quantium-task2

July 29, 2025

[ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
import datetime
import scipy.stats as stats
```

[ ]:
```
data = pd.read_csv('/content/QVI_data.csv')
data
```

[ ]:
```
        LYLTY_CARD_NBR        DATE  STORE_NBR  TXN_ID  PROD_NBR  \
0                 1000  2018-10-17          1       1         5
1                 1002  2018-09-16          1       2        58
2                 1003  2019-03-07          1       3        52
3                 1003  2019-03-08          1       4       106
4                 1004  2018-11-02          1       5        96
…                  …          …         …       …        …
264829         2370701  2018-12-08         88  240378        24
264830         2370751  2018-10-01         88  240394        60
264831         2370961  2018-10-24         88  240480        70
264832         2370961  2018-10-27         88  240481        65
264833         2373711  2018-12-14         88  241815        16

                                       PROD_NAME  PROD_QTY  TOT_SALES  \
0         Natural Chip        Compny SeaSalt175g         2        6.0
1          Red Rock Deli Chikn&Garlic Aioli 150g         1        2.7
2          Grain Waves Sour    Cream&Chives 210G         1        3.6
3         Natural ChipCo      Hony Soy Chckn175g         1        3.0
4                WW Original Stacked Chips 160g         1        1.9
…                                            …       …          …
264829     Grain Waves         Sweet Chilli 210g         2        7.2
264830      Kettle Tortilla ChpsFeta&Garlic 150g         2        9.2
264831   Tyrrells Crisps     Lightly Salted 165g         2        8.4
264832   Old El Paso Salsa   Dip Chnky Tom Ht300g         2       10.2
264833  Smiths Crinkle Chips Salt & Vinegar 330g         2       11.4
```

```
        PACK_SIZE       BRAND              LIFESTAGE PREMIUM_CUSTOMER
0             175     NATURAL  YOUNG SINGLES/COUPLES          Premium
1             150         RRD  YOUNG SINGLES/COUPLES       Mainstream
2             210     GRNWVES         YOUNG FAMILIES           Budget
3             175     NATURAL         YOUNG FAMILIES           Budget
4             160  WOOLWORTHS  OLDER SINGLES/COUPLES       Mainstream
…             …         …                    …                …
264829        210     GRNWVES         YOUNG FAMILIES       Mainstream
264830        150      KETTLE         YOUNG FAMILIES          Premium
264831        165    TYRRELLS         OLDER FAMILIES           Budget
264832        300         OLD         OLDER FAMILIES           Budget
264833        330      SMITHS  YOUNG SINGLES/COUPLES       Mainstream

[264834 rows x 12 columns]
```

[ ]:
```
data.shape
```

[ ]:
```
(264834, 12)
```

[ ]:
```
data.isnull().sum()
```

[ ]:
```
LYLTY_CARD_NBR      0
DATE                0
STORE_NBR           0
TXN_ID              0
PROD_NBR            0
PROD_NAME           0
PROD_QTY            0
TOT_SALES           0
PACK_SIZE           0
BRAND               0
LIFESTAGE           0
PREMIUM_CUSTOMER    0
dtype: int64
```

[ ]:
```
data.describe()
```

[ ]:
```
       LYLTY_CARD_NBR      STORE_NBR        TXN_ID       PROD_NBR  \
count    2.648340e+05  264834.000000  2.648340e+05  264834.000000
mean     1.355488e+05     135.079423  1.351576e+05      56.583554
std      8.057990e+04      76.784063  7.813292e+04      32.826444
min      1.000000e+03       1.000000  1.000000e+00       1.000000
25%      7.002100e+04      70.000000  6.760050e+04      28.000000
50%      1.303570e+05     130.000000  1.351365e+05      56.000000
75%      2.030940e+05     203.000000  2.026998e+05      85.000000
max      2.373711e+06     272.000000  2.415841e+06     114.000000
```

```
            PROD_QTY      TOT_SALES      PACK_SIZE
count  264834.000000  264834.000000  264834.000000
mean        1.905813       7.299346     182.425512
std         0.343436       2.527241      64.325148
min         1.000000       1.500000      70.000000
25%         2.000000       5.400000     150.000000
50%         2.000000       7.400000     170.000000
75%         2.000000       9.200000     175.000000
max         5.000000      29.500000     380.000000
```

[ ]:
```
data.info()
```

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

[ ]:
```
data["DATE"].dtype
```

[ ]:
```
dtype('O')
```

Create a month and year column

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

[ ]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 264834 entries, 0 to 264833
Data columns (total 12 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
```

```
 0   LYLTY_CARD_NBR    264834 non-null  int64
 1   DATE              264834 non-null  datetime64[ns]
 2   STORE_NBR         264834 non-null  int64
 3   TXN_ID            264834 non-null  int64
 4   PROD_NBR          264834 non-null  int64
 5   PROD_NAME         264834 non-null  object
 6   PROD_QTY          264834 non-null  int64
 7   TOT_SALES         264834 non-null  float64
 8   PACK_SIZE         264834 non-null  int64
 9   BRAND             264834 non-null  object
 10  LIFESTAGE         264834 non-null  object
 11  PREMIUM_CUSTOMER  264834 non-null  object
dtypes: datetime64[ns](1), float64(1), int64(6), object(4)
memory usage: 24.2+ MB
```

[ ]:
```
data['Month']=data['DATE'].dt.to_period('M')
```

[ ]:
```
data['MONTH_YEAR']=data['DATE'].dt.strftime('%m-%Y')
data['MONTH_YEAR']
```

[ ]:
```
0         10-2018
1         09-2018
2         03-2019
3         03-2019
4         11-2018
            …
264829    12-2018
264830    10-2018
264831    10-2018
264832    10-2018
264833    12-2018
Name: MONTH_YEAR, Length: 264834, dtype: object
```

[ ]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 264834 entries, 0 to 264833
Data columns (total 14 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   LYLTY_CARD_NBR    264834 non-null  int64
 1   DATE              264834 non-null  datetime64[ns]
 2   STORE_NBR         264834 non-null  int64
 3   TXN_ID            264834 non-null  int64
 4   PROD_NBR          264834 non-null  int64
 5   PROD_NAME         264834 non-null  object
 6   PROD_QTY          264834 non-null  int64
 7   TOT_SALES         264834 non-null  float64
```

```
 8   PACK_SIZE         264834 non-null  int64
 9   BRAND             264834 non-null  object
 10  LIFESTAGE         264834 non-null  object
 11  PREMIUM_CUSTOMER  264834 non-null  object
 12  Month             264834 non-null  period[M]
 13  MONTH_YEAR        264834 non-null  object
dtypes: datetime64[ns](1), float64(1), int64(6), object(5), period[M](1)
memory usage: 28.3+ MB
```

[ ]:
```
data['Month'].min()
```

[ ]:
```
Period('2018-07', 'M')
```

[ ]:
```
data['Month'].max()
```

[ ]:
```
Period('2019-06', 'M')
```

#Analysis on the store id 77,86 and 88

[ ]:
```
#Grouping by store num and month year
store_group=data.groupby(['STORE_NBR','MONTH_YEAR'])
total_group=store_group['TOT_SALES'].sum()
total_group
```

[ ]:
```
STORE_NBR  MONTH_YEAR
1          01-2019       154.80
           02-2019       225.40
           03-2019       192.90
           04-2019       192.90
           05-2019       221.40
                          …
272        08-2018       372.85
           09-2018       304.70
           10-2018       430.60
           11-2018       376.20
           12-2018       403.90
Name: TOT_SALES, Length: 3169, dtype: float64
```

[ ]:
```
#Looking at total sales by store number
store_sales=data.groupby(['STORE_NBR'])
total_sales=store_sales['TOT_SALES'].sum()
total_sales
```

[ ]:
```
STORE_NBR
1         2393.60
2         2005.80
3        12802.45
4        14647.65
```

```
5        9500.80
           …
268      2601.05
269     11221.80
270     11293.95
271      9721.80
272      4653.95
Name: TOT_SALES, Length: 272, dtype: float64
```

Looking for Total Sales in trial store

[ ]:
```
trial_store=total_sales[76:88]
trial_store
```

[ ]:
```
STORE_NBR
77      3040.00
78      9381.25
79     11831.20
80     11756.90
81     14361.95
82      4103.50
83      9924.90
84      5396.30
85        13.90
86     10635.35
87      3991.60
88     16333.25
Name: TOT_SALES, dtype: float64
```

Total sales in Trial Store 77-3040.00, store 86-10635.35 and store 88-16217.05.

Similar control stores will be identified using sales similarity and correlation analysis to measure trial impact accurately.

#Storing stores by total sales looking for a match for store 77.

[ ]:
```
total_sorted=total_sales.sort_values(ascending=True)
total_sorted.iloc[57:75]
```

[ ]:
```
STORE_NBR
41      2570.20
268     2601.05
195     2608.25
163     2635.70
6       2684.90
53      2715.05
214     2720.40
176     2752.90
233     2826.90
```

```
255    2835.30
185    2868.60
187    2909.70
205    2966.80
220    3008.20
50     3009.80
46     3023.45
141    3025.40
77     3040.00
Name: TOT_SALES, dtype: float64
```

#Isolating the stores

```
[ ]: stores_control_one=[41, 268, 195, 163, 6, 53,  214, 176, 233,  255, 185, 187, ␣
     ↪205, 220, 50, 46,  141, 77]
     control_one = pd.DataFrame({'value': total_group[stores_control_one]})
     print(control_one)
```

```
                      value
STORE_NBR MONTH_YEAR
41        01-2019     169.0
          02-2019     234.6
          03-2019     226.2
          04-2019     231.3
          05-2019     258.8
…                     …
77        08-2018     255.5
          09-2018     225.2
          10-2018     204.5
          11-2018     245.3
          12-2018     267.3

[216 rows x 1 columns]
```

#Putting the stores in a pivot chart format

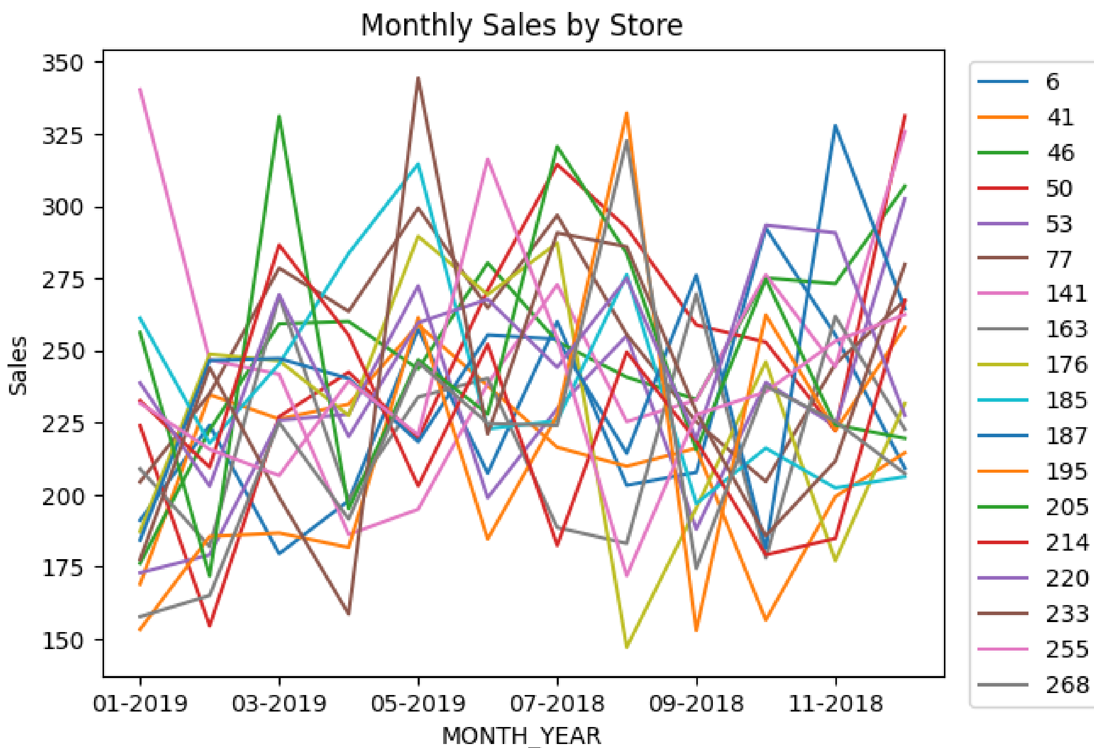
```
[ ]: pivot_chart=control_one.
     ↪pivot_table(index="MONTH_YEAR",columns="STORE_NBR",values="value")
     pivot_chart
```

```
[ ]: STORE_NBR     6      41      46      50      53      77     141    163    176  \
     MONTH_YEAR
     01-2019     191.1  169.0  176.20  223.9  172.90  204.4  340.3  208.9  187.2
     02-2019     224.0  234.6  222.40  154.5  179.10  235.0  246.7  182.0  248.7
     03-2019     179.5  226.2  259.20  227.0  225.80  278.5  241.7  268.8  246.4
     04-2019     197.9  231.3  260.00  242.4  227.80  263.5  186.2  198.3  227.4
     05-2019     257.3  258.8  243.55  219.5  272.35  299.3  194.9  233.8  289.5
     06-2019     207.4  237.7  280.30  270.8  198.90  264.7  238.4  240.3  269.3
```

07-2018	260.0	216.4	253.00	314.4	229.80	296.8	272.8	188.6	287.2
08-2018	203.2	209.8	240.70	292.4	255.10	255.5	225.3	183.1	147.1
09-2018	207.7	216.1	233.00	258.8	188.00	225.2	232.8	269.5	195.4
10-2018	292.4	156.5	275.10	252.8	238.90	204.5	276.2	178.0	246.0
11-2018	255.3	199.3	273.10	222.1	223.80	245.3	244.3	261.8	177.1
12-2018	209.1	214.5	306.90	331.2	302.60	267.3	325.8	222.6	231.6

STORE_NBR	185	187	195	205	214	220	233	255	268
MONTH_YEAR									
01-2019	261.1	184.2	153.30	256.1	232.5	238.7	177.5	231.7	157.70
02-2019	217.8	246.5	185.50	171.8	209.5	202.9	244.0	215.7	165.00
03-2019	245.3	247.2	186.70	331.1	286.5	269.3	199.1	206.6	225.00
04-2019	283.6	240.2	181.70	195.1	255.2	220.1	158.6	239.4	191.50
05-2019	314.6	218.3	261.30	246.7	203.0	259.6	344.4	220.7	245.80
06-2019	222.8	255.3	184.60	227.9	252.1	267.7	221.0	316.3	224.70
07-2018	225.6	253.9	227.50	320.6	182.4	244.1	290.7	254.1	224.00
08-2018	276.3	214.3	332.25	283.6	249.4	275.0	285.9	171.9	322.65
09-2018	196.9	276.1	152.90	215.5	218.6	219.3	228.6	227.7	174.40
10-2018	216.1	181.4	262.30	274.7	179.1	293.4	185.7	235.6	237.60
11-2018	202.3	327.9	222.20	224.2	184.8	290.7	211.6	253.2	225.40
12-2018	206.2	264.4	258.00	219.5	267.3	227.4	279.8	262.4	207.30

[ ]:
```python
pivot_chart.plot()
plt.title("Monthly Sales by Store")
plt.legend(loc="upper right",bbox_to_anchor=(1.20,1))
plt.ylabel("Sales")
plt.show()
```



Control stores show stable sales with seasonal peaks. Some stores have similar trends, making them good matches for comparison. Minor outliers may reflect local factors.

#Looking at correlation

[ ]:
```
control_pivot=pivot_chart.corr(method="pearson")
control_pivot
```

[ ]:

STORE_NBR	6	41	46	50	53	77 \
STORE_NBR						
6	1.000000	-0.247151	0.256520	0.006834	0.242594	-0.021268
41	-0.247151	1.000000	0.164603	-0.119241	0.167031	0.762292
46	0.256520	0.164603	1.000000	0.503370	0.650741	0.386913
50	0.006834	-0.119241	0.503370	1.000000	0.560896	0.304387
53	0.242594	0.167031	0.650741	0.560896	1.000000	0.526309
77	-0.021268	0.762292	0.386913	0.304387	0.526309	1.000000
141	-0.027162	-0.644727	-0.113383	0.277132	-0.042187	-0.413535
163	-0.295525	0.275608	0.165461	-0.068682	-0.074408	0.167020
176	0.345540	0.450519	0.269525	-0.021411	0.140227	0.531159
185	-0.155127	0.339814	-0.330201	-0.155053	0.238337	0.373824
187	-0.041647	0.349995	0.420943	0.052646	0.004825	0.285749
195	0.398130	-0.047535	0.374234	0.423526	0.763772	0.271905
205	0.088312	-0.237444	0.005459	0.374344	0.209564	0.291275

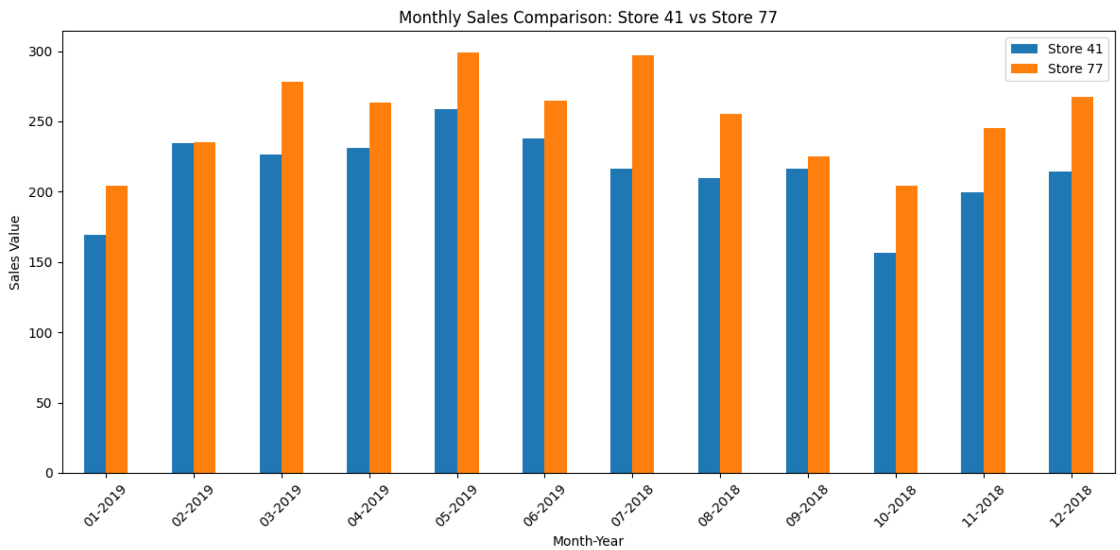
214	-0.878726	0.292472	0.133498	0.186751	0.141150	0.208531
220	0.416445	-0.341097	0.322455	0.141485	0.265352	0.013562
233	0.270639	0.500753	0.116010	0.284899	0.546609	0.613063
255	0.132702	0.069930	0.457896	0.264615	-0.080768	0.099836
268	0.219004	0.064578	0.348140	0.404818	0.583553	0.372558

STORE_NBR	141	163	176	185	187	195	\
STORE_NBR							
6	-0.027162	-0.295525	0.345540	-0.155127	-0.041647	0.398130	
41	-0.644727	0.275608	0.450519	0.339814	0.349995	-0.047535	
46	-0.113383	0.165461	0.269525	-0.330201	0.420943	0.374234	
50	0.277132	-0.068682	-0.021411	-0.155053	0.052646	0.423526	
53	-0.042187	-0.074408	0.140227	0.238337	0.004825	0.763772	
77	-0.413535	0.167020	0.531159	0.373824	0.285749	0.271905	
141	1.000000	-0.152094	-0.125022	-0.434634	-0.198275	-0.090739	
163	-0.152094	1.000000	-0.063802	-0.216258	0.600451	-0.399545	
176	-0.125022	-0.063802	1.000000	0.089027	-0.097138	-0.118839	
185	-0.434634	-0.216258	0.089027	1.000000	-0.520243	0.258070	
187	-0.198275	0.600451	-0.097138	-0.520243	1.000000	-0.197535	
195	-0.090739	-0.399545	-0.118839	0.258070	-0.197535	1.000000	
205	0.163641	0.010123	0.118971	0.157118	-0.283332	0.305944	
214	-0.004689	0.260964	-0.152799	0.205429	-0.030984	-0.121233	
220	-0.060033	0.137705	-0.108080	0.043941	-0.064919	0.490751	
233	-0.127935	-0.061831	0.292296	0.236836	0.063592	0.579931	
255	0.205388	0.217826	0.425772	-0.413567	0.337902	-0.344951	
268	-0.324463	-0.155875	-0.149257	0.328475	-0.119832	0.872191	

STORE_NBR	205	214	220	233	255	268
STORE_NBR						
6	0.088312	-0.878726	0.416445	0.270639	0.132702	0.219004
41	-0.237444	0.292472	-0.341097	0.500753	0.069930	0.064578
46	0.005459	0.133498	0.322455	0.116010	0.457896	0.348140
50	0.374344	0.186751	0.141485	0.284899	0.264615	0.404818
53	0.209564	0.141150	0.265352	0.546609	-0.080768	0.583553
77	0.291275	0.208531	0.013562	0.613063	0.099836	0.372558
141	0.163641	-0.004689	-0.060033	-0.127935	0.205388	-0.324463
163	0.010123	0.260964	0.137705	-0.061831	0.217826	-0.155875
176	0.118971	-0.152799	-0.108080	0.292296	0.425772	-0.149257
185	0.157118	0.205429	0.043941	0.236836	-0.413567	0.328475
187	-0.283332	-0.030984	-0.064919	0.063592	0.337902	-0.119832
195	0.305944	-0.121233	0.490751	0.579931	-0.344951	0.872191
205	1.000000	0.040112	0.547007	0.139341	-0.245834	0.493803
214	0.040112	1.000000	-0.200395	-0.178348	-0.054623	0.029985
220	0.547007	-0.200395	1.000000	-0.019430	0.020966	0.681772
233	0.139341	-0.178348	-0.019430	1.000000	-0.157443	0.450031
255	-0.245834	-0.054623	0.020966	-0.157443	1.000000	-0.308446
268	0.493803	0.029985	0.681772	0.450031	-0.308446	1.000000

store 41 and 77 has the strongest correlation at 0.762.

[ ]:
```
pivot_chart[[41, 77]].plot(kind='bar', figsize=(12,6))
plt.title("Monthly Sales Comparison: Store 41 vs Store 77")
plt.xlabel("Month-Year")
plt.ylabel("Sales Value")
plt.xticks(rotation=45)
plt.legend(["Store 41", "Store 77"])
plt.tight_layout()
plt.show()
```



Check correlations on entire table

[ ]:
```
total_grp_df=pd.DataFrame(total_group)
total_grp_pivot=total_grp_df.
↪pivot_table(index='MONTH_YEAR',columns='STORE_NBR',values='TOT_SALES')
total_grp_pivot_tb=total_grp_pivot.corr(method='pearson')
total_grp_pivot_tb[77].sort_values(ascending=False).head(10)
```

[ ]:
```
STORE_NBR
31     1.000000
77     1.000000
11     1.000000
41     0.762292
35     0.699708
167    0.696075
184    0.645118
63     0.633858
234    0.632204
```

```
20     0.620701
Name: 77, dtype: float64
```

These are the 10 correlations to store 77. Store 41 would be ranked in 3RD place

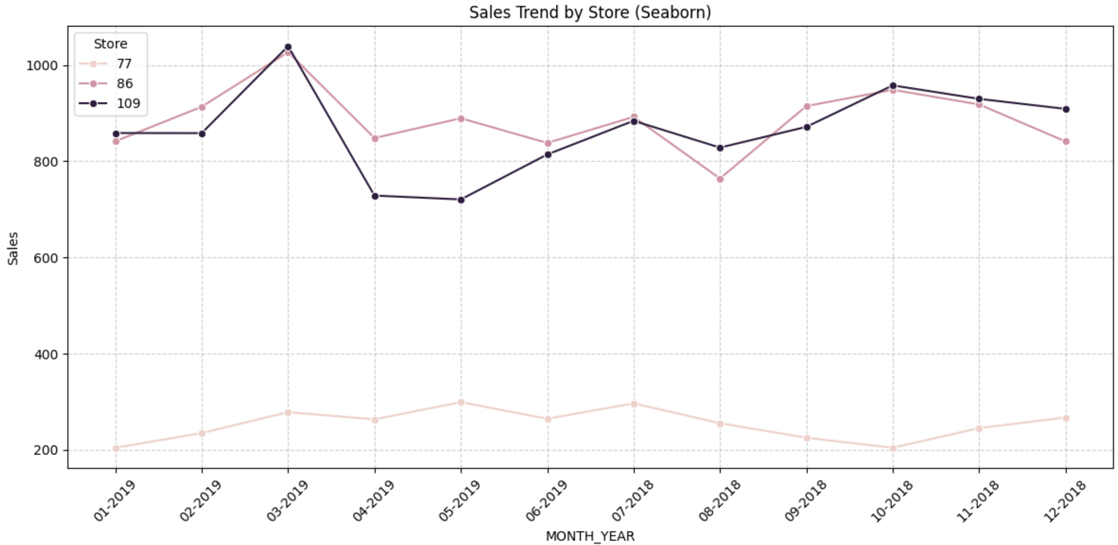
```
[ ]: total_sorted.loc[[31,11,41,35]]
```

```
[ ]: STORE_NBR
     31      14.8
     11       6.7
     41    2570.2
     35    1608.9
     Name: TOT_SALES, dtype: float64
```

Store 31 and 11 sales are way too low to use

```
[ ]: three_77 = total_group[[77, 109, 86]]  # Example: Replace with correct store␣
      ↪numbers
     amigos_77_df = pd.DataFrame(three_77)

     amigo_77_pivot = amigos_77_df.pivot_table(index='MONTH_YEAR',␣
      ↪columns='STORE_NBR', values='TOT_SALES')
```

```
[ ]: #store 41,35,77 from total group dataframe
     df_long = amigo_77_pivot.reset_index().melt(id_vars='MONTH_YEAR',␣
      ↪var_name='Store', value_name='Sales')

     plt.figure(figsize=(12, 6))
     sns.lineplot(data=df_long, x='MONTH_YEAR', y='Sales', hue='Store', marker='o')
     plt.title("Sales Trend by Store (Seaborn)")
     plt.xticks(rotation=45)
     plt.grid(True, linestyle='--', alpha=0.6)
     plt.tight_layout()
     plt.show()
```



Store 77 has the highest and most variable sales. Store 41 shows a similar trend at a lower range, while Store 35 is more stable and lower. Store 41 aligns better as a control.

#Sorting stores by total sales Looikng for a match for score 86

[ ]:
```
total_sorted.iloc[178:201]
```

[ ]:
```
STORE_NBR
109    10399.10
191    10404.70
196    10408.20
229    10417.90
97     10432.05
102    10440.70
105    10472.50
232    10485.30
57     10532.30
172    10545.60
113    10551.60
225    10566.60
62     10583.10
236    10621.00
227    10622.50
155    10628.95
86     10635.35
247    10651.50
13     10686.50
164    10718.90
106    10742.60
```

```
55     10760.15
138    10824.80
Name: TOT_SALES, dtype: float64
```

Isolating the stores

[ ]:
```
stores_control_two=[109,191,196,229,97,102,105,232,57,172,113,225,63,236,227,155,86,247,13,164
control_two = pd.DataFrame({"value": total_group[stores_control_two]})
print(control_two)
```

```
                     value
STORE_NBR MONTH_YEAR
109       01-2019      858.6
          02-2019      858.4
          03-2019     1039.2
          04-2019      728.6
          05-2019      720.6
…                      …
138       08-2018      707.4
          09-2018      913.6
          10-2018     1015.4
          11-2018      991.4
          12-2018      918.0

[276 rows x 1 columns]
```

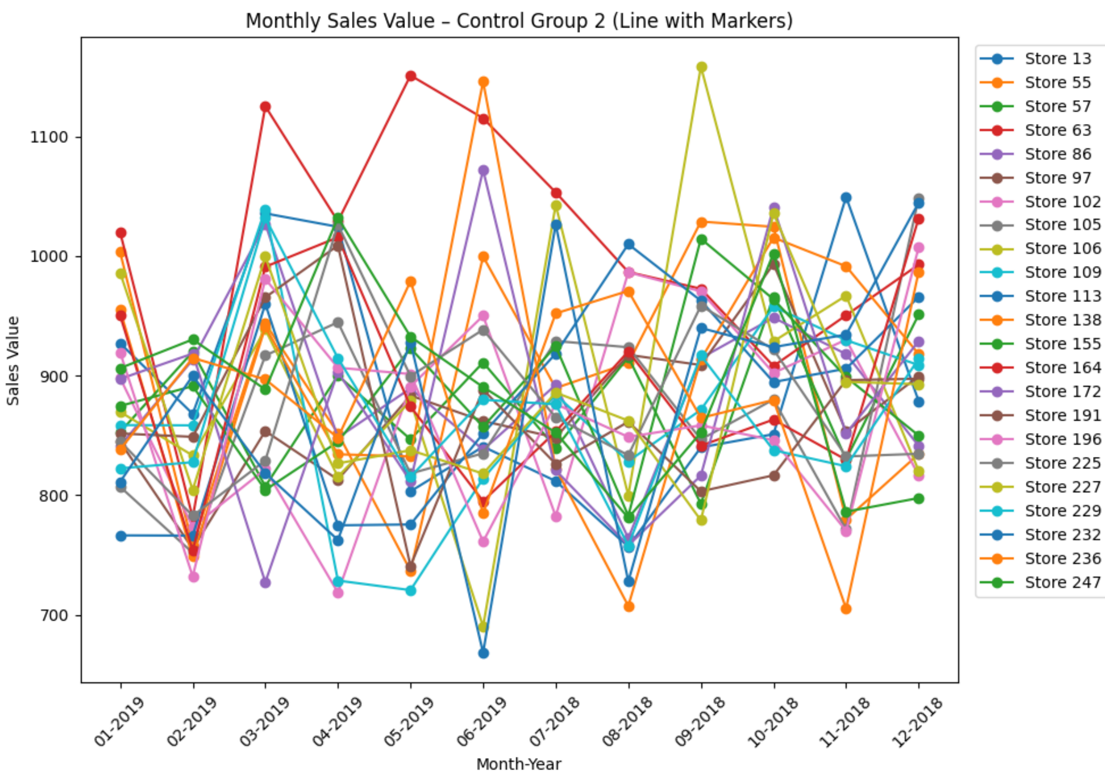
Putting the store in pivot table format

[ ]:
```
control_pivot_chart2=control_two.
 ↪pivot_table(index='MONTH_YEAR',columns='STORE_NBR',values='value')
control_pivot_chart2
```

[ ]:
```
STORE_NBR      13       55     57      63       86      97     102     105  \
MONTH_YEAR
01-2019      927.0  1003.20  852.8  1019.8   841.40  844.60  898.0   807.0
02-2019      868.0   757.80  919.8   781.2   913.20  755.20  773.4   751.8
03-2019     1035.6   943.60  807.4  1125.2  1026.80  853.60  821.8   916.8
04-2019     1024.4   851.80  900.0  1029.0   848.20  813.00  718.6   944.6
05-2019      803.2   736.85  846.7  1151.0   889.30  883.30  890.9   818.1
06-2019      840.6   999.60  911.0  1115.2   838.00  862.00  950.0   835.0
07-2018      811.8   889.60  839.6  1053.2   892.20  848.20  782.4   928.9
08-2018      756.9   910.30  915.4   986.6   764.05  917.35  986.4   923.7
09-2018      840.0  1028.80  792.8   972.6   914.60  908.80  970.4   846.6
10-2018      851.0  1024.40  965.8   908.0   948.40  993.20  902.2   880.0
11-2018     1049.4   779.80  830.0   950.4   918.00  853.40  930.0   771.4
12-2018      878.6   834.40  951.0   992.8   841.20  899.40  816.6  1048.6

STORE_NBR       106      109  …      164      172      191      196      225  \
```

```
MONTH_YEAR                           …
01-2019       869.60   858.6  …   950.2   897.2   851.6   919.4   845.0
02-2019       833.20   858.4  …   753.8   918.4   848.8   732.0   782.8
03-2019       938.60  1039.2  …   991.0   727.2   965.4   980.8   829.0
04-2019       815.40   728.6  …  1015.6   903.0  1008.8   906.6  1026.2
05-2019       878.75   720.6  …   874.1   811.6   740.9   901.3   899.6
06-2019       690.20   814.0  …   795.0  1072.0   888.2   761.2   938.4
07-2018      1042.80   884.0  …   853.2   820.8   826.2   876.2   865.0
08-2018       799.85   828.3  …   920.2   758.0   861.4   848.7   833.4
09-2018      1158.40   871.4  …   841.4   816.4   803.2   858.4   958.4
10-2018       928.60   957.6  …   863.2  1040.8   816.6   846.0   921.8
11-2018       966.80   929.6  …   829.6   851.4   896.2   770.2   832.4
12-2018       820.40   908.8  …  1031.6   928.8   897.4  1007.4   834.6

STORE_NBR       227     229     232    236     247
MONTH_YEAR
01-2019       986.0   822.4   811.2  838.2   906.2
02-2019       804.4   827.6   899.9  914.8   930.2
03-2019       999.4  1031.8   818.4  896.8   888.4
04-2019       826.6   914.6   762.6  848.0  1032.0
05-2019       837.6   815.3   928.0  979.0   932.5
06-2019       818.0   879.8   668.2  785.0   890.4
07-2018       885.8   876.0  1026.7  952.0   852.4
08-2018       862.3   757.8   727.9  970.8   781.0
09-2018       779.0   916.8   939.8  864.6   852.4
10-2018      1035.8   837.6   923.8  879.6  1002.2
11-2018       894.8   824.2   934.2  705.2   786.2
12-2018       892.8   914.0  1044.6  987.0   797.6

[12 rows x 23 columns]
```

[ ]:
```python
plt.figure(figsize=(10, 7))

for store in control_pivot_chart2.columns:
    plt.plot(control_pivot_chart2.index, control_pivot_chart2[store], marker='o', label=f'Store {store}')

plt.title('Monthly Sales Value - Control Group 2 (Line with Markers)')
plt.xlabel('Month-Year')
plt.ylabel('Sales Value')
plt.xticks(rotation=45)
plt.legend(loc='upper right', bbox_to_anchor=(1.18, 1))
plt.tight_layout()
plt.show()
```



Control Group 2 stores show sales mostly ranging between 800 and 1100. Despite some spikes (e.g., Store 106 above 1150), most stores follow a consistent monthly trend, making them reliable for comparison.

[ ]:
```
control_pivot_chart2.corr(method='pearson')
```

[ ]:
```
STORE_NBR        13        55        57        63        86        97   \
STORE_NBR
13         1.000000 -0.125341 -0.291218  0.032720  0.457947 -0.373037
55        -0.125341  1.000000 -0.039301  0.127272  0.043906  0.495256
57        -0.291218 -0.039301  1.000000 -0.394650 -0.402687  0.221201
63         0.032720  0.127272 -0.394650  1.000000 -0.015763  0.127964
86         0.457947  0.043906 -0.402687 -0.015763  1.000000 -0.015617
97        -0.373037  0.495256  0.221201  0.127964 -0.015617  1.000000
102       -0.377415  0.418809 -0.139586  0.094373 -0.226422  0.578719
105       -0.059766  0.124132  0.301428  0.277212 -0.202451  0.334039
106        0.049336  0.181864 -0.658612 -0.101670  0.510548  0.203434
109        0.324289  0.326968 -0.124668 -0.191460  0.643075  0.241536
113       -0.161963  0.306164 -0.087082  0.053762  0.043835  0.548974
138        0.284311  0.500047 -0.001387  0.279260  0.250447  0.286776
155       -0.228967  0.174382 -0.232252 -0.246829  0.326149  0.275949
164        0.357477  0.060884  0.060840  0.372032 -0.117970  0.140764
172       -0.091999  0.250338  0.665384 -0.240828 -0.156398  0.128774
```

191	0.733656	0.018181	0.081015	0.075825	0.043345	-0.359215
196	0.166098	0.101949	-0.113210	0.482153	0.081832	0.240357
225	0.043419	0.338013	-0.005863	0.314820	-0.109479	0.224941
227	0.289917	0.354941	0.106827	0.067017	0.393785	0.403000
229	0.508201	0.234072	-0.335684	0.356381	0.596886	-0.120038
232	-0.084443	-0.320462	-0.100878	-0.252797	0.327006	0.141757
236	-0.597718	-0.206578	0.237461	0.076634	-0.164982	0.162069
247	0.167139	0.096625	0.237256	-0.021229	0.250601	-0.106598

STORE_NBR	102	105	106	109	…	164	172	\
STORE_NBR					…			
13	-0.377415	-0.059766	0.049336	0.324289	…	0.357477	-0.091999	
55	0.418809	0.124132	0.181864	0.326968	…	0.060884	0.250338	
57	-0.139586	0.301428	-0.658612	-0.124668	…	0.060840	0.665384	
63	0.094373	0.277212	-0.101670	-0.191460	…	0.372032	-0.240828	
86	-0.226422	-0.202451	0.510548	0.643075	…	-0.117970	-0.156398	
97	0.578719	0.334039	0.203434	0.241536	…	0.140764	0.128774	
102	1.000000	-0.303843	0.088393	0.057036	…	-0.324841	0.000426	
105	-0.303843	1.000000	-0.084228	0.117184	…	0.754963	-0.099642	
106	0.088393	-0.084228	1.000000	0.363415	…	-0.132514	-0.452421	
109	0.057036	0.117184	0.363415	1.000000	…	0.065696	-0.115734	
113	0.388871	0.519296	0.331634	0.559222	…	0.190053	-0.359782	
138	0.317674	-0.120865	-0.089779	0.307574	…	-0.103974	0.600687	
155	0.171003	-0.345718	0.684674	-0.009058	…	-0.494106	0.214429	
164	-0.324841	0.754963	-0.132514	0.065696	…	1.000000	-0.275936	
172	0.000426	-0.099642	-0.452421	-0.115734	…	-0.275936	1.000000	
191	-0.454167	0.374381	-0.327944	0.179012	…	0.537125	-0.004155	
196	-0.283326	0.730895	0.110802	0.170177	…	0.894100	-0.363440	
225	-0.023039	0.169544	0.053068	-0.481235	…	0.134784	0.304429	
227	-0.009479	0.159843	0.054562	0.652795	…	0.365061	0.042652	
229	-0.406497	0.407354	0.233852	0.437263	…	0.417885	-0.147605	
232	-0.251850	0.176014	0.599607	0.287077	…	-0.019865	-0.136244	
236	-0.245020	0.520565	-0.022502	-0.130827	…	0.294461	-0.329880	
247	-0.460621	-0.131195	-0.155990	-0.296431	…	0.033993	0.403578	

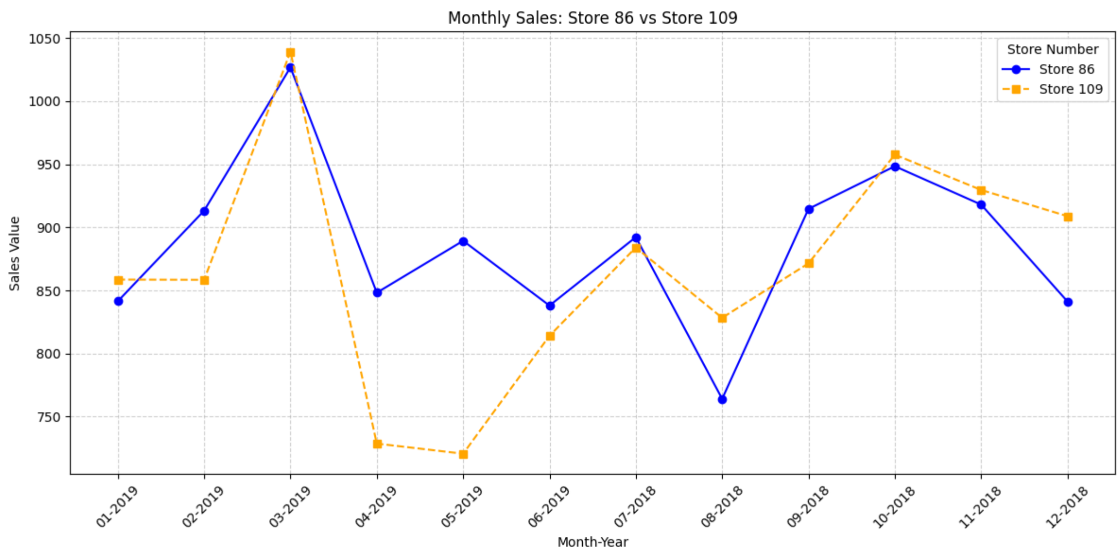
STORE_NBR	191	196	225	227	229	232	\
STORE_NBR							
13	0.733656	0.166098	0.043419	0.289917	0.508201	-0.084443	
55	0.018181	0.101949	0.338013	0.354941	0.234072	-0.320462	
57	0.081015	-0.113210	-0.005863	0.106827	-0.335684	-0.100878	
63	0.075825	0.482153	0.314820	0.067017	0.356381	-0.252797	
86	0.043345	0.081832	-0.109479	0.393785	0.596886	0.327006	
97	-0.359215	0.240357	0.224941	0.403000	-0.120038	0.141757	
102	-0.454167	-0.283326	-0.023039	-0.009479	-0.406497	-0.251850	
105	0.374381	0.730895	0.169544	0.159843	0.407354	0.176014	
106	-0.327944	0.110802	0.053068	0.054562	0.233852	0.599607	
109	0.179012	0.170177	-0.481235	0.652795	0.437263	0.287077	

```
113         0.062261  0.244345 -0.186722  0.139154  0.222562  0.181658
138         0.115548 -0.080774  0.269426  0.311361  0.298551 -0.160849
155        -0.637781 -0.243707  0.313403 -0.134664 -0.076662  0.547102
164         0.537125  0.894100  0.134784  0.365061  0.417885 -0.019865
172        -0.004155 -0.363440  0.304429  0.042652 -0.147605 -0.136244
191         1.000000  0.220600  0.141753  0.111003  0.533908 -0.402733
196         0.220600  1.000000  0.049180  0.402251  0.511064  0.247312
225         0.141753  0.049180  1.000000 -0.265452  0.207075 -0.264708
227         0.111003  0.402251 -0.265452  1.000000  0.141159  0.083682
229         0.533908  0.511064  0.207075  0.141159  1.000000  0.071268
232        -0.402733  0.247312 -0.264708  0.083682  0.071268  1.000000
236        -0.324656  0.489581 -0.219614 -0.037733 -0.033110  0.327655
247         0.105910 -0.013787  0.552432  0.131065  0.136757 -0.216490

STORE_NBR        236       247
STORE_NBR
13         -0.597718  0.167139
55         -0.206578  0.096625
57          0.237461  0.237256
63          0.076634 -0.021229
86         -0.164982  0.250601
97          0.162069 -0.106598
102        -0.245020 -0.460621
105         0.520565 -0.131195
106        -0.022502 -0.155990
109        -0.130827 -0.296431
113         0.190227 -0.679650
138        -0.641239  0.073685
155        -0.127099  0.201025
164         0.294461  0.033993
172        -0.329880  0.403578
191        -0.324656  0.105910
196         0.489581 -0.013787
225        -0.219614  0.552432
227        -0.037733  0.131065
229        -0.033110  0.136757
232         0.327655 -0.216490
236         1.000000 -0.045046
247        -0.045046  1.000000

[23 rows x 23 columns]
```

store 109 and 86 has the strongest correlation at 0.643. I show as Graph

[ ]:
```
control2_graph=control_pivot_chart2[[86,109]]
plt.figure(figsize=(12, 6))
```

```python
plt.plot(control2_graph.index, control2_graph[86], marker='o', linestyle='-',␣
    ↪color='blue', label='Store 86')
plt.plot(control2_graph.index, control2_graph[109], marker='s', linestyle='--',␣
    ↪color='orange', label='Store 109')

plt.title('Monthly Sales: Store 86 vs Store 109')
plt.xlabel('Month-Year')
plt.ylabel('Sales Value')
plt.xticks(rotation=45)
plt.grid(True, linestyle='--', alpha=0.6)
plt.legend(title='Store Number')
plt.tight_layout()
plt.show()
```



[ ]:
```python
total_grp_pivot_tb[86].sort_values(ascending=False).head(10)
```

[ ]:
```
STORE_NBR
31     1.000000
86     1.000000
193    0.933364
159    0.675773
231    0.674071
109    0.643075
132    0.629011
260    0.623775
61     0.617243
229    0.596886
Name: 86, dtype: float64
```

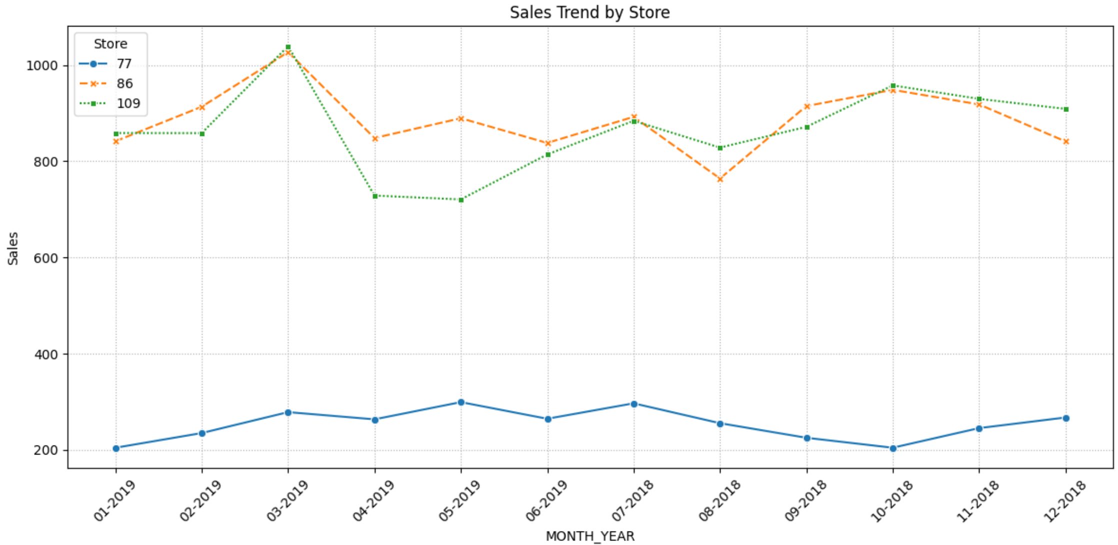
```
[ ]: #Total sales sorted series to see how the sales stack up for the top 5 above by␣
     ↪strongest correlation
     total_sales_sorted=total_sorted.loc[[31,193,159,231,109]]
     total_sales_sorted
```

```
[ ]: STORE_NBR
     31         14.8
     193        13.1
     159       338.9
     231     12996.0
     109     10399.1
     Name: TOT_SALES, dtype: float64
```

Store 31,159 and 193 sales are way too low to use

```
[ ]: three_86 = total_group[[231, 109, 86]]  # Example: Replace with correct store␣
     ↪numbers
     amigos_86_df = pd.DataFrame(three_86)

     amigo_86_pivot = amigos_86_df.pivot_table(index='MONTH_YEAR',␣
     ↪columns='STORE_NBR', values='TOT_SALES')
```

```
[ ]: plt.figure(figsize=(12, 6))
     sns.lineplot(
         data=df_long,
         x='MONTH_YEAR',
         y='Sales',
         hue='Store',
         style='Store',           # Different line styles
         markers=True,
         dashes=True,
         palette='tab10'
     )
     plt.title("Sales Trend by Store")
     plt.xticks(rotation=45)
     plt.grid(True, linestyle=':')
     plt.tight_layout()
     plt.show()
```



The line plot shows strong, consistent sales for Stores 109 and 231, while Store 86 lags behind. Seasonal peaks suggest promotional effects, making Store 86 ideal for uplift testing.

#Sorting stores by total sales looking for a match store 88

[ ]:
```
#total group pivot table to find top 10 correlated store
total_grp_pivot_tb[88].sort_values(ascending=False).head(10)
```

[ ]:
```
STORE_NBR
206    1.000000
88     1.000000
159    0.862608
193    0.836296
201    0.737583
188    0.733516
229    0.707309
228    0.697039
61     0.686658
140    0.613791
Name: 88, dtype: float64
```

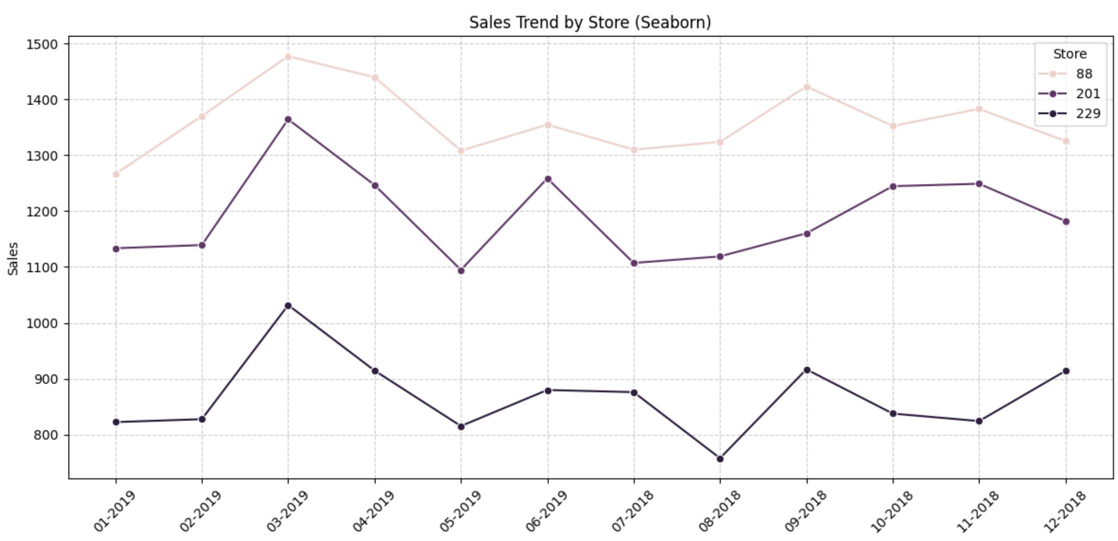
This are the top 10 correlations to store 88

[ ]:
```
#grouping the tortal sales stored series to see hoe the sales stack up for the
↪top 5 Above by strogest correlations

total_sorted.loc[[206,88,159,193,201,188,229,228,61,140]]
```

[ ]:
```
STORE_NBR
206        7.60
88      16333.25
159       338.90
193        13.10
201     14298.70
188      3086.00
229     10417.90
228      4236.30
61        562.90
140       244.90
Name: TOT_SALES, dtype: float64
```

[ ]:
```
three_88 = total_group[[201,229,88]]  # Example: Replace with correct store␣
 ↪numbers
amigos_88_df = pd.DataFrame(three_88)

amigo_88_pivot = amigos_88_df.pivot_table(index='MONTH_YEAR',␣
 ↪columns='STORE_NBR', values='TOT_SALES')
```

[ ]:
```
df_long = amigo_88_pivot.reset_index().melt(id_vars='MONTH_YEAR',␣
 ↪var_name='Store', value_name='Sales')

plt.figure(figsize=(12, 6))
sns.lineplot(data=df_long, x='MONTH_YEAR', y='Sales', hue='Store', marker='o')
plt.title("Sales Trend by Store (Seaborn)")
plt.xticks(rotation=45)
plt.grid(True, linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()
```



Store 201 comes close to the pattern to store 88

[ ]:
```
sorted_88=total_grp_pivot_tb[88].sort_values(ascending=False).head(10)
sorted_88[201]
```

[ ]:
```
np.float64(0.7375831241350634)
```

Store 229 even throught has a good correlation store 201 is a much better fit. store 206 even Though is a best match correlation wise it does not make sense volume. so i will go with store 201

for trial store 88, i will use store number 201 as a control strore . it's a 0.737 correlation.

[ ]:
```
#Creating new dataframe for tiral and control store
#selecting trail and control store from chips

trail_store_77=data.loc[data['STORE_NBR']==77]
control_store_41=data.loc[data['STORE_NBR']==41]

trail_store_86=data.loc[data['STORE_NBR']==86]
control_store_109=data.loc[data['STORE_NBR']==109]

trail_store_88=data.loc[data['STORE_NBR']==88]
control_store_201=data.loc[data['STORE_NBR']==201]
```

[ ]:
```
trail_store_77.head()
```

[ ]:
```
       LYLTY_CARD_NBR        DATE  STORE_NBR  TXN_ID  PROD_NBR  \
73365           77000  2019-03-28         77   74911        18
73366           77000  2019-04-13         77   74912        69
73367           77000  2018-09-26         77   74910        36
73368           77001  2019-02-27         77   74913         7
73369           77001  2019-01-21         77   74914         9

                                    PROD_NAME  PROD_QTY  TOT_SALES  \
73365          Cheetos Chs & Bacon Balls 190g         1        3.3
73366   Smiths Chip Thinly  S/Cream&Onion 175g        1        3.0
73367                       Kettle Chilli 175g         2       10.8
73368        Smiths Crinkle      Original 330g         2       11.4
73369  Kettle Tortilla ChpsBtroot&Ricotta 150g         2        9.2

       PACK_SIZE    BRAND              LIFESTAGE PREMIUM_CUSTOMER    Month  \
73365        190  CHEETOS  MIDAGE SINGLES/COUPLES           Budget  2019-03
73366        175   SMITHS  MIDAGE SINGLES/COUPLES           Budget  2019-04
73367        175   KETTLE  MIDAGE SINGLES/COUPLES           Budget  2018-09
73368        330   SMITHS          YOUNG FAMILIES       Mainstream  2019-02
73369        150   KETTLE          YOUNG FAMILIES       Mainstream  2019-01
```

```
      MONTH_YEAR
73365    03-2019
73366    04-2019
73367    09-2018
73368    02-2019
73369    01-2019
```

Start with store 77 and 41

[ ]:
```
trail_store_77[['TOT_SALES','PROD_QTY']].sum()
```

[ ]:
```
TOT_SALES    3040.0
PROD_QTY      872.0
dtype: float64
```

[ ]:
```
control_store_41[['TOT_SALES','PROD_QTY']].sum()
```

[ ]:
```
TOT_SALES    2570.2
PROD_QTY      723.0
dtype: float64
```

[ ]:
```
#Repeat customer for trail store
trail_store_77['LYLTY_CARD_NBR'].value_counts()
```

[ ]:
```
LYLTY_CARD_NBR
77476    5
77066    4
77313    4
77305    4
77093    4
        ..
77108    1
77298    1
77107    1
77105    1
77277    1
Name: count, Length: 356, dtype: int64
```

[ ]:
```
#Total Customer Transaction
trail_store_77[['LYLTY_CARD_NBR']].count()
```

[ ]:
```
LYLTY_CARD_NBR    563
dtype: int64
```

[ ]:
```
#Repeat customer for control store
control_store_41['LYLTY_CARD_NBR'].value_counts()
```

[ ]:
```
LYLTY_CARD_NBR
41497    4
41453    4
41466    4
41367    4
41359    4
        ..
41471    1
41499    1
41002    1
41001    1
41505    1
Name: count, Length: 344, dtype: int64
```

[ ]:
```
#Total Customer Transaction
control_store_41[['LYLTY_CARD_NBR']].count()
```

[ ]:
```
LYLTY_CARD_NBR    567
dtype: int64
```

[ ]:
```
#Customer repeat customers that purchased more than once
repeat_customers=trail_store_77['LYLTY_CARD_NBR'].value_counts()
repeat_customers.head(24)
```

[ ]:
```
LYLTY_CARD_NBR
77476    5
77066    4
77313    4
77305    4
77093    4
77338    4
77344    4
77205    4
77109    4
77454    4
77280    3
77271    3
77390    3
77402    3
77263    3
77258    3
77281    3
77308    3
77252    3
77049    3
77383    3
77069    3
```

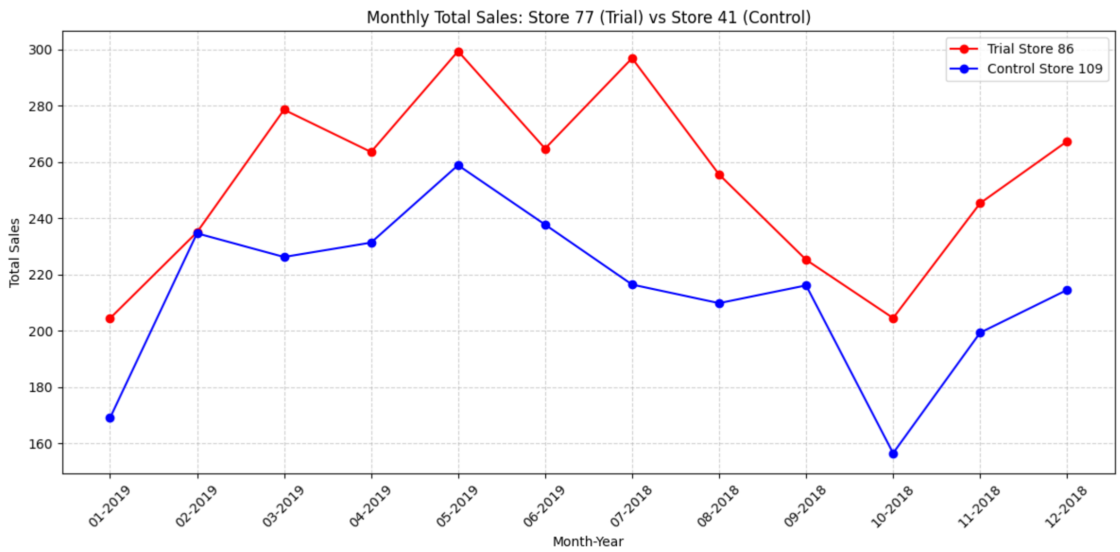
```
77044    3
77287    3
Name: count, dtype: int64
```

[ ]:
```
#Customer repeat customers that purchased more than once
repeat_customer2=control_store_41['LYLTY_CARD_NBR'].value_counts()
repeat_customer2.head(9)
```

[ ]:
```
LYLTY_CARD_NBR
41497    4
41453    4
41466    4
41367    4
41359    4
41368    4
41418    4
41423    4
41432    4
Name: count, dtype: int64
```

[ ]:
```
#Grouping store by Month
group_77=trail_store_77.groupby('MONTH_YEAR')
group_41=control_store_41.groupby('MONTH_YEAR')
```

[ ]:
```
# Grouping by Month-year and summing total sales
sales_77 = group_77['TOT_SALES'].sum()
sales_41 = group_41['TOT_SALES'].sum()

# Plotting
plt.figure(figsize=(12, 6))
plt.plot(sales_77.index, sales_77.values, marker='o', color='red', label="Trial␣
↪Store 86")
plt.plot(sales_41.index, sales_41.values, marker='o',  color='blue',␣
↪label="Control Store 109")

plt.title("Monthly Total Sales: Store 77 (Trial) vs Store 41 (Control)")
plt.xlabel("Month-Year")
plt.ylabel("Total Sales")
plt.xticks(rotation=45)
plt.legend()
plt.grid(True, linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()
```



For the first pair we can see a clear diffrence between the trail store and the control store.

#Start with Store 86 and 109

[ ]:
```
trail_store_86[['TOT_SALES','PROD_QTY']].sum()
```

[ ]:
```
TOT_SALES    10635.35
PROD_QTY      3066.00
dtype: float64
```

[ ]:
```
control_store_109[['TOT_SALES','PROD_QTY']].sum()
```

[ ]:
```
TOT_SALES    10399.1
PROD_QTY      2977.0
dtype: float64
```

[ ]:
```
#Repeat customer for trail store
trail_store_86['LYLTY_CARD_NBR'].value_counts()
```

[ ]:
```
LYLTY_CARD_NBR
86133     13
86112     13
86151     12
86075     12
86008     12
          ..
155000     1
155003     1
155004     1
```

```
155005     1
155510     1
Name: count, Length: 273, dtype: int64
```

[ ]:
```
#Total customer transaction
trail_store_86[['LYLTY_CARD_NBR']].count()
```

[ ]:
```
LYLTY_CARD_NBR    1538
dtype: int64
```

[ ]:
```
#customer repeat customers that purchaed more than once
repeat_customers_86=trail_store_86['LYLTY_CARD_NBR'].value_counts()
repeat_customers_86.iloc[:125]
```

[ ]:
```
LYLTY_CARD_NBR
86133    13
86112    13
86151    12
86075    12
86008    12
         ..
86208     6
86030     6
86031     6
86028     6
86016     6
Name: count, Length: 125, dtype: int64
```

[ ]:
```
#repeat customer for control store
control_store_109['LYLTY_CARD_NBR'].value_counts()
```

[ ]:
```
LYLTY_CARD_NBR
109036    16
109080    14
109086    13
109078    12
109212    12
          ..
109121     1
109017     1
109200     1
109214     1
109222     1
Name: count, Length: 261, dtype: int64
```

[ ]:
```
#total customer tansctions
control_store_109[['LYLTY_CARD_NBR']].count()
```

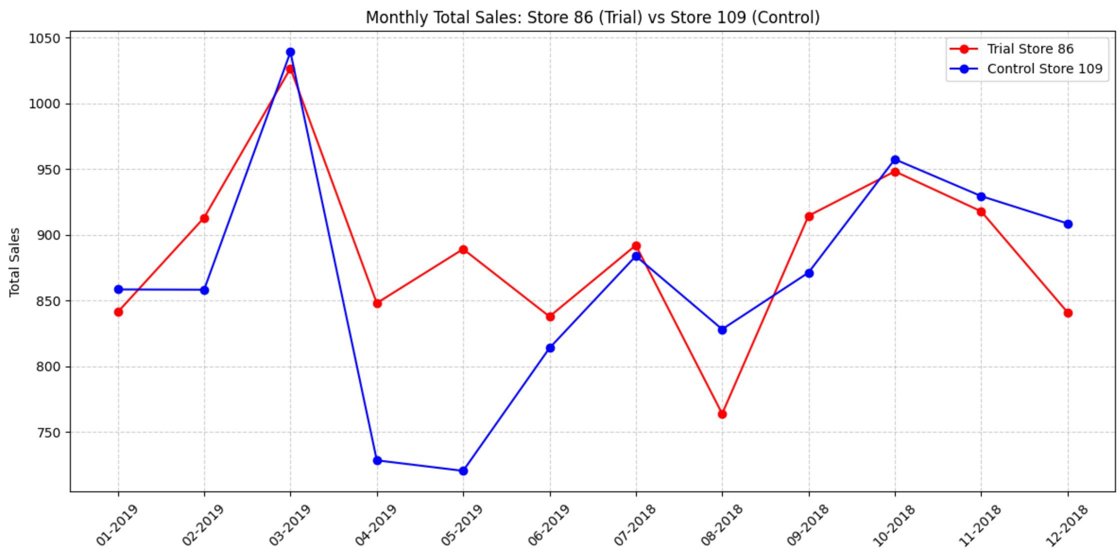
[ ]:
```
LYLTY_CARD_NBR    1505
dtype: int64
```

[ ]:
```
#customer repeat customers that purchaed more than once
repeat_customers_109=control_store_109['LYLTY_CARD_NBR'].value_counts()
repeat_customers_109.iloc[:115]
```

[ ]:
```
LYLTY_CARD_NBR
109036    16
109080    14
109086    13
109078    12
109212    12
          ..
109075     6
109066     6
109065     6
109148     6
109113     6
Name: count, Length: 115, dtype: int64
```

[ ]:
```
#Grouping store by month
group_86=trail_store_86.groupby('MONTH_YEAR')
group_109=control_store_109.groupby('MONTH_YEAR')
```

[ ]:
```
sales_86 = group_86['TOT_SALES'].sum()
sales_109 = group_109['TOT_SALES'].sum()

plt.figure(figsize=(12, 6))
plt.plot(sales_86.index, sales_86.values, marker='o', color='red', label="Trial␣
↪Store 86")
plt.plot(sales_109.index, sales_109.values, marker='o', color='blue',␣
↪label="Control Store 109")

plt.ylabel("Total Sales")
plt.title("Monthly Total Sales: Store 86 (Trial) vs Store 109 (Control)")
plt.xticks(rotation=45)
plt.grid(True, linestyle='--', alpha=0.6)
plt.legend()
plt.tight_layout()
plt.show()
```



For the second pair we can see a clear diffrence between the trail store and the control store.

#Start with store 88 and 201

[ ]:
```
trail_store_88[['TOT_SALES','PROD_QTY']].sum()
```

[ ]:
```
TOT_SALES    16333.25
PROD_QTY      3718.00
dtype: float64
```

[ ]:
```
control_store_201[['TOT_SALES','PROD_QTY']].sum()
```

[ ]:
```
TOT_SALES    14298.7
PROD_QTY      3262.0
dtype: float64
```

[ ]:
```
#Repeat customer for trail store
trail_store_88['LYLTY_CARD_NBR'].value_counts()
```

[ ]:
```
LYLTY_CARD_NBR
88105      13
88247      11
88358      11
88351      10
88348      10
           ..
88355       1
88372       1
2370701     1
```

```
2370751     1
2373711     1
Name: count, Length: 388, dtype: int64
```

[ ]:
```python
#Total customer transaction
trail_store_88[['LYLTY_CARD_NBR']].count()
```

[ ]:
```
LYLTY_CARD_NBR    1873
dtype: int64
```

[ ]:
```python
#Customer repeat customers that purchaed more than once
repeat_customers_88=trail_store_88['LYLTY_CARD_NBR'].value_counts()
repeat_customers_88.iloc[:146]
```

[ ]:
```
LYLTY_CARD_NBR
88105    13
88247    11
88358    11
88351    10
88348    10
         ..
88218     6
88134     6
88194     6
88188     6
88181     6
Name: count, Length: 146, dtype: int64
```

[ ]:
```python
#Total customer tansctions
control_store_201[['LYLTY_CARD_NBR']].count()
```

[ ]:
```
LYLTY_CARD_NBR    1654
dtype: int64
```

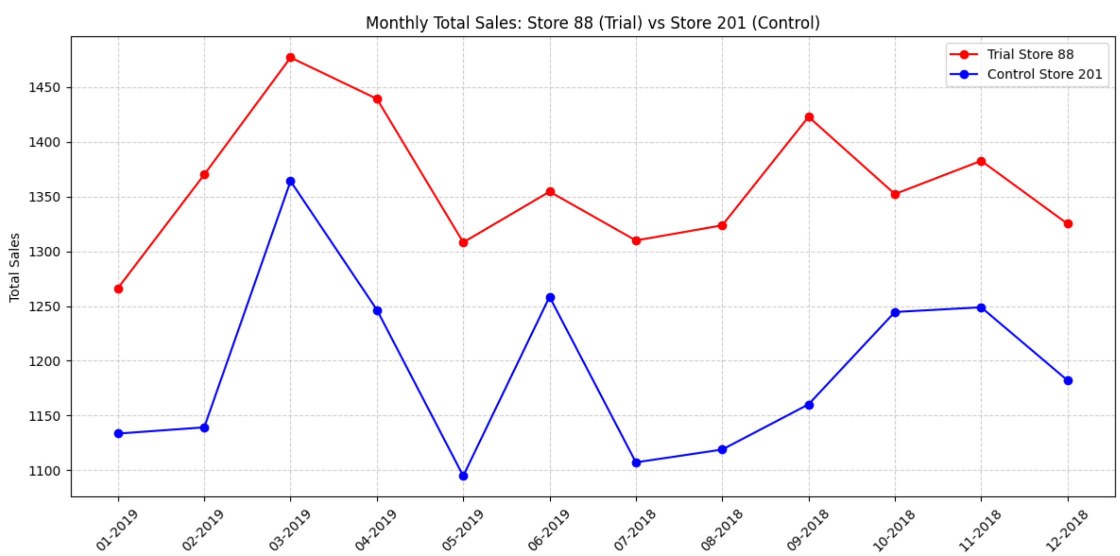
[ ]:
```python
#Customer repeat customers that purchaed more than once
repeat_customers_201=control_store_201['LYLTY_CARD_NBR'].value_counts()
repeat_customers_201.iloc[:110]
```

[ ]:
```
LYLTY_CARD_NBR
201294    13
201120    11
201186    11
201206    10
201018    10
          ..
201347     5
201348     5
```

```
201365     5
201318     5
201161     5
Name: count, Length: 110, dtype: int64
```

```
[ ]: #Grouping store by month

group_88=trail_store_88.groupby('MONTH_YEAR')
group_201=control_store_201.groupby('MONTH_YEAR')
```

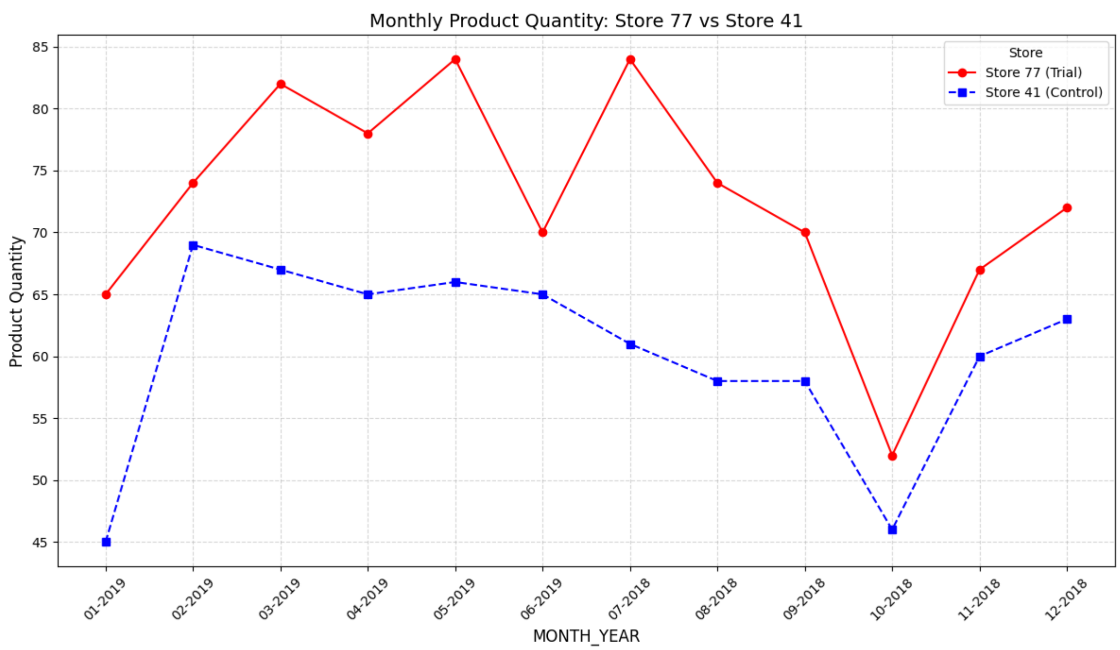
```
[ ]: sales_88= group_88['TOT_SALES'].sum()
sales_201 = group_201['TOT_SALES'].sum()

plt.figure(figsize=(12, 6))
plt.plot(sales_88.index, sales_88.values, marker='o', color='red', label="Trial␣
↪Store 88")
plt.plot(sales_201.index, sales_201.values, marker='o', color='blue',␣
↪label="Control Store 201")

plt.ylabel("Total Sales")
plt.title("Monthly Total Sales: Store 88 (Trial) vs Store 201 (Control)")
plt.xticks(rotation=45)
plt.grid(True, linestyle='--', alpha=0.6)
plt.legend()
plt.tight_layout()
plt.show()
```



For the third pair we can see a clear differnce between the trail store and the control store.

#Visualize the PRoduct Quantity sold

[ ]:
```
# Sum monthly product quantity
qty_77 = group_77['PROD_QTY'].sum()
qty_41 = group_41['PROD_QTY'].sum()

# Plot with styling
plt.figure(figsize=(12, 7))
plt.plot(qty_77.index, qty_77.values, label="Store 77 (Trial)", marker='o',␣
↪linestyle='-', color='red')
plt.plot(qty_41.index, qty_41.values, label="Store 41 (Control)", marker='s',␣
↪linestyle='--', color='blue')

plt.title("Monthly Product Quantity: Store 77 vs Store 41", fontsize=14)
plt.xlabel("MONTH_YEAR", fontsize=12)
plt.ylabel("Product Quantity", fontsize=12)
plt.xticks(rotation=45)
plt.grid(True, linestyle='--', alpha=0.5)
plt.legend(title="Store")
plt.tight_layout()
plt.show()
```

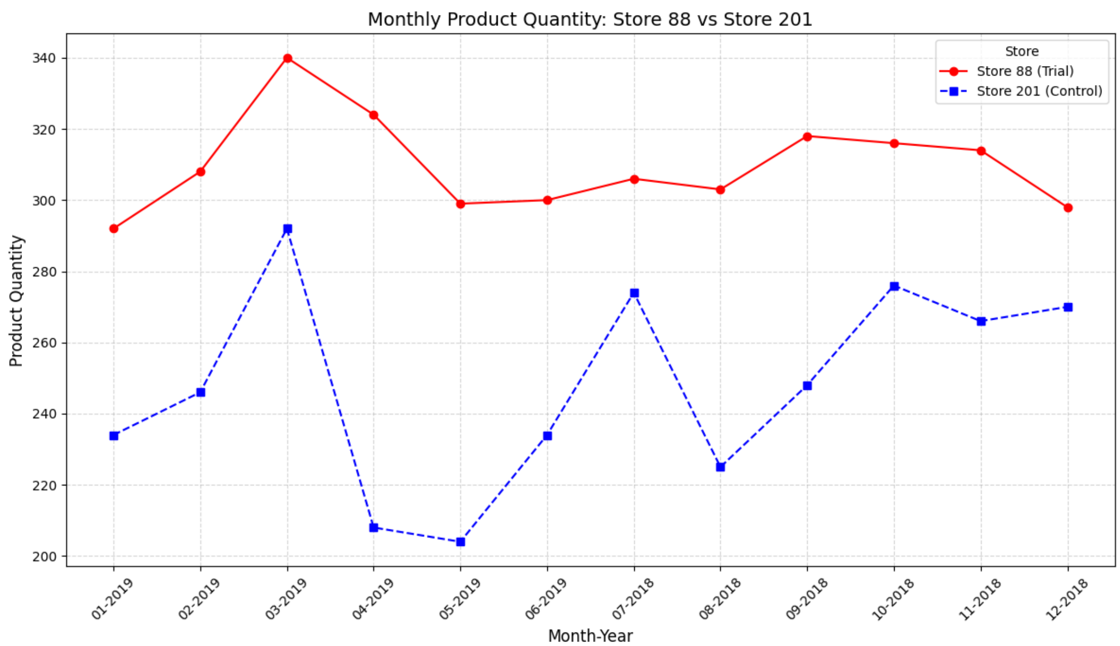


[ ]:
```
qty_86 = group_86['PROD_QTY'].sum()
qty_109 = group_109['PROD_QTY'].sum()

# Plot with styling
```

```
plt.figure(figsize=(12, 7))
plt.plot(qty_86.index, qty_86.values, label="Store 86 (Trial)", marker='o',␣
↪linestyle='-', color='RED')
plt.plot(qty_109.index, qty_109.values, label="Store 109 (Control)",␣
↪marker='s', linestyle='--', color='blue')

plt.title("Monthly Product Quantity: Store 86 vs Store 109", fontsize=14)
plt.xlabel("MONTH_YEAR", fontsize=12)
plt.ylabel("Product Quantity", fontsize=12)
plt.xticks(rotation=45)
plt.grid(True, linestyle='--', alpha=0.5)
plt.legend(title="Store")
plt.tight_layout()
plt.show()
```

[ ]:
```
qty_88 = group_88['PROD_QTY'].sum()
qty_201 = group_109['PROD_QTY'].sum()

# Plot with styling
plt.figure(figsize=(12, 7))
plt.plot(qty_88.index, qty_88.values, label="Store 88 (Trial)", marker='o',␣
↪linestyle='-', color='red')
plt.plot(qty_201.index, qty_201.values, label="Store 201 (Control)",␣
↪marker='s', linestyle='--', color='blue')
```

```python
plt.title("Monthly Product Quantity: Store 88 vs Store 201", fontsize=14)
plt.xlabel("Month-Year", fontsize=12)
plt.ylabel("Product Quantity", fontsize=12)
plt.xticks(rotation=45)
plt.grid(True, linestyle='--', alpha=0.5)
plt.legend(title="Store")
plt.tight_layout()
plt.show()
```



We can see by the graphs above the tral store outperformed the control strores by Quantity sold.

Average transaction per customer

[ ]:
```python
group_77['LYLTY_CARD_NBR'].value_counts().mean().round(3)
```

[ ]: np.float64(1.048)

[ ]:
```python
group_41['LYLTY_CARD_NBR'].value_counts().mean().round(3)
```

[ ]: np.float64(1.05)

[ ]:
```python
group_86['LYLTY_CARD_NBR'].value_counts().mean().round(3)
```

[ ]: np.float64(1.254)

[ ]:
```python
group_109['LYLTY_CARD_NBR'].value_counts().mean().round(3)
```

[ ]: np.float64(1.292)

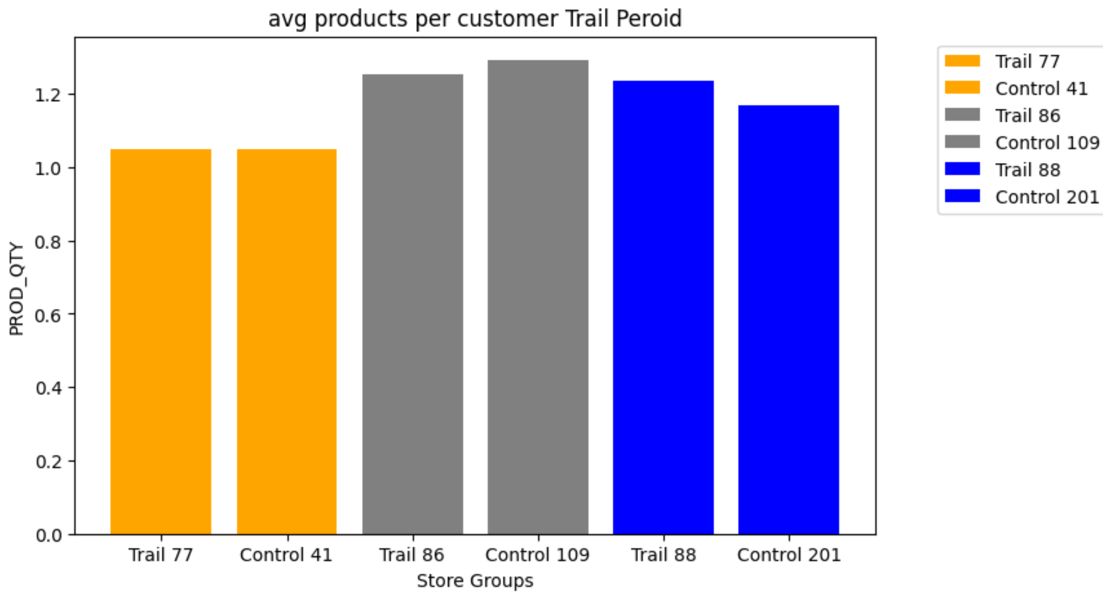
```
[ ]: group_88['LYLTY_CARD_NBR'].value_counts().mean().round(3)
```

[ ]: np.float64(1.236)

```
[ ]: group_201['LYLTY_CARD_NBR'].value_counts().mean().round(3)
```

[ ]: np.float64(1.169)

```
[ ]: group1=["Trail 77","Control 41"]
group2=["Trail 86","Control 109"]
group3=["Trail 88","Control 201"]
value_grp_1=[1.048, 1.05]
value_grp_2=[1.254,1.292]
value_grp_3=[1.236,1.169]


plt.figure(figsize=(8, 5))
plt.bar(group1,value_grp_1,label=group1,color='orange')
plt.bar(group2,value_grp_2,label=group2,color='gray')
plt.bar(group3,value_grp_3,label=group3,color='blue')

# Labels and titles
plt.xlabel("Store Groups")
plt.ylabel('PROD_QTY')
plt.title("avg products per customer Trail Peroid")

plt.legend(loc='upper right', bbox_to_anchor=(1.3, 1))

plt.show()
```



Insights

- **Best Performance-** Trial 86 and Control 109 achieved the highest product quantity per customer, showing strong engagement.
- **No Impact-** Trial 77 and Control 41 had nearly identical values (~1.05), indicating no trial effect.
- **Underperformance-** Trial 88 performed slightly worse than Control 201.
- **Overall-** Only the 86/109 pair shows a clear positive trial impact.