IMPORT REQUIRED LIBRARIES

In [256…

```
import numpy as np
import pandas as pd
import datetime
import matplotlib.pyplot as plt
import scipy.stats as stats
```

IMPORT THE FILES

In [257…

```
dataset = pd.read_csv("QVI_data.csv")
dataset
```

Out[257...

	LYLTY_CARD_NBR	DATE	STORE_NBR	TXN_ID	PROD_NBR	PROD_NAME	PI
0	1000	2018-10-17	1	1	5	Natural Chip Compny SeaSalt175g	
1	1002	2018-09-16	1	2	58	Red Rock Deli Chikn&Garlic Aioli 150g	
2	1003	2019-03-07	1	3	52	Grain Waves Sour Cream&Chives 210G	
3	1003	2019-03-08	1	4	106	Natural ChipCo Hony Soy Chckn175g	
4	1004	2018-11-02	1	5	96	WW Original Stacked Chips 160g	
...	...	...	...	...	...	...	
264829	2370701	2018-12-08	88	240378	24	Grain Waves Sweet Chilli 210g	
264830	2370751	2018-10-01	88	240394	60	Kettle Tortilla ChpsFeta&Garlic 150g	
264831	2370961	2018-10-24	88	240480	70	Tyrrells Crisps Lightly Salted 165g	
264832	2370961	2018-10-27	88	240481	65	Old El Paso Salsa Dip Chnky Tom Ht300g	
264833	2373711	2018-12-14	88	241815	16	Smiths Crinkle Chips Salt & Vinegar 330g	

264834 rows × 12 columns

In [258...

```
dataset["DATE"].dtype
```

Out[258...

```
dtype('O')
```

FIRST LET'S CREATE A MONTH AND YEAR COLUMN

In [259...

```
dataset["DATE"] = pd.to_datetime(dataset["DATE"])
dataset["MONTH_YEAR"] = dataset["DATE"].dt.strftime("%m/%Y")
dataset["MONTH_YEAR"]
```

Out[259…

```
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
```

GROUPING BY STORE NUMBER AND MONTH YEAR

In [260…

```
chips_grp_before = dataset.groupby(["STORE_NBR", "MONTH_YEAR"])
total_grp = chips_grp_before["TOT_SALES"].sum()
total_grp
```

Out[260…

```
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

In [261…

```
chips_grp_sales = dataset.groupby("STORE_NBR")
total_sales = chips_grp_sales["TOT_SALES"].sum()
total_sales
```

Out[261…

```
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 STORES

In [262…

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

Out[262…
```
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 STORES - STORE 77 : $3040.00 - STORE 86$:10635.35 - STORE 88 : $16333.25

NOW SINCE WE HAVE THE TOTAL SALES FOR THE TRIAL STORES. LETS LOOK FOR MATCHING CONTROL STORES FOR EACH. THERE ARE 272 STORES. I WILL USE 2 METHODS TO DETERMINE A CONTROL STORE. I WILL FIRST GO BY TOTAL SALES TO FIND STORES WITH SIMILAR TOTAL SALES. THEN I WILL USE THE PEARSON CORRELATIONS TEST TO DETERMINE HOW CORRELATED THE STORES ARE.

SORTING STORES BY TOTAL SALES LOOKING FOR A MATCH FOR STORE 77

In [263…
```
total_sorted = total_sales.sort_values(ascending=True)
total_sorted.iloc[57:75]
```

Out[263…
```
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

In [264…
```
stores_control_one = [41, 268, 195, 163, 6, 53, 214, 176, 233, 255, 185, 187, 20
control_one = pd.DataFrame({"Value" : total_grp[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

In [265…

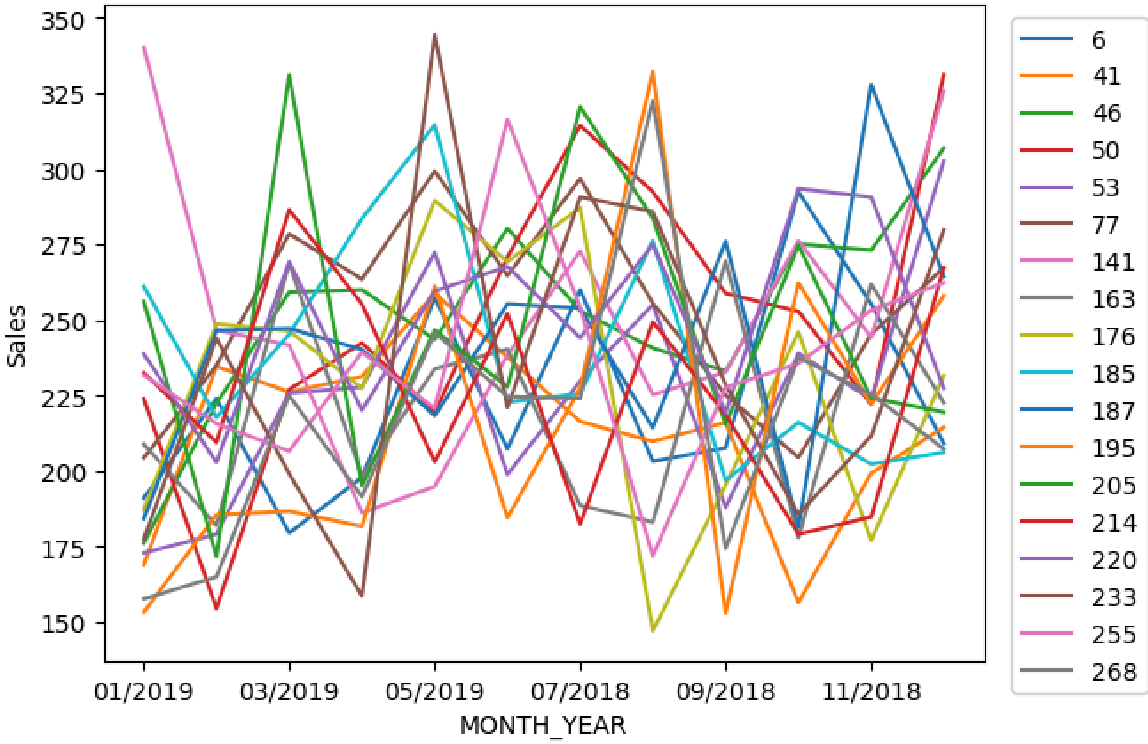
```
pivot_chips1 = control_one.pivot_table(index="MONTH_YEAR", columns="STORE_NBR",
pivot_chips1
```

Out[265…

STORE_NBR	6	41	46	50	53	77	141	163	176	185	1
MONTH_YEAR											
01/2019	191.1	169.0	176.20	223.9	172.90	204.4	340.3	208.9	187.2	261.1	18
02/2019	224.0	234.6	222.40	154.5	179.10	235.0	246.7	182.0	248.7	217.8	24
03/2019	179.5	226.2	259.20	227.0	225.80	278.5	241.7	268.8	246.4	245.3	24
04/2019	197.9	231.3	260.00	242.4	227.80	263.5	186.2	198.3	227.4	283.6	24
05/2019	257.3	258.8	243.55	219.5	272.35	299.3	194.9	233.8	289.5	314.6	21
06/2019	207.4	237.7	280.30	270.8	198.90	264.7	238.4	240.3	269.3	222.8	25
07/2018	260.0	216.4	253.00	314.4	229.80	296.8	272.8	188.6	287.2	225.6	25
08/2018	203.2	209.8	240.70	292.4	255.10	255.5	225.3	183.1	147.1	276.3	21
09/2018	207.7	216.1	233.00	258.8	188.00	225.2	232.8	269.5	195.4	196.9	27
10/2018	292.4	156.5	275.10	252.8	238.90	204.5	276.2	178.0	246.0	216.1	18
11/2018	255.3	199.3	273.10	222.1	223.80	245.3	244.3	261.8	177.1	202.3	32
12/2018	209.1	214.5	306.90	331.2	302.60	267.3	325.8	222.6	231.6	206.2	26

In [266…

```
pivot_chips1.plot()
plt.legend(loc = "upper right", bbox_to_anchor = (1.20, 1))
plt.ylabel("Sales")
plt.show()
```



THAT'S AN UGLY LOOKING LINE CHART. LETS TAKE A CLOSER LOOK AT THE CORRELATIONS BETWEEN THEM AND RECHART JUST THE TWO.

LOOKING AT CORRELATION

In [267…

```
pivot_chips1.corr(method="pearson")
```

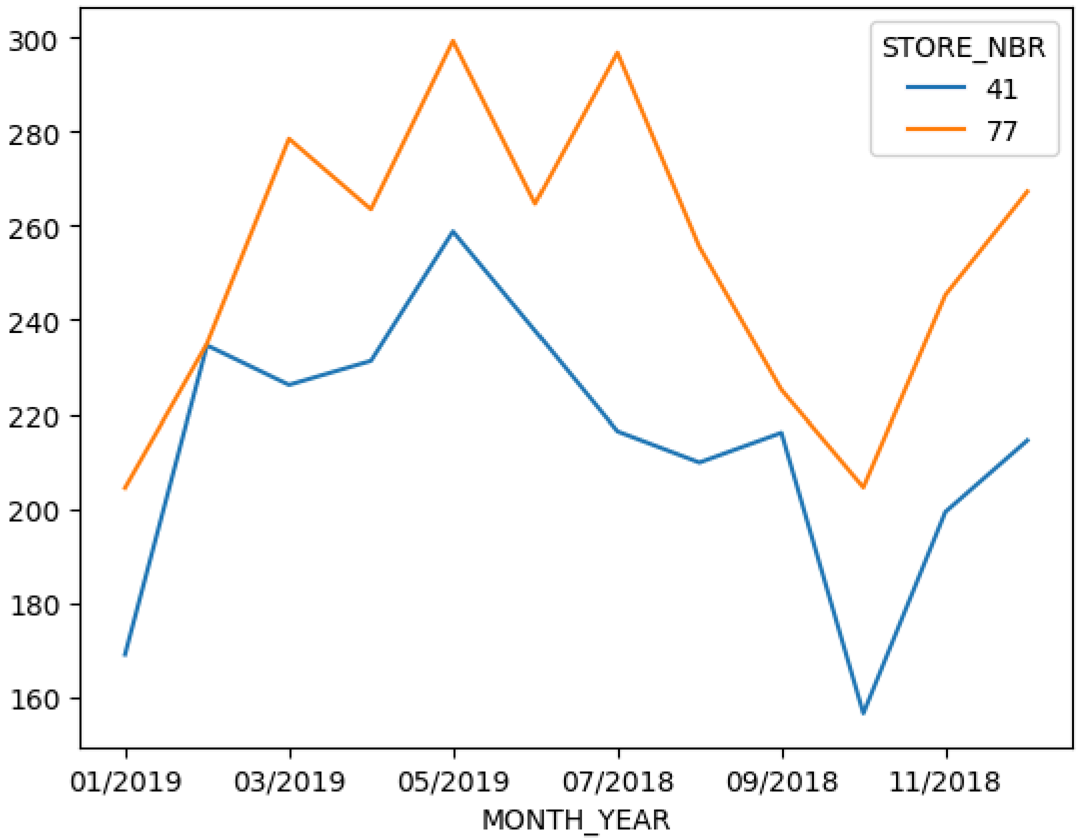
Out[267...

STORE_NBR	6	41	46	50	53	77	141
STORE_NBR							
6	1.000000	-0.247151	0.256520	0.006834	0.242594	-0.021268	-0.027162
41	-0.247151	1.000000	0.164603	-0.119241	0.167031	0.762292	-0.644727
46	0.256520	0.164603	1.000000	0.503370	0.650741	0.386913	-0.113383
50	0.006834	-0.119241	0.503370	1.000000	0.560896	0.304387	0.277132
53	0.242594	0.167031	0.650741	0.560896	1.000000	0.526309	-0.042187
77	-0.021268	0.762292	0.386913	0.304387	0.526309	1.000000	-0.413535
141	-0.027162	-0.644727	-0.113383	0.277132	-0.042187	-0.413535	1.000000
163	-0.295525	0.275608	0.165461	-0.068682	-0.074408	0.167020	-0.152094
176	0.345540	0.450519	0.269525	-0.021411	0.140227	0.531159	-0.125022
185	-0.155127	0.339814	-0.330201	-0.155053	0.238337	0.373824	-0.434634
187	-0.041647	0.349995	0.420943	0.052646	0.004825	0.285749	-0.198275
195	0.398130	-0.047535	0.374234	0.423526	0.763772	0.271905	-0.090739
205	0.088312	-0.237444	0.005459	0.374344	0.209564	0.291275	0.163641
214	-0.878726	0.292472	0.133498	0.186751	0.141150	0.208531	-0.004689
220	0.416445	-0.341097	0.322455	0.141485	0.265352	0.013562	-0.060033
233	0.270639	0.500753	0.116010	0.284899	0.546609	0.613063	-0.127935
255	0.132702	0.069930	0.457896	0.264615	-0.080768	0.099836	0.205388
268	0.219004	0.064578	0.348140	0.404818	0.583553	0.372558	-0.324463

STORE 41 AND 77 HAS HAS THE STRONGEST CORRELATION AT 0.762. LETS GRAPH IT.

In [268...

```
chips1_graph = pivot_chips1[[41, 77]]
chips1_graph.plot()
plt.show()
```



CHECKING CORRELATIONS ON ENTIRE TABLE

In [269…

```
total_grp_df = pd.DataFrame(total_grp)
total_grp_pivot = total_grp_df.pivot_table(index="MONTH_YEAR", columns="STORE_NB
total_grp_pivot_table = total_grp_pivot.corr(method="pearson")
total_grp_pivot_table[77].sort_values(ascending=False).head(10)
```

Out[269…

```
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 TOP 10 CORRELATIONS TO STORE 77. STORE 41 WOULD BE RANKED IN 3RD PLACE. LETS LOOK AT THE OTHER STORES BY TOTAL SALES BEFORE I MAKE A DECISION.

In [270…

```
# GRABBING THE TOTAL SALES SORTED SERIES TO SEE HOW THE SALES STACK UP FOR THE T
total_sorted.loc[[31, 11, 41, 35]]
```

Out[270…

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

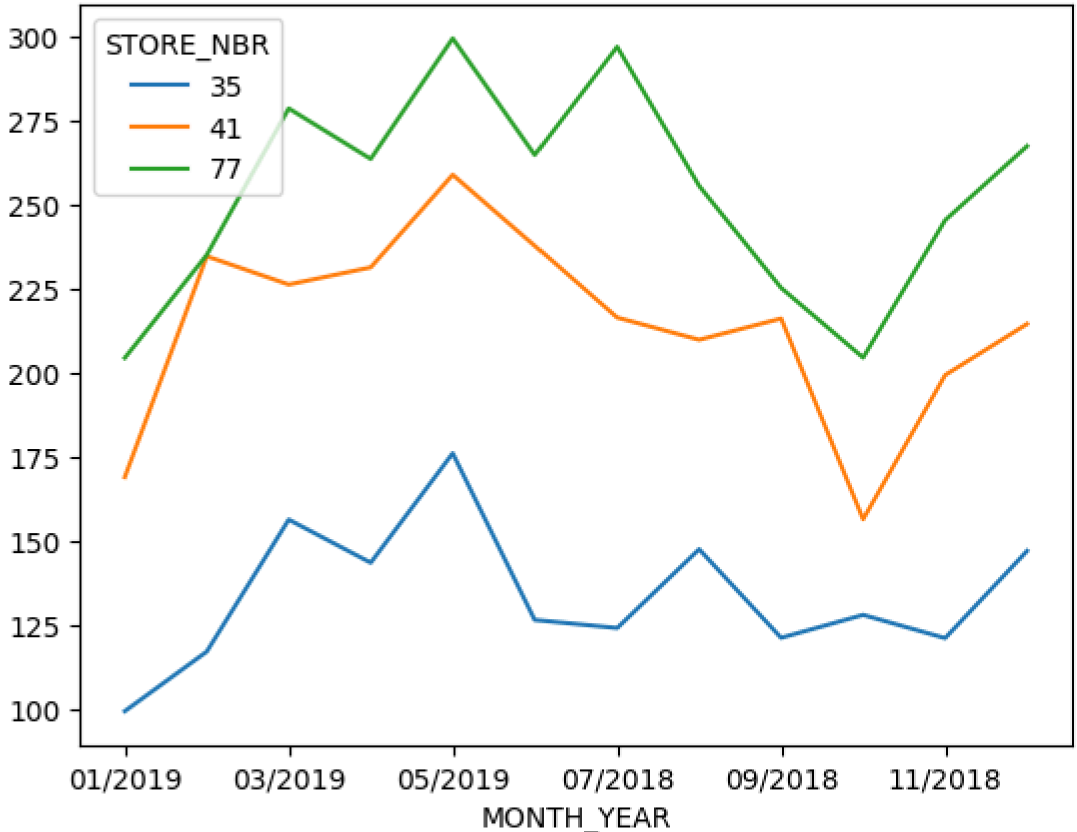
STORE 31 & 11 SALES ARE WAY TOO LOW TO USE.

In [271…

```
# GRABBING STORE 41, 35, 77 FROM TOTAL GROUP DATAFRAME
three_amigos_77 = total_grp[[41, 35, 77]]

# MAKING DATAFRAME
amigos_77_df = pd.DataFrame(three_amigos_77)

# PIVOTING THE DATAFRAME
amigos_77_pivot = amigos_77_df.pivot_table(index="MONTH_YEAR", columns="STORE_NB
amigos_77_pivot.plot()
plt.show()
```



STORE 35 EVEN THOUGH IT HAS A GOOD CORRELATION STORE 41 IS A MUCH BETTER FIT. STORE 31 & 11 EVEN THOUGH ARE A BEST MATCH CORRELATION WISE IT DOES NOT MAKE SENSE WITH SALES VOLUME. SO I WILL GO WITH STORE 41.

FOR TRIAL STORE 77, I WILL USE STORE NUMBER 41 AS A CONTROL STORE. IT'S A 0.76 CORRELATION.

SORTING STORES BY TOTAL SALES LOOKING FOR A MATCH FOR STORE 86

In [272…

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

Out[272…

```
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

In [273…

```
stores_control_two = [109, 191, 196, 229, 97, 102, 105, 232, 57, 172, 113, 225,
control_two = pd.DataFrame({"Value" : total_grp[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 STORES IN A PIVOT CHART FORMAT

In [274…

```
pivot_chips2 = control_two.pivot_table(index="MONTH_YEAR", columns="STORE_NBR",
pivot_chips2
```

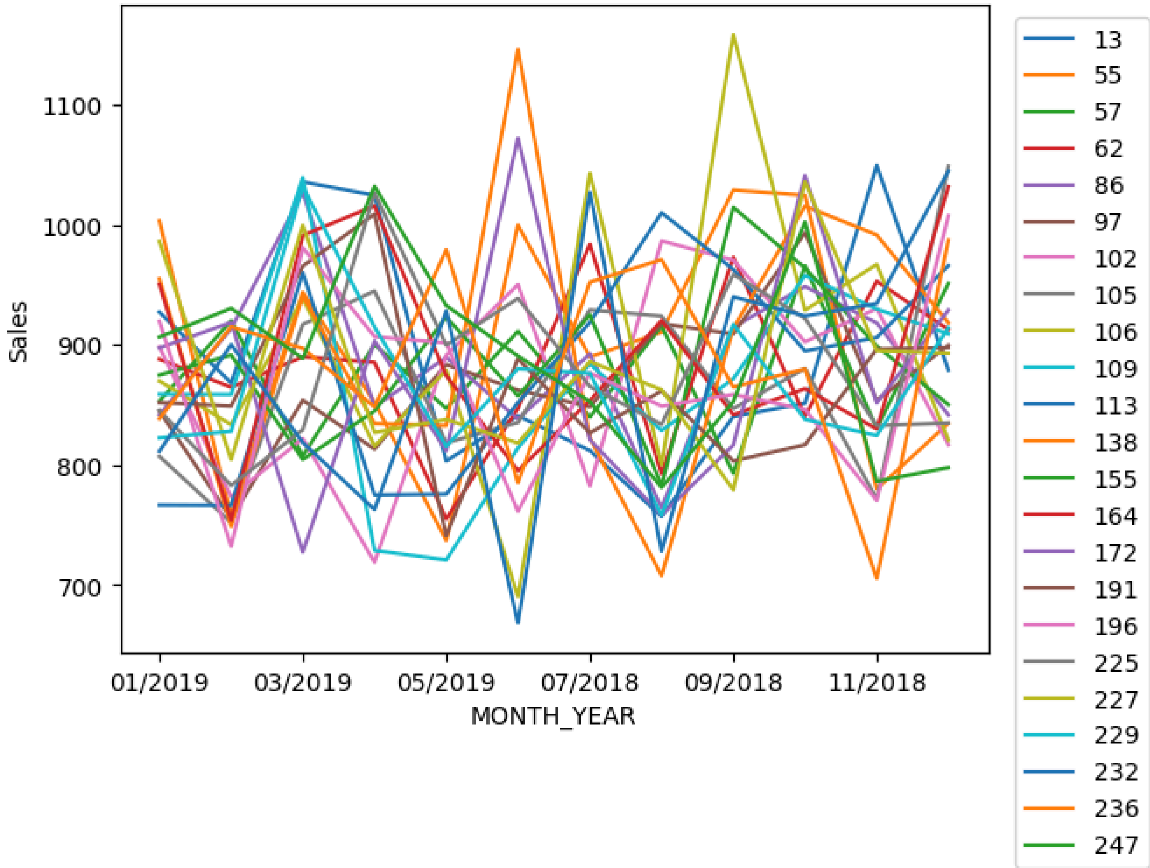
Out[274...

STORE_NBR	13	55	57	62	86	97	102	105	106
MONTH_YEAR									
01/2019	927.0	1003.20	852.8	887.8	841.40	844.60	898.0	807.0	869.60
02/2019	868.0	757.80	919.8	864.4	913.20	755.20	773.4	751.8	833.20
03/2019	1035.6	943.60	807.4	889.8	1026.80	853.60	821.8	916.8	938.60
04/2019	1024.4	851.80	900.0	885.2	848.20	813.00	718.6	944.6	815.40
05/2019	803.2	736.85	846.7	754.9	889.30	883.30	890.9	818.1	878.75
06/2019	840.6	999.60	911.0	846.8	838.00	862.00	950.0	835.0	690.20
07/2018	811.8	889.60	839.6	983.6	892.20	848.20	782.4	928.9	1042.80
08/2018	756.9	910.30	915.4	792.4	764.05	917.35	986.4	923.7	799.85
09/2018	840.0	1028.80	792.8	972.8	914.60	908.80	970.4	846.6	1158.40
10/2018	851.0	1024.40	965.8	840.2	948.40	993.20	902.2	880.0	928.60
11/2018	1049.4	779.80	830.0	952.8	918.00	853.40	930.0	771.4	966.80
12/2018	878.6	834.40	951.0	912.4	841.20	899.40	816.6	1048.6	820.40

12 rows × 23 columns

In [275...

```
pivot_chips2.plot()
plt.legend(loc = "upper right", bbox_to_anchor = (1.20, 1))
plt.ylabel("Sales")
plt.show()
```



THAT'S AN UGLY LOOKING LINE CHART. LETS TAKE A CLOSER LOOK AT THE CORRELATIONS BETWEEN THEM AND RECHART JUST THE TWO.

LOOKING AT CORRELATION

In [276…

```
pivot_chips2.corr(method="pearson")
```

Out[276…

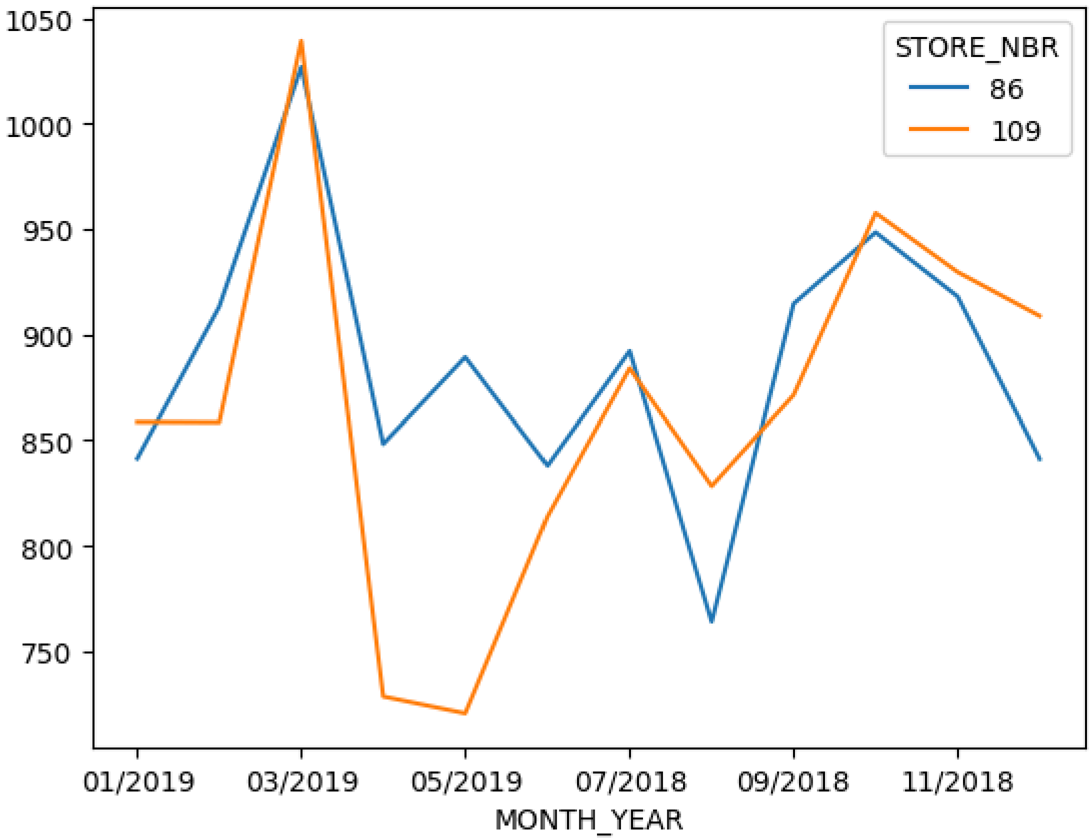
STORE_NBR	13	55	57	62	86	97	102
STORE_NBR							
13	1.000000	-0.125341	-0.291218	0.365314	0.457947	-0.373037	-0.377415
55	-0.125341	1.000000	-0.039301	0.181823	0.043906	0.495256	0.418809
57	-0.291218	-0.039301	1.000000	-0.428165	-0.402687	0.221201	-0.139586
62	0.365314	0.181823	-0.428165	1.000000	0.276452	-0.184301	-0.206387
86	0.457947	0.043906	-0.402687	0.276452	1.000000	-0.015617	-0.226422
97	-0.373037	0.495256	0.221201	-0.184301	-0.015617	1.000000	0.578719
102	-0.377415	0.418809	-0.139586	-0.206387	-0.226422	0.578719	1.000000
105	-0.059766	0.124132	0.301428	0.113294	-0.202451	0.334039	-0.303843
106	0.049336	0.181864	-0.658612	0.634354	0.510548	0.203434	0.088393
109	0.324289	0.326968	-0.124668	0.426023	0.643075	0.241536	0.057036
113	-0.161963	0.306164	-0.087082	0.287274	0.043835	0.548974	0.388871
138	0.284311	0.500047	-0.001387	0.172155	0.250447	0.286776	0.317674
155	-0.228967	0.174382	-0.232252	0.339800	0.326149	0.275949	0.171003
164	0.357477	0.060884	0.060840	-0.006044	-0.117970	0.140764	-0.324841
172	-0.091999	0.250338	0.665384	-0.100249	-0.156398	0.128774	0.000426
191	0.733656	0.018181	0.081015	0.227897	0.043345	-0.359215	-0.454167
196	0.166098	0.101949	-0.113210	0.049385	0.081832	0.240357	-0.283326
225	0.043419	0.338013	-0.005863	0.005783	-0.109479	0.224941	-0.023039
227	0.289917	0.354941	0.106827	-0.028706	0.393785	0.403000	-0.009479
229	0.508201	0.234072	-0.335684	0.426077	0.596886	-0.120038	-0.406497
232	-0.084443	-0.320462	-0.100878	0.461276	0.327006	0.141757	-0.251850
236	-0.597718	-0.206578	0.237461	-0.334550	-0.164982	0.162069	-0.245020
247	0.167139	0.096625	0.237256	-0.295701	0.250601	-0.106598	-0.460621

23 rows × 23 columns

STORE 109 AND 86 HAS HAS THE STRONGEST CORRELATION AT 0.643. LETS GRAPH IT.

In [277…

```
chips2_graph = pivot_chips2[[86, 109]]
chips2_graph.plot()
plt.show()
```



CHECKING CORRELATIONS ON ENTIRE TABLE

In [278…

```
total_grp_pivot_table[86].sort_values(ascending=False).head(10)
```

Out[278…

```
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
```

THESE ARE THE TOP 10 CORRELATIONS TO STORE 86. STORE 109 WOULD BE RANKED IN 5TH PLACE. LETS LOOK AT THE OTHER STORES BY TOTAL SALES BEFORE I MAKE A DECISION.

In [279…

```
# GRABBING THE TOTAL SALES SORTED SERIES TO SEE HOW THE SALES STACK UP FOR THE T
total_sorted.loc[[31, 193, 159, 231, 109]]
```

Out[279…

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

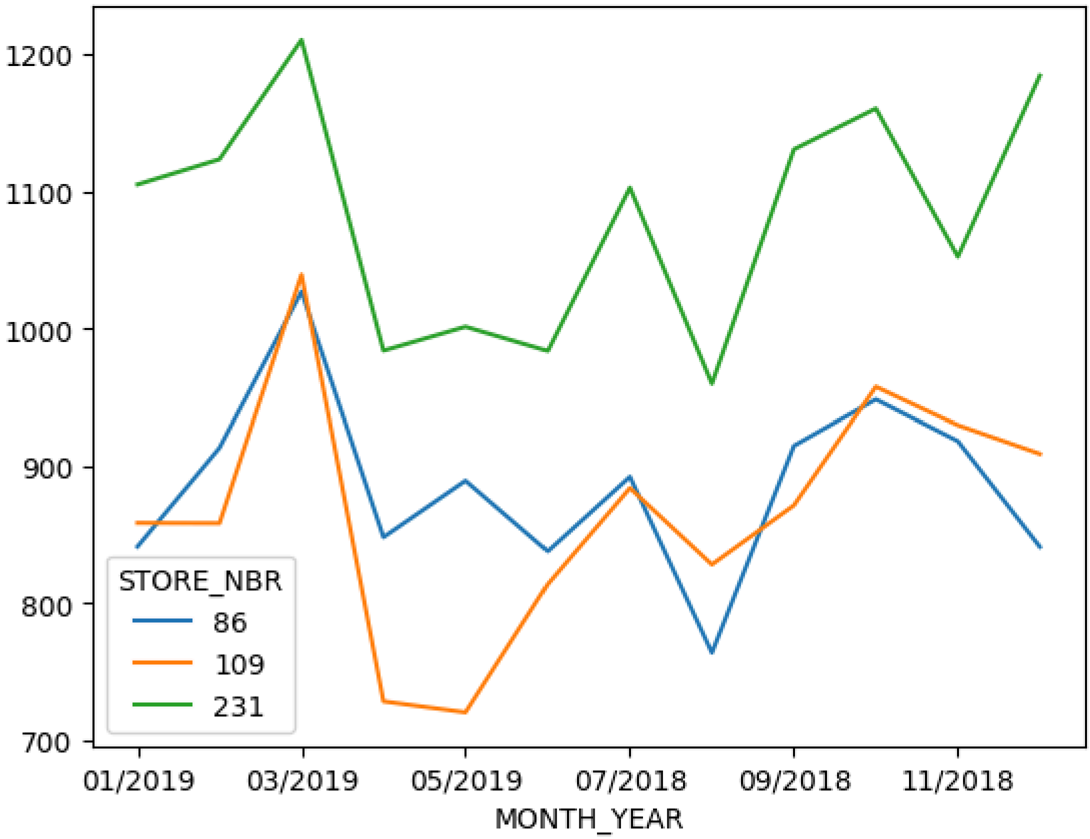
STORE 31, 159, & 193 SALES ARE WAY TOO LOW TO USE.

In [280…

```
# GRABBING STORE 231, 109, 86 FROM TOTAL GROUP DATAFRAME
three_amigos_86 = total_grp[[231, 109, 86]]

# MAKING DATAFRAME
amigos_86_df = pd.DataFrame(three_amigos_86)

# PIVOTING THE DATAFRAME
amigos_86_pivot = amigos_86_df.pivot_table(index="MONTH_YEAR", columns="STORE_NB
amigos_86_pivot.plot()
plt.show()
```



STORE 231 EVEN THOUGH IT HAS A GOOD CORRELATION STORE 109 IS A MUCH BETTER FIT. STORE 31 EVEN THOUGH is A BEST MATCH CORRELATION WISE IT DOES NOT MAKE SENSE WITH SALES VOLUME. SO I WILL GO WITH STORE 41.

FOR TRIAL STORE 86, I WILL USE STORE NUMBER 109 AS A CONTROL STORE. IT'S A 0.643 CORRELATION.

SORTING STORES BY TOTAL SALES LOOKING FOR A MATCH FOR STORE 88

In [281…

```
# LOOKING FOR CONTROL STORE FOR STORE 88
# REUSING TOTAL GROUP PIVOT TABLE TO FIND TOP 10 CORRELATED STORES
total_grp_pivot_table[88].sort_values(ascending=False).head(10)
```

Out[281…

```
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
```

THESE ARE THE TOP 10 CORRELATIONS TO STORE 88. LETS LOOK AT THE OTHER STORES BY TOTAL SALES BEFORE I MAKE A DECISION.

In [282…

```
# GRABBING THE TOTAL SALES SORTED SERIES TO SEE HOW THE SALES STACK UP FOR THE T
total_sorted.loc[[206, 88, 159, 193, 201, 188, 229, 228, 61, 140]]
```

Out[282…

```
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
```

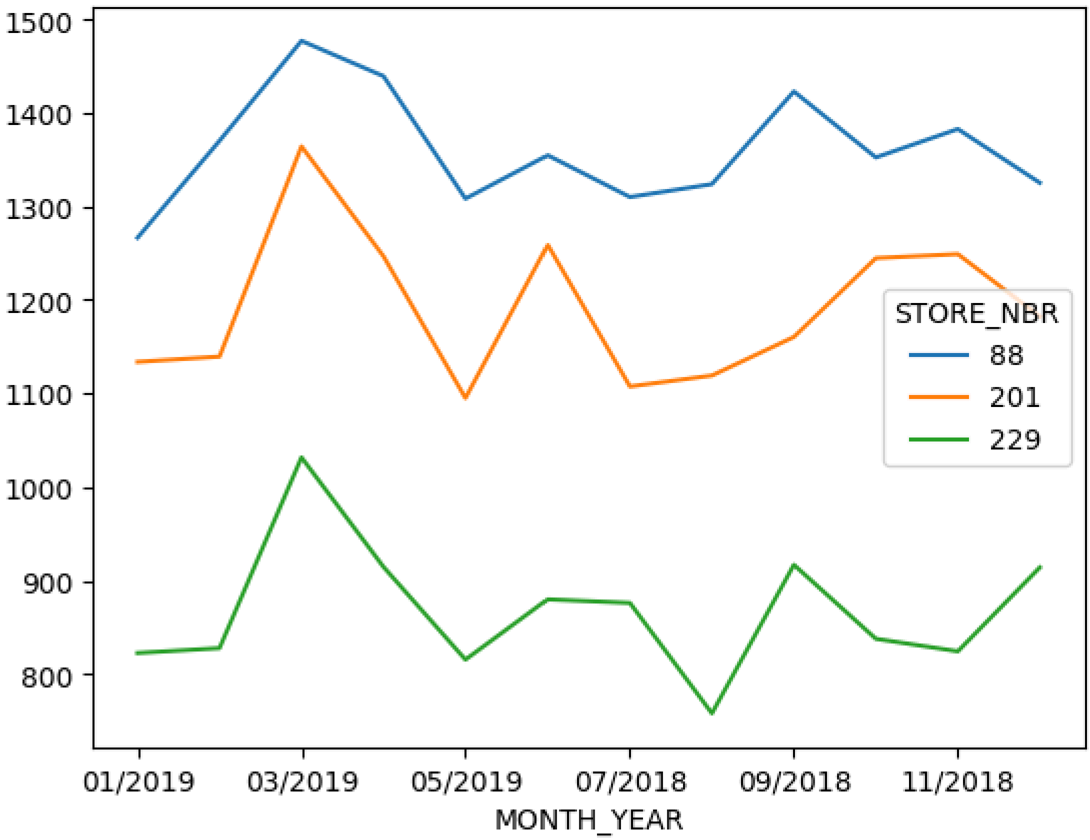
STORE 206, 159, 188, 228, 61, 140, & 193 SALES ARE WAY TOO LOW TO USE.

In [294…

```
# GRABBING STORE 201, 229, 88 FROM TOTAL GROUP DATAFRAME
three_amigos_88 = total_grp[[201, 229, 88]]

# MAKING DATAFRAME
amigos_88_df = pd.DataFrame(three_amigos_88)

# PIVOTING THE DATAFRAME
amigos_88_pivot = amigos_88_df.pivot_table(index="MONTH_YEAR", columns="STORE_NB
amigos_88_pivot.plot()
plt.show()
```



STORE 201 COMES CLOSE TO THE PATTERN OF STORE 88

In [295…

```
sorted_88 = total_grp_pivot_table[88].sort_values(ascending=False)
sorted_88[201]
```

Out[295…

```
np.float64(0.7375831241350634)
```

STORE 229 EVEN THOUGH IT 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 WITH SALES VOLUME. SO I WILL GO WITH STORE 201.

FOR TRIAL STORE 88, I WILL USE STORE NUMBER 201 AS A CONTROL STORE. IT'S A 0.737 CORRELATION.

In [300…

```
# CREATING NEW DATAFRAME FOR TRIAL & CONTROL STORE
# Selecting trial and control stores from chips_trial
trial_store_77 = dataset.loc[dataset["STORE_NBR"] == 77]
control_store_41 = dataset.loc[dataset["STORE_NBR"] == 41]

trial_store_86 = dataset.loc[dataset["STORE_NBR"] == 86]
control_store_109 = dataset.loc[dataset["STORE_NBR"] == 109]

trial_store_88 = dataset.loc[dataset["STORE_NBR"] == 88]
control_store_201 = dataset.loc[dataset["STORE_NBR"] == 201]

trial_store_77
```

Out[300…

	LYLTY_CARD_NBR	DATE	STORE_NBR	TXN_ID	PROD_NBR	PROD_NAME
73365	77000	2019-03-28	77	74911	18	Cheetos Chs & Bacon Balls 190g
73366	77000	2019-04-13	77	74912	69	Smiths Chip Thinly S/Cream&Onion 175g
73367	77000	2018-09-26	77	74910	36	Kettle Chilli 175g
73368	77001	2019-02-27	77	74913	7	Smiths Crinkle Original 330g
73369	77001	2019-01-21	77	74914	9	Kettle Tortilla ChpsBtroot&Ricotta 150g
...	...	...	...	...	...	...
264818	2330321	2018-07-30	77	236756	71	Twisties Cheese Burger 250g
264819	2330331	2018-11-18	77	236760	95	Sunbites Whlegrn Crisps Frch/Onin 90g
264820	2330431	2018-07-31	77	236770	50	Tostitos Lightly Salted 175g
264821	2330461	2018-07-21	77	236777	87	Infuzions BBQ Rib Prawn Crackers 110g
264822	2330501	2019-06-20	77	236780	63	Kettle 135g Swt Pot Sea Salt

563 rows × 13 columns

LETS START WITH STORE 77 & 41

In [301…

```
# LOOKING AT TOTAL SALES & PRODUCTS SOLD
trial_store_77[["TOT_SALES", "PROD_QTY"]].sum()
```

Out[301…

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

In [302…

```
# LOOKING AT TOTAL SALES & PRODUCTS SOLD
control_store_41[["TOT_SALES", "PROD_QTY"]].sum()
```

Out[302…

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

In [ ]:

```
# LOOKING AT REPEAT CUSTOMERS FOR TRIAL STORE
trial_store_77["LYLTY_CARD_NBR"].value_counts()
```

Out[ ]:
```
LYLTY_CARD_NBR
77476    5
77109    4
77205    4
77066    4
77093    4
        ..
77023    1
77024    1
77025    1
77187    1
77003    1
Name: count, Length: 356, dtype: int64
```

In [304…
```
# TOTAL CUSTOMER TRANSACTIONS
trial_store_77[["LYLTY_CARD_NBR"]].count()
```

Out[304…
```
LYLTY_CARD_NBR    563
dtype: int64
```

In [305…
```
# LOOKING AT REPEAT CUSTOMERS FOR CONTROL STORE
control_store_41["LYLTY_CARD_NBR"].value_counts()
```

Out[305…
```
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
```

In [306…
```
# TOTAL CUSTOMER TRANSACTIONS
control_store_41[["LYLTY_CARD_NBR"]].count()
```

Out[306…
```
LYLTY_CARD_NBR    567
dtype: int64
```

In [307…
```
# COUNTING REPEAT CUSTOMERS THAT PURCHASED MORE THAN ONCE
repeat_customers = trial_store_77["LYLTY_CARD_NBR"].value_counts()
print(repeat_customers.head(24))

repeats_total = 24
```

```
LYLTY_CARD_NBR
77476    5
77109    4
77205    4
77066    4
77093    4
77305    4
77313    4
77338    4
77344    4
77454    4
77206    3
77102    3
77480    3
77238    3
77136    3
77044    3
77207    3
77111    3
77080    3
77114    3
77049    3
77077    3
77263    3
77069    3
Name: count, dtype: int64
```

In [308…

```
# COUNTING REPEAT CUSTOMERS THAT PURCHASED MORE THAN ONCE
repeat_customers2 = control_store_41["LYLTY_CARD_NBR"].value_counts()
print(repeat_customers2.head(9))

repeats_total_two = 9
```

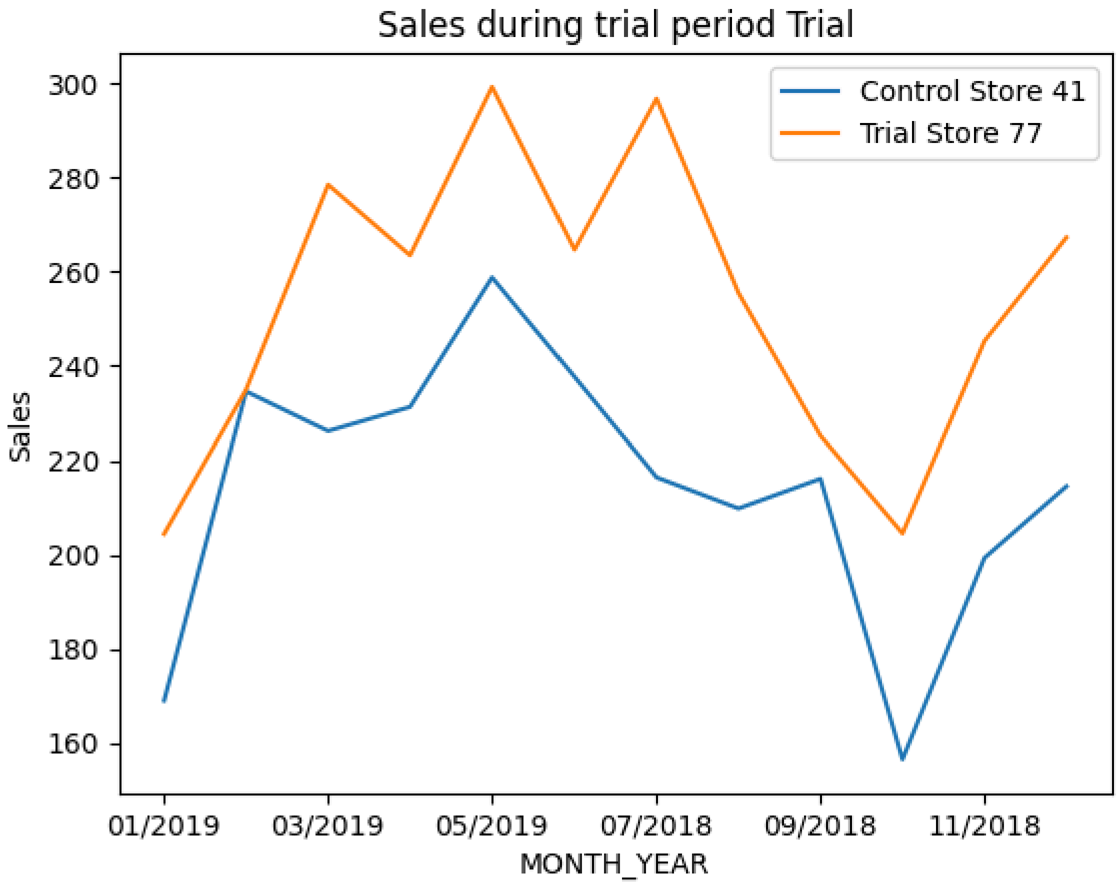
```
LYLTY_CARD_NBR
41497    4
41453    4
41466    4
41367    4
41359    4
41368    4
41418    4
41423    4
41432    4
Name: count, dtype: int64
```

In [309…

```
# GROUPING STORES BY MONTH
grouped77 = trial_store_77.groupby("MONTH_YEAR")
grouped41 = control_store_41.groupby("MONTH_YEAR")
```

In [310…

```
grouped41["TOT_SALES"].sum().plot(label = "Control Store 41")
grouped77["TOT_SALES"].sum().plot(label = "Trial Store 77")
plt.ylabel("Sales")
plt.legend()
plt.title("Sales during trial period Trial")
plt.show()
```



FOR THE FIRST PAIR WE CAN SEE A CLEAR DIFFERENCE BETWEEN THE TRIAL STORE AND THE CONTROL STORE. LETS LOOK AT THE NEXT PAIR OF STORES.

LETS START WITH STORE 86 & 109

In [311…

```
# LOOKING AT TOTAL SALES & PRODUCTS SOLD
trial_store_86[["TOT_SALES", "PROD_QTY"]].sum()
```

Out[311…

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

In [312…

```
# LOOKING AT TOTAL SALES & PRODUCTS SOLD
control_store_109[["TOT_SALES", "PROD_QTY"]].sum()
```

Out[312…

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

In [313…

```
# LOOKING AT REPEAT CUSTOMERS FOR TRIAL STORE
trial_store_86["LYLTY_CARD_NBR"].value_counts()
```

Out[313…
```
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
```

In [315…
```python
# TOTAL CUSTOMER TRANSACTIONS
trial_store_86[["LYLTY_CARD_NBR"]].count()
```

Out[315…
```
LYLTY_CARD_NBR    1538
dtype: int64
```

In [316…
```python
# WE HAVE 123 REPEAT CUSTOMERS FOR STORE 86
repeat_customers_86 = trial_store_86["LYLTY_CARD_NBR"].value_counts()
repeat_customers_86.iloc[:125]
```

Out[316…
```
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
```

In [317…
```python
# LOOKING AT REPEAT CUSTOMERS FOR CONTROL STORE
control_store_109["LYLTY_CARD_NBR"].value_counts()
```

Out[317…
```
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
```

In [318…
```python
# TOTAL CUSTOMER TRANSACTIONS
control_store_109[["LYLTY_CARD_NBR"]].count()
```

Out[318…
```
LYLTY_CARD_NBR    1505
dtype: int64
```

In [319…

```
# WE HAVE 111 REPEAT CUSTOMERS FOR STORE 86
repeat_customers_109 = control_store_109["LYLTY_CARD_NBR"].value_counts()
repeat_customers_109.iloc[:115]
```

Out[319…

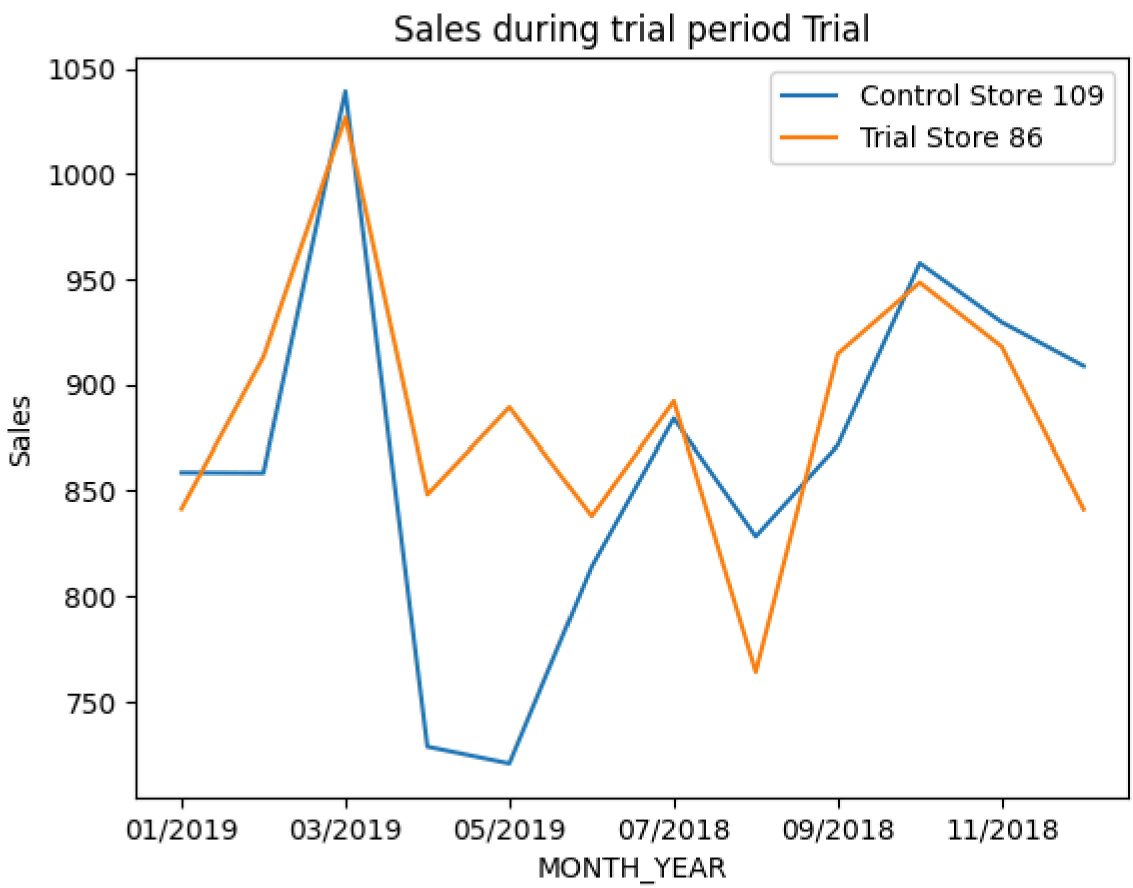
```
LYLTY_CARD_NBR
109036    16
109080    14
109086    13
109078    12
109212    12
          ..
109075     6
109066     6
109065     6
109148     6
109113     6
Name: count, Length: 115, dtype: int64
```

In [321…

```
# GROUPING STORES BY MONTH
grouped86 = trial_store_86.groupby("MONTH_YEAR")
grouped109 = control_store_109.groupby("MONTH_YEAR")
```

In [322…

```
grouped109["TOT_SALES"].sum().plot(label = "Control Store 109")
grouped86["TOT_SALES"].sum().plot(label = "Trial Store 86")
plt.ylabel("Sales")
plt.legend()
plt.title("Sales during trial period Trial")
plt.show()
```



FOR THE SECOND PAIR WE CAN SEE A CLEAR DIFFERENCE BETWEEN THE TRIAL STORE AND THE CONTROL STORE. LETS LOOK AT THE NEXT PAIR OF STORES.

LETS START WITH STORE 88 & 201

In [323…
```
# LOOKING AT TOTAL SALES & PRODUCTS SOLD
trial_store_88[["TOT_SALES", "PROD_QTY"]].sum()
```

Out[323…
```
TOT_SALES    16333.25
PROD_QTY      3718.00
dtype: float64
```

In [324…
```
# LOOKING AT TOTAL SALES & PRODUCTS SOLD
control_store_201[["TOT_SALES", "PROD_QTY"]].sum()
```

Out[324…
```
TOT_SALES    14298.7
PROD_QTY      3262.0
dtype: float64
```

In [325…
```
# LOOKING AT REPEAT CUSTOMERS FOR TRIAL STORE
trial_store_88["LYLTY_CARD_NBR"].value_counts()
```

Out[325…
```
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
```

In [326…
```
# TOTAL CUSTOMER TRANSACTIONS
trial_store_88[["LYLTY_CARD_NBR"]].count()
```

Out[326…
```
LYLTY_CARD_NBR    1873
dtype: int64
```

In [327…
```
# WE HAVE 145 REPEAT CUSTOMERS FOR STORE 86
repeat_customers_88 = trial_store_88["LYLTY_CARD_NBR"].value_counts()
repeat_customers_88.iloc[:146]
```

Out[327…
```
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
```

In [328…
```
# LOOKING AT REPEAT CUSTOMERS FOR CONTROL STORE
control_store_201["LYLTY_CARD_NBR"].value_counts()
```

Out[328…
```
LYLTY_CARD_NBR
201294    13
201120    11
201186    11
201206    10
201018    10
          ..
201057     1
201037     1
201043     1
201356     1
201005     1
Name: count, Length: 376, dtype: int64
```

In [329…
```
# TOTAL CUSTOMER TRANSACTIONS
control_store_201[["LYLTY_CARD_NBR"]].count()
```

Out[329…
```
LYLTY_CARD_NBR    1654
dtype: int64
```

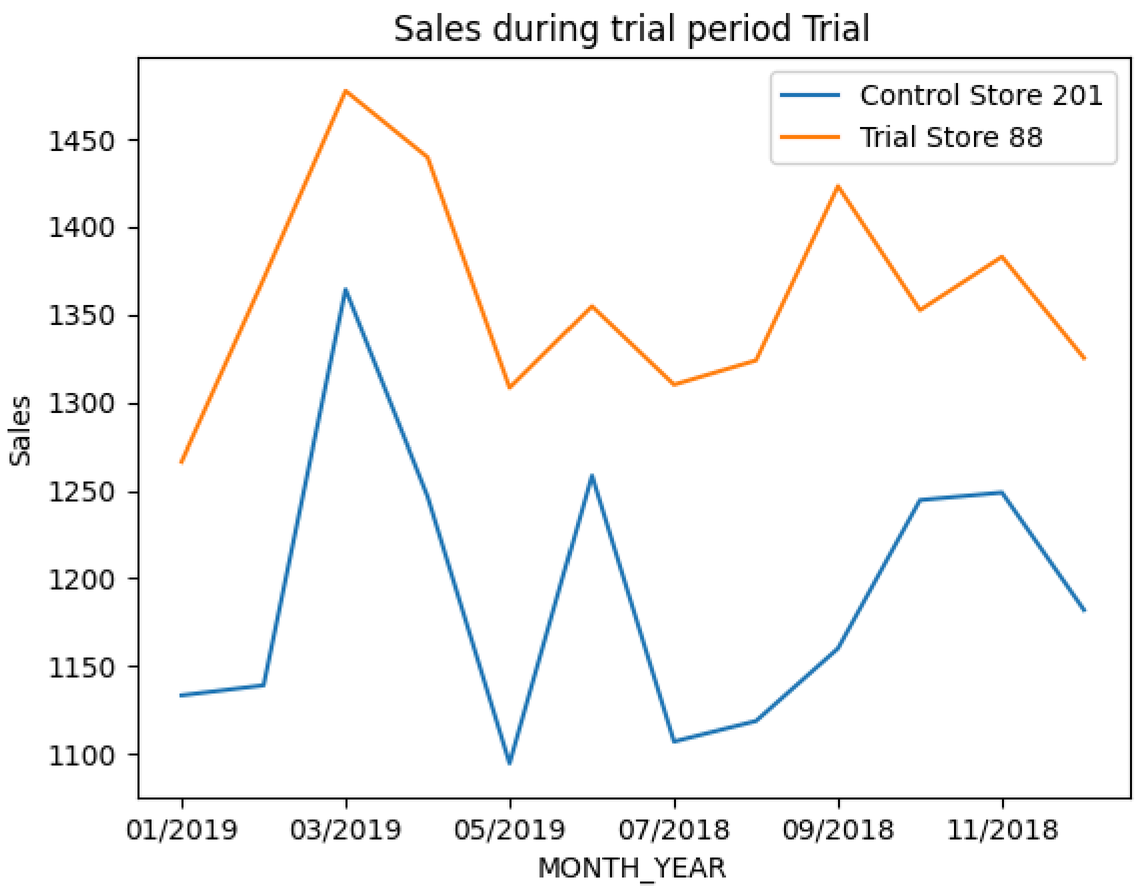
In [330…
```
# WE HAVE 109 REPEAT CUSTOMERS FOR STORE 86
repeat_customers_109 = control_store_109["LYLTY_CARD_NBR"].value_counts()
repeat_customers_109.iloc[:110]
```

Out[330…
```
LYLTY_CARD_NBR
109036    16
109080    14
109086    13
109078    12
109212    12
          ..
109202     6
109095     6
109077     6
109073     6
109074     6
Name: count, Length: 110, dtype: int64
```

In [331…
```
# GROUPING STORES BY MONTH
grouped88 = trial_store_88.groupby("MONTH_YEAR")
grouped201 = control_store_201.groupby("MONTH_YEAR")
```

In [332…

```
grouped201["TOT_SALES"].sum().plot(label = "Control Store 201")
grouped88["TOT_SALES"].sum().plot(label = "Trial Store 88")
plt.ylabel("Sales")
plt.legend()
plt.title("Sales during trial period Trial")
plt.show()
```

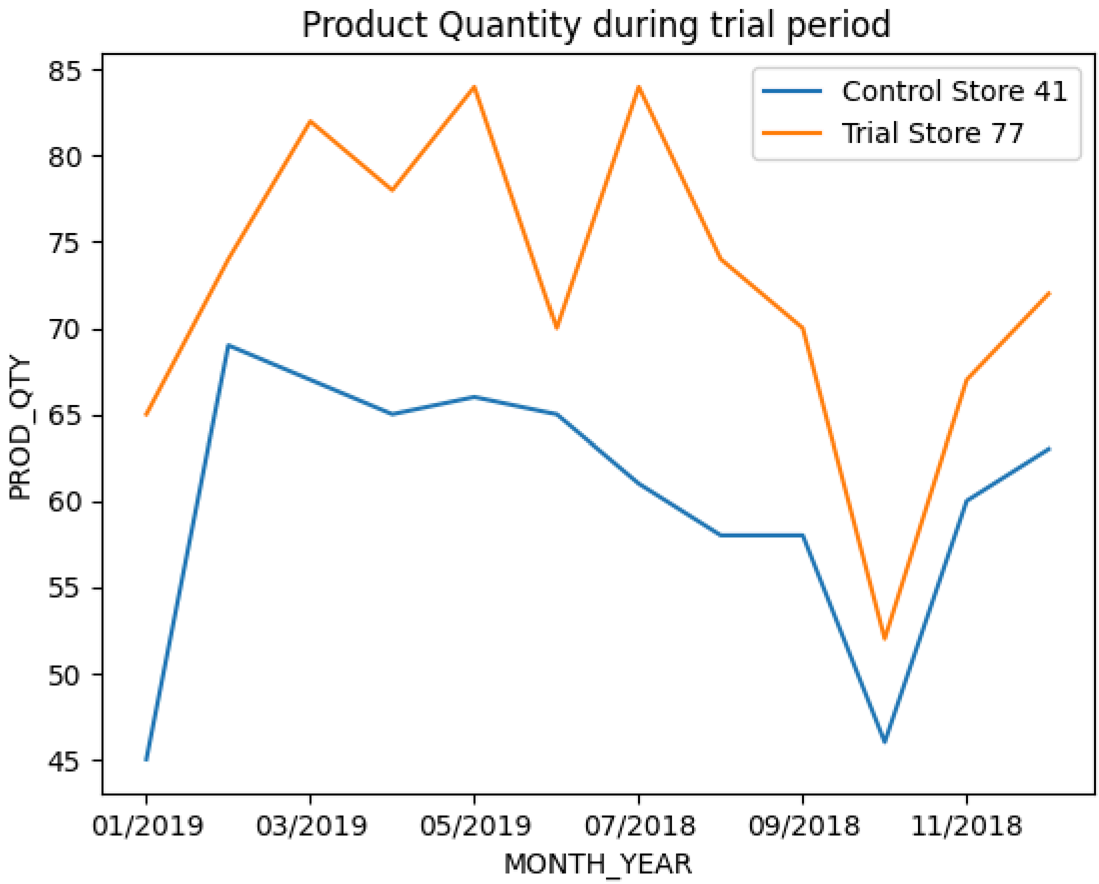


FOR THE THIRD PAIR WE CAN SEE A CLEAR DIFFERENCE BETWEEN THE TRIAL STORE AND THE CONTROL STORE. LETS LOOK AT THE NEXT PAIR OF STORES.

LETS VISUALIZE THE PRODUCT QUANTITY SOLD

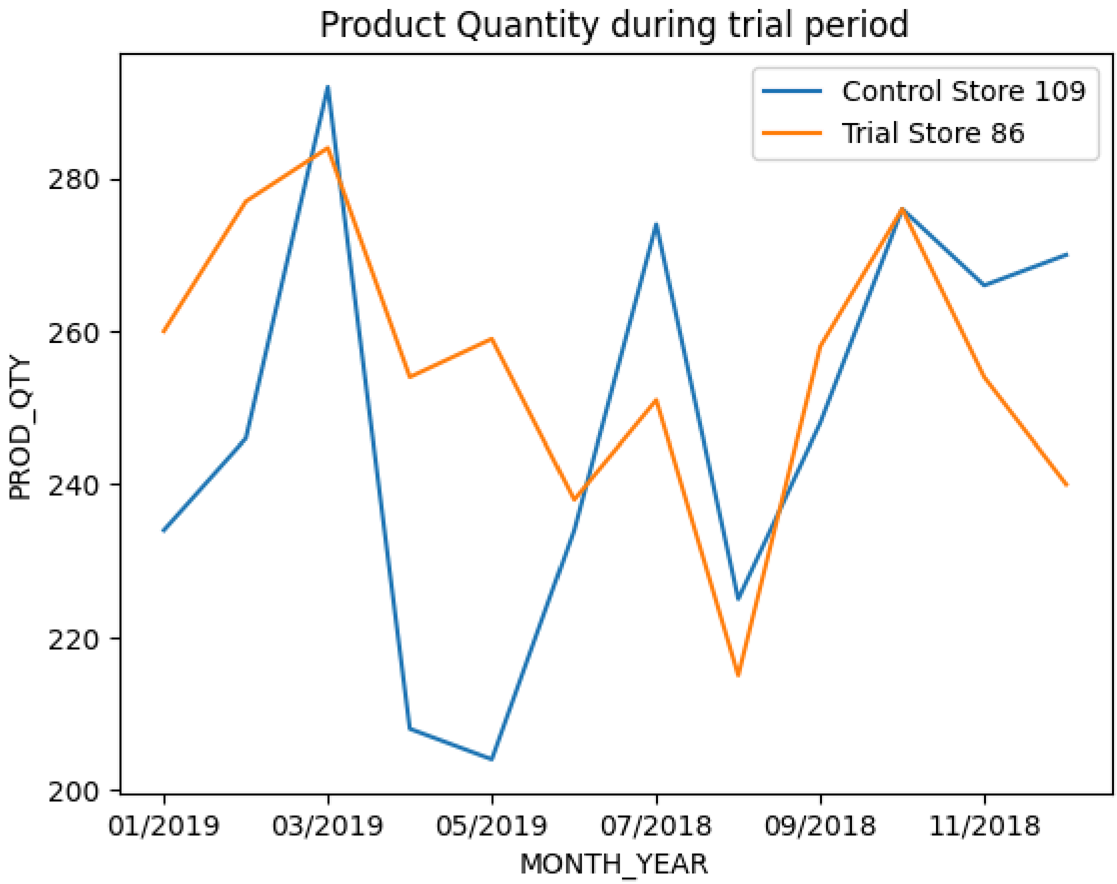
In [335…

```
grouped41["PROD_QTY"].sum().plot(label = "Control Store 41")
grouped77["PROD_QTY"].sum().plot(label = "Trial Store 77")
plt.ylabel("PROD_QTY")
plt.legend()
plt.title("Product Quantity during trial period")
plt.show()
```



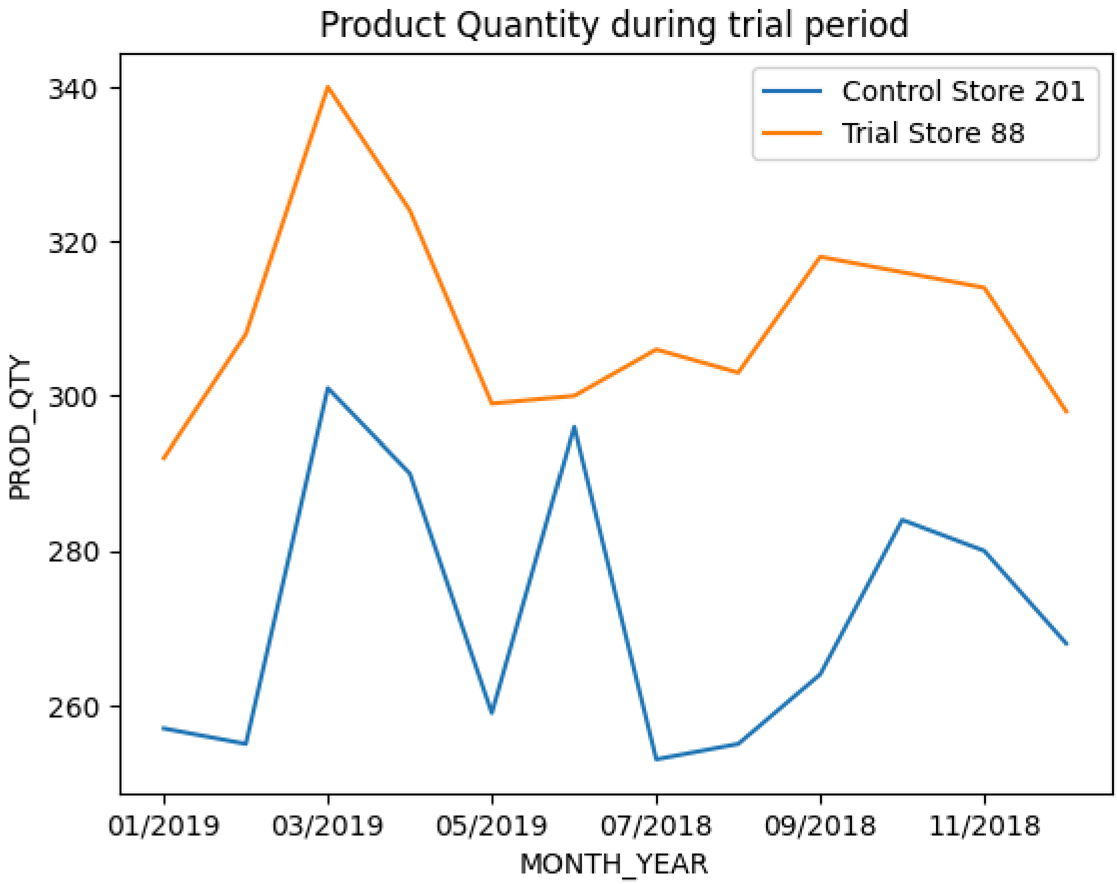
In [334...

```
grouped109["PROD_QTY"].sum().plot(label = "Control Store 109")
grouped86["PROD_QTY"].sum().plot(label = "Trial Store 86")
plt.ylabel("PROD_QTY")
plt.legend()
plt.title("Product Quantity during trial period")
plt.show()
```



In [333…

```
grouped201["PROD_QTY"].sum().plot(label = "Control Store 201")
grouped88["PROD_QTY"].sum().plot(label = "Trial Store 88")
plt.ylabel("PROD_QTY")
plt.legend()
plt.title("Product Quantity during trial period")
plt.show()
```



AS WE CAN SEE BY THE GRAPHS ABOVE THE TRIAL STORES OUTPERFORMED THE CONTROL STORES BY QUANTITY SOLD.

LETS SEE HOW THEY STACK UP WITH AVERAGE TRANSACTIONS PER CUSTOMER

In [336…
```
grouped77["LYLTY_CARD_NBR"].value_counts().mean()
```
Out[336…
```
np.float64(1.048417132216015)
```

In [337…
```
grouped41["LYLTY_CARD_NBR"].value_counts().mean()
```
Out[337…
```
np.float64(1.05)
```

In [338…
```
grouped86["LYLTY_CARD_NBR"].value_counts().mean()
```
Out[338…
```
np.float64(1.2544861337683524)
```

In [339…
```
grouped109["LYLTY_CARD_NBR"].value_counts().mean()
```
Out[339…
```
np.float64(1.2918454935622317)
```

In [340…
```
grouped88["LYLTY_CARD_NBR"].value_counts().mean()
```
Out[340…
```
np.float64(1.2363036303630364)
```

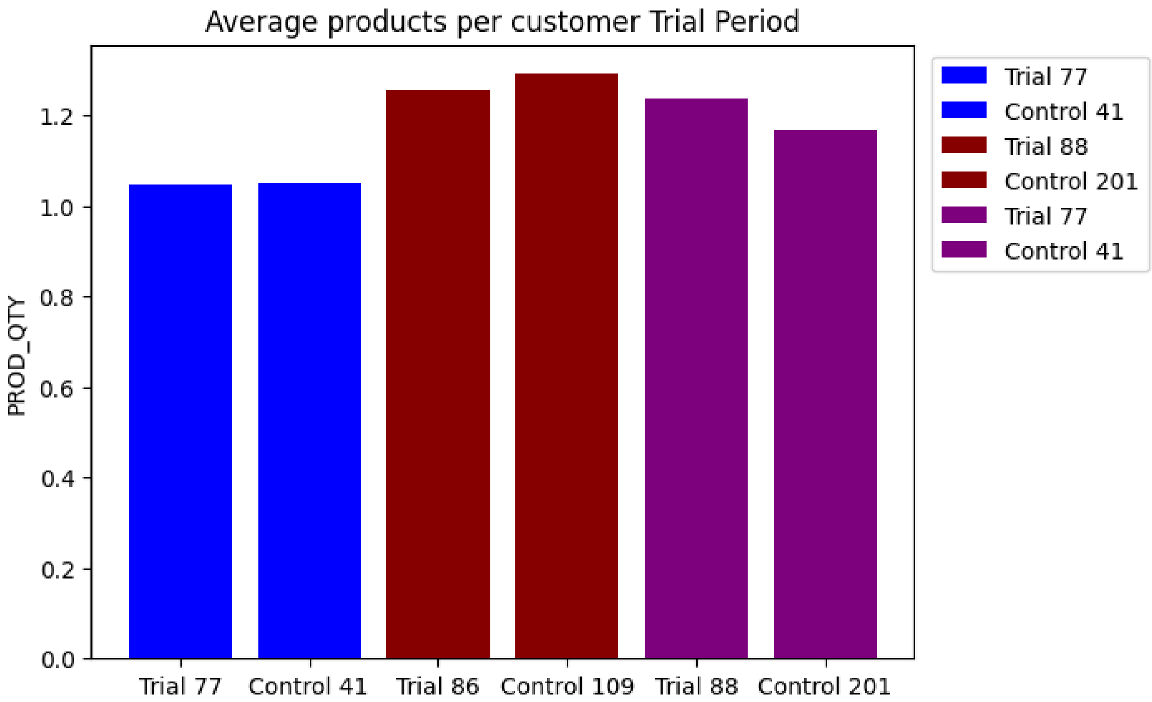
In [341…
```
grouped201["LYLTY_CARD_NBR"].value_counts().mean()
```
Out[341…
```
np.float64(1.1689045936395759)
```

In [342…

```
group1 = ["Trial 77", "Control 41"]
group2 = ["Trial 86", "Control 109"]
group3 = ["Trial 88", "Control 201"]
values_grp_1 = [1.048417132216015, 1.05]
values_grp_2 = [1.2544861337683524, 1.2918454935622317]
values_grp_3 = [1.2363036303630364, 1.1689045936395759]

plt.bar(group1, values_grp_1, label = group1, color = "blue")
plt.bar(group2, values_grp_2, label = group3, color = "darkred")
plt.bar(group3, values_grp_3, label = group1, color = "purple")

plt.ylabel("PROD_QTY")
plt.legend(loc = "upper right", bbox_to_anchor = (1.3, 1))
plt.title("Average products per customer Trial Period")
plt.show()
```



AS WE CAN SEE THE AVERAGE TRANSACTIONS WERE SLIGHTLY HIGHER FOR 1 OF THE 3 TRIAL STORES.

I BELIEVE THE NEW LAYOUT IS WORKING TO INCREASE SALES. SALES, PRODUCTS SOLD, AMOUNT OF REPEAT CUSTOMERS AND AVERAGE TRANSACTIONS PER CUSTOMER ALL SHOW SIGNS THATS THE TRIAL STORES ARE OUTPERFORMING THE CONTROL STORES.

MY RECOMMENDATION WOULD BE TO INCREASE THE AMOUNT OF TRIAL STORES AND TO RUN ANOTHER ANALYSIS IN 3 MONTHS TO SEE IF THE INCREASED SALES STAY TRUE AND STABILIZE AT A HIGHER POINT.