QUANTIUM FORAGE PROJECT: TASK 2

The objective of task 2 is to examine the performance in trial vs control stores to provide a recommendation for each location bases on our insight.

Imagine a company wants to test a new discount offer in some stores to see if it increases sales.

To do this, they divide stores into two groups:

Trial Stores → These stores get the new discount (or any other change).

Control Stores → These stores stay the same, with no changes.

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
```

In [3]:
```
chips_final = pd.read_csv('chips_final.csv')
chips_final.head()
```

Out[3]:

	Unnamed: 0	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	Weight	Weight in g	PROD_BRAND	LIFESTAGE	PREMIUM_CUSTOMER
0	0	2018-10-17	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0	175g	175	Natural	YOUNG SINGLES/COUPLES	Premium
1	1	2019-05-14	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3	175g	175	CCs	MIDAGE SINGLES/COUPLES	Budget
2	2	2019-05-20	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9	170g	170	Smiths	MIDAGE SINGLES/COUPLES	Budget
3	3	2018-08-17	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	15.0	175g	175	Smiths	MIDAGE SINGLES/COUPLES	Budget
4	4	2018-08-18	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	13.8	150g	150	Kettle	MIDAGE SINGLES/COUPLES	Budget

In [4]:
```
chips_final.drop('Unnamed: 0', axis=1, inplace=True)
```

In [5]:
```
chips_final.head(2)
```

Out[5]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	Weight	Weight in g	PROD_BRAND	LIFESTAGE	PREMIUM_CUSTOMER
0	2018-10-17	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0	175g	175	Natural	YOUNG SINGLES/COUPLES	Premium
1	2019-05-14	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3	175g	175	CCs	MIDAGE SINGLES/COUPLES	Budget

In [6]:
```
chips_final.dtypes
```

Out[6]:
```
DATE                object
STORE_NBR            int64
LYLTY_CARD_NBR       int64
TXN_ID               int64
PROD_NBR             int64
PROD_NAME           object
PROD_QTY             int64
TOT_SALES          float64
Weight              object
Weight in g          int64
PROD_BRAND          object
LIFESTAGE           object
PREMIUM_CUSTOMER    object
dtype: object
```

In [7]:
```
chips_final['DATE'] = pd.to_datetime(chips_final['DATE'])
```

In [8]:
```
chips_final['DATE'].head()
```

Out[8]:
```
0   2018-10-17
1   2019-05-14
2   2019-05-20
3   2018-08-17
4   2018-08-18
Name: DATE, dtype: datetime64[ns]
```

In [9]:
```
# Create a month and year column i.e with no date

chips_final['MONTH_YEAR'] = chips_final['DATE'].dt.strftime("%m/%Y")
```

In [10]:
```
chips_final['MONTH_YEAR'].head()
```

Out[10]:
```
0    10/2018
1    05/2019
2    05/2019
3    08/2018
4    08/2018
Name: MONTH_YEAR, dtype: object
```

In [11]:
```
sorted(chips_final['MONTH_YEAR'].unique())
```

Out[11]:
```
['01/2019',
 '02/2019',
 '03/2019',
 '04/2019',
 '05/2019',
 '06/2019',
 '07/2018',
 '08/2018',
 '09/2018',
 '10/2018',
 '11/2018',
 '12/2018']
```

In [12]:
```
chips_final['MONTH_YEAR'].value_counts().reset_index()
# chips_monthly_count
```

Out[12]:

	MONTH_YEAR	count
0	12/2018	21106
1	03/2019	21013
2	07/2018	20889
3	08/2018	20745
4	05/2019	20715
5	10/2018	20660
6	01/2019	20490
7	06/2019	20242
8	11/2018	20211
9	09/2018	20182
10	04/2019	20154
11	02/2019	18895

In [13]:
```
chips_final.dtypes
```

Out[13]:
```
DATE                datetime64[ns]
STORE_NBR                    int64
LYLTY_CARD_NBR               int64
TXN_ID                       int64
PROD_NBR                     int64
PROD_NAME                   object
PROD_QTY                     int64
TOT_SALES                  float64
Weight                      object
Weight in g                  int64
PROD_BRAND                  object
LIFESTAGE                   object
PREMIUM_CUSTOMER            object
MONTH_YEAR                  object
dtype: object
```

In [14]:
```
chips_final['MONTH_YEAR'] = pd.to_datetime(chips_final['MONTH_YEAR'], format='%m/%Y')
```

In [15]:
```
chips_before = chips_final[(chips_final['MONTH_YEAR'] >= '07/2018') & (chips_final['MONTH_YEAR'] <= '01/2019')]
chips_before['MONTH_YEAR'].value_counts().reset_index()
```

Out[15]:

	MONTH_YEAR	count
0	2018-12-01	21106
1	2018-07-01	20889
2	2018-08-01	20745
3	2018-10-01	20660
4	2019-01-01	20490
5	2018-11-01	20211
6	2018-09-01	20182

In [16]:
```
# Date wise total sales

chips_grp_before = chips_before.groupby(['STORE_NBR', 'MONTH_YEAR'])['TOT_SALES'].sum()
chips_grp_before
```

Out[16]:
```
STORE_NBR  MONTH_YEAR
1          2018-07-01    188.9
           2018-08-01    168.4
           2018-09-01    268.1
           2018-10-01    175.4
           2018-11-01    184.8
                         ...
272        2018-09-01    294.5
           2018-10-01    405.1
           2018-11-01    355.8
           2018-12-01    363.1
           2019-01-01    392.4
Name: TOT_SALES, Length: 1846, dtype: float64
```

In [17]:
```
total_sales = chips_before.groupby('STORE_NBR')['TOT_SALES'].sum()
total_sales
```

Out[17]:
```
STORE_NBR
1      1290.70
2      1044.00
3      7184.45
4      8576.20
5      5261.90
        ...
268    1416.15
269    6212.30
270    6173.35
271    5255.30
272    2530.15
Name: TOT_SALES, Length: 270, dtype: float64
```

TRIAL STORES INCLUDE 77, 86, 88 (Given in email)

In [18]:
```
# Total Sales for trial stores from July 2018 to January 2019

total_sales.loc[[77,86,88]]
```

Out[18]:
```
STORE_NBR
77    1587.70
86    5759.25
88    8832.80
Name: TOT_SALES, dtype: float64
```

CONTROL STORES (Find out)

Use 2 methods

- Find stores with similar total sales
- Pearson Correlation Test to check if sales trend is similar

In [19]:
```
#Find out loc range for store 77

total_sales_sorted = total_sales.sort_values(ascending=True)
range_77 = total_sales_sorted.index.get_loc(77)
range_77
```

Out[19]:
```
68
```

So use Range as 63:73

In [20]:
```
# Finding match with store number 77

total_sorted_77 = total_sales_sorted.iloc[63:73].reset_index()
total_sorted_77
```

Out[20]:

	STORE_NBR	TOT_SALES
0	6	1489.50
1	188	1497.00
2	195	1500.25
3	233	1511.10
4	46	1577.60
5	77	1587.70
6	90	1611.30
7	187	1625.70
8	220	1635.10
9	205	1692.50

In [21]:
```
control_store_one = total_sorted_77['STORE_NBR'].sort_values(ascending=True).tolist()
control_store_one
```

Out[21]:
```
[6, 46, 77, 90, 187, 188, 195, 205, 220, 233]
```

In [22]:
```
control_one = pd.DataFrame({'Value': chips_grp_before[control_store_one]})
print(control_one)
```

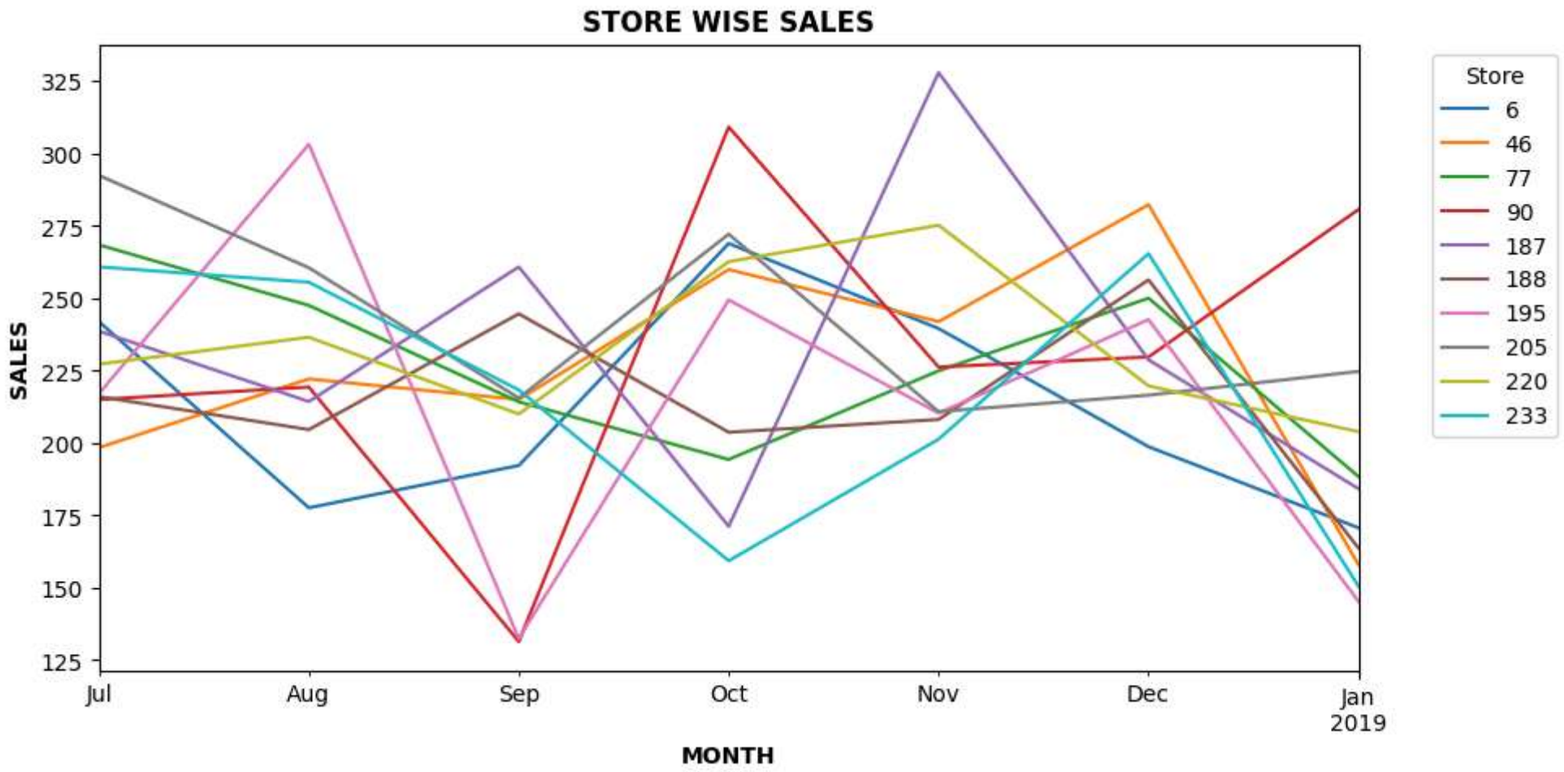
```
                     Value
STORE_NBR MONTH_YEAR
6         2018-07-01  241.9
          2018-08-01  177.6
          2018-09-01  192.2
          2018-10-01  268.9
          2018-11-01  239.5
...                     ...
233       2018-09-01  218.3
          2018-10-01  159.3
          2018-11-01  201.3
          2018-12-01  265.4
          2019-01-01  150.5

[70 rows x 1 columns]
```

In [23]:
```
# Display above informationm in pivot chart format.
# MOnthly total sales for each store

pivot_chips1 = control_one.pivot_table(index='MONTH_YEAR', columns='STORE_NBR', values='Value')
pivot_chips1
```

Out[23]:

STORE_NBR	6	46	77	90	187	188	195	205	220	233
MONTH_YEAR										
2018-07-01	241.9	198.4	268.4	215.0	238.6	216.0	217.00	292.4	227.2	260.8
2018-08-01	177.6	222.1	247.5	219.3	214.3	204.6	303.15	260.5	236.5	255.5
2018-09-01	192.2	215.0	214.2	131.4	260.8	244.6	132.50	215.5	210.0	218.3
2018-10-01	268.9	259.9	194.3	309.1	171.2	203.7	249.40	272.1	262.6	159.3
2018-11-01	239.5	241.9	224.9	226.2	327.9	208.1	210.30	210.8	275.2	201.3
2018-12-01	198.7	282.3	250.0	229.7	228.7	256.3	242.60	216.5	219.7	265.4
2019-01-01	170.7	158.0	188.4	280.6	184.2	163.7	145.30	224.7	203.9	150.5

In [24]:
```
pivot_chips1.plot(figsize=(10,5))

plt.title('STORE WISE SALES', fontweight = 'bold')
plt.xlabel('MONTH', fontweight = 'bold')
plt.ylabel('SALES', fontweight = 'bold')
plt.legend(title="Store", bbox_to_anchor=(1.05, 1), loc='upper left')

plt.show()
```



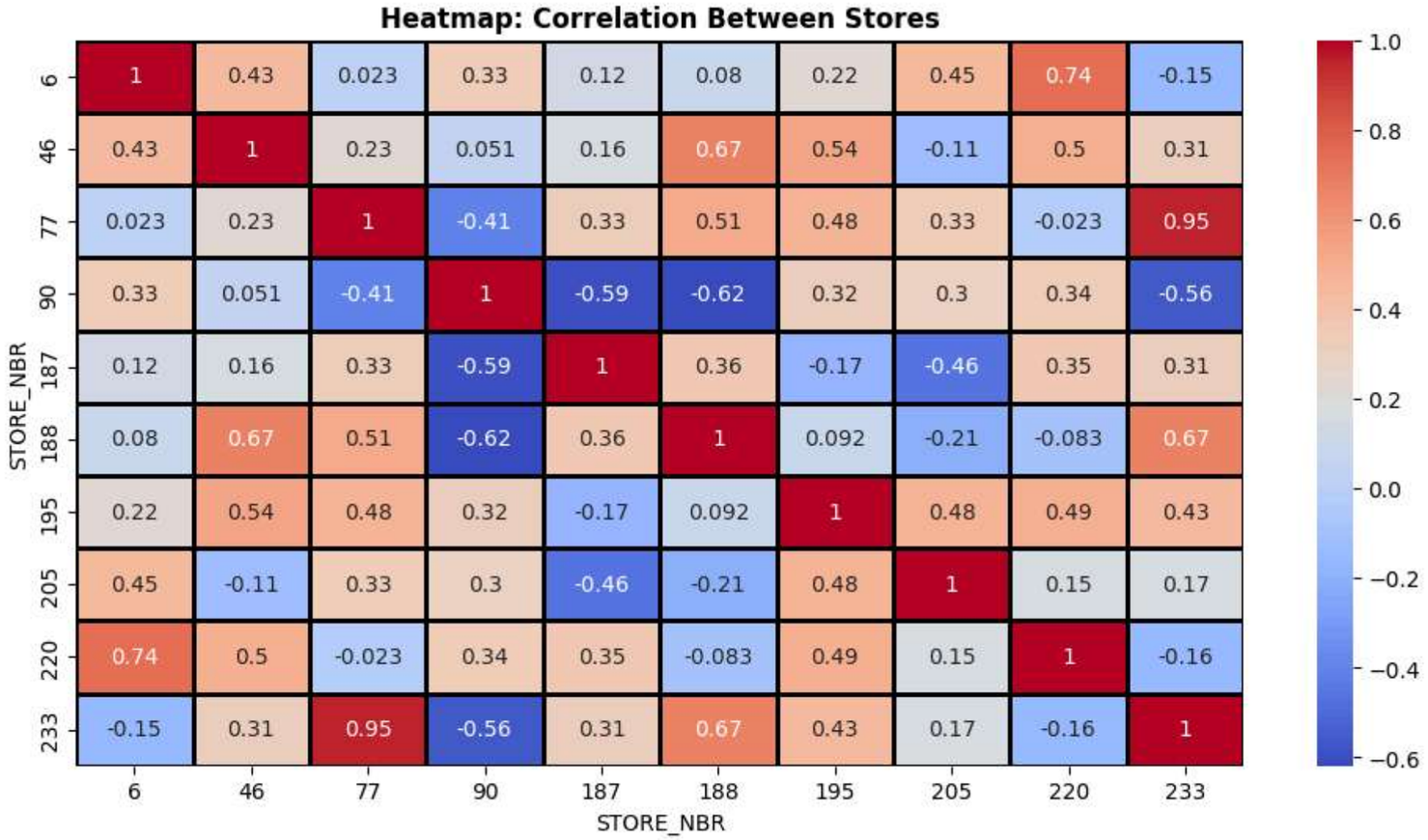
In [25]:
```
correlation_77 = pivot_chips1.corr(method='pearson')
correlation_77
```

Out[25]:

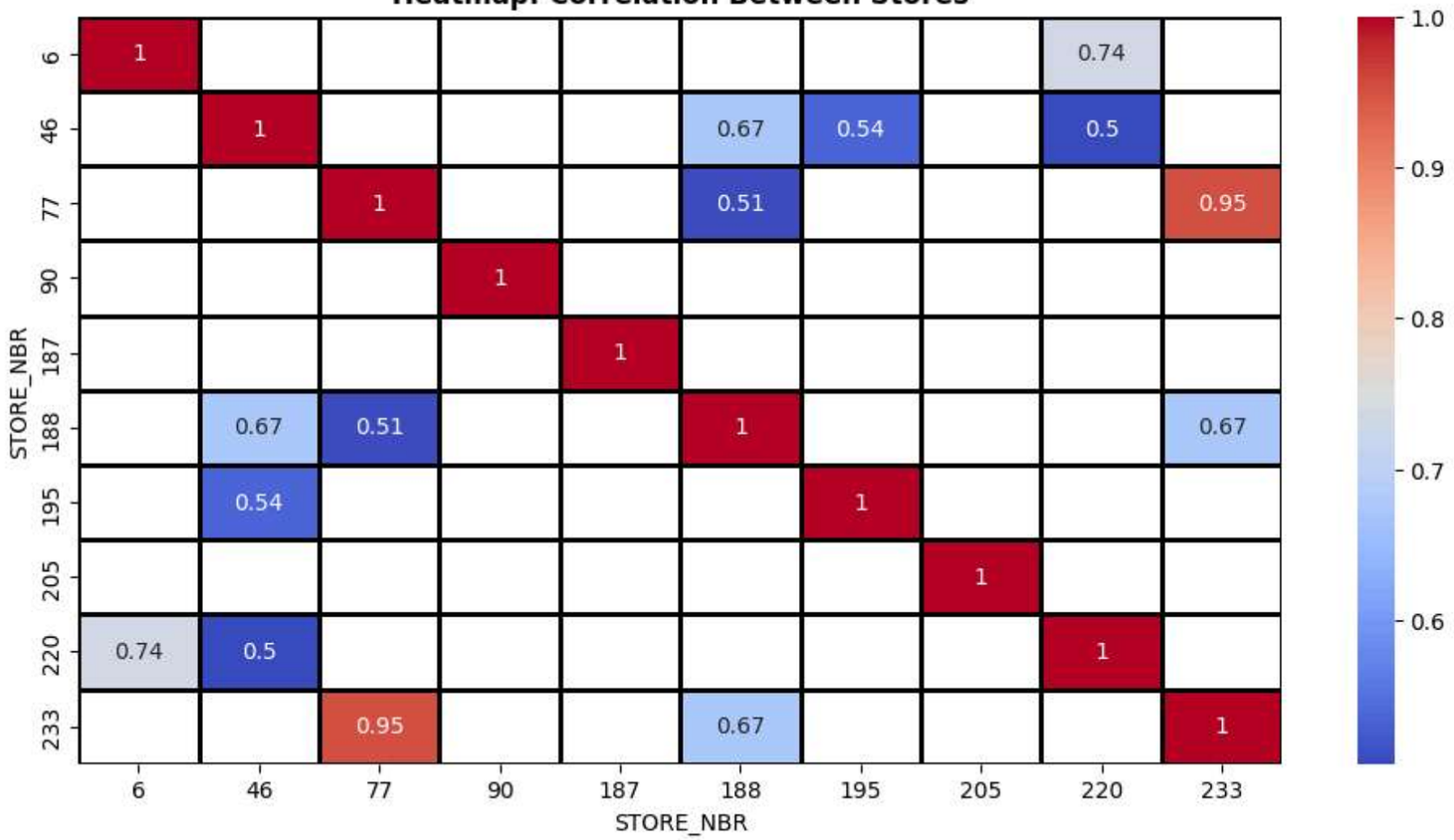
STORE_NBR	6	46	77	90	187	188	195	205	220	233
STORE_NBR										
6	1.000000	0.431184	0.023417	0.333076	0.122390	0.080477	0.224847	0.451831	0.735655	-0.153826
46	0.431184	1.000000	0.229845	0.051025	0.163323	0.672259	0.536910	-0.112817	0.504705	0.311149
77	0.023417	0.229845	1.000000	-0.407929	0.329746	0.507757	0.479952	0.334585	-0.023472	0.951196
90	0.333076	0.051025	-0.407929	1.000000	-0.587924	-0.618321	0.316578	0.298973	0.337917	-0.563654
187	0.122390	0.163323	0.329746	-0.587924	1.000000	0.357183	-0.172191	-0.455580	0.349718	0.311305
188	0.080477	0.672259	0.507757	-0.618321	0.357183	1.000000	0.091987	-0.205394	-0.082877	0.673034
195	0.224847	0.536910	0.479952	0.316578	-0.172191	0.091987	1.000000	0.478059	0.490338	0.433282
205	0.451831	-0.112817	0.334585	0.298973	-0.455580	-0.205394	0.478059	1.000000	0.153120	0.165469
220	0.735655	0.504705	-0.023472	0.337917	0.349718	-0.082877	0.490338	0.153120	1.000000	-0.160096
233	-0.153826	0.311149	0.951196	-0.563654	0.311305	0.673034	0.433282	0.165469	-0.160096	1.000000

In [26]:
```
plt.figure(figsize=(12,6))

sns.heatmap(correlation_77, annot=True, cmap="coolwarm", linewidths=1, linecolor='black')
plt.title("Heatmap: Correlation Between Stores", fontsize=12, fontweight='bold')
plt.show()
```

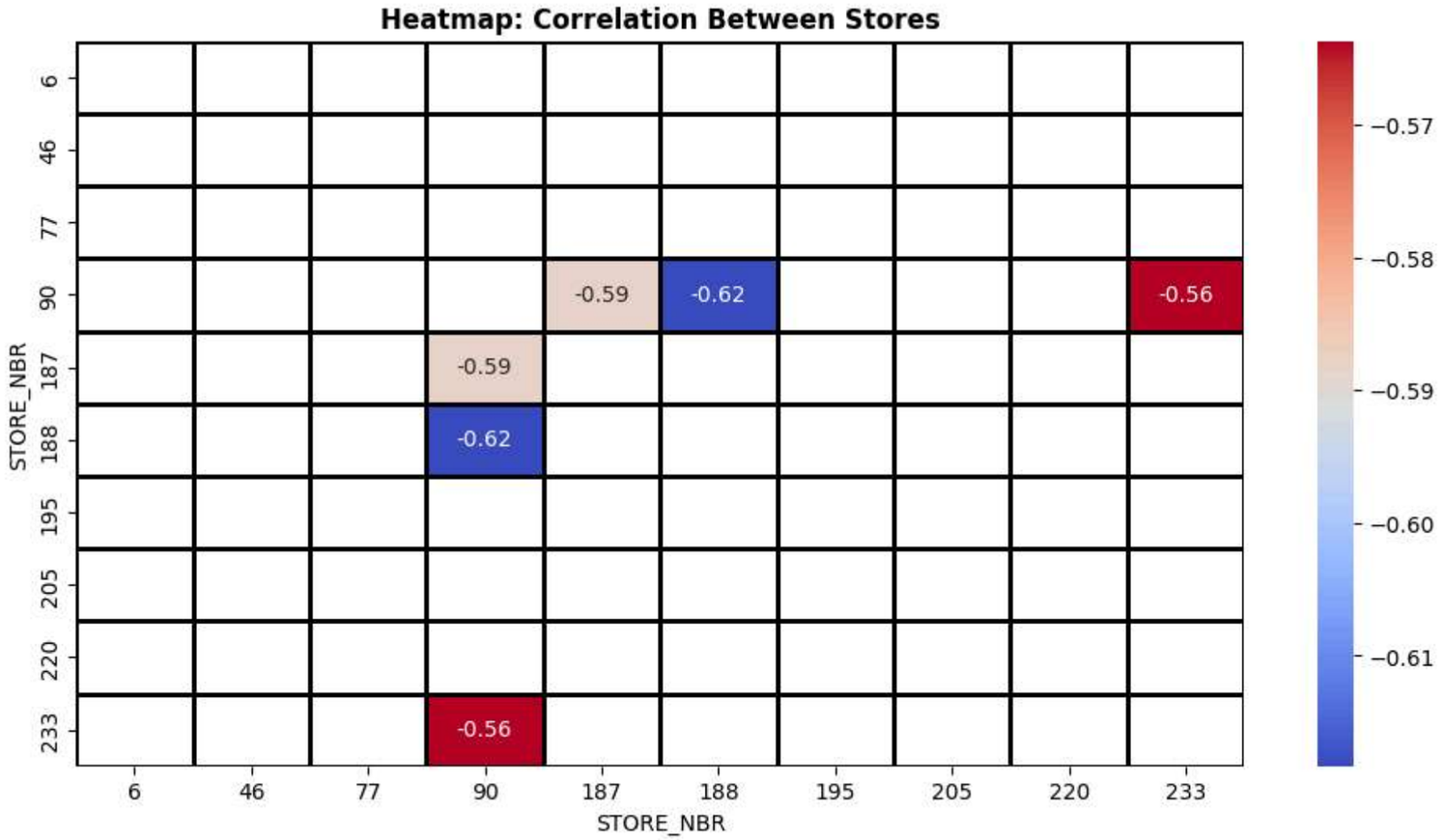
```
# Darker color means strong correlation
```



In [27]:
```
plt.figure(figsize=(12,6))

mask = (correlation_77 <= 0.5)

sns.heatmap(correlation_77, annot=True, cmap="coolwarm", linewidths=1, linecolor='black', mask = mask)
plt.title("Heatmap: Correlation Between Stores", fontsize=12, fontweight='bold')
plt.show()

# Darker color means strong correlation
```



In [28]:
```
plt.figure(figsize=(12,6))

mask = (correlation_77 >= -0.5)

sns.heatmap(correlation_77, annot=True, cmap="coolwarm", linewidths=1, linecolor='black', mask = mask)
plt.title("Heatmap: Correlation Between Stores", fontsize=12, fontweight='bold')
plt.show()

# Darker color means strong correlation
```



In [29]:
```
chips1_graph = pivot_chips1[[77,233]]

chips1_graph.plot()

plt.title('STORE WISE SALES', fontweight = 'bold')
plt.xlabel('MONTH', fontweight = 'bold')
plt.ylabel('SALES', fontweight = 'bold')

plt.legend(title="Store", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```

In [30]:
```
# Check correlation on entire table

chips_grp_before_df = pd.DataFrame(chips_grp_before)
chips_grp_before_pivot = chips_grp_before_df.pivot_table(index="MONTH_YEAR", columns="STORE_NBR", values="TOT_SALES")
chips_grp_before_corr = chips_grp_before_pivot.corr(method="pearson")
```

In [31]:
```
chips_grp_before_corr[77].sort_values(ascending=False).head(10).reset_index(name="Correlation").round(2)
```

Out[31]:

	STORE_NBR	Correlation
0	31	1.00
1	77	1.00
2	11	1.00
3	233	0.95
4	50	0.88
5	162	0.84
6	71	0.84
7	157	0.78
8	119	0.78
9	113	0.70

In [32]:
```
total_sales_sorted[[77,11,31,233,50,162,71]].reset_index()
```

Out[32]:

	STORE_NBR	TOT_SALES
0	77	1587.7
1	11	6.7
2	31	14.8
3	233	1511.1
4	50	1781.1
5	162	5432.0
6	71	6234.7

Store number 77 and 233 are highly correlated and Total Sales have a match. So we match 77 and 233 as Trial and Control stores resp.

In [33]:
```
# Finding match with store 86

range_86 = total_sales_sorted.index.get_loc(86)
range_86
```

Out[33]: 187

In [34]:
```
total_sorted_86 = total_sales_sorted.iloc[187:197].reset_index()
total_sorted_86
```

Out[34]:

	STORE_NBR	TOT_SALES
0	86	5759.25
1	138	5759.90
2	109	5763.10
3	227	5795.00
4	62	5797.60
5	91	5806.70
6	10	5818.35
7	102	5819.80
8	160	5891.70
9	257	5899.50

In [35]:
```
control_store_two = total_sorted_86['STORE_NBR'].tolist()
control_store_two
```

Out[35]: [86, 138, 109, 227, 62, 91, 10, 102, 160, 257]

In [36]:
```
chips_grp_before[control_store_two]
```

Out[36]:
```
STORE_NBR  MONTH_YEAR
86         2018-07-01     845.80
           2018-08-01     721.65
           2018-09-01     849.80
           2018-10-01     893.60
           2018-11-01     846.00
                             ...
257        2018-09-01     877.00
           2018-10-01     693.60
           2018-11-01    1005.80
           2018-12-01     801.60
           2019-01-01     712.40
Name: TOT_SALES, Length: 70, dtype: float64
```

In [37]:
```
control_two = pd.DataFrame({'Value': chips_grp_before[control_store_two]})
print(control_two)
```

```
                        Value
STORE_NBR MONTH_YEAR
86        2018-07-01   845.80
          2018-08-01   721.65
          2018-09-01   849.80
          2018-10-01   893.60
          2018-11-01   846.00
...                       ...
257       2018-09-01   877.00
          2018-10-01   693.60
          2018-11-01  1005.80
          2018-12-01   801.60
          2019-01-01   712.40

[70 rows x 1 columns]
```

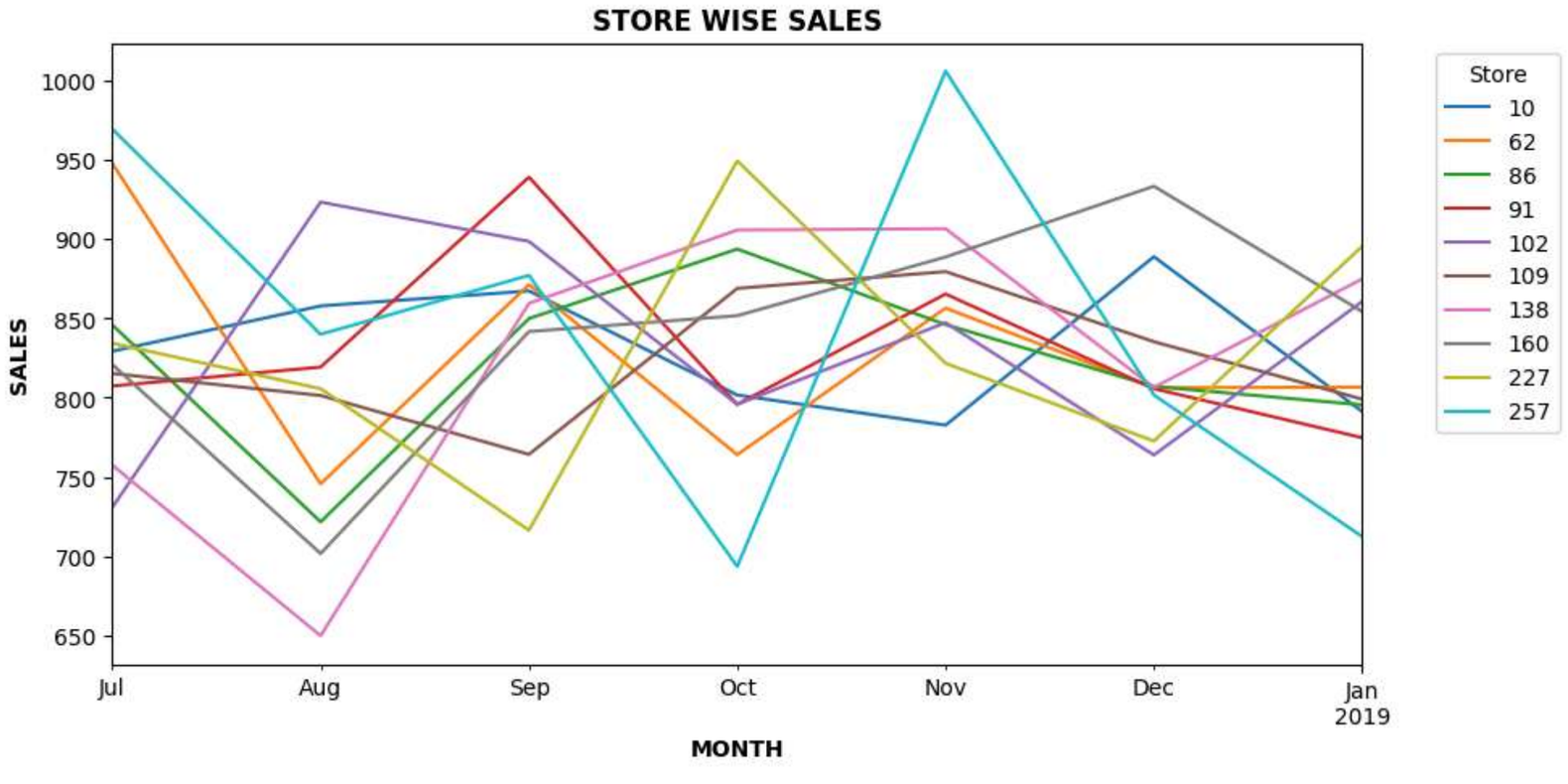
In [38]:
```
pivot_chips2 = control_two.pivot_table(index="MONTH_YEAR", columns="STORE_NBR",values='Value')
pivot_chips2
```

Out[38]:

STORE_NBR	10	62	86	91	102	109	138	160	227	257
MONTH_YEAR										
2018-07-01	829.20	947.6	845.80	807.3	730.8	815.2	757.4	820.8	834.2	969.2
2018-08-01	857.75	745.6	721.65	819.2	923.2	801.3	649.9	701.7	805.6	839.9
2018-09-01	867.20	871.2	849.80	938.9	898.4	764.2	859.4	841.6	716.4	877.0
2018-10-01	801.60	764.0	893.60	795.7	795.8	868.8	905.6	851.6	949.2	693.6
2018-11-01	782.60	856.4	846.00	865.3	847.2	879.4	906.4	888.6	821.6	1005.8
2018-12-01	888.80	806.2	807.00	805.6	763.8	835.2	806.4	933.2	772.6	801.6
2019-01-01	791.20	806.6	795.40	774.7	860.6	799.0	874.8	854.2	895.4	712.4

In [39]:
```
pivot_chips2.plot(figsize=(10,5))

plt.title('STORE WISE SALES', fontweight = 'bold')
plt.xlabel('MONTH', fontweight = 'bold')
plt.ylabel('SALES', fontweight = 'bold')
plt.legend(title="Store", bbox_to_anchor=(1.05, 1), loc='upper left')
```

```
plt.show()
```



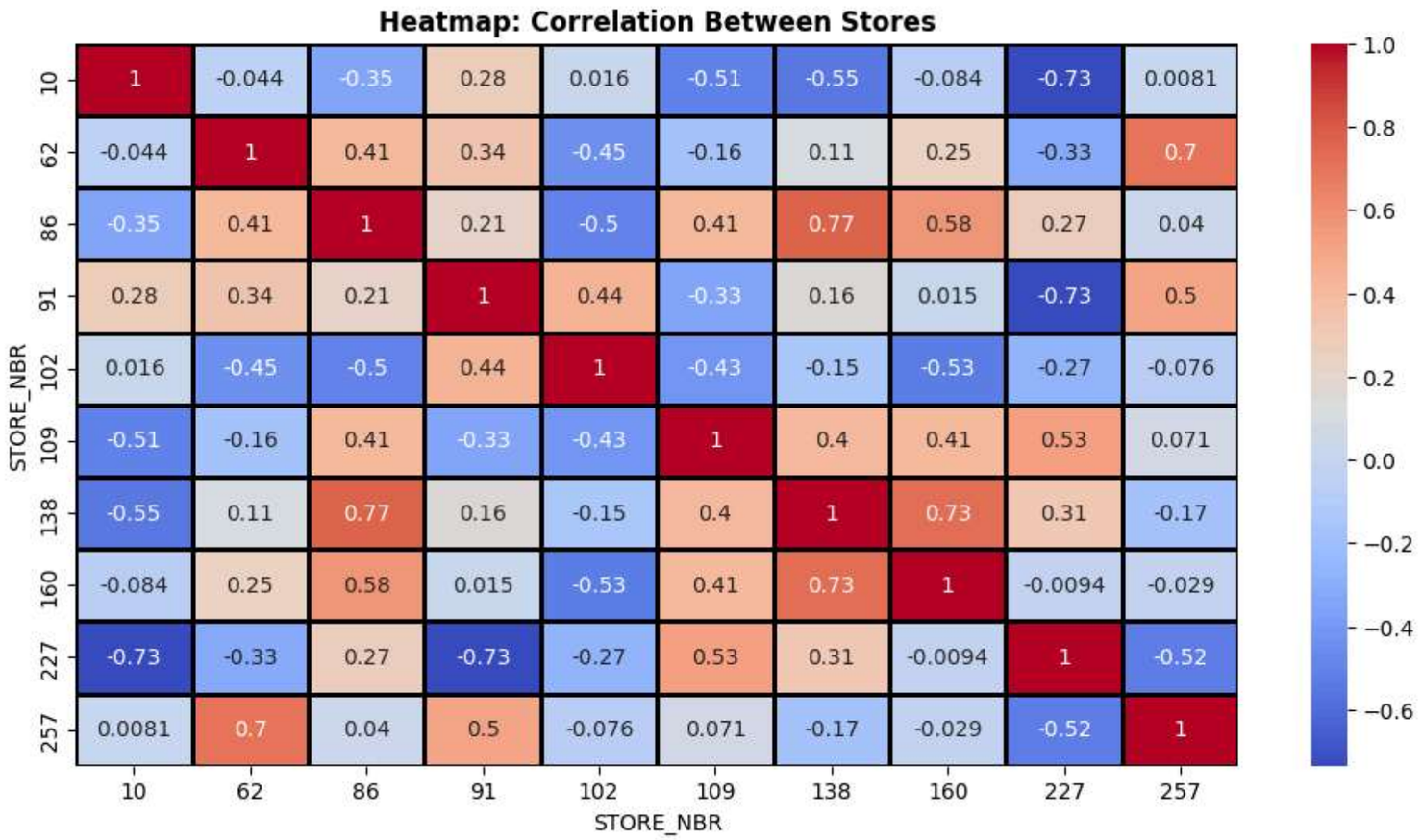
In [40]:
```
correlation_86 = pivot_chips2.corr(method='pearson')
correlation_86
```

Out[40]:

STORE_NBR	10	62	86	91	102	109	138	160	227	257
STORE_NBR										
10	1.000000	-0.044269	-0.354294	0.275602	0.015735	-0.507275	-0.549926	-0.083620	-0.727664	0.008141
62	-0.044269	1.000000	0.408494	0.337057	-0.446761	-0.163139	0.111441	0.250517	-0.326242	0.702696
86	-0.354294	0.408494	1.000000	0.207730	-0.502103	0.405479	0.768387	0.579684	0.271544	0.040067
91	0.275602	0.337057	0.207730	1.000000	0.440935	-0.333434	0.159561	0.015357	-0.734420	0.503743
102	0.015735	-0.446761	-0.502103	0.440935	1.000000	-0.428267	-0.147467	-0.532215	-0.272227	-0.075943
109	-0.507275	-0.163139	0.405479	-0.333434	-0.428267	1.000000	0.400986	0.405553	0.526680	0.070572
138	-0.549926	0.111441	0.768387	0.159561	-0.147467	0.400986	1.000000	0.733046	0.309212	-0.170430
160	-0.083620	0.250517	0.579684	0.015357	-0.532215	0.405553	0.733046	1.000000	-0.009367	-0.028780
227	-0.727664	-0.326242	0.271544	-0.734420	-0.272227	0.526680	0.309212	-0.009367	1.000000	-0.522826
257	0.008141	0.702696	0.040067	0.503743	-0.075943	0.070572	-0.170430	-0.028780	-0.522826	1.000000

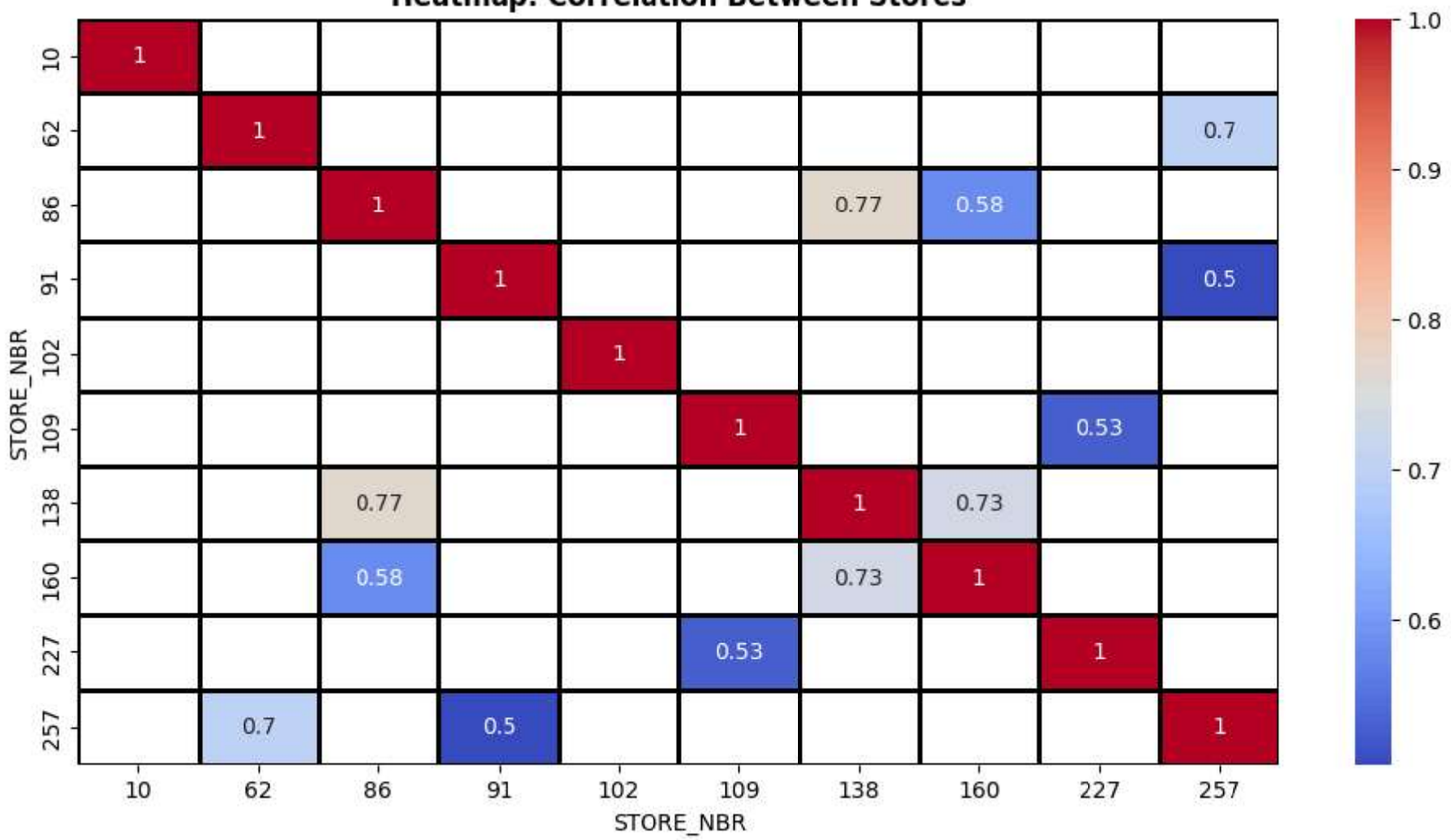
In [41]:
```
plt.figure(figsize=(12,6))
```

```
sns.heatmap(correlation_86, annot=True, cmap="coolwarm", linewidths=1, linecolor='black')
plt.title("Heatmap: Correlation Between Stores", fontsize=12, fontweight='bold')
plt.show()
```

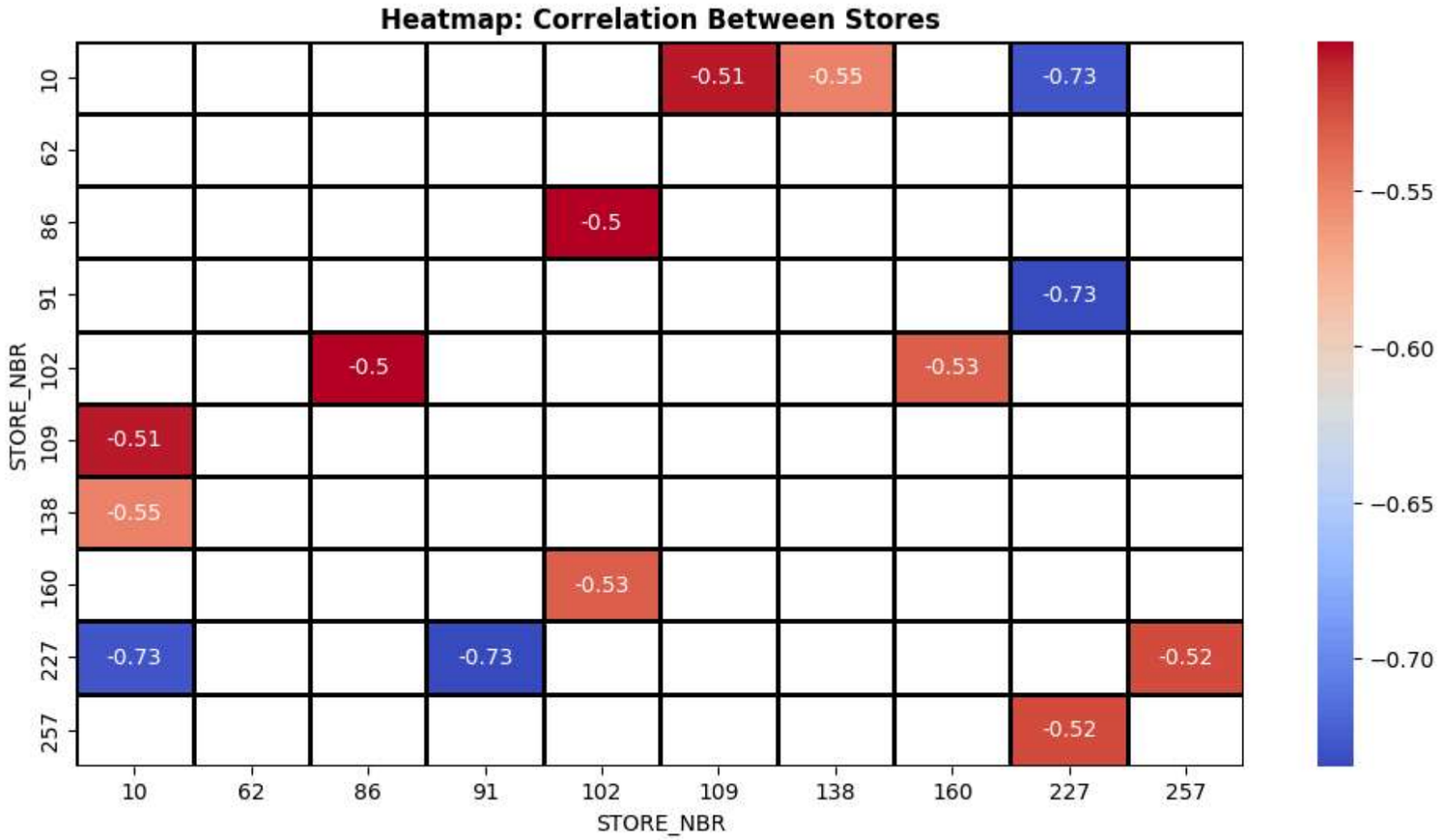


In [42]:

```
plt.figure(figsize=(12,6))

mask = (correlation_86 <= 0.5)

sns.heatmap(correlation_86, annot=True, cmap="coolwarm", linewidths=1, linecolor='black', mask = mask)
plt.title("Heatmap: Correlation Between Stores", fontsize=12, fontweight='bold')
plt.show()
```



In [43]:
```
plt.figure(figsize=(12,6))

mask = (correlation_86 >= -0.5)

sns.heatmap(correlation_86, annot=True, cmap="coolwarm", linewidths=1, linecolor='black', mask = mask)
plt.title("Heatmap: Correlation Between Stores", fontsize=12, fontweight='bold')
plt.show()
```



In [44]:
```
chips2_graph = pivot_chips2[[86,138]]

chips2_graph.plot()

plt.title('STORE WISE SALES (86 and 138)', fontweight = 'bold')
plt.xlabel('MONTH', fontweight = 'bold')
plt.ylabel('SALES', fontweight = 'bold')

plt.legend(title="Store", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```

In [45]:
```
chips_grp_before_corr[86].sort_values(ascending=False).head(10).reset_index(name="Correlation").round(2)
```

Out[45]:

	STORE_NBR	Correlation
0	86	1.00
1	155	0.84
2	260	0.83
3	6	0.79
4	132	0.78
5	138	0.77
6	114	0.73
7	269	0.73
8	222	0.73
9	166	0.73

In [46]:
```
total_sales_sorted[[86,155,260,6,132,138]].reset_index()
```

Out[46]:

	STORE_NBR	TOT_SALES
0	86	5759.25
1	155	5915.50
2	260	2832.20
3	6	1489.50
4	132	252.80
5	138	5759.90

In [47]:
```
# From above table, it looks like store 86 has the best match with store 155, 138. Lets visualize

chips3_graph = chips_grp_before_pivot[[86,155]]
chips3_graph.plot()

plt.title('STORE WISE SALES (86 and 155)', fontweight = 'bold')
plt.xlabel('MONTH', fontweight = 'bold')
plt.ylabel('SALES', fontweight = 'bold')

plt.legend(title="Store", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```

Store number 86 and 155 are highly correlated and Total Sales have a match. So we match 86 and 155 as Trial and Control stores resp.

In [48]:
```
# Finding matching store for 88
```

```
chips_grp_before_corr[88].sort_values(ascending=False).head(10).reset_index(name="Correlation")
```

Out[48]:

	STORE_NBR	Correlation
0	88	1.000000
1	91	0.911986
2	159	0.911447
3	204	0.880141
4	1	0.842511
5	240	0.817245
6	61	0.777007
7	79	0.771989
8	225	0.721337
9	7	0.698846

In [49]:
```
total_sales_sorted[[88,91,159,204,1,240,61,79,225,7]].reset_index()
```

Out[49]:

	STORE_NBR	TOT_SALES
0	88	8832.80
1	91	5806.70
2	159	162.00
3	204	261.70
4	1	1290.70
5	240	2606.50
6	61	322.90
7	79	6673.95
8	225	5536.00
9	7	6925.90

In [50]:
```
total_sales_sorted.index.get_loc(88)
```

Out[50]: 267

In [51]:
```
total_sorted_88 = total_sales_sorted.iloc[260:].reset_index()
control_store_three = total_sorted_88['STORE_NBR'].sort_values(ascending=True).tolist()
control_store_three
```

Out[51]: [4, 26, 40, 58, 88, 165, 199, 203, 226, 237]

In [52]:
```
chips_grp_before_corr[88][control_store_three].sort_values(ascending=False).head(10).reset_index()
```

Out[52]:

	STORE_NBR	88
0	88	1.000000
1	26	0.158180
2	237	0.109878
3	58	0.042724
4	203	0.000674
5	226	0.000075
6	40	-0.139776
7	165	-0.289887
8	4	-0.506130
9	199	-0.822121

In [53]:
```
total_sales_sorted[control_store_three].reset_index()
```

Out[53]:

	STORE_NBR	TOT_SALES
0	4	8576.20
1	26	7958.50
2	40	8224.20
3	58	8576.95
4	88	8832.80
5	165	8748.20
6	199	8200.50
7	203	8301.10
8	226	9657.75
9	237	8915.10

In [54]:
```
chips4_graph = chips_grp_before_pivot[[88,237]]
chips4_graph.plot()

plt.title('STORE WISE SALES (86 and 155)', fontweight = 'bold')
plt.xlabel('MONTH', fontweight = 'bold')
plt.ylabel('SALES', fontweight = 'bold')

plt.legend(title="Store", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```

Store number 88 and 237 are slightly correlated but Total Sales have a match. So we match 88 and 237 as Trial and Control stores resp.

In [55]:
```
trial_stores = total_sales_sorted[[77,86,88]].reset_index()
trial_stores.rename(columns={"STORE_NBR": "TRIAL STORES"}, inplace=True)
```

In [56]:
```
trial_stores
```

Out[56]:

	TRIAL STORES	TOT_SALES
0	77	1587.70
1	86	5759.25
2	88	8832.80

In [61]:
```
control_stores = total_sales_sorted[[233,155,237]].reset_index()
control_stores.rename(columns={"STORE_NBR": "CONTROL STORES"}, inplace=True)
```

In [62]:
```
control_stores
```

Out[62]:

	CONTROL STORES	TOT_SALES
0	233	1511.1
1	155	5915.5
2	237	8915.1

In [68]:
```
stores = pd.concat([trial_stores, control_stores], axis=1)
stores
```

Out[68]:

	TRIAL STORES	TOT_SALES	CONTROL STORES	TOT_SALES
0	77	1587.70	233	1511.1
1	86	5759.25	155	5915.5
2	88	8832.80	237	8915.1

In [93]:
```
chips_final.head(1)
```

Out[93]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	Weight	Weight in g	PROD_BRAND	LIFESTAGE	PREMIUM_CUSTOMER	MONTH_YEAR
0	2018-10-17	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0	175g	175	Natural	YOUNG SINGLES/COUPLES	Premium	2018-10-01

In [89]:
```
chips_trial_2019 = chips_final[(chips_final['MONTH_YEAR'] >= '2019-01') & (chips_final['MONTH_YEAR'] <= '2019-06')]
chips_trial_2019['MONTH_YEAR'].value_counts().reset_index()
```

Out[89]:

	MONTH_YEAR	count
0	2019-03-01	21013
1	2019-05-01	20715
2	2019-01-01	20490
3	2019-06-01	20242
4	2019-04-01	20154
5	2019-02-01	18895

In [94]:
```
tstore_77 = chips_trial_2019[chips_trial_2019['STORE_NBR'] == 77]
cstore_233 = chips_trial_2019[chips_trial_2019['STORE_NBR'] == 233]

tstore_86 = chips_trial_2019[chips_trial_2019['STORE_NBR'] == 86]
cstore_155 = chips_trial_2019[chips_trial_2019['STORE_NBR'] == 155]

tstore_88 = chips_trial_2019[chips_trial_2019['STORE_NBR'] == 88]
cstore_237 = chips_trial_2019[chips_trial_2019['STORE_NBR'] == 237]
```

In [95]:
```
tstore_77.head(2)
```

Out[95]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	Weight	Weight in g	PROD_BRAND	LIFESTAGE	PREMIUM_CUSTOMER	MONTH_YEAR
185	2019-06-17	77	77069	74987	70	Tyrrells Crisps Lightly Salted 165g	2	8.4	165g	165	Tyrrells	MIDAGE SINGLES/COUPLES	Budget	2019-06-01
1330	2019-03-28	77	77000	74911	18	Cheetos Chs & Bacon Balls 190g	1	3.3	190g	190	Cheetos	MIDAGE SINGLES/COUPLES	Budget	2019-03-01

STORES COMPARISON

TRIAL STORE 77 & CONTROL STORE 233

In [173…
```
# Looking at total sales and products sold

total_p_s1 = tstore_77[['TOT_SALES', 'PROD_QTY']].sum().reset_index(name = 'Values')
total_p_s1
```

Out[173…

	index	Values
0	TOT_SALES	1421.7
1	PROD_QTY	416.0

In [175…
```
total_cus1 = pd.Series({'Total_Customers': tstore_77['LYLTY_CARD_NBR'].shape[0]}).reset_index(name= 'Values')
total_cus1
```

Out[175…

	index	Values
0	Total_Customers	264

In [177…
```
repeat_cust1 = tstore_77['LYLTY_CARD_NBR'].value_counts().reset_index()
total_rep1 = pd.Series({'Repeat_Customers': repeat_cust1[repeat_cust1['count'] > 1].shape[0]}).reset_index(name = 'Values')
total_rep1
```

Out[177…

	index	Values
0	Repeat_Customers	51

In [206…
```
summary_77 = pd.concat([total_p_s1, total_cus1, total_rep1]).reset_index(drop=True)
summary_77
```

Out[206…

	index	Values
0	TOT_SALES	1421.7
1	PROD_QTY	416.0
2	Total_Customers	264.0
3	Repeat_Customers	51.0

In [212…
```
total_p_s2 = cstore_233[['TOT_SALES', 'PROD_QTY']].sum().reset_index(name="Values")
total_p_s2
```

Out[212…

	index	Values
0	TOT_SALES	1202.5
1	PROD_QTY	332.0

In [213…
```
total_cus2 = pd.Series({'Total_Customers': cstore_233['LYLTY_CARD_NBR'].shape[0]}).reset_index(name="Values")
total_cus2
```

Out[213…

	index	Values
0	Total_Customers	228

In [214…
```
repeat_cust2 = cstore_233['LYLTY_CARD_NBR'].value_counts().reset_index()
total_rep2 = pd.Series({'Repeat_Customers': repeat_cust2[repeat_cust2['count'] > 1].shape[0]}).reset_index(name="Values")
total_rep2
```

Out[214…

	index	Values
0	Repeat_Customers	37

In [217…
```
summary_233 = pd.concat([total_p_s2, total_cus2, total_rep2]).reset_index(drop=True)
summary_233
```

Out[217…

	index	Values
0	TOT_SALES	1202.5
1	PROD_QTY	332.0
2	Total_Customers	228.0
3	Repeat_Customers	37.0

In [219…
```
display(summary_77)
display(summary_233)
```

	index	Values
0	TOT_SALES	1421.7
1	PROD_QTY	416.0
2	Total_Customers	264.0
3	Repeat_Customers	51.0

	index	Values
0	TOT_SALES	1202.5
1	PROD_QTY	332.0
2	Total_Customers	228.0
3	Repeat_Customers	37.0

In [311…
```
grouped77 = tstore_77.groupby(['MONTH_YEAR'])['TOT_SALES'].sum().reset_index()
grouped77
```

Out[311…

	MONTH_YEAR	TOT_SALES
0	2019-01-01	188.4
1	2019-02-01	211.6
2	2019-03-01	255.1
3	2019-04-01	258.1
4	2019-05-01	272.3
5	2019-06-01	236.2

In [314…

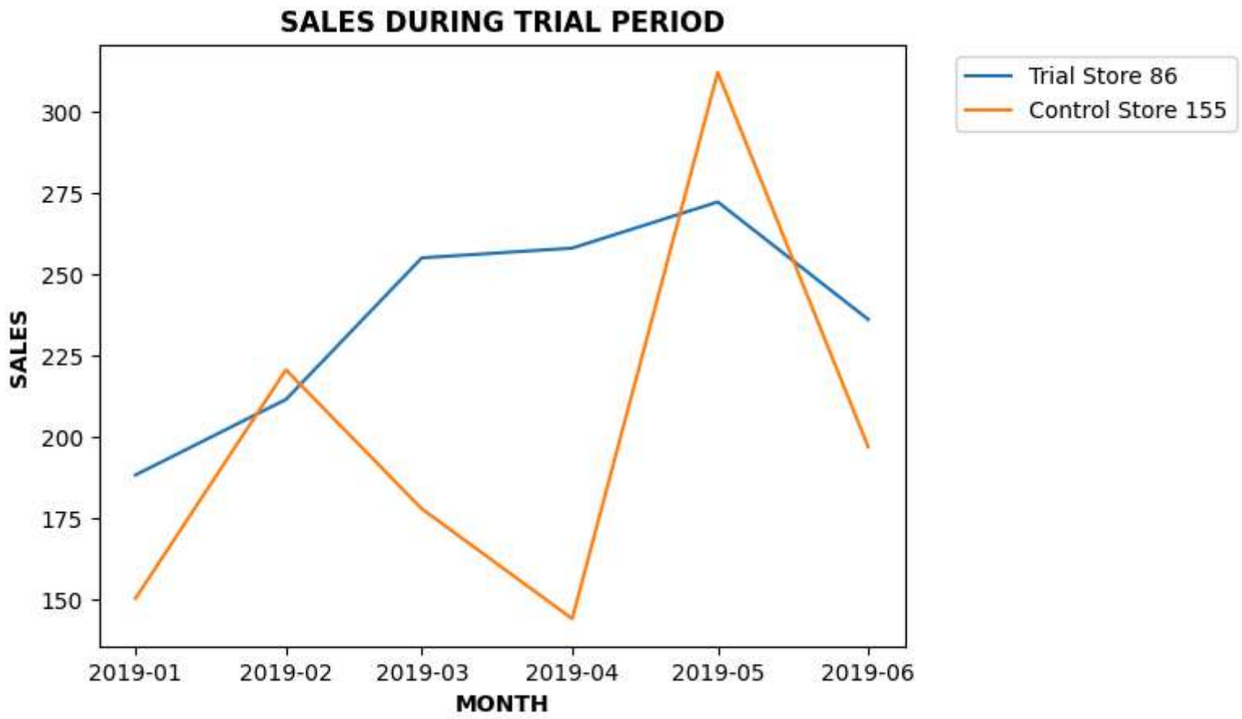
```
grouped233 = cstore_233.groupby(['MONTH_YEAR'])['TOT_SALES'].sum().reset_index()
grouped233
```

Out[314…

	MONTH_YEAR	TOT_SALES
0	2019-01-01	150.5
1	2019-02-01	220.7
2	2019-03-01	178.0
3	2019-04-01	144.2
4	2019-05-01	312.1
5	2019-06-01	197.0

In [315…

```
sns.lineplot(x = grouped77['MONTH_YEAR'] , y = grouped77['TOT_SALES'], label = 'Trial Store 86')
sns.lineplot(x = grouped233['MONTH_YEAR'] , y = grouped233['TOT_SALES'], label = 'Control Store 155')

plt.title('SALES DURING TRIAL PERIOD', fontweight = 'bold')
plt.xlabel('MONTH', fontweight = 'bold')
plt.ylabel('SALES', fontweight = 'bold')

plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



In [302…

```
import warnings
warnings.filterwarnings("ignore", category=FutureWarning)
```

In [305…

```
x_labels = ['Trial Store 77', 'Control Store 233']
y_values = [summary_77['Values'][0], summary_233['Values'][0]]

ax = sns.barplot(x=x_labels, y=y_values, palette='pastel')

plt.title('SALES TRIAL VS CONTROL', fontweight = 'bold')
plt.xlabel('STORE', fontweight = 'bold')
plt.ylabel('SALES', fontweight = 'bold')

for x in ax.containers:
    ax.bar_label(x)

plt.show()
```

Total Sales has increased after conducting the trials

TRIAL STORE 86 & CONTROL STORE 155

In [225…
```
total_p_s3 = tstore_86[['TOT_SALES', 'PROD_QTY']].sum().reset_index(name="Values")
total_p_s3
```

Out[225…

	index	Values
0	TOT_SALES	5000.1
1	PROD_QTY	1449.0

In [239…
```
total_cus3 = pd.Series({"Total_Customer": tstore_86['LYLTY_CARD_NBR'].shape[0]}).reset_index(name="Values")
total_cus3
```

Out[239…

	index	Values
0	Total_Customer	725

In [259…
```
repeat_cust3 = tstore_77['LYLTY_CARD_NBR'].value_counts().reset_index()
total_rep3 = pd.Series({'Repeat_Customers': repeat_cust3[repeat_cust3['count'] > 1].shape[0]}).reset_index(name = 'Values')
total_rep3
```

Out[259…

	index	Values
0	Repeat_Customers	51

In [260…
```
summary_86 = pd.concat([total_p_s3, total_cus3, total_rep3]).reset_index(drop=True)
summary_86
```

Out[260…

	index	Values
0	TOT_SALES	5000.1
1	PROD_QTY	1449.0
2	Total_Customer	725.0
3	Repeat_Customers	51.0

In [256…
```
total_p_s4 = cstore_155[['TOT_SALES', 'PROD_QTY']].sum().reset_index(name='Values')
total_p_s4
```

Out[256…

	index	Values
0	TOT_SALES	4861.25
1	PROD_QTY	1378.00

In [258…
```
total_cus4 = pd.Series({"Total_Customer": cstore_155['LYLTY_CARD_NBR'].shape[0]}).reset_index(name="Values")
total_cus4
```

Out[258…

	index	Values
0	Total_Customer	695

In [261…
```
repeat_cust4 = cstore_155['LYLTY_CARD_NBR'].value_counts().reset_index()
total_rep4 = pd.Series({'Repeat_Customers': repeat_cust4[repeat_cust4['count'] > 1].shape[0]}).reset_index(name = 'Values')
total_rep4
```

Out[261…

	index	Values
0	Repeat_Customers	192

In [265…
```
summary_155 = pd.concat([total_p_s4, total_cus4, total_rep4]).reset_index(drop=True)
summary_155
```

Out[265…

	index	Values
0	TOT_SALES	4861.25
1	PROD_QTY	1378.00
2	Total_Customer	695.00
3	Repeat_Customers	192.00

In [ ]:
```
display(summary_86)
display(summary_155)
```

	index	Values
0	TOT_SALES	5000.1
1	PROD_QTY	1449.0
2	Total_Customer	725.0
3	Repeat_Customers	51.0

	index	Values
0	TOT_SALES	4861.25
1	PROD_QTY	1378.00
2	Total_Customer	695.00
3	Repeat_Customers	192.00

In [306…
```
grouped86 = tstore_86.groupby(['MONTH_YEAR'])['TOT_SALES'].sum().reset_index()
grouped86
```

Out[306…

	MONTH_YEAR	TOT_SALES
0	2019-01-01	795.4
1	2019-02-01	872.8
2	2019-03-01	945.4
3	2019-04-01	798.8
4	2019-05-01	826.9
5	2019-06-01	760.8

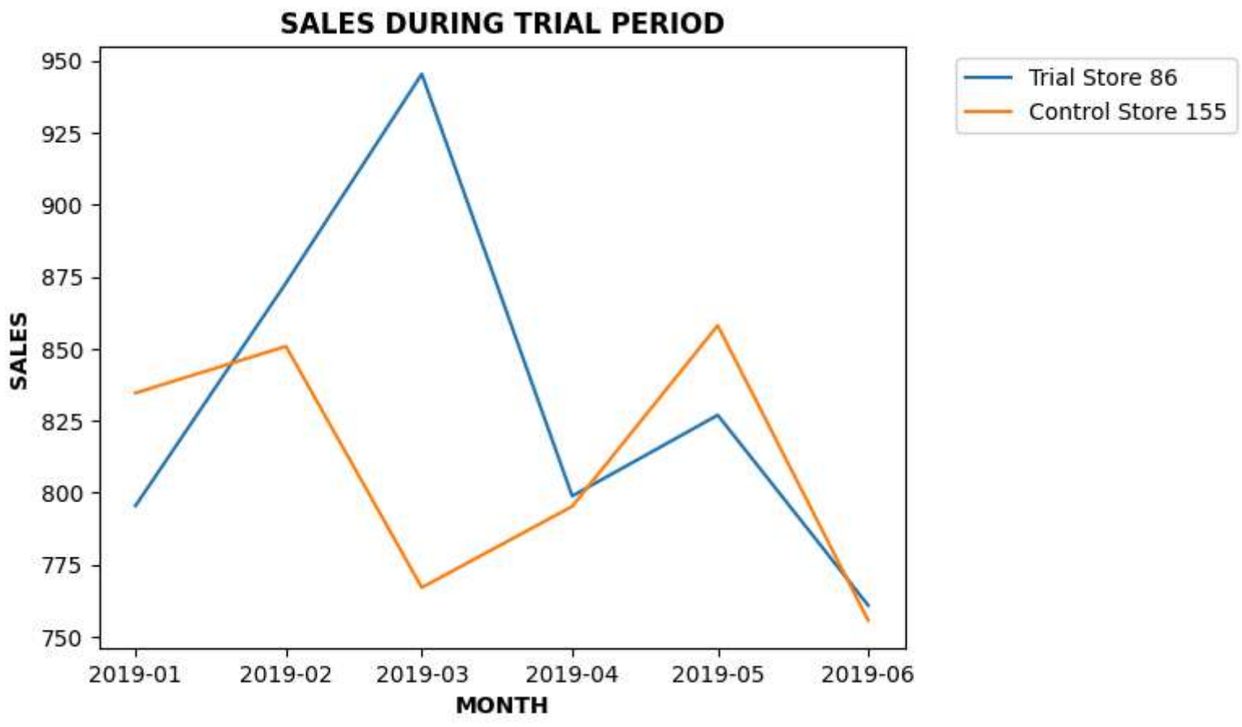
In [307…
```
grouped155 = cstore_155.groupby(['MONTH_YEAR'])['TOT_SALES'].sum().reset_index()
grouped155
```

Out[307…

	MONTH_YEAR	TOT_SALES
0	2019-01-01	834.60
1	2019-02-01	850.80
2	2019-03-01	767.00
3	2019-04-01	795.20
4	2019-05-01	858.05
5	2019-06-01	755.60

In [308…
```
sns.lineplot(x = grouped86['MONTH_YEAR'] , y = grouped86['TOT_SALES'], label = 'Trial Store 86')
sns.lineplot(x = grouped155['MONTH_YEAR'] , y = grouped155['TOT_SALES'], label = 'Control Store 155')

plt.title('SALES DURING TRIAL PERIOD', fontweight = 'bold')
plt.xlabel('MONTH', fontweight = 'bold')
plt.ylabel('SALES', fontweight = 'bold')
```

```
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



In [309…

```
x_labels = ['Trial Store 86', 'Control Store 155']
y_values = [summary_86['Values'][0], summary_155['Values'][0]]

ax = sns.barplot(x=x_labels, y=y_values, palette='pastel')

plt.title('SALES TRIAL VS CONTROL', fontweight = 'bold')
plt.xlabel('STORE', fontweight = 'bold')
plt.ylabel('SALES', fontweight = 'bold')

for x in ax.containers:
    ax.bar_label(x)

plt.show()
```

Total sales has increased after conducting trials

TRIAL STORE 88 & CONTROL STORE 237

In [277…
```
stores
```

Out[277…

	TRIAL STORES	TOT_SALES	CONTROL STORES	TOT_SALES
0	77	1587.70	233	1511.1
1	86	5759.25	155	5915.5
2	88	8832.80	237	8915.1

In [273…
```
total_p_s5 = tstore_88[['TOT_SALES', 'PROD_QTY']].sum().reset_index(name='Values')
total_p_s5
```

Out[273…

	index	Values
0	TOT_SALES	7828.45
1	PROD_QTY	1787.00

In [274…
```
total_cus5 = pd.Series({"Total_Customer": tstore_88['LYLTY_CARD_NBR'].shape[0]}).reset_index(name="Values")
total_cus5
```

Out[274...

	index	Values
0	Total_Customer	899

In [275...
```
repeat_cust5 = tstore_88['LYLTY_CARD_NBR'].value_counts().reset_index()
total_rep5 = pd.Series({'Repeat_Customers': repeat_cust5[repeat_cust5['count'] > 1].shape[0]}).reset_index(name = 'Values')
total_rep5
```

Out[275...

	index	Values
0	Repeat_Customers	253

In [276...
```
summary_88 = pd.concat([total_p_s5, total_cus5, total_rep5]).reset_index(drop=True)
summary_88
```

Out[276...

	index	Values
0	TOT_SALES	7828.45
1	PROD_QTY	1787.00
2	Total_Customer	899.00
3	Repeat_Customers	253.00

In [278...
```
total_p_s6 = cstore_237[['TOT_SALES', 'PROD_QTY']].sum().reset_index(name='Values')
total_p_s6
```

Out[278...

	index	Values
0	TOT_SALES	7033.2
1	PROD_QTY	1601.0

In [279...
```
total_cus6 = pd.Series({"Total_Customer": cstore_237['LYLTY_CARD_NBR'].shape[0]}).reset_index(name="Values")
total_cus6
```

Out[279...

	index	Values
0	Total_Customer	815

In [280...
```
repeat_cust6 = cstore_237['LYLTY_CARD_NBR'].value_counts().reset_index()
total_rep6 = pd.Series({'Repeat_Customers': repeat_cust6[repeat_cust6['count'] > 1].shape[0]}).reset_index(name = 'Values')
total_rep6
```

Out[280...

	index	Values
0	Repeat_Customers	240

In [282...
```
summary_237 = pd.concat([total_p_s6, total_cus6, total_rep6]).reset_index(drop=True)
summary_237
```

Out[282...

	index	Values
0	TOT_SALES	7033.2
1	PROD_QTY	1601.0
2	Total_Customer	815.0
3	Repeat_Customers	240.0

In [283...

```
display(summary_88)
display(summary_237)
```

	index	Values
0	TOT_SALES	7828.45
1	PROD_QTY	1787.00
2	Total_Customer	899.00
3	Repeat_Customers	253.00

	index	Values
0	TOT_SALES	7033.2
1	PROD_QTY	1601.0
2	Total_Customer	815.0
3	Repeat_Customers	240.0

In [284...

```
grouped88 = tstore_88.groupby(['MONTH_YEAR'])['TOT_SALES'].sum().reset_index()
grouped88
```

Out[284...

	MONTH_YEAR	TOT_SALES
0	2019-01-01	1215.40
1	2019-02-01	1339.60
2	2019-03-01	1467.00
3	2019-04-01	1317.00
4	2019-05-01	1236.85
5	2019-06-01	1252.60

In [285...

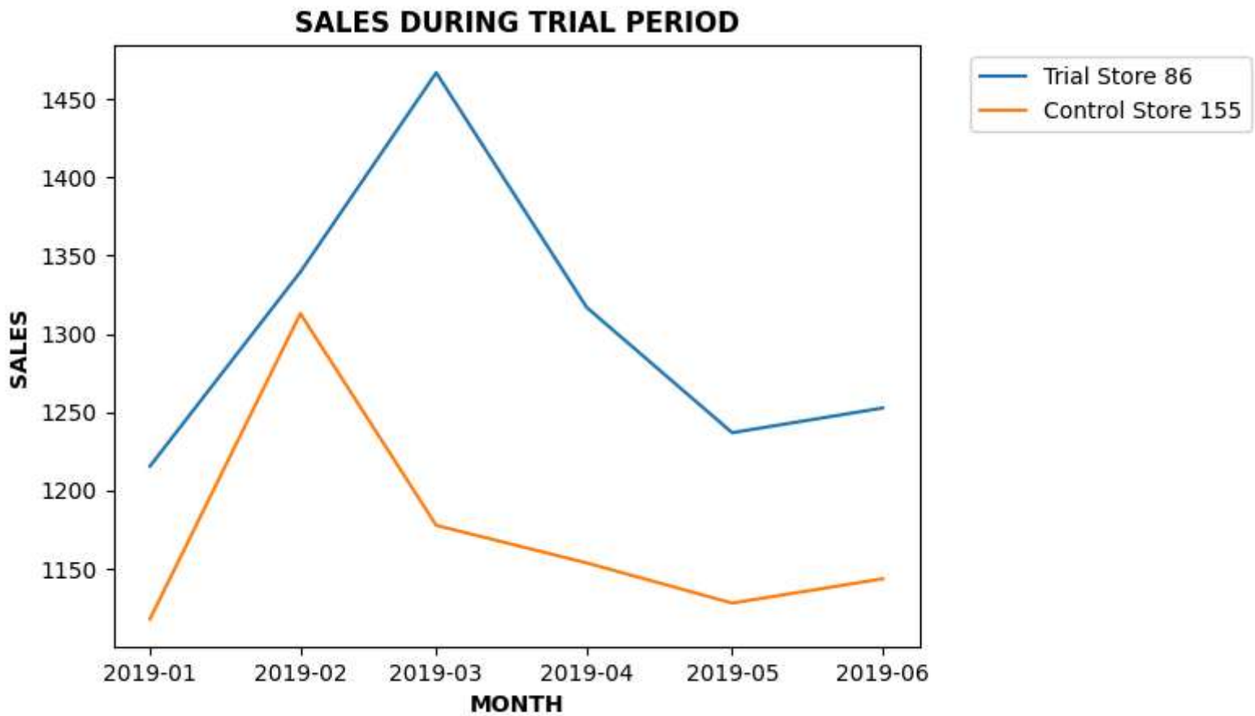
```
grouped237 = cstore_237.groupby(['MONTH_YEAR'])['TOT_SALES'].sum().reset_index()
grouped237
```

Out[285…

	MONTH_YEAR	TOT_SALES
0	2019-01-01	1117.7
1	2019-02-01	1313.0
2	2019-03-01	1177.6
3	2019-04-01	1153.6
4	2019-05-01	1127.9
5	2019-06-01	1143.4

In [286…

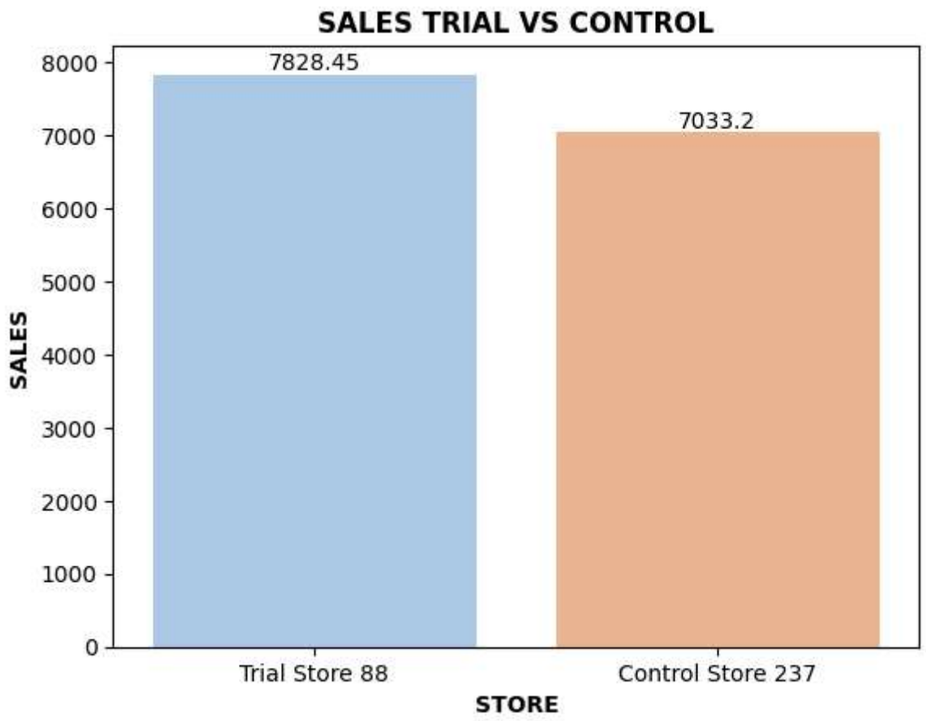
```
sns.lineplot(x = grouped88['MONTH_YEAR'] , y = grouped88['TOT_SALES'], label = 'Trial Store 86')
sns.lineplot(x = grouped237['MONTH_YEAR'] , y = grouped237['TOT_SALES'], label = 'Control Store 155')

plt.title('SALES DURING TRIAL PERIOD', fontweight = 'bold')
plt.xlabel('MONTH', fontweight = 'bold')
plt.ylabel('SALES', fontweight = 'bold')

plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



In [310…

```
x_labels = ['Trial Store 88', 'Control Store 237']
y_values = [summary_88['Values'][0], summary_237['Values'][0]]

ax = sns.barplot(x=x_labels, y=y_values, palette='pastel')

plt.title('SALES TRIAL VS CONTROL', fontweight = 'bold')
plt.xlabel('STORE', fontweight = 'bold')
plt.ylabel('SALES', fontweight = 'bold')
```

```
for x in ax.containers:
    ax.bar_label(x)

plt.show()
```



Total sales has increased after conducting the trials

In [316…
```
stores
```

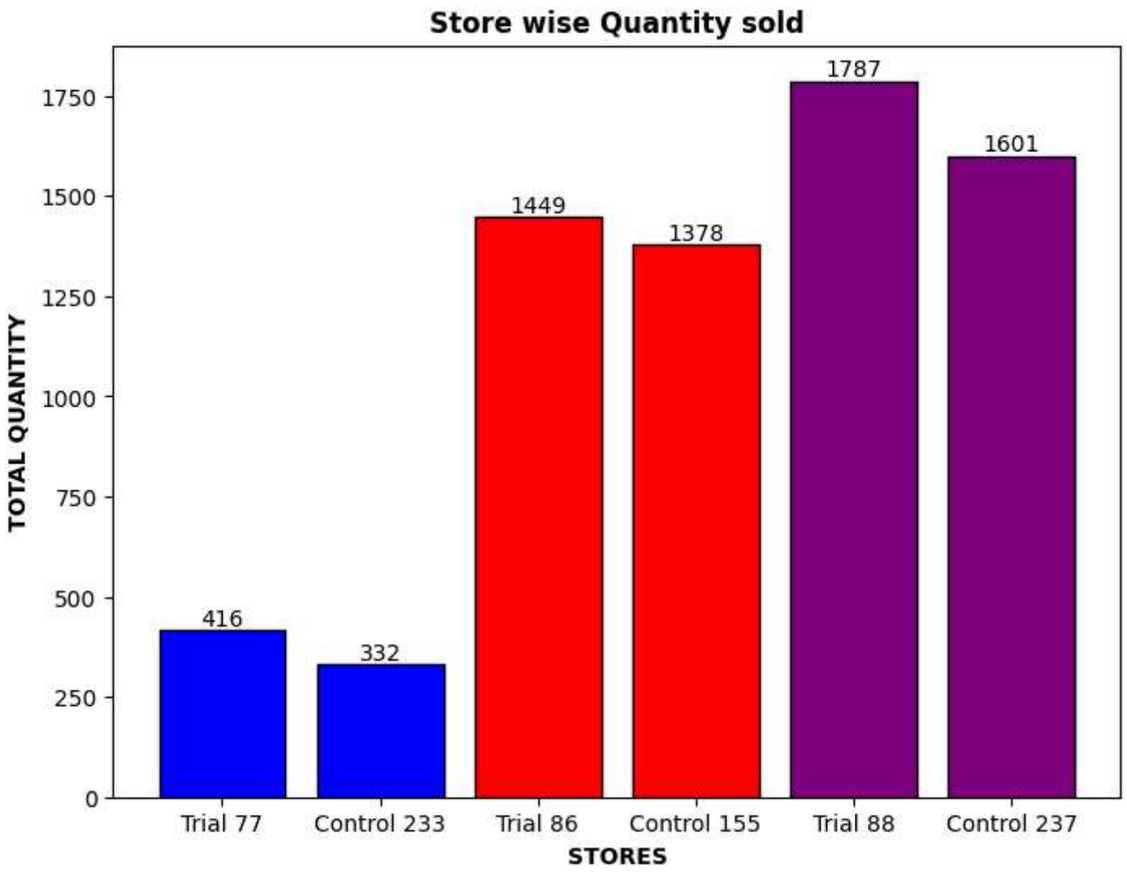
Out[316…

	TRIAL STORES	TOT_SALES	CONTROL STORES	TOT_SALES
0	77	1587.70	233	1511.1
1	86	5759.25	155	5915.5
2	88	8832.80	237	8915.1

In [371…
```
group1 = ["Trial 77", "Control 233"]
group2 = ["Trial 86", "Control 155"]
group3 = ["Trial 88", "Control 237"]

values_grp1 = [summary_77['Values'][1], summary_233['Values'][1]]  # Extract the actual values
values_grp2 = [summary_86['Values'][1], summary_155['Values'][1]]
values_grp3 = [summary_88['Values'][1], summary_237['Values'][1]]
```

In [372…
```
fig, ax = plt.subplots(figsize=(8, 6))

ax.bar(group1, values_grp1, label= group1, color = 'blue', edgecolor = 'black')
ax.bar(group2, values_grp2, label= group2, color = 'red', edgecolor = 'black')
ax.bar(group3, values_grp3, label= group3, color = 'purple', edgecolor = 'black')
```

```
plt.xlabel("STORES", fontweight = 'bold')
plt.ylabel("TOTAL QUANTITY", fontweight = 'bold')
plt.title("Store wise Quantity sold", fontweight = 'bold')

for bars in ax.containers:
    ax.bar_label(bars)
```



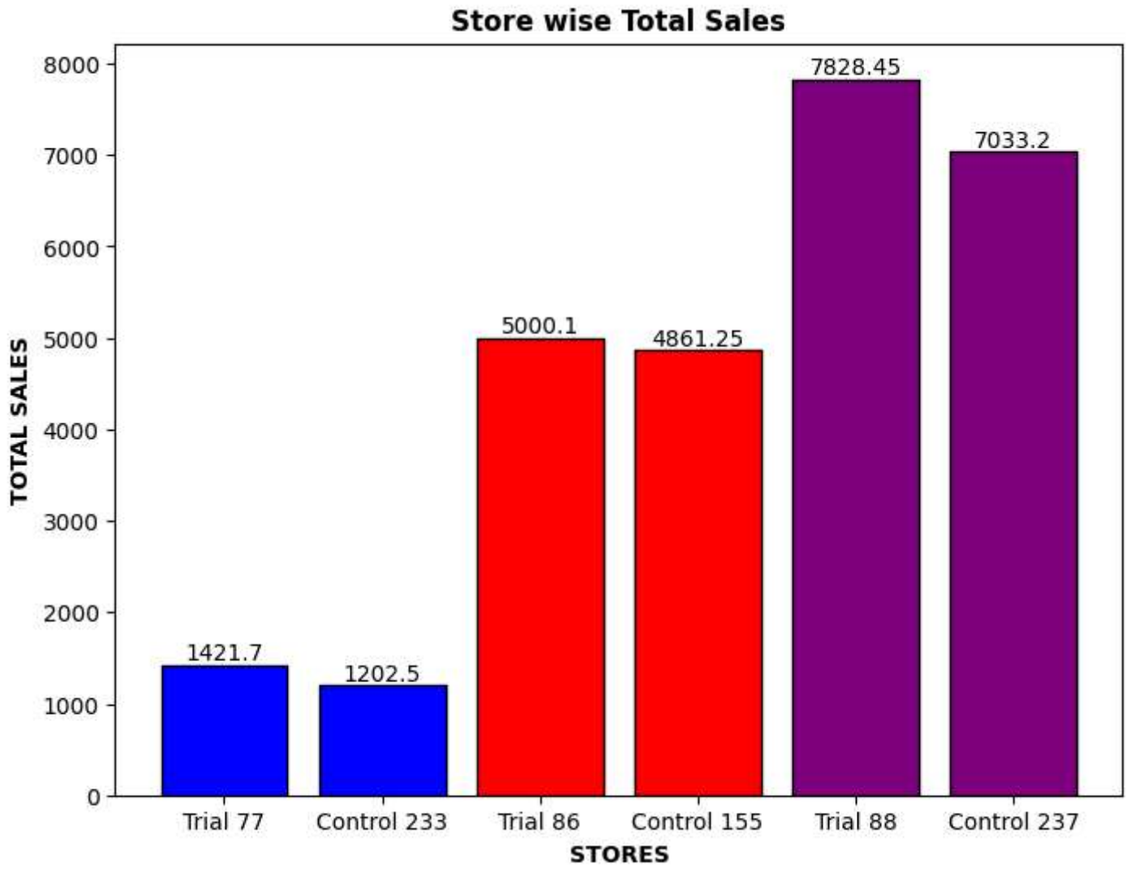
In [373…

```
group1 = ["Trial 77", "Control 233"]
group2 = ["Trial 86", "Control 155"]
group3 = ["Trial 88", "Control 237"]

values_grp1 = [summary_77['Values'][0], summary_233['Values'][0]]  # Extract the actual values
values_grp2 = [summary_86['Values'][0], summary_155['Values'][0]]
values_grp3 = [summary_88['Values'][0], summary_237['Values'][0]]
```

In [375…

```
fig, ax = plt.subplots(figsize=(8, 6))

ax.bar(group1, values_grp1, label= group1, color = 'blue', edgecolor = 'black')
ax.bar(group2, values_grp2, label= group2, color = 'red', edgecolor = 'black')
ax.bar(group3, values_grp3, label= group3, color = 'purple', edgecolor = 'black')

plt.xlabel("STORES", fontweight = 'bold')
plt.ylabel("TOTAL SALES", fontweight = 'bold')
plt.title("Store wise Total Sales", fontweight = 'bold')
```

```
for bars in ax.containers:
    ax.bar_label(bars)
```



OBSERVATION

- New layout is working out to increase sales
- Both the Total Quantity Sold and Total Sales are higher for the trial stores
- Hence, the number of trail stores must be increased and continue the incentives in control stores as well