161.115874
2710	201905	344.4	349.863221
2711	201906	221.0	224.505725

In [ ]:
```
str1_sales.query('month_ID > "201902"')
```

Out[ ]:

	month_ID	TOT_SALES
889	201903	278.5
890	201904	263.5
891	201905	299.3
892	201906	264.7

In [ ]:
```
str1_sales
```

Out[ ]:

	month_ID	TOT_SALES
881	201807	296.8
882	201808	255.5
883	201809	225.2
884	201810	204.5
885	201811	245.3
886	201812	267.3
887	201901	204.4
888	201902	235.0
889	201903	278.5
890	201904	263.5
891	201905	299.3
892	201906	264.7

In [ ]:
```
str1_c_sales
```

Out[ ]:

	month_ID	TOT_SALES	Scaled_Sales
2700	201807	290.7	295.311377
2701	201808	285.9	290.435235
2702	201809	228.6	232.226284
2703	201810	185.7	188.645761
2704	201811	211.6	214.956613
2705	201812	279.8	284.238470
2706	201901	177.5	180.315684
2707	201902	244.0	247.870575
2708	201903	199.1	202.258325
2709	201904	158.6	161.115874
2710	201905	344.4	349.863221
2711	201906	221.0	224.505725

In [ ]:
```
str1_sales = str1_sales.merge(str1_c_sales[['month_ID','Scaled_Sales']], on='month_
```

In [ ]:
```
((str1_sales['TOT_SALES'] - str1_sales['Scaled_Sales'] ) / str1_sales['Scaled_Sales
```

Out[ ]:
```
0      0.504086
1    -12.028580
2     -3.025620
3      8.404238
4     14.116052
5     -5.959246
6     13.356750
7     -5.192458
8     37.695197
9     63.546890
10   -14.452282
11    17.903452
dtype: float64
```

In [ ]:
```
str1_sales['%_diff'] = abs(((str1_sales['TOT_SALES'] - str1_sales['Scaled_Sales'] )
```

In [ ]:
```
str1_sales
```

Out[ ]:

	month_ID	TOT_SALES	Scaled_Sales	%_diff
0	201807	296.8	295.311377	0.504086
1	201808	255.5	290.435235	12.028580
2	201809	225.2	232.226284	3.025620
3	201810	204.5	188.645761	8.404238
4	201811	245.3	214.956613	14.116052
5	201812	267.3	284.238470	5.959246
6	201901	204.4	180.315684	13.356750
7	201902	235.0	247.870575	5.192458
8	201903	278.5	202.258325	37.695197
9	201904	263.5	161.115874	63.546890
10	201905	299.3	349.863221	14.452282
11	201906	264.7	224.505725	17.903452

In [ ]:
```
sd = str1_sales.query('month_ID < "201903"')['%_diff'].std()
sd
```

Out[ ]:
```
np.float64(5.003203638285844)
```

In [ ]:
```
str1_sales.query('month_ID < "201903"')
```

Out[ ]:

	month_ID	TOT_SALES	Scaled_Sales	%_diff
0	201807	296.8	295.311377	0.504086
1	201808	255.5	290.435235	12.028580
2	201809	225.2	232.226284	3.025620
3	201810	204.5	188.645761	8.404238
4	201811	245.3	214.956613	14.116052
5	201812	267.3	284.238470	5.959246
6	201901	204.4	180.315684	13.356750
7	201902	235.0	247.870575	5.192458

In [ ]:
```
str1_sales['t_values'] =  (str1_sales['%_diff'] - 0) / sd
str1_sales
```

Out[ ]:

	month_ID	TOT_SALES	Scaled_Sales	%_diff	t_values
0	201807	296.8	295.311377	0.504086	0.100753
1	201808	255.5	290.435235	12.028580	2.404176
2	201809	225.2	232.226284	3.025620	0.604736
3	201810	204.5	188.645761	8.404238	1.679771
4	201811	245.3	214.956613	14.116052	2.821403
5	201812	267.3	284.238470	5.959246	1.191086
6	201901	204.4	180.315684	13.356750	2.669640
7	201902	235.0	247.870575	5.192458	1.037827
8	201903	278.5	202.258325	37.695197	7.534212
9	201904	263.5	161.115874	63.546890	12.701240
10	201905	299.3	349.863221	14.452282	2.888606
11	201906	264.7	224.505725	17.903452	3.578398

In [ ]:
```
str1_sales.query(' month_ID > "201902"')
```

Out[ ]:

	month_ID	TOT_SALES	Scaled_Sales	%_diff	t_values
8	201903	278.5	202.258325	37.695197	7.534212
9	201904	263.5	161.115874	63.546890	12.701240
10	201905	299.3	349.863221	14.452282	2.888606
11	201906	264.7	224.505725	17.903452	3.578398

95 percentile t-value of degree of freedom 7 is 1.894579

t-value of the samples during the trial period is higher.

In [ ]:
```
str1_comp_df = str1_sales.merge(str1_c_sales[['month_ID','TOT_SALES']], on='month_I
```

In [ ]:
```
str1_comp_df.drop(columns=['Scaled_Sales','%_diff'], inplace=True)
```

In [ ]:
```
str1_comp_df
```

Out[ ]:

	month_ID	TOT_SALES	t_values	TOT_SALES_C
0	201807	296.8	0.100753	290.7
1	201808	255.5	2.404176	285.9
2	201809	225.2	0.604736	228.6
3	201810	204.5	1.679771	185.7
4	201811	245.3	2.821403	211.6
5	201812	267.3	1.191086	279.8
6	201901	204.4	2.669640	177.5
7	201902	235.0	1.037827	244.0
8	201903	278.5	7.534212	199.1
9	201904	263.5	12.701240	158.6
10	201905	299.3	2.888606	344.4
11	201906	264.7	3.578398	221.0

In [ ]:
```
str1_comp_df['p_95'] = str1_comp_df['TOT_SALES_C'] + pow(sd,2)
str1_comp_df['p_5'] = str1_comp_df['TOT_SALES_C'] - pow(sd,2)
str1_comp_df
```

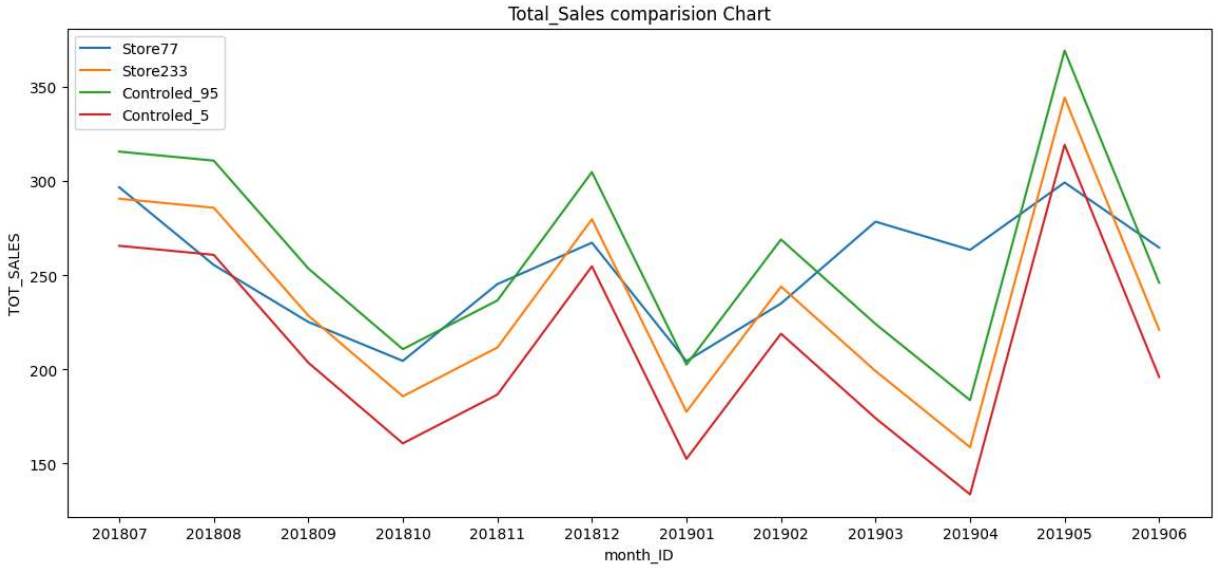
Out[ ]:

	month_ID	TOT_SALES	t_values	TOT_SALES_C	p_95	p_5
0	201807	296.8	0.100753	290.7	315.732047	265.667953
1	201808	255.5	2.404176	285.9	310.932047	260.867953
2	201809	225.2	0.604736	228.6	253.632047	203.567953
3	201810	204.5	1.679771	185.7	210.732047	160.667953
4	201811	245.3	2.821403	211.6	236.632047	186.567953
5	201812	267.3	1.191086	279.8	304.832047	254.767953
6	201901	204.4	2.669640	177.5	202.532047	152.467953
7	201902	235.0	1.037827	244.0	269.032047	218.967953
8	201903	278.5	7.534212	199.1	224.132047	174.067953
9	201904	263.5	12.701240	158.6	183.632047	133.567953
10	201905	299.3	2.888606	344.4	369.432047	319.367953
11	201906	264.7	3.578398	221.0	246.032047	195.967953

In [ ]:
```
plt.figure(figsize=(14,6))
sns.lineplot(data=str1_comp_df, x='month_ID', y='TOT_SALES', label='Store77')
sns.lineplot(data=str1_comp_df, x='month_ID', y='TOT_SALES_C', label='Store233')
```

```
sns.lineplot(data=str1_comp_df, x='month_ID', y='p_95', label='Controled_95')
sns.lineplot(data=str1_comp_df, x='month_ID', y='p_5', label='Controled_5')
plt.title('Total_Sales comparision Chart')
```

Out[ ]: Text(0.5, 1.0, 'Total_Sales comparision Chart')



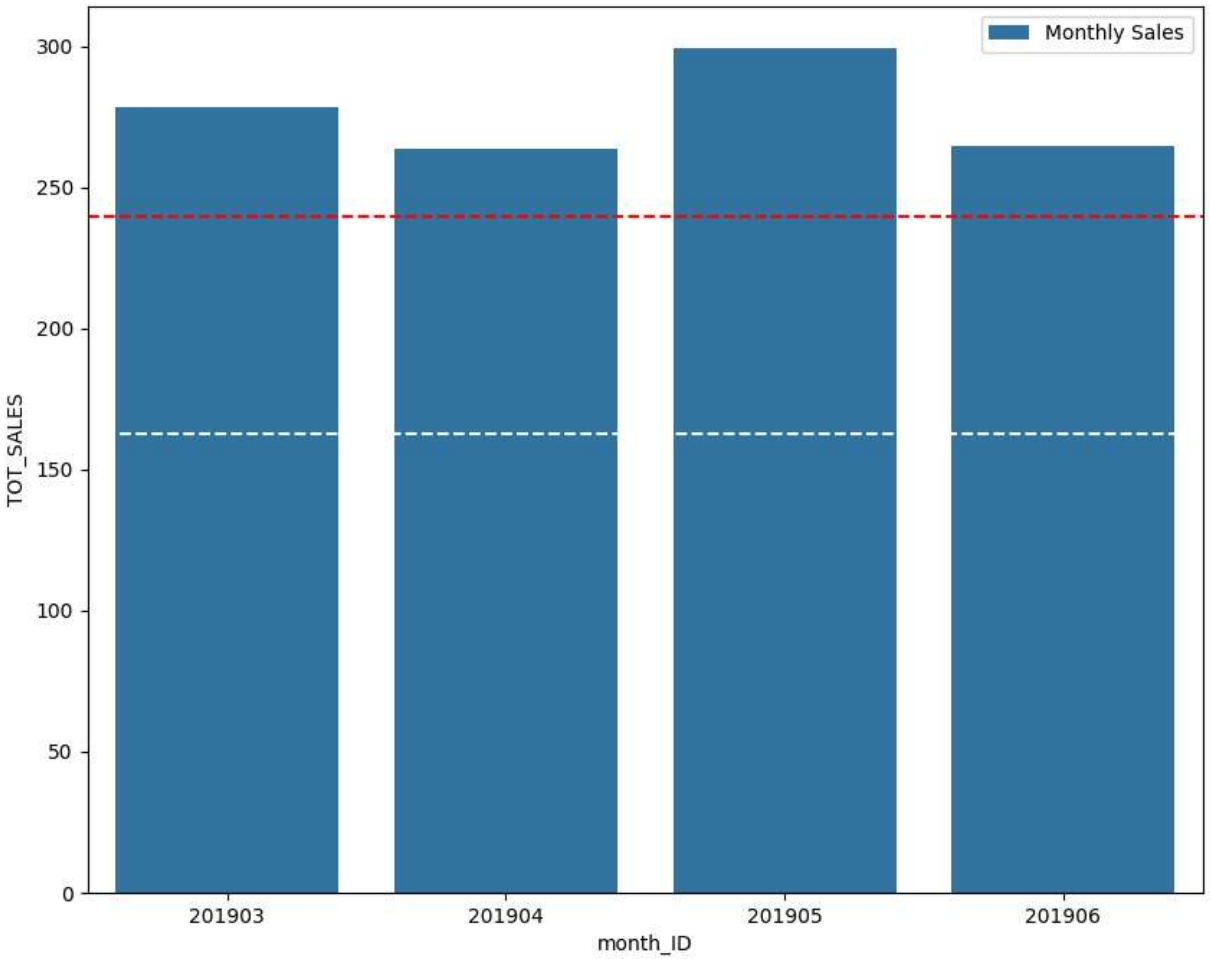
In [ ]:
```
str1_c_95 = str1_comp_df.query('"201901"< month_ID < "201905"')['TOT_SALES_C'].quan
```

In [ ]:
```
str1_c_5 = str1_comp_df.query('"201901"< month_ID < "201905"')['TOT_SALES_C'].quant
```

In [ ]:
```
plt.figure(figsize=(10, 8))
sns.barplot(data=str1_comp_df.query('month_ID >"201902"'), x='month_ID', y='TOT_SAL

plt.axhline(y=str1_c_95, color='red', linestyle='--', label='95th Percentile')
plt.axhline(y=str1_c_5, color='white', linestyle='--', label='95th Percentile')
```

Out[ ]: <matplotlib.lines.Line2D at 0x28905667a40>



customer data analysis for store 77

In [ ]:
```
str1c_cmr_factor = df_po_st1['TOT_Customers'].sum()/df_po_st1_c['TOT_Customers'].su
str1c_cmr_factor
```

Out[ ]:
```
np.float64(1.0029154518950438)
```

In [ ]:
```
str1_customer = df_agg.query('STORE_NBR == 77')[['month_ID','TOT_Customers']]
str1_customer
```

Out[ ]:

	month_ID	TOT_Customers
881	201807	51
882	201808	47
883	201809	42
884	201810	37
885	201811	41
886	201812	46
887	201901	35
888	201902	45
889	201903	50
890	201904	47
891	201905	55
892	201906	41

In [ ]:
```
str1_c_customer = df_agg.query('STORE_NBR == 233')[['month_ID','TOT_Customers']]
str1_c_customer
```

Out[ ]:

	month_ID	TOT_Customers
2700	201807	51
2701	201808	48
2702	201809	42
2703	201810	35
2704	201811	40
2705	201812	47
2706	201901	35
2707	201902	45
2708	201903	40
2709	201904	30
2710	201905	57
2711	201906	41

In [ ]:
```
str1_customer = str1_customer.merge(str1_c_customer, on='month_ID', how='inner').re
str1_customer
```

Out[ ]:

	month_ID	T_Customers	T_Customers_C
0	201807	51	51
1	201808	47	48
2	201809	42	42
3	201810	37	35
4	201811	41	40
5	201812	46	47
6	201901	35	35
7	201902	45	45
8	201903	50	40
9	201904	47	30
10	201905	55	57
11	201906	41	41

In [ ]:
```
str1_customer['Scaled_Customers'] = str1_customer['T_Customers_C'] * str1c_factor
str1_customer
```

Out[ ]:

	month_ID	T_Customers	T_Customers_C	Scaled_Customers
0	201807	51	51	51.809014
1	201808	47	48	48.761425
2	201809	42	42	42.666246
3	201810	37	35	35.555205
4	201811	41	40	40.634520
5	201812	46	47	47.745562
6	201901	35	35	35.555205
7	201902	45	45	45.713835
8	201903	50	40	40.634520
9	201904	47	30	30.475890
10	201905	55	57	57.904192
11	201906	41	41	41.650383

In [ ]:
```
str1_customer['%_diff'] = abs(((str1_customer['T_Customers'] - str1_customer['Scale
str1_customer
```

Out[ ]:

	month_ID	T_Customers	T_Customers_C	Scaled_Customers	%_diff
0	201807	51	51	51.809014	1.561531
1	201808	47	48	48.761425	3.612332
2	201809	42	42	42.666246	1.561531
3	201810	37	35	35.555205	4.063525
4	201811	41	40	40.634520	0.899431
5	201812	46	47	47.745562	3.655966
6	201901	35	35	35.555205	1.561531
7	201902	45	45	45.713835	1.561531
8	201903	50	40	40.634520	23.048087
9	201904	47	30	30.475890	54.220269
10	201905	55	57	57.904192	5.015512
11	201906	41	41	41.650383	1.561531

In [ ]:
```
str1_customer.query('month_ID < "201903"')
```

Out[ ]:

	month_ID	T_Customers	T_Customers_C	Scaled_Customers	%_diff
0	201807	51	51	51.809014	1.561531
1	201808	47	48	48.761425	3.612332
2	201809	42	42	42.666246	1.561531
3	201810	37	35	35.555205	4.063525
4	201811	41	40	40.634520	0.899431
5	201812	46	47	47.745562	3.655966
6	201901	35	35	35.555205	1.561531
7	201902	45	45	45.713835	1.561531

In [ ]:
```
sd = str1_customer.query('month_ID < "201903"')['%_diff'].std()
sd
```

Out[ ]:
```
np.float64(1.242869510442506)
```

In [ ]:
```
str1_customer['t_values'] =  (str1_customer['%_diff'] - 0) / sd
str1_customer
```

Out[ ]:

	month_ID	T_Customers	T_Customers_C	Scaled_Customers	%_diff	t_values
0	201807	51	51	51.809014	1.561531	1.256391
1	201808	47	48	48.761425	3.612332	2.906445
2	201809	42	42	42.666246	1.561531	1.256391
3	201810	37	35	35.555205	4.063525	3.269470
4	201811	41	40	40.634520	0.899431	0.723673
5	201812	46	47	47.745562	3.655966	2.941553
6	201901	35	35	35.555205	1.561531	1.256391
7	201902	45	45	45.713835	1.561531	1.256391
8	201903	50	40	40.634520	23.048087	18.544253
9	201904	47	30	30.475890	54.220269	43.625070
10	201905	55	57	57.904192	5.015512	4.035429
11	201906	41	41	41.650383	1.561531	1.256391

In [ ]:
```
str1_customer.query(' month_ID > "201902"')
```

Out[ ]:

	month_ID	T_Customers	T_Customers_C	Scaled_Customers	%_diff	t_values
8	201903	50	40	40.634520	23.048087	18.544253
9	201904	47	30	30.475890	54.220269	43.625070
10	201905	55	57	57.904192	5.015512	4.035429
11	201906	41	41	41.650383	1.561531	1.256391

95 percentile t-value of degree of freedom 7 is 1.894579

t-value of the samples during the trial period is higher in March, April & May months.

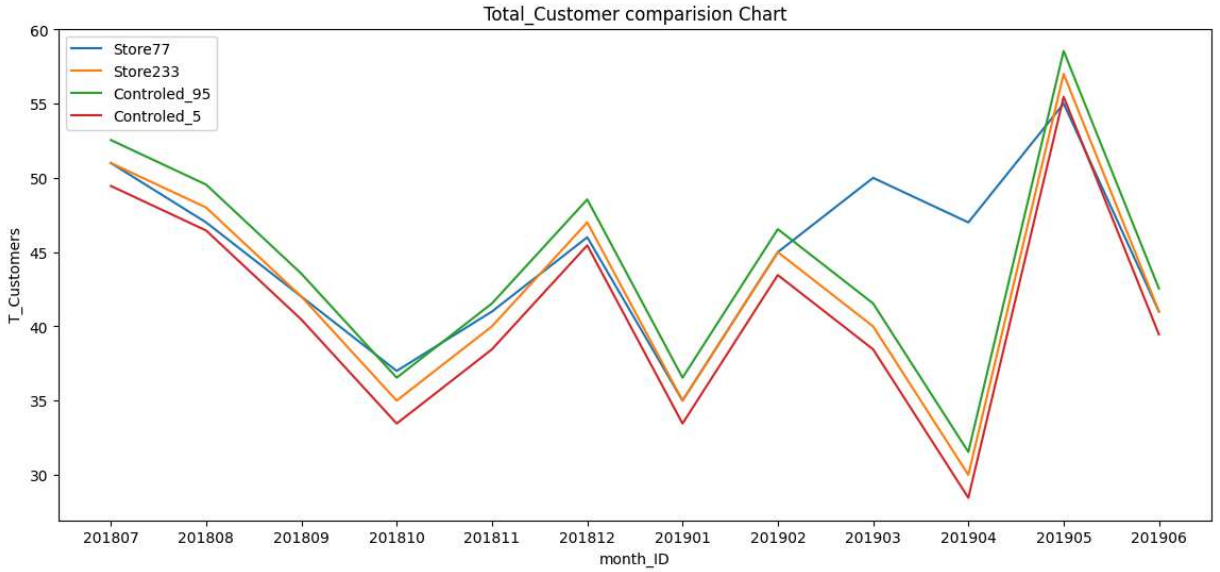
In [ ]:
```
str1_customer['p_95'] = str1_customer['T_Customers_C'] + pow(sd,2)
str1_customer['p_5'] = str1_customer['T_Customers_C'] - pow(sd,2)
str1_customer
```

Out[ ]:

	month_ID	T_Customers	T_Customers_C	Scaled_Customers	%_diff	t_values	
0	201807	51	51	51.809014	1.561531	1.256391	52.54
1	201808	47	48	48.761425	3.612332	2.906445	49.54
2	201809	42	42	42.666246	1.561531	1.256391	43.54
3	201810	37	35	35.555205	4.063525	3.269470	36.54
4	201811	41	40	40.634520	0.899431	0.723673	41.54
5	201812	46	47	47.745562	3.655966	2.941553	48.54
6	201901	35	35	35.555205	1.561531	1.256391	36.54
7	201902	45	45	45.713835	1.561531	1.256391	46.54
8	201903	50	40	40.634520	23.048087	18.544253	41.54
9	201904	47	30	30.475890	54.220269	43.625070	31.54
10	201905	55	57	57.904192	5.015512	4.035429	58.54
11	201906	41	41	41.650383	1.561531	1.256391	42.54

In [ ]:
```
plt.figure(figsize=(14,6))
sns.lineplot(data=str1_customer, x='month_ID', y='T_Customers', label='Store77')
sns.lineplot(data=str1_customer, x='month_ID', y='T_Customers_C', label='Store233')
sns.lineplot(data=str1_customer, x='month_ID', y='p_95', label='Controled_95')
sns.lineplot(data=str1_customer, x='month_ID', y='p_5', label='Controled_5')
plt.title('Total_Customer comparision Chart')
```

Out[ ]:
```
Text(0.5, 1.0, 'Total_Customer comparision Chart')
```



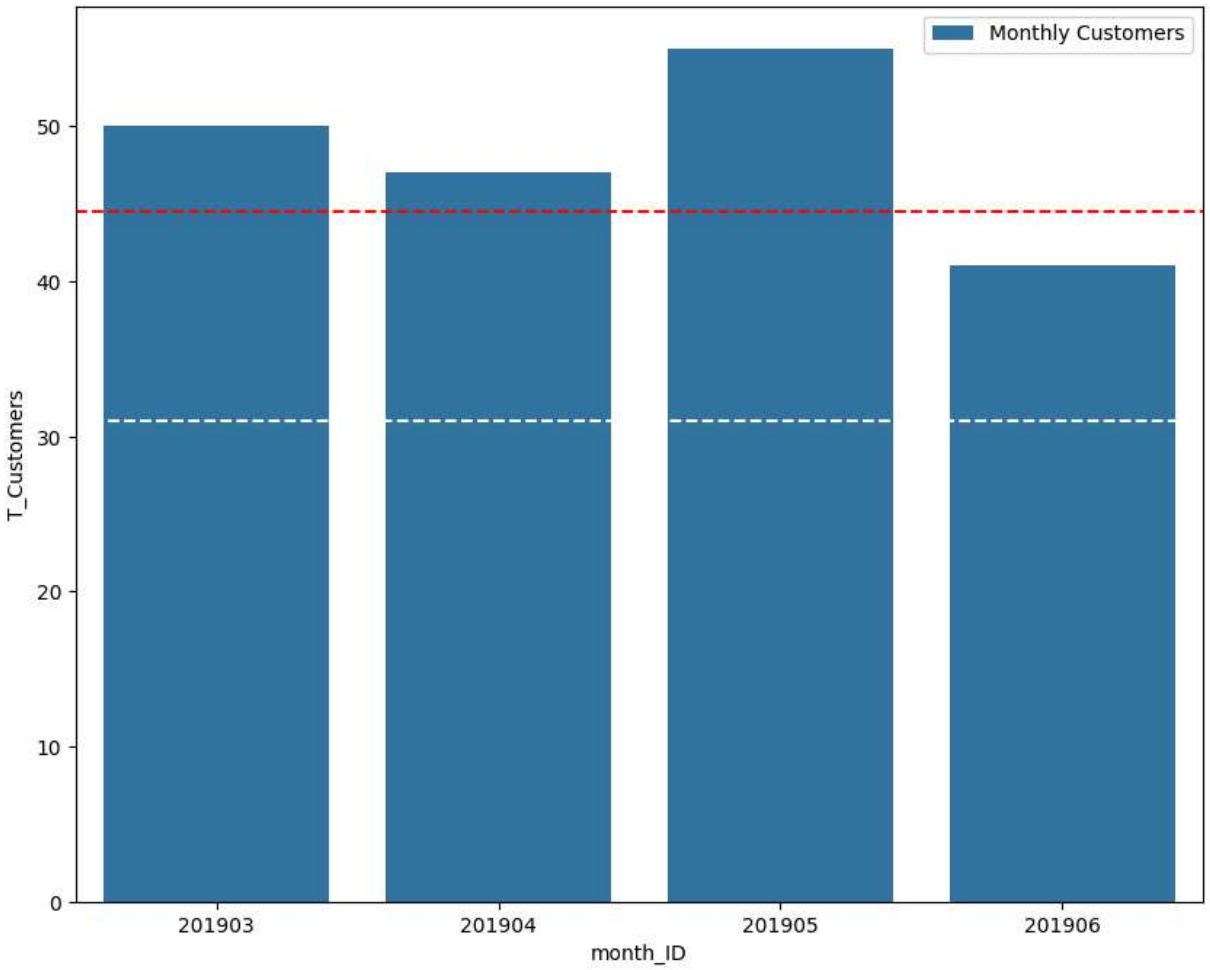
In [ ]:
```
str1_c_c95 = str1_customer.query('"201901"< month_ID < "201905"')['T_Customers_C'].
```

In [ ]:
```
str1_c_c5 = str1_customer.query('"201901"< month_ID < "201905"')['T_Customers_C'].q
```

In [ ]:
```
plt.figure(figsize=(10, 8))
sns.barplot(data=str1_customer.query('month_ID >"201902"'), x='month_ID', y='T_Cust

plt.axhline(y=str1_c_c95, color='red', linestyle='--', label='95th Percentile')
plt.axhline(y=str1_c_c5, color='white', linestyle='--', label='5th Percentile')
```

Out[ ]:
```
<matplotlib.lines.Line2D at 0x28905c384d0>
```



During the trial period we can see significant increase number of customers in 3/4 months and for the total sales 4/4 months of the trial store compare to the control store.

The trial was a successfull one for Store 77

Data Analysis for Trial Store 86

In [ ]:
```
df_agg_po.query('STORE_NBR == [86]')
```

Out[ ]:

	STORE_NBR	month_ID	TOT_SALES	TOT_QTY	TOT_Customers	TOT_Transactions	TRN
978	86	201807	892.20	251	99	126	
979	86	201808	764.05	215	94	110	
980	86	201809	914.60	258	103	128	
981	86	201810	948.40	276	109	138	
982	86	201811	918.00	254	100	125	
983	86	201812	841.20	240	98	120	
984	86	201901	841.40	260	94	129	
985	86	201902	913.20	277	107	138	

In [ ]:
```
df_agg_po_str2 = df_agg_po.query('STORE_NBR == [86]')[['TOT_Customers', 'TOT_SALES'
```

In [ ]:
```
#str2_sales_mean = df_agg_po_str2[['TOT_SALES']].mean()
#str2_sales_mean
```

In [ ]:
```
def tpl(n):
    np =df_agg_po.query('STORE_NBR == @n')[['TOT_SALES']].reset_index(drop=True)
    r = df_agg_po_str2['TOT_SALES'].corr(np['TOT_SALES'])
    return r

sales_list = []
for i in range(1,273):
        coe = tpl(i)  # Call the custom function
        sales_list.append(coe)
```

```
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
```

```
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
```

In [ ]:
```
sales_list = pd.DataFrame(sales_list).rename(columns={0:'Sales_coe'})
sales_list
```

Out[ ]:

	Sales_coe
0	0.478355
1	-0.453556
2	-0.081104
3	-0.204603
4	0.067263
...	...
267	-0.494257
268	0.681775
269	-0.747037
270	0.409585
271	0.011432

272 rows × 1 columns

In [ ]:
```
sales_list.query('Sales_coe >= 0.8')
```

Out[ ]:

	Sales_coe
85	1.000000
154	0.841589

In [ ]:
```
def tpl(n):
    np = df_agg_po.query('STORE_NBR == @n')[['TOT_Customers']].reset_index(drop=Tru
    r = df_agg_po_str2['TOT_Customers'].corr(np['TOT_Customers'])
    return r

customer_list = []
for i in range(1,273):
        cmr_coe = tpl(i)  # Call the custom function
        customer_list.append(cmr_coe)
```

```
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2922: RuntimeWarning: invalid value encountered in divid
e
  c /= stddev[:, None]
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2923: RuntimeWarning: invalid value encountered in divid
e
  c /= stddev[None, :]
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2922: RuntimeWarning: invalid value encountered in divid
e
  c /= stddev[:, None]
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2923: RuntimeWarning: invalid value encountered in divid
e
  c /= stddev[None, :]
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2922: RuntimeWarning: invalid value encountered in divid
e
  c /= stddev[:, None]
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2923: RuntimeWarning: invalid value encountered in divid
e
  c /= stddev[None, :]
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
```

```
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
```

In [ ]:
```
customer_list = pd.DataFrame(customer_list).rename(columns={0:'Customer_coe'})
```

In [ ]:
```
customer_list.query('Customer_coe >= 0.5')
```

Out[ ]:

	Customer_coe
0	0.559062
56	0.581417
73	0.701439
85	1.000000
98	0.640103
146	0.728606
154	0.646118
167	0.540929
175	0.710471
228	0.575004
259	0.772289

Index 154 (Store 155) is the only common indexes and has strogest positive relation

with Store 86's total sales and total customer data is moderately similar thus we will select #Store155 as the control store for trial store 86

In [ ]:
```
df_agg_po.query('STORE_NBR == 86')[['month_ID','TOT_Customers', 'TOT_SALES']]
```

Out[ ]:

	month_ID	TOT_Customers	TOT_SALES
978	201807	99	892.20
979	201808	94	764.05
980	201809	103	914.60
981	201810	109	948.40
982	201811	100	918.00
983	201812	98	841.20
984	201901	94	841.40
985	201902	107	913.20

In [ ]:
```
df_po_st2 = df_agg_po.query('STORE_NBR == 86')[['month_ID','TOT_Customers', 'TOT_SA
```

In [ ]:
```
df_agg_po.query('STORE_NBR == 155')[['month_ID','TOT_Customers', 'TOT_SALES']]
```

Out[ ]:

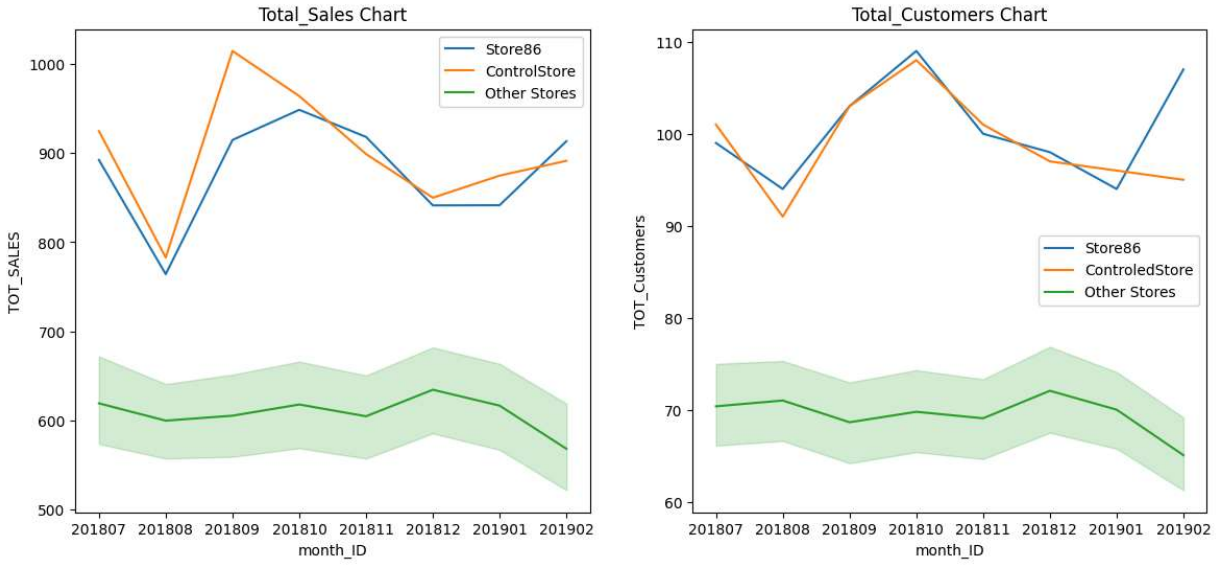
	month_ID	TOT_Customers	TOT_SALES
1794	201807	101	924.6
1795	201808	91	782.7
1796	201809	103	1014.4
1797	201810	108	963.8
1798	201811	101	898.8
1799	201812	97	849.8
1800	201901	96	874.6
1801	201902	95	891.2

In [ ]:
```
df_po_st2_c = df_agg_po.query('STORE_NBR == 155')[['month_ID','TOT_Customers', 'TOT
```

We are selecting Store 155 to be the control store for the trial store 86

In [ ]:
```
plt.figure(figsize=(14,6))
plt.subplot(1,2,1)
sns.lineplot(data=df_po_st2, x='month_ID', y='TOT_SALES', label='Store86')
sns.lineplot(data=df_po_st2_c, x='month_ID', y='TOT_SALES', label='ControlStore')
sns.lineplot(data=df_agg_po.query('STORE_NBR != [86,155]'), x='month_ID', y='TOT_SA
plt.title('Total_Sales Chart')
plt.subplot(1,2,2)
sns.lineplot(data=df_po_st2, x='month_ID', y='TOT_Customers', label='Store86')
sns.lineplot(data=df_po_st2_c, x='month_ID', y='TOT_Customers', label='ControledSto
sns.lineplot(data=df_agg_po.query('STORE_NBR != [86,155]'), x='month_ID', y='TOT_Cu
plt.title('Total_Customers Chart')
```

Out[ ]:
```
Text(0.5, 1.0, 'Total_Customers Chart')
```



Store86 Statistical analysis

In [ ]:
```
str2c_factor = df_po_st2['TOT_SALES'].sum()/df_po_st2_c['TOT_SALES'].sum()
str2c_factor
```

Out[ ]:
```
np.float64(0.9768260670287088)
```

In [ ]:
```
str2_sales = df_agg.query('STORE_NBR == 86')[['month_ID','TOT_SALES']]
str2_sales
```

Out[ ]:

	month_ID	TOT_SALES
978	201807	892.20
979	201808	764.05
980	201809	914.60
981	201810	948.40
982	201811	918.00
983	201812	841.20
984	201901	841.40
985	201902	913.20
986	201903	1026.80
987	201904	848.20
988	201905	889.30
989	201906	838.00

In [ ]:
```
str2_c_sales = df_agg.query('STORE_NBR == 155')[['month_ID','TOT_SALES']]
str2_c_sales
```

Out[ ]:

	month_ID	TOT_SALES
1794	201807	924.60
1795	201808	782.70
1796	201809	1014.40
1797	201810	963.80
1798	201811	898.80
1799	201812	849.80
1800	201901	874.60
1801	201902	891.20
1802	201903	804.40
1803	201904	844.60
1804	201905	922.85
1805	201906	857.20

In [ ]:
```
str2_c_ScaledSales = str2_c_sales['TOT_SALES'] * str2c_factor
str2_c_ScaledSales
```

Out[ ]:
```
1794    903.173382
1795    764.561763
1796    990.892362
1797    941.464963
1798    877.971269
1799    830.106792
1800    854.332078
1801    870.547391
1802    785.758888
1803    825.027296
1804    901.463936
1805    837.335305
Name: TOT_SALES, dtype: float64
```

In [ ]:
```
str2_c_sales['Scaled_Sales'] = str2_c_ScaledSales
```

In [ ]:
```
str2_sales = str2_sales.merge(str2_c_sales[['month_ID','Scaled_Sales']], on='month_
str2_sales
```

Out[ ]:

	month_ID	TOT_SALES	Scaled_Sales
0	201807	892.20	903.173382
1	201808	764.05	764.561763
2	201809	914.60	990.892362
3	201810	948.40	941.464963
4	201811	918.00	877.971269
5	201812	841.20	830.106792
6	201901	841.40	854.332078
7	201902	913.20	870.547391
8	201903	1026.80	785.758888
9	201904	848.20	825.027296
10	201905	889.30	901.463936
11	201906	838.00	837.335305

In [ ]:
```
str2_sales['%_diff'] = abs(((str2_sales['TOT_SALES'] - str2_sales['Scaled_Sales'] )
str2_sales
```

Out[ ]:

	month_ID	TOT_SALES	Scaled_Sales	%_diff
0	201807	892.20	903.173382	1.214981
1	201808	764.05	764.561763	0.066935
2	201809	914.60	990.892362	7.699359
3	201810	948.40	941.464963	0.736622
4	201811	918.00	877.971269	4.559230
5	201812	841.20	830.106792	1.336359
6	201901	841.40	854.332078	1.513706
7	201902	913.20	870.547391	4.899516
8	201903	1026.80	785.758888	30.676218
9	201904	848.20	825.027296	2.808720
10	201905	889.30	901.463936	1.349354
11	201906	838.00	837.335305	0.079382

In [ ]:
```
sd = str2_sales.query('month_ID < "201902"')['%_diff'].std()
sd
```

Out[ ]:
```
np.float64(2.716042645389702)
```

In [ ]:
```
str2_sales['t_values'] =  (str2_sales['%_diff'] - 0) / sd
str2_sales
```

Out[ ]:

	month_ID	TOT_SALES	Scaled_Sales	%_diff	t_values
0	201807	892.20	903.173382	1.214981	0.447335
1	201808	764.05	764.561763	0.066935	0.024644
2	201809	914.60	990.892362	7.699359	2.834771
3	201810	948.40	941.464963	0.736622	0.271211
4	201811	918.00	877.971269	4.559230	1.678630
5	201812	841.20	830.106792	1.336359	0.492024
6	201901	841.40	854.332078	1.513706	0.557321
7	201902	913.20	870.547391	4.899516	1.803917
8	201903	1026.80	785.758888	30.676218	11.294454
9	201904	848.20	825.027296	2.808720	1.034122
10	201905	889.30	901.463936	1.349354	0.496809
11	201906	838.00	837.335305	0.079382	0.029227

In [ ]:
```
str2_sales.query(' month_ID > "201902"')
```

Out[ ]:

	month_ID	TOT_SALES	Scaled_Sales	%_diff	t_values
8	201903	1026.8	785.758888	30.676218	11.294454
9	201904	848.2	825.027296	2.808720	1.034122
10	201905	889.3	901.463936	1.349354	0.496809
11	201906	838.0	837.335305	0.079382	0.029227

In [ ]:
```
str2_comp_df = str2_sales.merge(str2_c_sales[['month_ID','TOT_SALES']], on=['month_
.rename(columns={'TOT_SALES_x':'TOT_SALES', 'TOT_SALES_y':'TOT_SALES_C'})
```

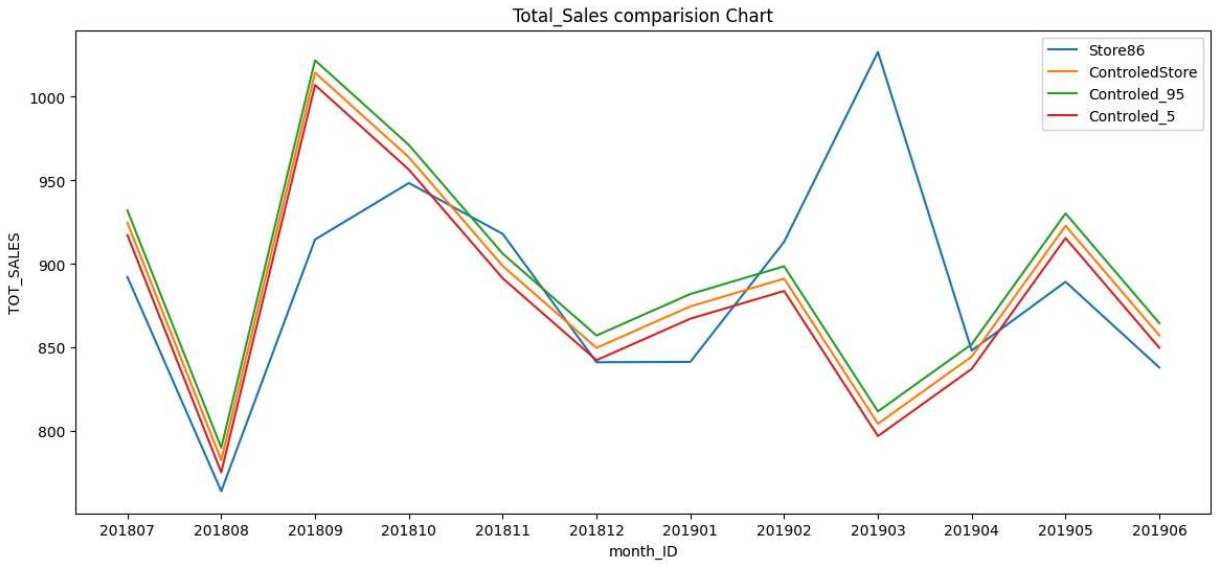
In [ ]:
```
str2_comp_df['p_95'] = str2_comp_df['TOT_SALES_C'] + pow(sd,2)
str2_comp_df['p_5'] = str2_comp_df['TOT_SALES_C'] - pow(sd,2)
str2_comp_df
```

Out[ ]:

	month_ID	TOT_SALES	Scaled_Sales	%_diff	t_values	TOT_SALES_C	p_95
0	201807	892.20	903.173382	1.214981	0.447335	924.60	931.976888
1	201808	764.05	764.561763	0.066935	0.024644	782.70	790.076888
2	201809	914.60	990.892362	7.699359	2.834771	1014.40	1021.776888
3	201810	948.40	941.464963	0.736622	0.271211	963.80	971.176888
4	201811	918.00	877.971269	4.559230	1.678630	898.80	906.176888
5	201812	841.20	830.106792	1.336359	0.492024	849.80	857.176888
6	201901	841.40	854.332078	1.513706	0.557321	874.60	881.976888
7	201902	913.20	870.547391	4.899516	1.803917	891.20	898.576888
8	201903	1026.80	785.758888	30.676218	11.294454	804.40	811.776888
9	201904	848.20	825.027296	2.808720	1.034122	844.60	851.976888
10	201905	889.30	901.463936	1.349354	0.496809	922.85	930.226888
11	201906	838.00	837.335305	0.079382	0.029227	857.20	864.576888

In [ ]:
```
plt.figure(figsize=(14,6))
sns.lineplot(data=str2_comp_df, x='month_ID', y='TOT_SALES', label='Store86')
sns.lineplot(data=str2_comp_df, x='month_ID', y='TOT_SALES_C', label='ControledStor
sns.lineplot(data=str2_comp_df, x='month_ID', y='p_95', label='Controled_95')
sns.lineplot(data=str2_comp_df, x='month_ID', y='p_5', label='Controled_5')
plt.title('Total_Sales comparision Chart')
```

Out[ ]:
```
Text(0.5, 1.0, 'Total_Sales comparision Chart')
```



In [ ]:
```
str2_c_95 = str2_comp_df.query('"201901"< month_ID < "201905"')['TOT_SALES_C'].quan
str2_c_95
```

Out[ ]: np.float64(886.5400000000001)

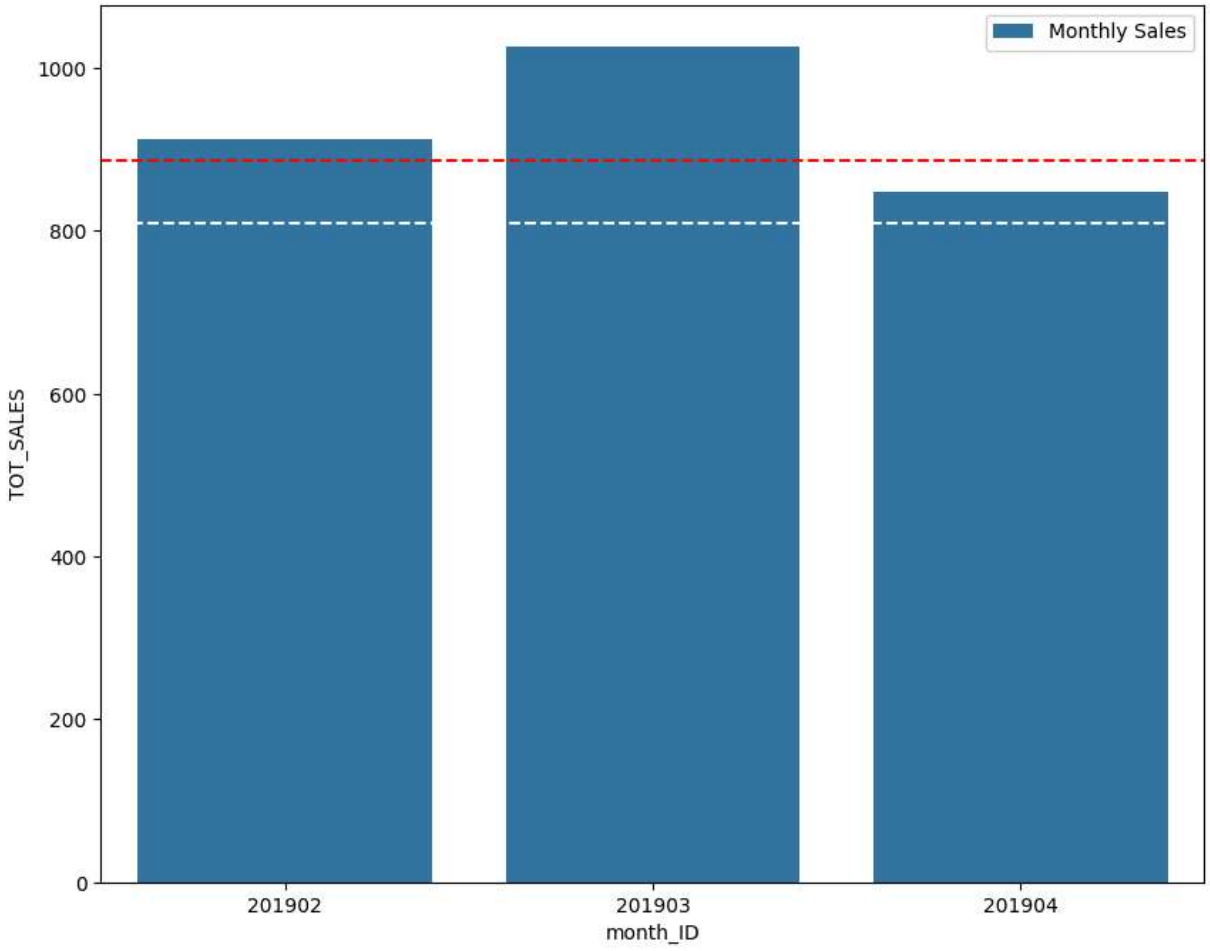
In [ ]:
```
str2_c_5 = str2_comp_df.query('"201901"<= month_ID < "201905"')['TOT_SALES_C'].quan
str2_c_5
```

Out[ ]: np.float64(810.43)

In [ ]:
```
plt.figure(figsize=(10, 8))
sns.barplot(data=str2_comp_df.query('"201902"<= month_ID <"201905"'), x='month_ID',

plt.axhline(y=str2_c_95, color='red', linestyle='--', label='95th Percentile')
plt.axhline(y=str2_c_5, color='white', linestyle='--', label='95th Percentile')
```

Out[ ]: <matplotlib.lines.Line2D at 0x28905e89bb0>



customer data analysis on Store 86

In [ ]:
```
str2c_cmr_factor = df_po_st2['TOT_Customers'].sum()/df_po_st2_c['TOT_Customers'].su
str2c_cmr_factor
```

Out[ ]: np.float64(1.0151515151515151)

In [ ]:
```
str2_customer = df_agg.query('STORE_NBR == 86')[['month_ID','TOT_Customers']]
str2_customer
```

Out[ ]:

	month_ID	TOT_Customers
978	201807	99
979	201808	94
980	201809	103
981	201810	109
982	201811	100
983	201812	98
984	201901	94
985	201902	107
986	201903	115
987	201904	105
988	201905	104
989	201906	98

In [ ]:
```
str2_c_customer = df_agg.query('STORE_NBR == 155')[['month_ID','TOT_Customers']]
str2_c_customer
```

Out[ ]:

	month_ID	TOT_Customers
1794	201807	101
1795	201808	91
1796	201809	103
1797	201810	108
1798	201811	101
1799	201812	97
1800	201901	96
1801	201902	95
1802	201903	94
1803	201904	99
1804	201905	106
1805	201906	95

In [ ]:
```
str2_comp_df = str2_customer.merge(str2_c_customer, on='month_ID', how='inner').ren
str2_comp_df
```

Out[ ]:

	month_ID	T_Customers	T_Customers_C
0	201807	99	101
1	201808	94	91
2	201809	103	103
3	201810	109	108
4	201811	100	101
5	201812	98	97
6	201901	94	96
7	201902	107	95
8	201903	115	94
9	201904	105	99
10	201905	104	106
11	201906	98	95

As the factor is 1 we will treat the control total customer as the adjusted total customer.

In [ ]:
```
str2_comp_df['%_diff'] = abs(((str2_comp_df['T_Customers'] - str2_comp_df['T_Custom
str2_comp_df
```

Out[ ]:

	month_ID	T_Customers	T_Customers_C	%_diff
0	201807	99	101	1.980198
1	201808	94	91	3.296703
2	201809	103	103	0.000000
3	201810	109	108	0.925926
4	201811	100	101	0.990099
5	201812	98	97	1.030928
6	201901	94	96	2.083333
7	201902	107	95	12.631579
8	201903	115	94	22.340426
9	201904	105	99	6.060606
10	201905	104	106	1.886792
11	201906	98	95	3.157895

In [ ]:
```
sd = str2_comp_df.query('month_ID < "201902"')['%_diff'].std()
sd
```

Out[ ]:
```
np.float64(1.0687444701395237)
```

In [ ]:
```
str2_comp_df['t_values'] =  (str2_comp_df['%_diff'] - 0) / sd
str2_comp_df
```

Out[ ]:

	month_ID	T_Customers	T_Customers_C	%_diff	t_values
0	201807	99	101	1.980198	1.852826
1	201808	94	91	3.296703	3.084651
2	201809	103	103	0.000000	0.000000
3	201810	109	108	0.925926	0.866368
4	201811	100	101	0.990099	0.926413
5	201812	98	97	1.030928	0.964616
6	201901	94	96	2.083333	1.949328
7	201902	107	95	12.631579	11.819082
8	201903	115	94	22.340426	20.903430
9	201904	105	99	6.060606	5.670772
10	201905	104	106	1.886792	1.765429
11	201906	98	95	3.157895	2.954771

In [ ]:
```
str2_comp_df.query(' month_ID > "201902"')
```

Out[ ]:

	month_ID	T_Customers	T_Customers_C	%_diff	t_values
8	201903	115	94	22.340426	20.903430
9	201904	105	99	6.060606	5.670772
10	201905	104	106	1.886792	1.765429
11	201906	98	95	3.157895	2.954771

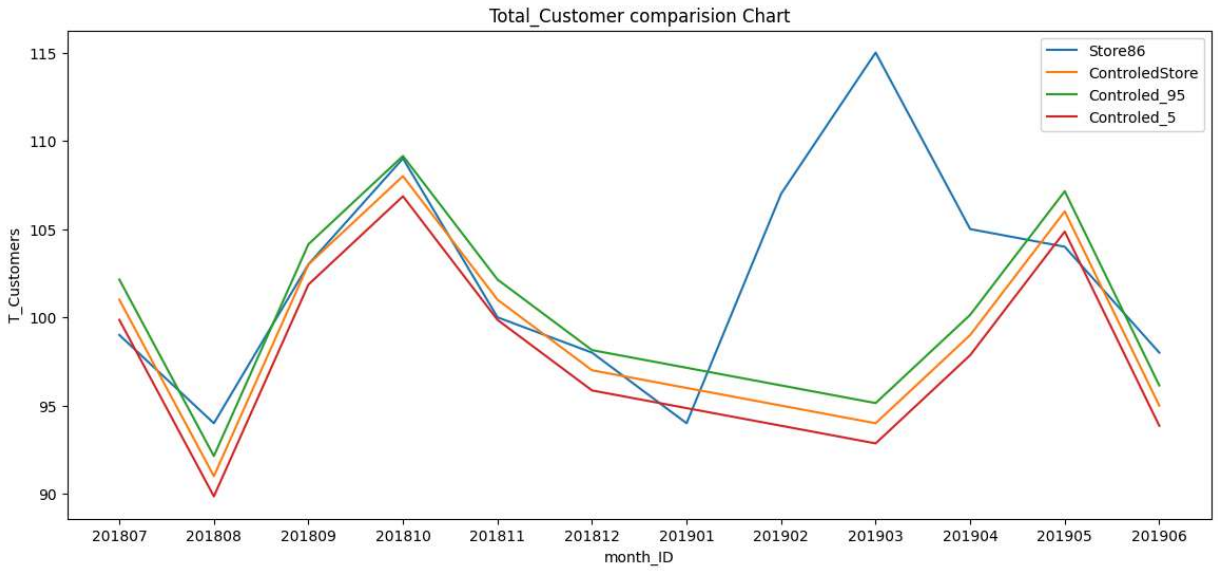
In [ ]:
```
str2_comp_df['p_95'] = str2_comp_df['T_Customers_C'] + pow(sd,2)
str2_comp_df['p_5'] = str2_comp_df['T_Customers_C'] - pow(sd,2)
str2_comp_df
```

Out[ ]:

	month_ID	T_Customers	T_Customers_C	%_diff	t_values	p_95	p_5
0	201807	99	101	1.980198	1.852826	102.142215	99.857785
1	201808	94	91	3.296703	3.084651	92.142215	89.857785
2	201809	103	103	0.000000	0.000000	104.142215	101.857785
3	201810	109	108	0.925926	0.866368	109.142215	106.857785
4	201811	100	101	0.990099	0.926413	102.142215	99.857785
5	201812	98	97	1.030928	0.964616	98.142215	95.857785
6	201901	94	96	2.083333	1.949328	97.142215	94.857785
7	201902	107	95	12.631579	11.819082	96.142215	93.857785
8	201903	115	94	22.340426	20.903430	95.142215	92.857785
9	201904	105	99	6.060606	5.670772	100.142215	97.857785
10	201905	104	106	1.886792	1.765429	107.142215	104.857785
11	201906	98	95	3.157895	2.954771	96.142215	93.857785

In [ ]:
```
plt.figure(figsize=(14,6))
sns.lineplot(data=str2_comp_df, x='month_ID', y='T_Customers', label='Store86')
sns.lineplot(data=str2_comp_df, x='month_ID', y='T_Customers_C', label='ControledSt
sns.lineplot(data=str2_comp_df, x='month_ID', y='p_95', label='Controled_95')
sns.lineplot(data=str2_comp_df, x='month_ID', y='p_5', label='Controled_5')
plt.title('Total_Customer comparision Chart')
```

Out[ ]:
```
Text(0.5, 1.0, 'Total_Customer comparision Chart')
```



In [ ]:
```
str2_c_c95 = str2_comp_df.query('"201901"< month_ID < "201905"')['T_Customers_C'].q
str2_c_c95
```

Out[ ]:
```
np.float64(98.6)
```

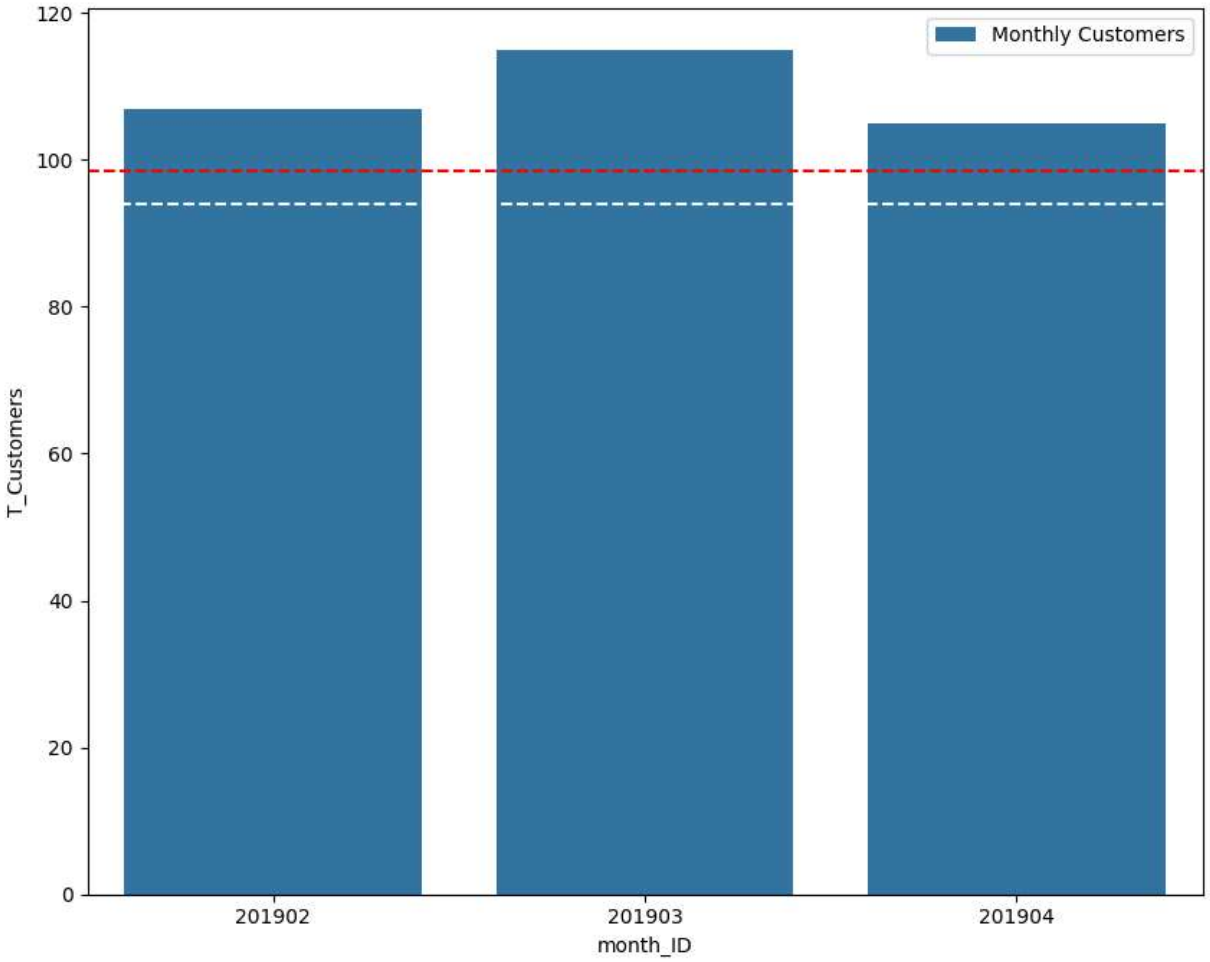
In [ ]:
```
str2_c_c5 = str2_comp_df.query('"201901"< month_ID < "201905"')['T_Customers_C'].qu
str2_c_c5
```

Out[ ]:
```
np.float64(94.1)
```

In [ ]:
```
plt.figure(figsize=(10, 8))
sns.barplot(data=str2_comp_df.query('"201901"< month_ID <"201905"'), x='month_ID',

plt.axhline(y=str2_c_c95, color='red', linestyle='--', label='95th Percentile')
plt.axhline(y=str2_c_c5, color='white', linestyle='--', label='95th Percentile')
```

Out[ ]:
```
<matplotlib.lines.Line2D at 0x28905fa86b0>
```



During the trial period we can see significant increase number of customers but the sales is almost similar to the control store.

Analysis on trial Store 88

In [ ]:
```
df_agg_po.query('STORE_NBR == [88]')
```

Out[ ]:

	STORE_NBR	month_ID	TOT_SALES	TOT_QTY	TOT_Customers	TOT_Transactions	TR
1002	88	201807	1310.0	306	129	153	
1003	88	201808	1323.8	303	131	158	
1004	88	201809	1423.0	318	124	157	
1005	88	201810	1352.4	316	123	155	
1006	88	201811	1382.8	314	130	156	
1007	88	201812	1325.2	298	126	148	
1008	88	201901	1266.4	292	117	144	
1009	88	201902	1370.2	308	124	153	

In [ ]:
```
df_agg_po_str3 = df_agg_po.query('STORE_NBR == 88')[['TOT_Customers', 'TOT_SALES']]
```

In [ ]:
```
str3_sales_mean = df_agg_po_str3[['TOT_SALES']].mean()
str3_sales_mean
```

Out[ ]:
```
TOT_SALES    1344.225
dtype: float64
```

In [ ]:
```
def tpl(n):
    np =df_agg_po.query('STORE_NBR == @n')[['TOT_SALES']].reset_index(drop=True)
    r = df_agg_po_str3['TOT_SALES'].corr(np['TOT_SALES'])
    return r

sales_list = []
for i in range(1,273):
        coe = tpl(i)  # Call the custom function
        sales_list.append(coe)
```

```
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
```

In [ ]:
```
sales_list = pd.DataFrame(sales_list).rename(columns={0:'Sales_coe'})
sales_list
```

Out[ ]:

	Sales_coe
0	0.823306
1	-0.151853
2	-0.288716
3	-0.619880
4	0.038594
...	...
267	-0.100489
268	-0.165132
269	-0.737748
270	-0.166019
271	-0.747878

272 rows × 1 columns

In [ ]:
```
sales_list.query('Sales_coe >= 0.8')
```

Out[ ]:

	Sales_coe
0	0.823306
10	1.000000
30	1.000000
87	1.000000
158	0.895637
192	1.000000

Customer Data Analysis

In [ ]:
```
def tpl(n):
    np = df_agg_po.query('STORE_NBR == @n')[['TOT_Customers']].reset_index(drop=Tru
    r = df_agg_po_str3['TOT_Customers'].corr(np['TOT_Customers'])
    return r

customer_list = []
for i in range(1,273):
        cmr_coe = tpl(i)  # Call the custom function
        customer_list.append(cmr_coe)
```

```
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2922: RuntimeWarning: invalid value encountered in divid
e
  c /= stddev[:, None]
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2923: RuntimeWarning: invalid value encountered in divid
e
  c /= stddev[None, :]
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2922: RuntimeWarning: invalid value encountered in divid
e
  c /= stddev[:, None]
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2923: RuntimeWarning: invalid value encountered in divid
e
  c /= stddev[None, :]
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2922: RuntimeWarning: invalid value encountered in divid
e
  c /= stddev[:, None]
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2923: RuntimeWarning: invalid value encountered in divid
e
  c /= stddev[None, :]
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
```

```
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2922: RuntimeWarning: invalid value encountered in divid
e
  c /= stddev[:, None]
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2923: RuntimeWarning: invalid value encountered in divid
e
  c /= stddev[None, :]
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
```

```
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divi
de
  c *= np.true_divide(1, fact)
c:\Users\KNIGHT GOKU\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy
\lib\_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multi
ply
  c *= np.true_divide(1, fact)
```

In [ ]:
```
customer_list = pd.DataFrame(customer_list).rename(columns={0:'Customer_coe'})
```

In [ ]:
```
customer_list.query('Customer_coe >= 0.9')
```

Out[ ]:

	Customer_coe
13	0.928985
34	0.901397
87	1.000000
236	0.942232

In [ ]:
```
sales_list.query('Sales_coe >= 0.8')
```

Out[ ]:

	Sales_coe
0	0.823306
10	1.000000
30	1.000000
87	1.000000
158	0.895637
192	1.000000

Here we will use the meanvalue technique for sales data of store 88

In [ ]:
```
def tpl(n):
    np = df_agg_po.query('STORE_NBR == @n')[['TOT_SALES']].mean()
    return np

sales_list = []
for i in range(1,273):
    str_mean = tpl(i)
    sales_list.append(str_mean)
```

In [ ]:
```
sales_list = pd.DataFrame(sales_list).rename(columns={'TOT_SALES':'Sales_mean'})
sales_list
```

Out[ ]:

	Sales_mean
0	201.53750
1	158.48750
2	1090.48125
3	1251.30000
4	808.33750
...	...
267	214.25625
268	952.46250
269	945.79375
270	813.91250
271	392.48125

272 rows × 1 columns

In [ ]:
```
sales_diff = []
for i in sales_list['Sales_mean']:
    mean_d = i - str3_sales_mean
    sales_diff.append(mean_d)

sales_diff = pd.DataFrame(sales_diff).rename(columns={'TOT_SALES':'Mean_diff'})
```

In [ ]:
```
sales_diff
```

Out[ ]:

	Mean_diff
0	-1142.68750
1	-1185.73750
2	-253.74375
3	-92.92500
4	-535.88750
...	...
267	-1129.96875
268	-391.76250
269	-398.43125
270	-530.31250
271	-951.74375

272 rows × 1 columns

In [ ]:
```
sales_diff.query('-10<=Mean_diff<10')
```

Out[ ]:

	Mean_diff
87	0.0
236	2.5

In [ ]:
```
customer_list.query('Customer_coe >= 0.9')
```

Out[ ]:

	Customer_coe
13	0.928985
34	0.901397
87	1.000000
236	0.942232

Index 236 (Store 237) is the only common indexes and has strogest positive relation with Store 88's total customer and total sales data thus we will select #Store237 as the control store for trial store 88

In [ ]:
```
df_agg_po.query('STORE_NBR == 88')[['month_ID','TOT_Customers', 'TOT_SALES']]
```

Out[ ]:

	month_ID	TOT_Customers	TOT_SALES
1002	201807	129	1310.0
1003	201808	131	1323.8
1004	201809	124	1423.0
1005	201810	123	1352.4
1006	201811	130	1382.8
1007	201812	126	1325.2
1008	201901	117	1266.4
1009	201902	124	1370.2

In [ ]:
```
df_po_st3 = df_agg_po.query('STORE_NBR == 88')[['month_ID','TOT_Customers', 'TOT_SA
```

In [ ]:
```
df_agg_po.query('STORE_NBR == 237')[['month_ID','TOT_Customers', 'TOT_SALES']]
```

Out[ ]:

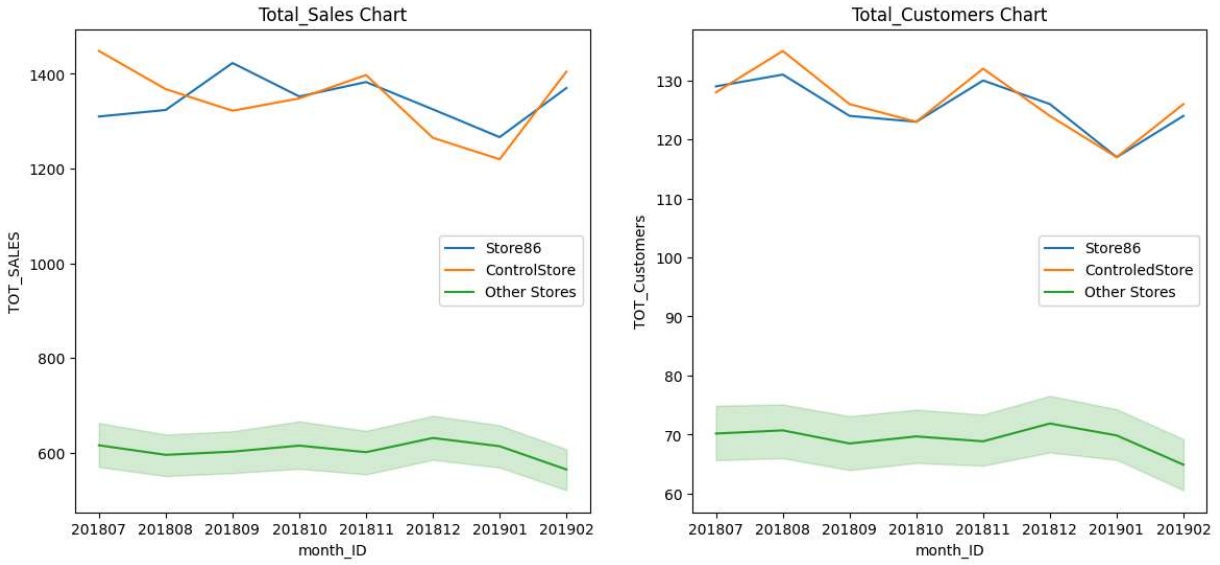
	month_ID	TOT_Customers	TOT_SALES
2748	201807	128	1448.4
2749	201808	135	1367.8
2750	201809	126	1322.2
2751	201810	123	1348.3
2752	201811	132	1397.6
2753	201812	124	1265.0
2754	201901	117	1219.7
2755	201902	126	1404.8

In [ ]:
```
df_po_st3_c = df_agg_po.query('STORE_NBR == 237')[['month_ID','TOT_Customers', 'TOT
```

We are selecting Store 237 to be the control store for the trial store 88

In [ ]:
```
plt.figure(figsize=(14,6))
plt.subplot(1,2,1)
sns.lineplot(data=df_po_st3, x='month_ID', y='TOT_SALES', label='Store86')
sns.lineplot(data=df_po_st3_c, x='month_ID', y='TOT_SALES', label='ControlStore')
sns.lineplot(data=df_agg_po.query('STORE_NBR != [88,237]'), x='month_ID', y='TOT_SA
plt.title('Total_Sales Chart')
plt.subplot(1,2,2)
sns.lineplot(data=df_po_st3, x='month_ID', y='TOT_Customers', label='Store86')
sns.lineplot(data=df_po_st3_c, x='month_ID', y='TOT_Customers', label='ControledSto
sns.lineplot(data=df_agg_po.query('STORE_NBR != [88,237]'), x='month_ID', y='TOT_Cu
plt.title('Total_Customers Chart')
```

Out[ ]:
```
Text(0.5, 1.0, 'Total_Customers Chart')
```



Store88 statistical analysis

In [ ]:
```
str3c_factor = df_po_st3['TOT_SALES'].sum()/df_po_st3_c['TOT_SALES'].sum()
str3c_factor
```

Out[ ]:
```
np.float64(0.998143644767863)
```

In [ ]:
```
str3_sales = df_agg.query('STORE_NBR == 88')[['month_ID','TOT_SALES']]
str3_sales
```

Out[ ]:

	month_ID	TOT_SALES
1002	201807	1310.00
1003	201808	1323.80
1004	201809	1423.00
1005	201810	1352.40
1006	201811	1382.80
1007	201812	1325.20
1008	201901	1266.40
1009	201902	1370.20
1010	201903	1477.20
1011	201904	1439.40
1012	201905	1308.25
1013	201906	1354.60

In [ ]:
```
str3_c_sales = df_agg.query('STORE_NBR == 237')[['month_ID','TOT_SALES']]
str3_c_sales
```

Out[ ]:

	month_ID	TOT_SALES
2748	201807	1448.4
2749	201808	1367.8
2750	201809	1322.2
2751	201810	1348.3
2752	201811	1397.6
2753	201812	1265.0
2754	201901	1219.7
2755	201902	1404.8
2756	201903	1208.2
2757	201904	1204.6
2758	201905	1199.3
2759	201906	1153.6

In [ ]:
```
str3_sales.merge(str3_c_sales, on='month_ID', how='inner').rename(columns={'TOT_SAL
```

Out[ ]:

	month_ID	TOT_SALES	TOT_SALES_C
0	201807	1310.00	1448.4
1	201808	1323.80	1367.8
2	201809	1423.00	1322.2
3	201810	1352.40	1348.3
4	201811	1382.80	1397.6
5	201812	1325.20	1265.0
6	201901	1266.40	1219.7
7	201902	1370.20	1404.8
8	201903	1477.20	1208.2
9	201904	1439.40	1204.6
10	201905	1308.25	1199.3
11	201906	1354.60	1153.6

In [ ]:
```
str3_c_ScaledSales = str3_c_sales['TOT_SALES'] * str3c_factor
str3_c_ScaledSales
```

Out[ ]:
```
2748    1445.711255
2749    1365.260877
2750    1319.745527
2751    1345.797076
2752    1395.005558
2753    1262.651711
2754    1217.435804
2755    1402.192192
2756    1205.957152
2757    1202.363834
2758    1197.073673
2759    1151.458509
Name: TOT_SALES, dtype: float64
```

In [ ]:
```
str3_c_sales['Scaled_Sales'] = str3_c_ScaledSales
```

In [ ]:
```
str3_sales = str3_sales.merge(str3_c_sales, on='month_ID', how='inner').rename(colu
str3_sales
```

Out[ ]:

	month_ID	TOT_SALES	TOT_SALES_C	Scaled_Sales
0	201807	1310.00	1448.4	1445.711255
1	201808	1323.80	1367.8	1365.260877
2	201809	1423.00	1322.2	1319.745527
3	201810	1352.40	1348.3	1345.797076
4	201811	1382.80	1397.6	1395.005558
5	201812	1325.20	1265.0	1262.651711
6	201901	1266.40	1219.7	1217.435804
7	201902	1370.20	1404.8	1402.192192
8	201903	1477.20	1208.2	1205.957152
9	201904	1439.40	1204.6	1202.363834
10	201905	1308.25	1199.3	1197.073673
11	201906	1354.60	1153.6	1151.458509

In [ ]:
```
str3_sales['%_diff'] = abs(((str3_sales['TOT_SALES'] - str3_sales['Scaled_Sales'] )
str3_sales
```

Out[ ]:

	month_ID	TOT_SALES	TOT_SALES_C	Scaled_Sales	%_diff
0	201807	1310.00	1448.4	1445.711255	9.387162
1	201808	1323.80	1367.8	1365.260877	3.036847
2	201809	1423.00	1322.2	1319.745527	7.823817
3	201810	1352.40	1348.3	1345.797076	0.490633
4	201811	1382.80	1397.6	1395.005558	0.874947
5	201812	1325.20	1265.0	1262.651711	4.953725
6	201901	1266.40	1219.7	1217.435804	4.021912
7	201902	1370.20	1404.8	1402.192192	2.281584
8	201903	1477.20	1208.2	1205.957152	22.491914
9	201904	1439.40	1204.6	1202.363834	19.714180
10	201905	1308.25	1199.3	1197.073673	9.287342
11	201906	1354.60	1153.6	1151.458509	17.642103

In [ ]:
```
sd = str3_sales.query('month_ID < "201902"')['%_diff'].std()
sd
```

Out[ ]: np.float64(3.3317874687050675)

In [ ]:
```
str3_sales['t_values'] =  (str3_sales['%_diff'] - 0) / sd
str3_sales
```

Out[ ]:

	month_ID	TOT_SALES	TOT_SALES_C	Scaled_Sales	%_diff	t_values
0	201807	1310.00	1448.4	1445.711255	9.387162	2.817455
1	201808	1323.80	1367.8	1365.260877	3.036847	0.911477
2	201809	1423.00	1322.2	1319.745527	7.823817	2.348234
3	201810	1352.40	1348.3	1345.797076	0.490633	0.147258
4	201811	1382.80	1397.6	1395.005558	0.874947	0.262606
5	201812	1325.20	1265.0	1262.651711	4.953725	1.486807
6	201901	1266.40	1219.7	1217.435804	4.021912	1.207133
7	201902	1370.20	1404.8	1402.192192	2.281584	0.684793
8	201903	1477.20	1208.2	1205.957152	22.491914	6.750705
9	201904	1439.40	1204.6	1202.363834	19.714180	5.916998
10	201905	1308.25	1199.3	1197.073673	9.287342	2.787495
11	201906	1354.60	1153.6	1151.458509	17.642103	5.295086

In [ ]:
```
str3_sales.query(' month_ID > "201902"')
```

Out[ ]:

	month_ID	TOT_SALES	TOT_SALES_C	Scaled_Sales	%_diff	t_values
8	201903	1477.20	1208.2	1205.957152	22.491914	6.750705
9	201904	1439.40	1204.6	1202.363834	19.714180	5.916998
10	201905	1308.25	1199.3	1197.073673	9.287342	2.787495
11	201906	1354.60	1153.6	1151.458509	17.642103	5.295086

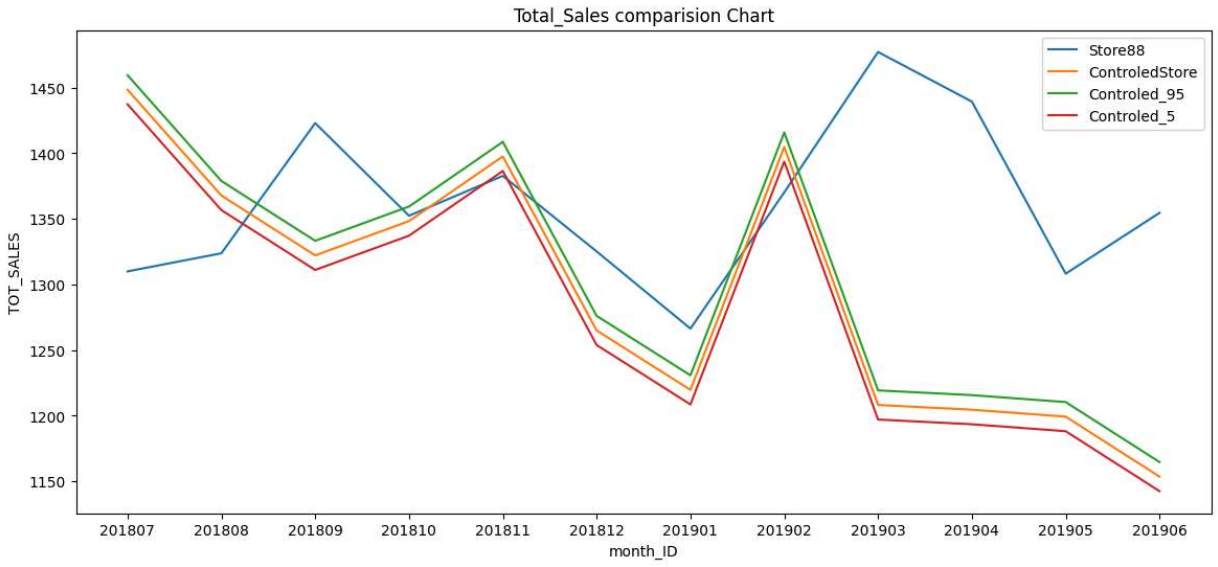
In [ ]:
```
str3_sales['p_95'] = str3_sales['TOT_SALES_C'] + pow(sd,2)
str3_sales['p_5'] = str3_sales['TOT_SALES_C'] - pow(sd,2)
str3_sales
```

Out[ ]:

	month_ID	TOT_SALES	TOT_SALES_C	Scaled_Sales	%_diff	t_values	p_95	
0	201807	1310.00	1448.4	1445.711255	9.387162	2.817455	1459.500808	1
1	201808	1323.80	1367.8	1365.260877	3.036847	0.911477	1378.900808	1
2	201809	1423.00	1322.2	1319.745527	7.823817	2.348234	1333.300808	1
3	201810	1352.40	1348.3	1345.797076	0.490633	0.147258	1359.400808	1
4	201811	1382.80	1397.6	1395.005558	0.874947	0.262606	1408.700808	1
5	201812	1325.20	1265.0	1262.651711	4.953725	1.486807	1276.100808	1
6	201901	1266.40	1219.7	1217.435804	4.021912	1.207133	1230.800808	1
7	201902	1370.20	1404.8	1402.192192	2.281584	0.684793	1415.900808	1
8	201903	1477.20	1208.2	1205.957152	22.491914	6.750705	1219.300808	1
9	201904	1439.40	1204.6	1202.363834	19.714180	5.916998	1215.700808	1
10	201905	1308.25	1199.3	1197.073673	9.287342	2.787495	1210.400808	1
11	201906	1354.60	1153.6	1151.458509	17.642103	5.295086	1164.700808	1

In [ ]:
```
plt.figure(figsize=(14,6))
sns.lineplot(data=str3_sales, x='month_ID', y='TOT_SALES', label='Store88')
sns.lineplot(data=str3_sales, x='month_ID', y='TOT_SALES_C', label='ControledStore'
sns.lineplot(data=str3_sales, x='month_ID', y='p_95', label='Controled_95')
sns.lineplot(data=str3_sales, x='month_ID', y='p_5', label='Controled_5')
plt.title('Total_Sales comparision Chart')
```

Out[ ]:
```
Text(0.5, 1.0, 'Total_Sales comparision Chart')
```



In [ ]:
```
str3_c_95 = str3_sales.query('"201902"<= month_ID < "201905"')['TOT_SALES_C'].quant
str3_c_95
```

Out[ ]: np.float64(1385.1399999999999)

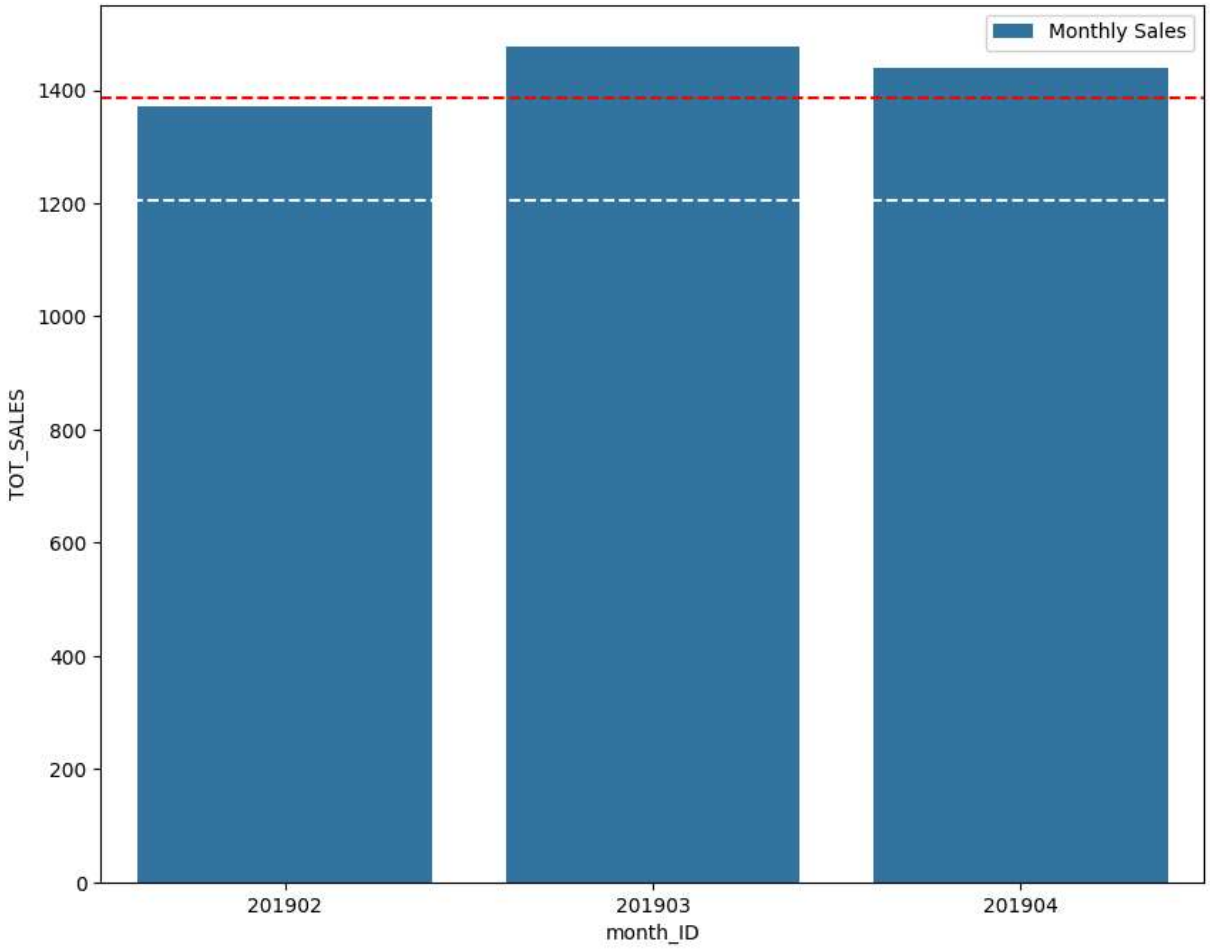
In [ ]:
```
str3_c_5 = str3_sales.query('"201902"<= month_ID < "201905"')['TOT_SALES_C'].quanti
str3_c_5
```

Out[ ]: np.float64(1204.96)

In [ ]:
```
plt.figure(figsize=(10, 8))
sns.barplot(data=str3_sales.query('"201902"<= month_ID <"201905"'), x='month_ID', y

plt.axhline(y=str3_c_95, color='red', linestyle='--', label='95th Percentile')
plt.axhline(y=str3_c_5, color='white', linestyle='--', label='95th Percentile')
```

Out[ ]: <matplotlib.lines.Line2D at 0x28908a3ade0>



Customer data analysis for Store 88

In [ ]:
```
str3c_cmr_factor = df_po_st3['TOT_Customers'].sum()/df_po_st3_c['TOT_Customers'].su
str3c_cmr_factor
```

Out[ ]: np.float64(0.9930761622156281)

In [ ]:
```
str3_customer = df_agg.query('STORE_NBR == 88')[['month_ID','TOT_Customers']]
str3_customer
```

Out[ ]:

	month_ID	TOT_Customers
1002	201807	129
1003	201808	131
1004	201809	124
1005	201810	123
1006	201811	130
1007	201812	126
1008	201901	117
1009	201902	124
1010	201903	134
1011	201904	128
1012	201905	128
1013	201906	121

In [ ]:
```
str3_c_customer = df_agg.query('STORE_NBR == 237')[['month_ID','TOT_Customers']]
str3_c_customer
```

Out[ ]:

	month_ID	TOT_Customers
2748	201807	128
2749	201808	135
2750	201809	126
2751	201810	123
2752	201811	132
2753	201812	124
2754	201901	117
2755	201902	126
2756	201903	119
2757	201904	120
2758	201905	129
2759	201906	119

In [ ]:
```
str3_customer = str3_customer.merge(str3_c_customer, on='month_ID', how='inner').re
str3_customer
```

Out[ ]:

	month_ID	T_Customers	T_Customers_C
0	201807	129	128
1	201808	131	135
2	201809	124	126
3	201810	123	123
4	201811	130	132
5	201812	126	124
6	201901	117	117
7	201902	124	126
8	201903	134	119
9	201904	128	120
10	201905	128	129
11	201906	121	119

In [ ]:
```
str3_customer['Scaled_Customers'] = str3_customer['T_Customers_C'] * str3c_factor
str3_customer
```

Out[ ]:

	month_ID	T_Customers	T_Customers_C	Scaled_Customers
0	201807	129	128	127.762387
1	201808	131	135	134.749392
2	201809	124	126	125.766099
3	201810	123	123	122.771668
4	201811	130	132	131.754961
5	201812	126	124	123.769812
6	201901	117	117	116.782806
7	201902	124	126	125.766099
8	201903	134	119	118.779094
9	201904	128	120	119.777237
10	201905	128	129	128.760530
11	201906	121	119	118.779094

In [ ]:
```
str3_customer['%_diff'] = abs(((str3_customer['T_Customers'] - str3_customer['Scale
str3_customer
```

Out[ ]:

	month_ID	T_Customers	T_Customers_C	Scaled_Customers	%_diff
0	201807	129	128	127.762387	0.968684
1	201808	131	135	134.749392	2.782493
2	201809	124	126	125.766099	1.404273
3	201810	123	123	122.771668	0.185981
4	201811	130	132	131.754961	1.331989
5	201812	126	124	123.769812	1.801884
6	201901	117	117	116.782806	0.185981
7	201902	124	126	125.766099	1.404273
8	201903	134	119	118.779094	12.814466
9	201904	128	120	119.777237	6.865046
10	201905	128	129	128.760530	0.590655
11	201906	121	119	118.779094	1.869779

In [ ]:
```
sd = str3_customer.query('month_ID < "201902"')['%_diff'].std()
sd
```

Out[ ]:
```
np.float64(0.9152340453291177)
```

In [ ]:
```
str3_customer['t_values'] =  (str3_customer['%_diff'] - 0) / sd
str3_customer
```

Out[ ]:

	month_ID	T_Customers	T_Customers_C	Scaled_Customers	%_diff	t_values
0	201807	129	128	127.762387	0.968684	1.058400
1	201808	131	135	134.749392	2.782493	3.040198
2	201809	124	126	125.766099	1.404273	1.534332
3	201810	123	123	122.771668	0.185981	0.203206
4	201811	130	132	131.754961	1.331989	1.455353
5	201812	126	124	123.769812	1.801884	1.968768
6	201901	117	117	116.782806	0.185981	0.203206
7	201902	124	126	125.766099	1.404273	1.534332
8	201903	134	119	118.779094	12.814466	14.001299
9	201904	128	120	119.777237	6.865046	7.500864
10	201905	128	129	128.760530	0.590655	0.645359
11	201906	121	119	118.779094	1.869779	2.042951

In [ ]:
```
str3_customer.query(' month_ID > "201902"')
```

Out[ ]:

	month_ID	T_Customers	T_Customers_C	Scaled_Customers	%_diff	t_values
8	201903	134	119	118.779094	12.814466	14.001299
9	201904	128	120	119.777237	6.865046	7.500864
10	201905	128	129	128.760530	0.590655	0.645359
11	201906	121	119	118.779094	1.869779	2.042951

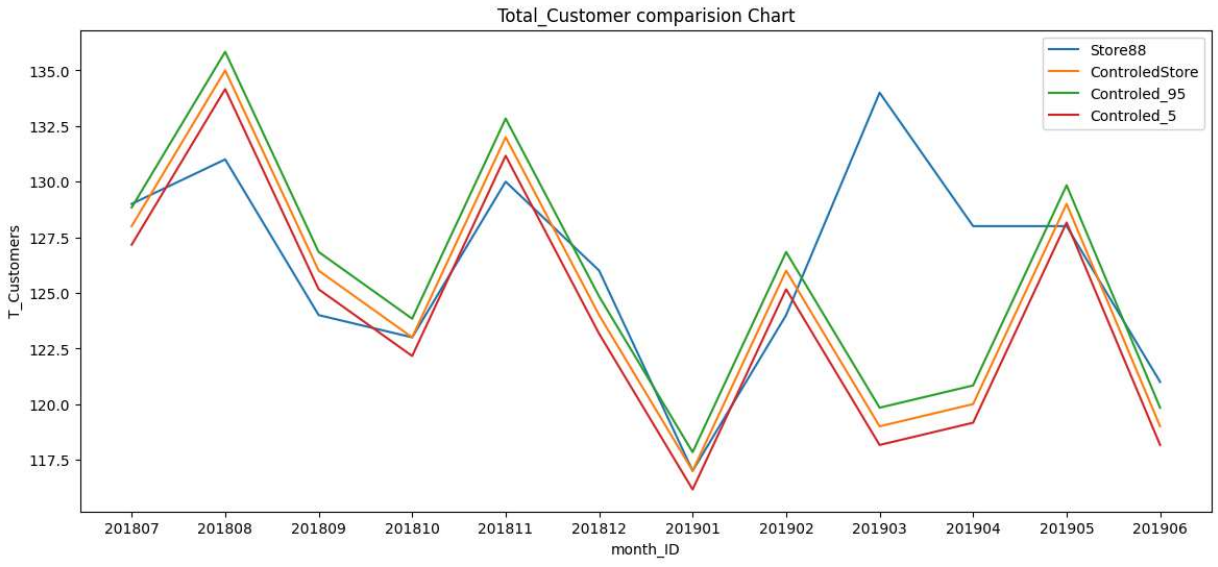
In [ ]:
```
str3_customer['p_95'] = str3_customer['T_Customers_C'] + pow(sd,2)
str3_customer['p_5'] = str3_customer['T_Customers_C'] - pow(sd,2)
str3_customer
```

Out[ ]:

	month_ID	T_Customers	T_Customers_C	Scaled_Customers	%_diff	t_values	
0	201807	129	128	127.762387	0.968684	1.058400	128.8
1	201808	131	135	134.749392	2.782493	3.040198	135.8
2	201809	124	126	125.766099	1.404273	1.534332	126.8
3	201810	123	123	122.771668	0.185981	0.203206	123.8
4	201811	130	132	131.754961	1.331989	1.455353	132.8
5	201812	126	124	123.769812	1.801884	1.968768	124.8
6	201901	117	117	116.782806	0.185981	0.203206	117.8
7	201902	124	126	125.766099	1.404273	1.534332	126.8
8	201903	134	119	118.779094	12.814466	14.001299	119.8
9	201904	128	120	119.777237	6.865046	7.500864	120.8
10	201905	128	129	128.760530	0.590655	0.645359	129.8
11	201906	121	119	118.779094	1.869779	2.042951	119.8

In [ ]:
```
plt.figure(figsize=(14,6))
sns.lineplot(data=str3_customer, x='month_ID', y='T_Customers', label='Store88')
sns.lineplot(data=str3_customer, x='month_ID', y='T_Customers_C', label='ControledS
sns.lineplot(data=str3_customer, x='month_ID', y='p_95', label='Controled_95')
sns.lineplot(data=str3_customer, x='month_ID', y='p_5', label='Controled_5')
plt.title('Total_Customer comparision Chart')
```

Out[ ]:
```
Text(0.5, 1.0, 'Total_Customer comparision Chart')
```



In [ ]:
```
str3_c_c95 = str3_customer.query('"201902"<= month_ID < "201905"')['T_Customers_C']
str3_c_c95
```

Out[ ]: np.float64(125.4)

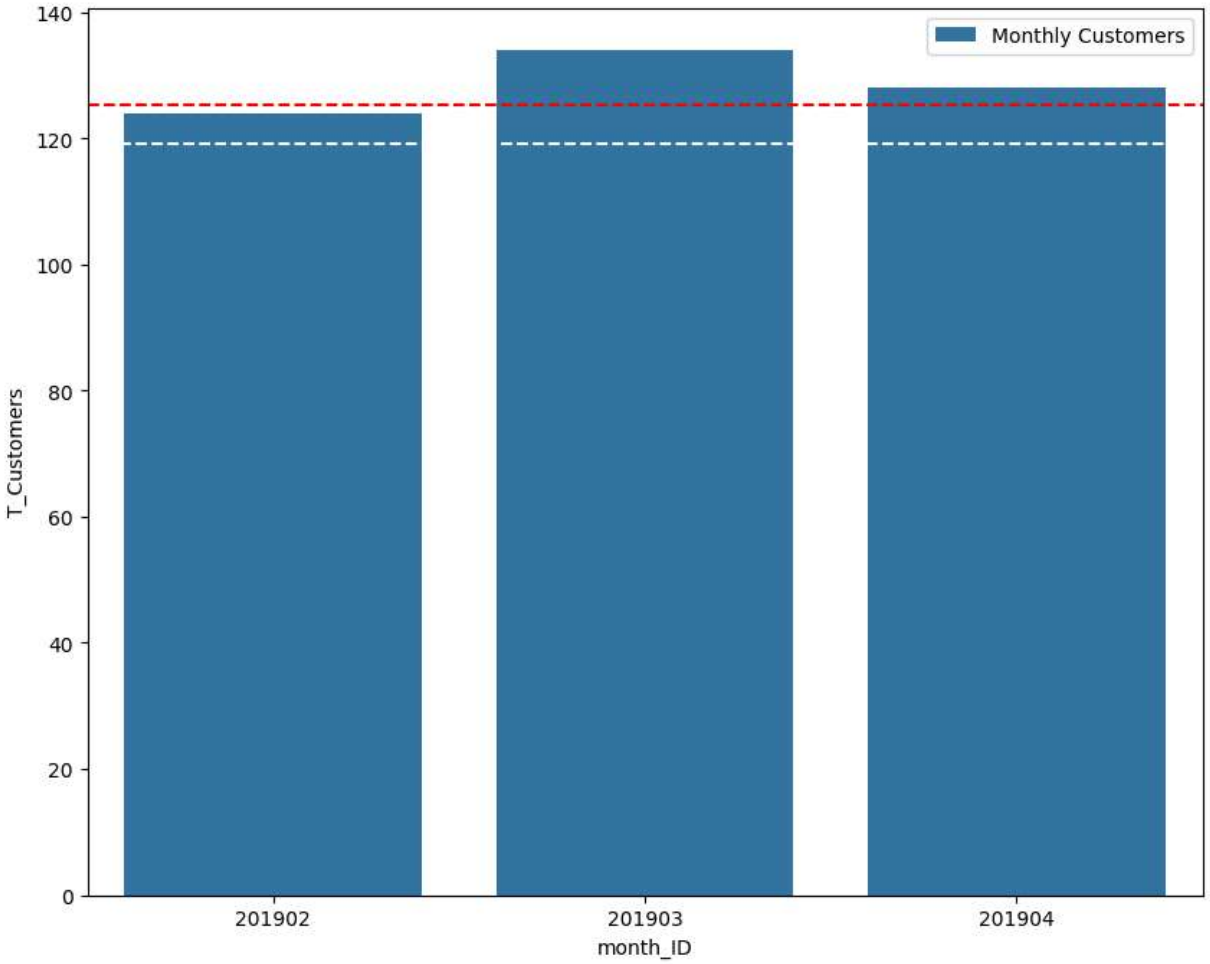
In [ ]:
```
str3_c_c5 = str3_customer.query('"201902"<= month_ID < "201905"')['T_Customers_C'].
str3_c_c5
```

Out[ ]: np.float64(119.1)

In [ ]:
```
plt.figure(figsize=(10, 8))
sns.barplot(data=str3_customer.query('"201902"<= month_ID <"201905"'), x='month_ID'

plt.axhline(y=str3_c_c95, color='red', linestyle='--', label='95th Percentile')
plt.axhline(y=str3_c_c5, color='white', linestyle='--', label='95th Percentile')
```

Out[ ]: <matplotlib.lines.Line2D at 0x289090732c0>



During the trial period we can see significant increase number of customers and same for the total sales of the store compare to the control store.

The trial was a successfull one for Store 88

Conclusion

We've found control stores 233, 155, 237 for trial stores 77, 86 and 88 respectively. The results for trial stores 77 and 88 during the trial period show a significant difference in at least two of the three trial months but this is not the case for trial store 86. We can check with the client if the implementation of the trial was different in trial store 86 but overall, the trial shows a significant increase in sales. Now that we have finished our analysis, we can prepare our presentation to the Category Manager.