Here is your task

Julia has asked us to evaluate the performance of a store trial which was performed in stores 77, 86 and 88.

To get started, use the QVI_data dataset below or your output from task 1 and consider the monthly sales experience of each store.

This can be broken down by:

total sales revenue total number of customers average number of transactions per customer

Create a measure to compare different control stores to each of the trial stores to do this write a function to reduce having to re-do the analysis for each trial store. Consider using Pearson correlations or a metric such as a magnitude distance e.g. 1- (Observed distance – minimum distance)/(Maximum distance – minimum distance) as a measure.

Once you have selected your control stores, compare each trial and control pair during the trial period. You want to test if total sales are significantly different in the trial period and if so, check if the driver of change is more purchasing customers or more purchases per customers etc.

```
import pandas as pd, numpy as np
import matplotlib.pyplot as plt, seaborn as sns
import scipy.stats as stats

qvi_merged = pd.read_csv('qvi_merged.csv')

qvi_merged
```

```
        Unnamed: 0  LYLTY_CARD_NBR               LIFESTAGE
PREMIUM_CUSTOMER  \
0                0            1000   YOUNG SINGLES/COUPLES
Premium
1                1            1002   YOUNG SINGLES/COUPLES
Mainstream
2                2            1003         YOUNG FAMILIES
Budget
3                3            1003         YOUNG FAMILIES
Budget
4                4            1004   OLDER SINGLES/COUPLES
Mainstream
...            ...             ...                    ...
...
258592      258592         2370651  MIDAGE SINGLES/COUPLES
Mainstream
258593      258593         2370701         YOUNG FAMILIES
Mainstream
258594      258594         2370751         YOUNG FAMILIES
Premium
258595      258595         2370961         OLDER FAMILIES
```

```
Budget
258596      258596         2373711   YOUNG SINGLES/COUPLES
Mainstream

               DATE  STORE_NBR  TXN_ID  PROD_NBR  \
0       2018-10-17          1       1         5
1       2018-09-16          1       2        58
2       2019-03-07          1       3        52
3       2019-03-08          1       4       106
4       2018-11-02          1       5        96
...            ...        ...     ...       ...
258592  2018-08-03         88  240350         4
258593  2018-12-08         88  240378        24
258594  2018-10-01         88  240394        60
258595  2018-10-24         88  240480        70
258596  2018-12-14         88  241815        16

                                        PROD_NAME  PROD_QTY  TOT_SALES
WEIGHT  \
0         Natural Chip        Compny SeaSalt175g         2        6.0
175g
1          Red Rock Deli Chikn&Garlic Aioli 150g         1        2.7
150g
2          Grain Waves Sour    Cream&Chives 210G         1        3.6
210g
3         Natural ChipCo      Hony Soy Chckn175g         1        3.0
175g
4                 WW Original Stacked Chips 160g         1        1.9
160g
...                                          ...       ...        ...
...
258592          Dorito Corn Chp     Supreme 380g         2       13.0
380g
258593     Grain Waves         Sweet Chilli 210g         2        7.2
210g
258594      Kettle Tortilla ChpsFeta&Garlic 150g         2        9.2
150g
258595   Tyrrells Crisps     Lightly Salted 165g         2        8.4
165g
258596  Smiths Crinkle Chips Salt & Vinegar 330g         2       11.4
330g

           BRAND
0        Natural
1            Red
2          Grain
3        Natural
4             WW
...          ...
258592   Doritos
```

```
258593     Grain
258594    Kettle
258595  Tyrrells
258596    Smiths

[258597 rows x 13 columns]
```

```
# removing unncessary columns

qvi_merged = qvi_merged.drop('Unnamed: 0' , axis = 1)

qvi_merged.to_csv('qvi_merged.csv')

# Date column
qvi_merged['DATE'].dtype
```

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

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

qvi_merged['MONTH_YEAR']
```

```
0         10/2018
1         09/2018
2         03/2019
3         03/2019
4         11/2018
          ...
258592    08/2018
258593    12/2018
258594    10/2018
258595    10/2018
258596    12/2018
Name: MONTH_YEAR, Length: 258597, dtype: object
```

```
# Now we want to find comparable stores. I will isolate the timeframe
from July 2018 to January 31st 2019

qvi_merged['MONTH_YEAR'] = pd.to_datetime(qvi_merged['MONTH_YEAR'])

qvi_isolate = qvi_merged[(qvi_merged['MONTH_YEAR'] >= '07/2018') &
(qvi_merged['MONTH_YEAR'] <= '01/2019')]

qvi_isolate['MONTH_YEAR'].value_counts()
```

```
2018-12-01    22283
2018-07-01    22053
2018-08-01    21838
2018-10-01    21767
2019-01-01    21623
2018-11-01    21332
```

```
2018-09-01    21221
Name: MONTH_YEAR, dtype: int64
```

```
# grouping by store number and month year

qvi_isolate.groupby(['STORE_NBR','MONTH_YEAR'])['TOT_SALES'].sum()
```

```
STORE_NBR  MONTH_YEAR
1          2018-07-01    191.6
           2018-08-01    171.0
           2018-09-01    273.7
           2018-10-01    188.1
           2018-11-01    187.5
                         ...
272        2018-09-01    304.7
           2018-10-01    420.4
           2018-11-01    366.0
           2018-12-01    383.5
           2019-01-01    402.6
Name: TOT_SALES, Length: 1848, dtype: float64
```

```
# grouping by store number and total sales

qvi_isolate.groupby('STORE_NBR')['TOT_SALES'].sum()
```

```
STORE_NBR
1      1330.80
2      1113.20
3      7255.85
4      8759.80
5      5571.40
        ...
268    1492.95
269    6511.50
270    6555.15
271    5612.10
272    2611.75
Name: TOT_SALES, Length: 271, dtype: float64
```

Julia has asked us to evaluate performance of store trials in stores 77 , 86 and 88

```
qvi_isolate.groupby('STORE_NBR')['TOT_SALES'].sum().iloc[76:88]
```

```
STORE_NBR
77    1678.60
78    5405.20
79    6826.95
80    6555.60
81    7913.50
```

```
82    2259.30
83    5627.60
84    3116.10
85      13.90
86    6007.65
87    2242.70
88    9077.60
Name: TOT_SALES, dtype: float64
```

Total Sales for trials stores between July 2018 and Jan 2019

Store 77: 1678.60

Store 86: 6007.65

Store 88: 9077.60

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 sales during this period. Then i will use the Pearsons correlation test to determine how correlated the 2 stores are

Store 77

```
# Sorting stores by TOT_SALES and looking for similar match with store 77
qvi_isolate.groupby('STORE_NBR')
['TOT_SALES'].sum().sort_values(ascending = True).iloc[63:73]
```

```
STORE_NBR
255    1565.20
6      1567.80
195    1588.05
233    1608.80
188    1632.50
187    1651.20
46     1671.30
77     1678.60
90     1685.40
220    1727.40
Name: TOT_SALES, dtype: float64
```

```
# isolating the 10 stores
stores_10 = [255,6,195,233,188,187,46,77,90,220]
qvi_10 = qvi_isolate[qvi_isolate['STORE_NBR'].isin(stores_10)]

qvi_10.groupby('STORE_NBR')['TOT_SALES'].sum()
```

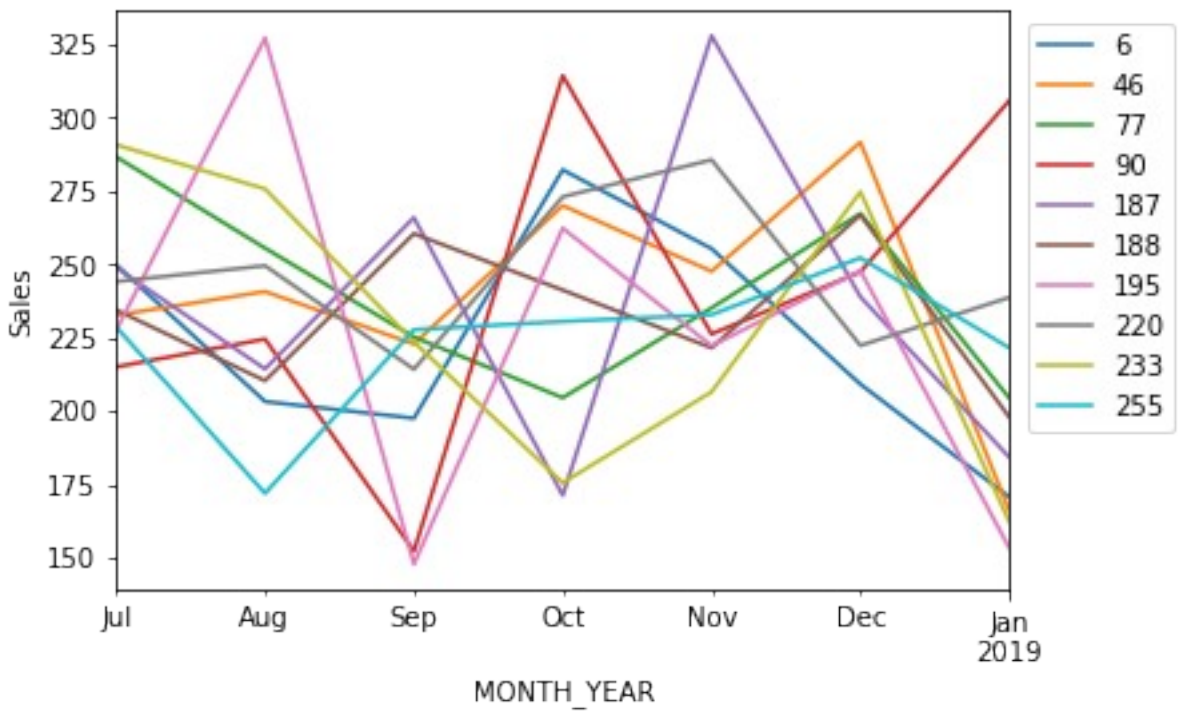
```
STORE_NBR
6      1567.80
46     1671.30
```

```
77     1678.60
90     1685.40
187    1651.20
188    1632.50
195    1588.05
220    1727.40
233    1608.80
255    1565.20
Name: TOT_SALES, dtype: float64
```

```
qvi_10.pivot_table(index = 'MONTH_YEAR', columns = 'STORE_NBR', values
= 'TOT_SALES', aggfunc = 'sum')
```

```
STORE_NBR     6      46     77     90     187    188     195    220
233  \
MONTH_YEAR

2018-07-01  249.8  232.6  286.6  215.0  248.8  234.4  227.50  244.1
290.7
2018-08-01  203.2  240.7  255.5  224.5  214.3  210.3  327.15  249.5
275.7
2018-09-01  197.5  222.8  225.2  152.2  265.9  260.4  147.80  214.2
223.5
2018-10-01  282.2  270.0  204.5  314.3  171.2  241.2  262.30  273.0
175.5
2018-11-01  255.3  247.6  235.1  226.2  327.9  221.5  222.20  285.6
206.5
2018-12-01  209.1  291.6  267.3  247.4  238.9  266.8  247.80  222.3
274.7
2019-01-01  170.7  166.0  204.4  305.8  184.2  197.9  153.30  238.7
162.2

STORE_NBR     255
MONTH_YEAR
2018-07-01  228.6
2018-08-01  171.9
2018-09-01  227.7
2018-10-01  230.5
2018-11-01  232.8
2018-12-01  252.2
2019-01-01  221.5
```

```
# line chart
qvi_pivot = qvi_10.pivot_table(index = 'MONTH_YEAR', columns =
'STORE_NBR', values = 'TOT_SALES', aggfunc = 'sum')

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



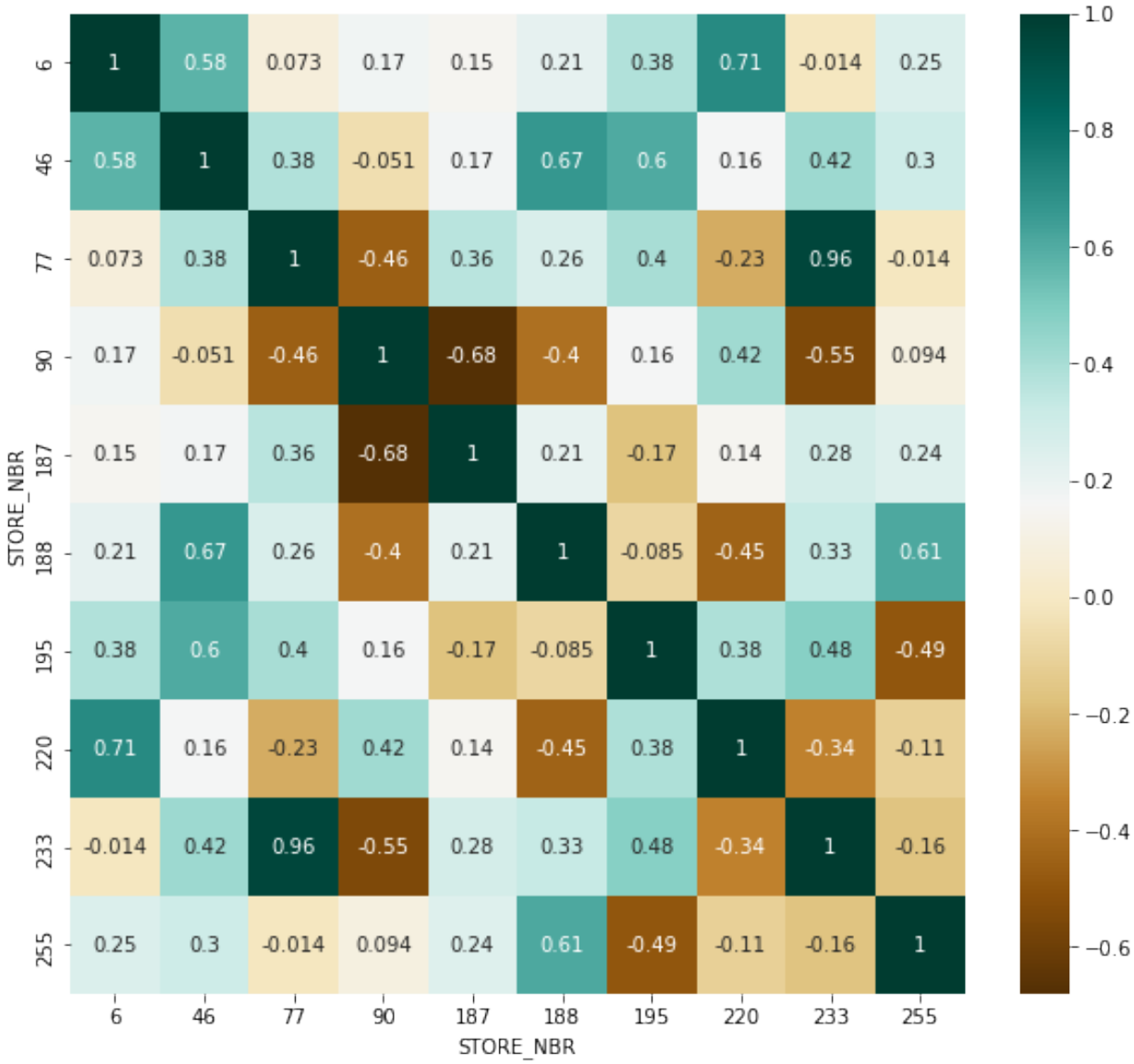

```
# Lets take a closer look at the correlations between them

# looking at correlation
qvi_corr = qvi_pivot.corr(method = 'pearson')

qvi_corr
```

STORE_NBR	6	46	77	90	187	188
6	1.000000	0.579596	0.073448	0.173641	0.145235	0.212723
46	0.579596	1.000000	0.382361	-0.050651	0.167422	0.665080
77	0.073448	0.382361	1.000000	-0.457788	0.359139	0.261052
90	0.173641	-0.050651	-0.457788	1.000000	-0.680752	-0.399336
187	0.145235	0.167422	0.359139	-0.680752	1.000000	0.205976
188	0.212723	0.665080	0.261052	-0.399336	0.205976	1.000000
195	0.377418	0.603646	0.399575	0.161756	-0.170295	-0.085156
220	0.709190	0.163844	-0.228226	0.418310	0.140991	-0.445830
233	-0.014185	0.424350	0.958719	-0.551455	0.280892	0.330255

```
255         0.252768  0.299840 -0.013620  0.093712  0.241345  0.608366
```

```
STORE_NBR       195       220       233       255
STORE_NBR
6          0.377418  0.709190 -0.014185  0.252768
46         0.603646  0.163844  0.424350  0.299840
77         0.399575 -0.228226  0.958719 -0.013620
90         0.161756  0.418310 -0.551455  0.093712
187       -0.170295  0.140991  0.280892  0.241345
188       -0.085156 -0.445830  0.330255  0.608366
195        1.000000  0.382883  0.478041 -0.490752
220        0.382883  1.000000 -0.342800 -0.113872
233        0.478041 -0.342800  1.000000 -0.164438
255       -0.490752 -0.113872 -0.164438  1.000000
```

```
# understanding the correlation through a heatmap
plt.figure(figsize = [10,9])
sns.heatmap(qvi_corr, cmap = 'BrBG', annot = True)
plt.show()
```



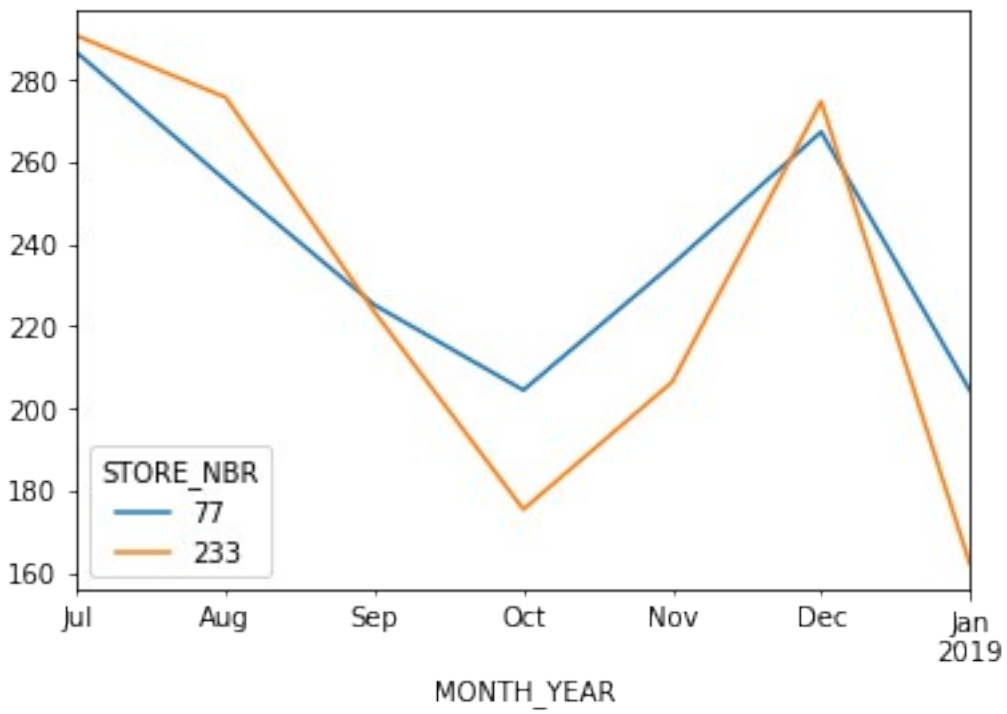

```
# It can be seen that STORE_NBR 233 correlates to store 77 the most
```

Difference Between Correlation & Causation

Correlation: Correlation simply means that two or more things tend to change together. When one thing changes, the other tends to change in a related way. Ex: Ice cream sales and crime rates might be positively correlated. When ice cream sales go up, crime rates might also go up.

Causation: Causation means that one thing directly causes another thing to happen. One event is the direct result of another event. Ex: Smoking causes an increased risk of lung cancer.

```
# Line chart showing the relation between store 77 and store 233
qvi_pivot[[77,233]].plot()
plt.show()
```



Store 86

```
# looking for stores with sales similar to store 86
qvi_isolate.groupby('STORE_NBR')
['TOT_SALES'].sum().sort_values(ascending = True).iloc[176:186]
```

```
STORE_NBR
13     5946.40
39     5954.30
172    5960.40
91     5969.90
57     5974.00
236    5983.20
30     6000.80
86     6007.65
32     6007.80
207    6011.70
Name: TOT_SALES, dtype: float64
```

```
# Isolating the above 10 stores
stalls_10 = [13,39,172,91,57,236,30,86,32,207]
qvi_stalls = qvi_isolate[qvi_isolate['STORE_NBR'].isin(stalls_10)]

qvi_stalls.groupby('STORE_NBR')['TOT_SALES'].sum()
```

```
STORE_NBR
13     5946.40
30     6000.80
```

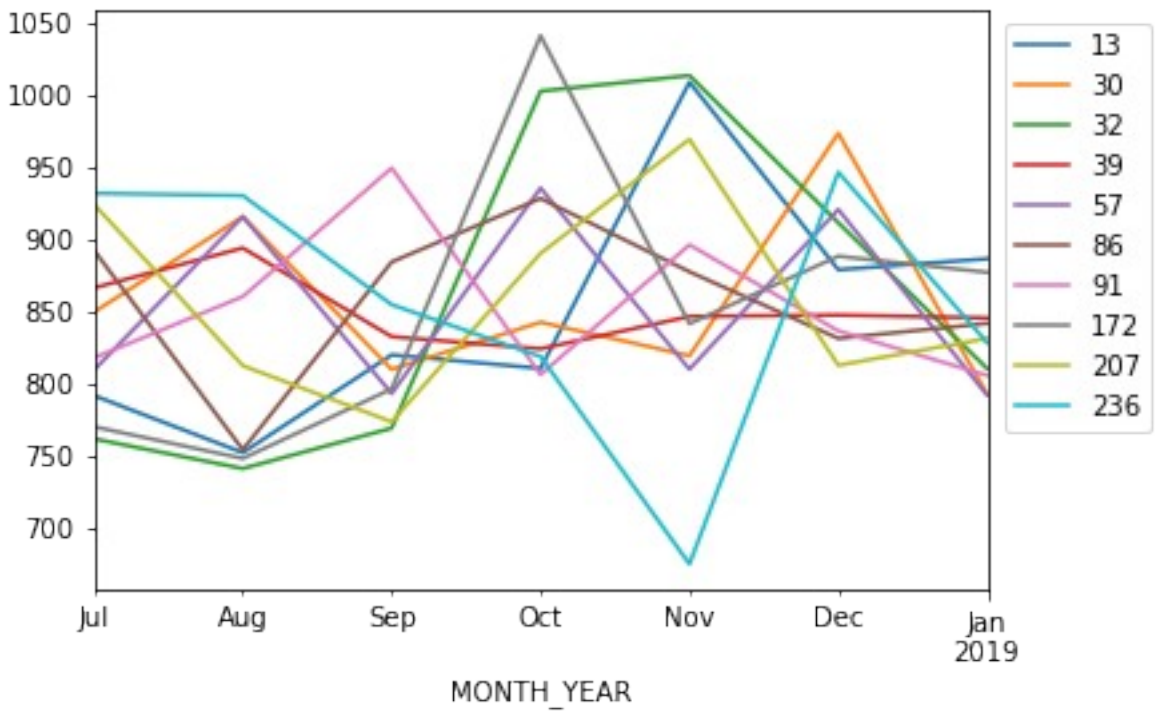
```
32     6007.80
39     5954.30
57     5974.00
86     6007.65
91     5969.90
172    5960.40
207    6011.70
236    5983.20
Name: TOT_SALES, dtype: float64
```

```
qvistalls_pivot = qvi_stalls.pivot_table(index = 'MONTH_YEAR', columns
= 'STORE_NBR',
                                         values = 'TOT_SALES', aggfunc
= 'sum')
```

```
qvistalls_pivot
```

```
STORE_NBR      13     30      32     39     57      86     91      172
207  \
MONTH_YEAR

2018-07-01   791.4  849.2   761.4  866.0  809.0  892.20  817.5   769.8
924.0
2018-08-01   751.8  915.4   740.8  893.7  915.4  753.85  860.0   747.8
812.1
2018-09-01   819.6  809.6   768.8  832.2  792.8  884.00  949.1   796.0
773.2
2018-10-01   810.2  842.2  1002.4  823.8  935.2  928.00  805.9  1040.8
890.0
2018-11-01  1008.6  819.0  1013.2  846.2  809.6  877.20  895.9   841.2
969.0
2018-12-01   878.6  973.2   911.2  847.2  920.4  831.00  836.2   888.0
812.4
2019-01-01   886.2  792.2   810.0  845.2  791.6  841.40  805.3   876.8
831.0

STORE_NBR     236
MONTH_YEAR
2018-07-01  931.6
2018-08-01  930.0
2018-09-01  854.4
2018-10-01  818.4
2018-11-01  674.6
2018-12-01  946.2
2019-01-01  828.0
```

```
qvistalls_pivot.plot()
plt.legend(loc = 'upper right', bbox_to_anchor = (1.2,1))
plt.show()
```



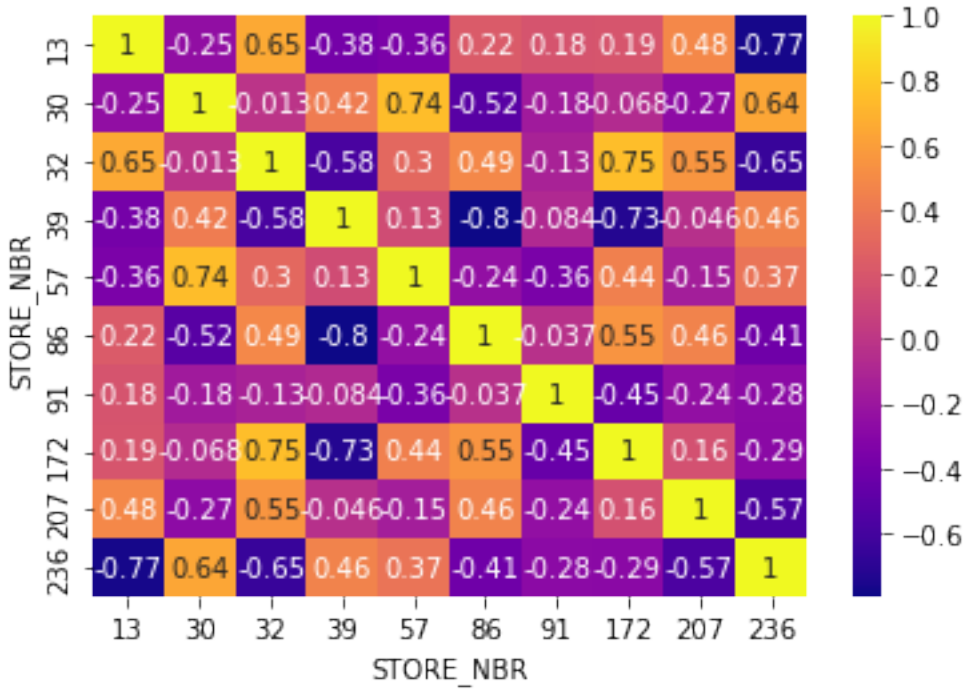

```
# lets take a closer look on the correlations between them

# checking for correlation
qvistalls_corr = qvistalls_pivot.corr(method = 'pearson')

qvistalls_corr
```

STORE_NBR	13	30	32	39	57	86
13	1.000000	-0.253009	0.653562	-0.383470	-0.361657	0.221650
30	-0.253009	1.000000	-0.013434	0.422020	0.741519	-0.523107
32	0.653562	-0.013434	1.000000	-0.580733	0.301956	0.485270
39	-0.383470	0.422020	-0.580733	1.000000	0.131380	-0.796925
57	-0.361657	0.741519	0.301956	0.131380	1.000000	-0.243514
86	0.221650	-0.523107	0.485270	-0.796925	-0.243514	1.000000
91	0.182220	-0.176581	-0.126263	-0.084188	-0.362802	-0.037260
172	0.188841	-0.067829	0.747626	-0.729685	0.442481	0.547095
207	0.483183	-0.271318	0.554998	-0.045670	-0.152152	0.457724

```
236       -0.773507  0.637554 -0.651337  0.456373  0.368571 -0.412316
```

STORE_NBR	91	172	207	236
STORE_NBR				
13	0.182220	0.188841	0.483183	-0.773507
30	-0.176581	-0.067829	-0.271318	0.637554
32	-0.126263	0.747626	0.554998	-0.651337
39	-0.084188	-0.729685	-0.045670	0.456373
57	-0.362802	0.442481	-0.152152	0.368571
86	-0.037260	0.547095	0.457724	-0.412316
91	1.000000	-0.452799	-0.243380	-0.278526
172	-0.452799	1.000000	0.159153	-0.291327
207	-0.243380	0.159153	1.000000	-0.570103
236	-0.278526	-0.291327	-0.570103	1.000000

```
# plotting a heatmap
sns.heatmap(qvistalls_corr, cmap = 'plasma', annot = True)
plt.show()
```




```
# From the graph, we can observe that the strongest correlation to
store 86 is store 172 with 0.55$

# Line chart showing the correlation between store 86 and 172

qvistalls_pivot[[86,172]].plot()
plt.show()
```


```
# from the chart, it can be seen that the correlation between the two
is pretty bad. So lets look further

# looking for stores with sales similar to store 86
qvi_isolate.groupby('STORE_NBR')
['TOT_SALES'].sum().sort_values(ascending = True).iloc[181:195]
```

```
STORE_NBR
236    5983.20
30     6000.80
86     6007.65
32     6007.80
207    6011.70
105    6032.80
221    6039.60
180    6087.20
97     6091.55
102    6092.20
62     6102.30
109    6105.70
164    6116.00
227    6122.30
Name: TOT_SALES, dtype: float64
```

```
# isolating 10 stores from above
Stores_10 = [236,30,86,32,207,105,221,180,97,102,62,109,164,227]

QVI_STALLS = qvi_isolate[qvi_isolate['STORE_NBR'].isin(Stores_10)]
```

```
QVI_STALLS.groupby('STORE_NBR')
['TOT_SALES'].sum().sort_values(ascending = True)
```

```
STORE_NBR
236    5983.20
30     6000.80
86     6007.65
32     6007.80
207    6011.70
105    6032.80
221    6039.60
180    6087.20
97     6091.55
102    6092.20
62     6102.30
109    6105.70
164    6116.00
227    6122.30
Name: TOT_SALES, dtype: float64
```

```
QVI_STALLS_PIVOT = QVI_STALLS.pivot_table(index = 'MONTH_YEAR',
columns = 'STORE_NBR',
                                          values = 'TOT_SALES',
aggfunc = 'sum')
QVI_STALLS_PIVOT
```

STORE_NBR	30	32	62	86	97	102	105	109
MONTH_YEAR								
2018-07-01	849.2	761.4	963.2	892.20	848.20	762.0	888.1	873.8
2018-08-01	915.4	740.8	766.9	753.85	907.15	955.8	913.5	818.1
2018-09-01	809.6	768.8	932.0	884.00	857.80	939.8	826.2	830.6
2018-10-01	842.2	1002.4	819.8	928.00	962.60	851.2	869.8	927.0
2018-11-01	819.0	1013.2	891.6	877.20	833.00	889.2	730.6	909.2
2018-12-01	973.2	911.2	871.6	831.00	858.60	806.4	1007.8	898.6
2019-01-01	792.2	810.0	857.2	841.40	824.20	887.8	796.8	848.4

STORE_NBR	164	180	207	221	227	236
MONTH_YEAR						
2018-07-01	812.4	796.2	924.0	915.5	875.6	931.6
2018-08-01	910.0	773.2	812.1	906.2	852.1	930.0
2018-09-01	810.8	860.4	773.2	691.9	758.6	854.4

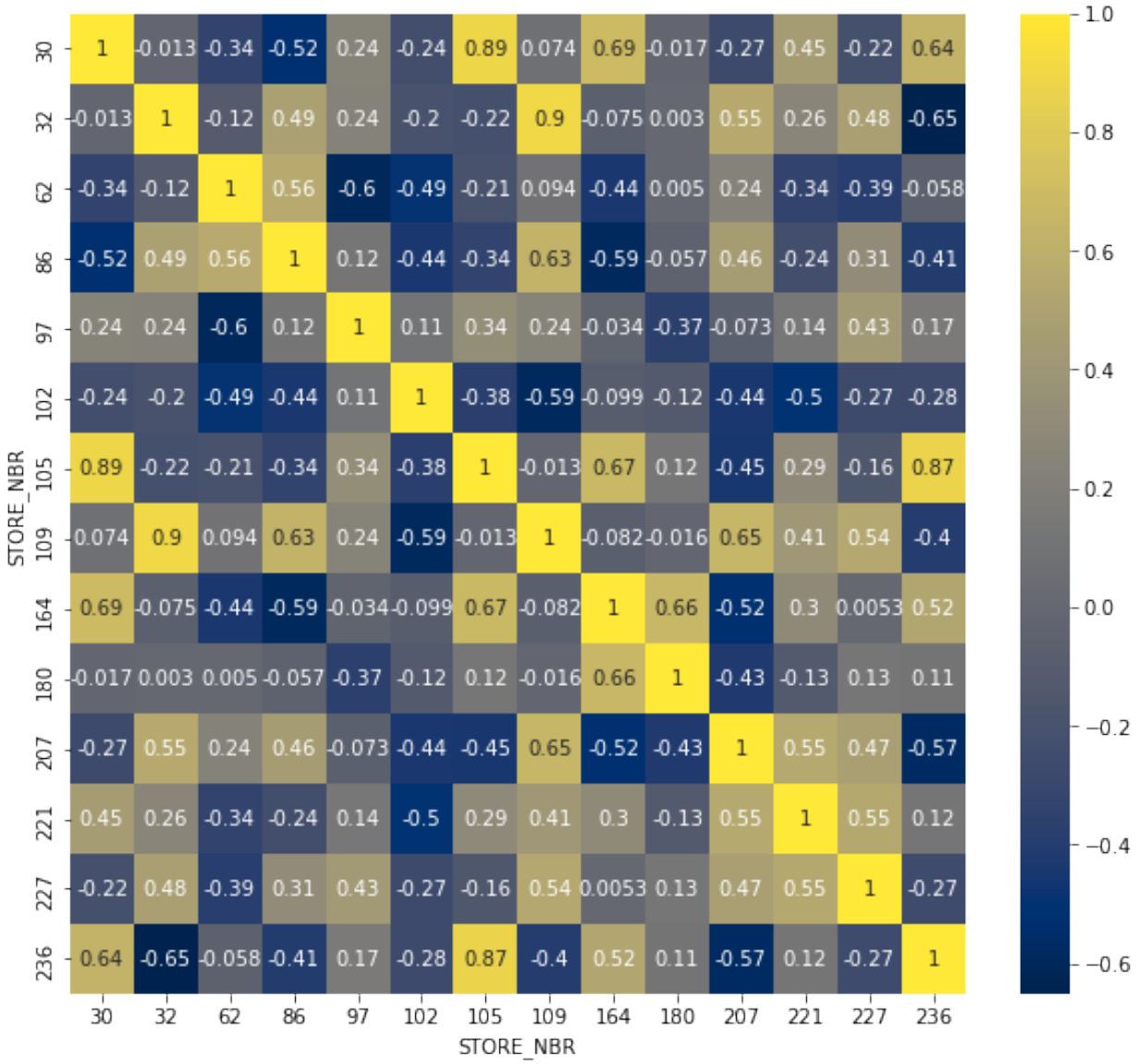
```
2018-10-01   842.8   840.8  890.0  881.3  995.0  818.4
2018-11-01   799.0   793.4  969.0  889.4  874.4  674.6
2018-12-01  1011.2   974.6  812.4  899.0  831.6  946.2
2019-01-01   929.8  1048.6  831.0  856.3  935.0  828.0
```

```
# defining the correlations between the stores
QVI_STALLS_PIVOT_corr = QVI_STALLS_PIVOT.corr(method = 'pearson')
QVI_STALLS_PIVOT_corr
```

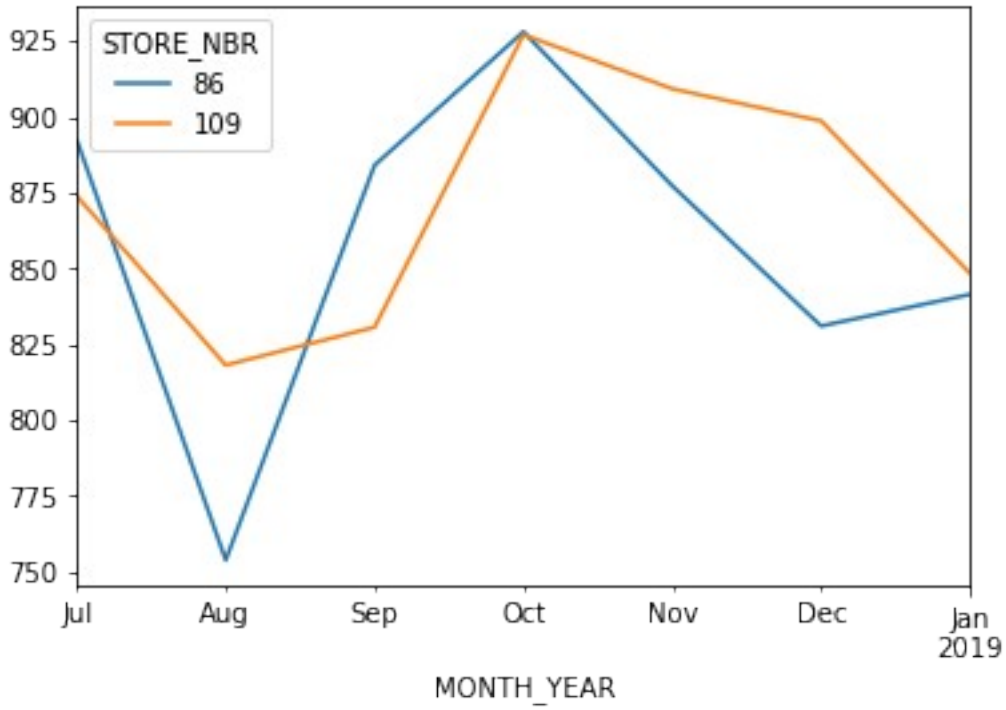
```
STORE_NBR       30        32        62        86        97        102
\
STORE_NBR

30          1.000000 -0.013434 -0.340832 -0.523107  0.243752 -0.239273

32         -0.013434  1.000000 -0.122370  0.485270  0.237743 -0.202612

62         -0.340832 -0.122370  1.000000  0.564459 -0.595740 -0.491208

86         -0.523107  0.485270  0.564459  1.000000  0.116329 -0.441087

97          0.243752  0.237743 -0.595740  0.116329  1.000000  0.105994

102        -0.239273 -0.202612 -0.491208 -0.441087  0.105994  1.000000

105         0.885422 -0.221564 -0.205450 -0.341106  0.335046 -0.377572

109         0.074223  0.903239  0.093739  0.632358  0.244801 -0.587082

164         0.690723 -0.075231 -0.442488 -0.589771 -0.033867 -0.098587

180        -0.017346  0.003018  0.004960 -0.057050 -0.367572 -0.124919

207        -0.271318  0.554998  0.238630  0.457724 -0.072884 -0.443634

221         0.449814  0.260066 -0.335869 -0.237343  0.139467 -0.498185

227        -0.223008  0.477464 -0.390559  0.313038  0.430175 -0.273915

236         0.637554 -0.651337 -0.058034 -0.412316  0.165856 -0.275359


STORE_NBR       105       109       164       180       207       221
\
STORE_NBR

30          0.885422  0.074223  0.690723 -0.017346 -0.271318  0.449814

32         -0.221564  0.903239 -0.075231  0.003018  0.554998  0.260066

62         -0.205450  0.093739 -0.442488  0.004960  0.238630 -0.335869
```

```
86         -0.341106  0.632358 -0.589771 -0.057050  0.457724 -0.237343

97          0.335046  0.244801 -0.033867 -0.367572 -0.072884  0.139467

102        -0.377572 -0.587082 -0.098587 -0.124919 -0.443634 -0.498185

105         1.000000 -0.013055  0.674212  0.124308 -0.447501  0.291178

109        -0.013055  1.000000 -0.081690 -0.015733  0.651704  0.411624

164         0.674212 -0.081690  1.000000  0.662787 -0.520946  0.301814

180         0.124308 -0.015733  0.662787  1.000000 -0.425262 -0.132393

207        -0.447501  0.651704 -0.520946 -0.425262  1.000000  0.548938

221         0.291178  0.411624  0.301814 -0.132393  0.548938  1.000000

227        -0.155004  0.538161  0.005284  0.128391  0.474542  0.546341

236         0.872923 -0.403035  0.523709  0.113057 -0.570103  0.121443


STORE_NBR       227       236
STORE_NBR
30         -0.223008  0.637554
32          0.477464 -0.651337
62         -0.390559 -0.058034
86          0.313038 -0.412316
97          0.430175  0.165856
102        -0.273915 -0.275359
105        -0.155004  0.872923
109         0.538161 -0.403035
164         0.005284  0.523709
180         0.128391  0.113057
207         0.474542 -0.570103
221         0.546341  0.121443
227         1.000000 -0.266682
236        -0.266682  1.000000
```

```
# presenting it through a heatmap
plt.figure(figsize = [10,9])
sns.heatmap(QVI_STALLS_PIVOT_corr, cmap = 'cividis', annot = True)
plt.show()
```




```
QVI_STALLS_PIVOT[[86,109]].plot()
plt.show()
```




```
# It can be seen that the strongest correlation to store 86 is store
109
```

Store 88

```
# looking for stores with sales similar to store 86
qvi_isolate.groupby('STORE_NBR')
['TOT_SALES'].sum().sort_values(ascending = True).iloc[260:]
```

```
STORE_NBR
181     8050.55
26      8111.50
199     8368.80
203     8474.50
40      8530.20
4       8759.80
58      8801.35
165     8957.30
88      9077.60
237     9088.50
226    10481.15
Name: TOT_SALES, dtype: float64
```

```
# isolating individual stores
individual_stores = [181,26,199,203,40,4,58,165,88,237,226]
qvi_individual =
qvi_isolate[qvi_isolate['STORE_NBR'].isin(individual_stores)]
```

```
qvi_individual.groupby('STORE_NBR')
['TOT_SALES'].sum().sort_values(ascending = True)
```

```
STORE_NBR
181     8050.55
26      8111.50
199     8368.80
203     8474.50
40      8530.20
4       8759.80
58      8801.35
165     8957.30
88      9077.60
237     9088.50
226    10481.15
Name: TOT_SALES, dtype: float64
```

```
qvi_individual_pivot = qvi_individual.pivot_table(index =
'MONTH_YEAR', columns = 'STORE_NBR',
                                                  values = 'TOT_SALES',
aggfunc = 'sum')
qvi_individual_pivot
```

```
STORE_NBR      4       26      40       58      88      165      181
199  \
MONTH_YEAR

2018-07-01  1328.5  1214.4  1281.0  1586.40  1259.0  1406.0  1323.80
1258.8
2018-08-01  1218.7   985.7  1187.6   992.75  1272.8  1181.1  1010.15
1194.8
2018-09-01  1178.2  1222.0  1271.2  1369.00  1372.0  1250.6  1104.20
1103.0
2018-10-01  1336.2  1093.2  1252.2  1368.80  1311.6  1203.8  1218.50
1242.2
2018-11-01  1099.8  1186.8  1058.6  1103.80  1352.2  1250.4  1121.60
1100.8
2018-12-01  1155.0  1276.2  1264.0  1144.80  1284.4  1274.0  1169.40
1285.4
2019-01-01  1443.4  1133.2  1215.6  1235.80  1225.6  1391.4  1102.90
1183.8

STORE_NBR      203      226     237
MONTH_YEAR
2018-07-01  1215.8  1408.80  1438.2
2018-08-01  1135.2  1819.25  1332.1
2018-09-01  1215.8  1339.00  1281.4
2018-10-01  1241.5  1468.00  1317.7
2018-11-01  1265.2  1561.80  1336.4
```

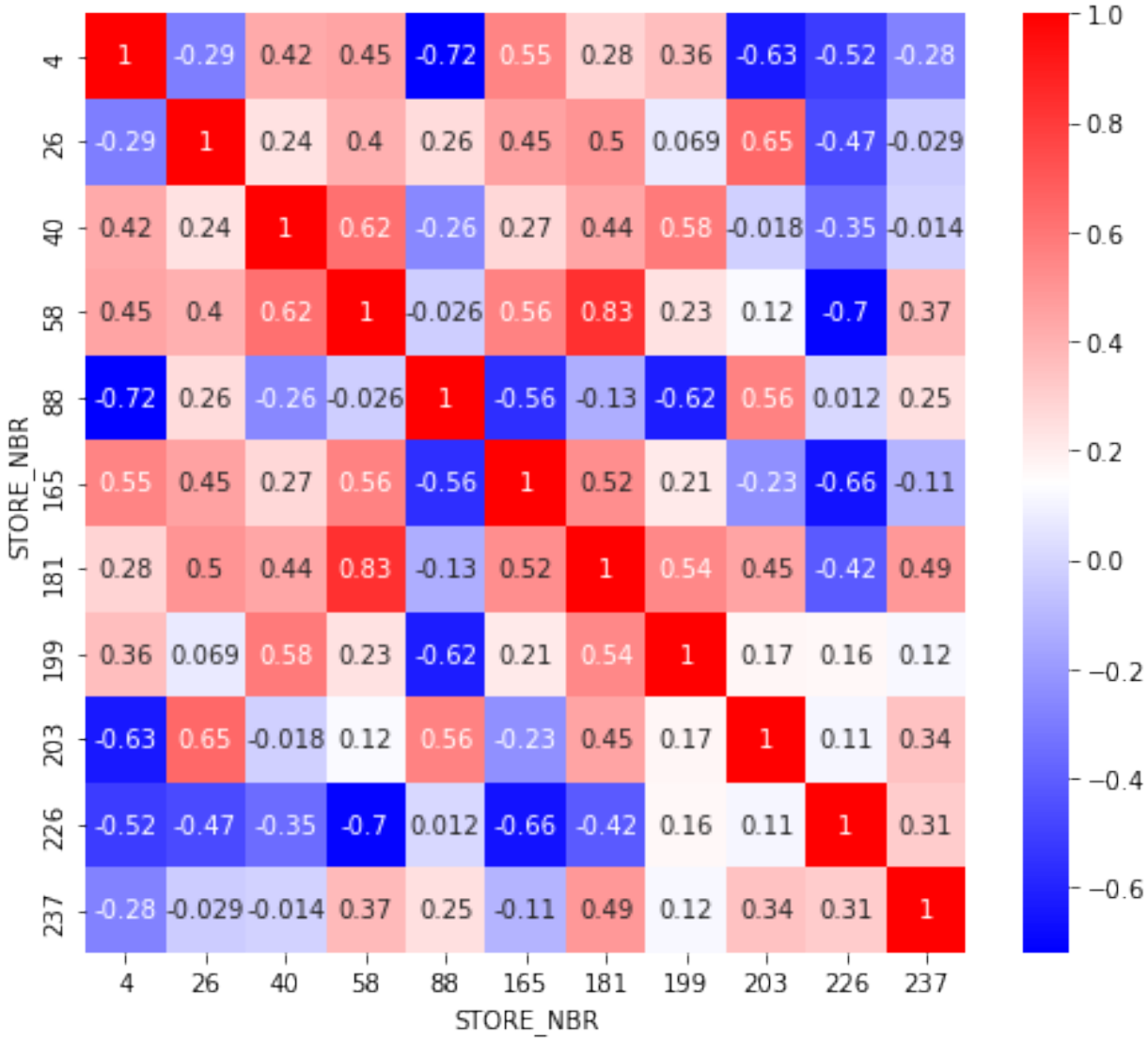
```
2018-12-01  1295.8  1598.40  1244.6
2019-01-01  1105.2  1285.90  1138.1
```

```
# applying correlation pearson method
qvi_individual_pivot_corr = qvi_individual_pivot.corr(method =
'pearson')
qvi_individual_pivot_corr
```

```
STORE_NBR       4         26        40        58        88        165
\
STORE_NBR

4          1.000000 -0.292481  0.424487  0.447593 -0.719568  0.551497

26        -0.292481  1.000000  0.239741  0.402430  0.256310  0.450239

40         0.424487  0.239741  1.000000  0.619188 -0.262696  0.268573

58         0.447593  0.402430  0.619188  1.000000 -0.025992  0.560698

88        -0.719568  0.256310 -0.262696 -0.025992  1.000000 -0.560361

165        0.551497  0.450239  0.268573  0.560698 -0.560361  1.000000

181        0.281470  0.500875  0.441673  0.833063 -0.131801  0.522489

199        0.358477  0.069014  0.584823  0.234988 -0.623525  0.213898

203       -0.634010  0.649536 -0.018088  0.121735  0.560412 -0.230972

226       -0.517834 -0.470889 -0.353580 -0.700040  0.011568 -0.656478

237       -0.278409 -0.028741 -0.014068  0.369428  0.246313 -0.110443


STORE_NBR       181       199       203       226       237
STORE_NBR
4          0.281470  0.358477 -0.634010 -0.517834 -0.278409
26         0.500875  0.069014  0.649536 -0.470889 -0.028741
40         0.441673  0.584823 -0.018088 -0.353580 -0.014068
58         0.833063  0.234988  0.121735 -0.700040  0.369428
88        -0.131801 -0.623525  0.560412  0.011568  0.246313
165        0.522489  0.213898 -0.230972 -0.656478 -0.110443
181        1.000000  0.537530  0.448574 -0.415320  0.488507
199        0.537530  1.000000  0.169293  0.162880  0.117025
203        0.448574  0.169293  1.000000  0.107146  0.344615
226       -0.415320  0.162880  0.107146  1.000000  0.313553
237        0.488507  0.117025  0.344615  0.313553  1.000000
```

```
# plotting a heatmap
plt.figure(figsize = [8,7])
```

```
sns.heatmap(qvi_individual_pivot_corr, cmap = 'bwr', annot = True)
plt.show()
```




```
qvi_individual_pivot[[88,203]].plot()
plt.show()
```


```
# store 203 is the only store that strongly correlates to store 88
```

Comparing the trial stores vs control stores

Now that i have the control stores and the trial stores its time to run some analysis on the trial sales period.

First i will start with an overview comparing them separately. I will isolate the trial periods and compare how they performed to the control stores during the same months. I will look at the total sales, total customers, total purchases for each

```
qvi_merged.head(4)
```

```
   LYLTY_CARD_NBR               LIFESTAGE PREMIUM_CUSTOMER
DATE  \
0            1000  YOUNG SINGLES/COUPLES          Premium 2018-10-17

1            1002  YOUNG SINGLES/COUPLES       Mainstream 2018-09-16

2            1003         YOUNG FAMILIES           Budget 2019-03-07

3            1003         YOUNG FAMILIES           Budget 2019-03-08


   STORE_NBR  TXN_ID  PROD_NBR                                PROD_NAME
\
0          1       1         5  Natural Chip        Compny SeaSalt175g
```

```
1           1       2        58   Red Rock Deli Chikn&Garlic Aioli 150g

2           1       3        52   Grain Waves Sour    Cream&Chives 210G

3           1       4       106  Natural ChipCo      Hony Soy Chckn175g


   PROD_QTY  TOT_SALES WEIGHT    BRAND MONTH_YEAR
0         2        6.0   175g  Natural 2018-10-01
1         1        2.7   150g      Red 2018-09-01
2         1        3.6   210g    Grain 2019-03-01
3         1        3.0   175g  Natural 2019-03-01
```

```
qvi_2019 = qvi_merged[(qvi_merged['MONTH_YEAR'] >= '02/2019') &
(qvi_merged['MONTH_YEAR'] <= '04/2019')]

qvi_2019['MONTH_YEAR'].value_counts()
```

```
2019-03-01    22091
2019-04-01    21284
2019-02-01    19905
Name: MONTH_YEAR, dtype: int64
```

Store 77 and Store 233

```
# creating new df for trial and control stores

qvi_77 = qvi_2019[qvi_2019['STORE_NBR'] == 77]
qvi_233 = qvi_2019[qvi_2019['STORE_NBR'] == 233]

# looking at products sold and total sales
qvi_77[['TOT_SALES','PROD_QTY']].sum()
```

```
TOT_SALES    766.8
PROD_QTY     232.0
dtype: float64
```

```
qvi_233[['TOT_SALES','PROD_QTY']].sum()
```

```
TOT_SALES    581.3
PROD_QTY     171.0
dtype: float64
```

```
# looking for customers that repeated purchasing from the stores
qvi_77['LYLTY_CARD_NBR'].count()
```

```
146
```

```
qvi_77['LYLTY_CARD_NBR'].value_counts().head(25)
# 24 customers that have purchased twice
```

```
77000    2
77206    2
77207    2
77350    2
77420    2
77424    2
77123    2
77402    2
77450    2
77077    2
77359    2
77454    2
77069    2
77341    2
77462    2
77466    2
77338    2
77045    2
77482    2
77139    2
77389    2
77009    2
77007    2
77115    2
77363    1
Name: LYLTY_CARD_NBR, dtype: int64
```

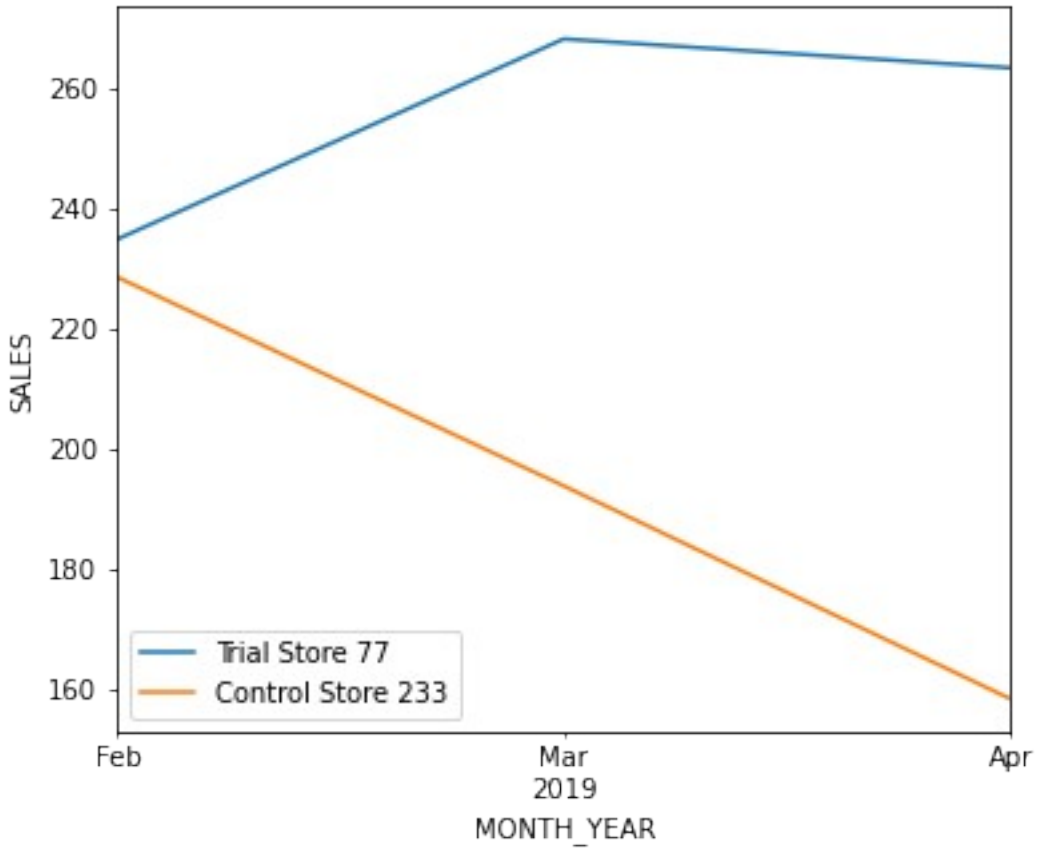
```
qvi_233['LYLTY_CARD_NBR'].count()
```

```
118
```

```
qvi_233['LYLTY_CARD_NBR'].value_counts().head(15)
# 9 customers have repeated purchasing from store 233
```

```
233071    2
233186    2
233227    2
233327    2
233111    2
233398    2
233449    2
233284    2
233341    2
233276    1
233313    1
233236    1
233336    1
233238    1
233322    1
Name: LYLTY_CARD_NBR, dtype: int64
```

```
plt.figure(figsize = [6,5])
qvi_77.groupby('MONTH_YEAR')['TOT_SALES'].sum().plot(label = 'Trial
Store 77')
qvi_233.groupby('MONTH_YEAR')['TOT_SALES'].sum().plot(label = 'Control
Store 233')
plt.ylabel('SALES')
plt.legend()
plt.show()
```



Summary

Store 77 :

-- No. of customers between feb and april = 146

-- No. of repeated customers = 24

-- TOT_SALES = 766.8

-- PROD_QTY = 232.0

Store 233 :

-- No. of customers between feb and april = 118

-- No. of repeated customers = 9

-- TOT_SALES = 581.3

-- PROD_QTY = 171.0

Store 86 and 109

```
qvi_86 = qvi_2019[qvi_2019['STORE_NBR'] == 86]
qvi_109 = qvi_2019[qvi_2019['STORE_NBR'] == 109]

qvi_86.groupby('MONTH_YEAR')['TOT_SALES'].sum()
```

```
MONTH_YEAR
2019-02-01     903.0
2019-03-01    1006.4
2019-04-01     838.0
Name: TOT_SALES, dtype: float64
```

```
qvi_109.groupby('MONTH_YEAR')['TOT_SALES'].sum()
```

```
MONTH_YEAR
2019-02-01     858.4
2019-03-01    1029.0
2019-04-01     718.4
Name: TOT_SALES, dtype: float64
```

```
# looking at the total sales and product quantity
qvi_86[['TOT_SALES','PROD_QTY']].sum()
```

```
TOT_SALES    2747.4
PROD_QTY      807.0
dtype: float64
```

```
qvi_109[['TOT_SALES','PROD_QTY']].sum()
```

```
TOT_SALES    2605.8
PROD_QTY      742.0
dtype: float64
```

```
# checking for repeated and number of customers
qvi_86['LYLTY_CARD_NBR'].count()
```

```
404
```

```
qvi_86['LYLTY_CARD_NBR'].value_counts().head(125)
# 122 are repeated customers
```

```
86112    6
86075    5
86230    5
86116    5
86172    5
```

```
        ..
86048    2
86248    2
86196    1
86199    1
86203    1
Name: LYLTY_CARD_NBR, Length: 125, dtype: int64
```

```
qvi_109['LYLTY_CARD_NBR'].count()
```

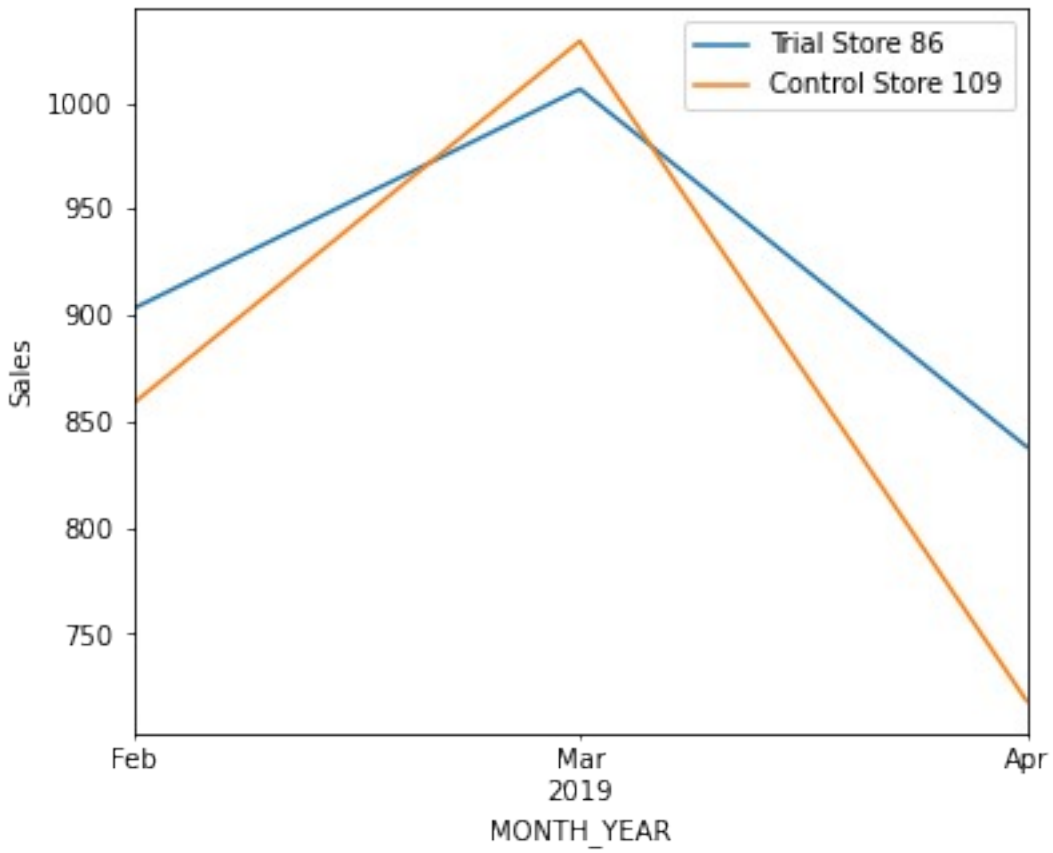
```
371
```

```
qvi_109['LYLTY_CARD_NBR'].value_counts().head(105)
# 102 customers have repeated purchasing from the store
```

```
109255    6
109207    6
109227    5
109139    5
109045    5
         ..
109095    2
109004    2
109212    1
109217    1
109216    1
Name: LYLTY_CARD_NBR, Length: 105, dtype: int64
```

```
plt.figure(figsize = [6,5])
qvi_86.groupby('MONTH_YEAR')['TOT_SALES'].sum().plot(label = 'Trial
Store 86')
qvi_109.groupby('MONTH_YEAR')['TOT_SALES'].sum().plot(label = 'Control
Store 109')
plt.ylabel('Sales')
plt.legend()
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Summary

Store 86 : -- No. of customers between feb and april = 404

-- No. of repeated customers = 122

-- TOT_SALES = 2747.4

-- PROD_QTY = 807.0

Store 109 :

-- No. of customers between feb and april = 371

-- No. of repeated customers = 102

-- TOT_SALES = 2605.8

-- PROD_QTY = 742.0

Store 88 & 203

Mind you we are taking the data between feb 2019 to april 2019

```
qvi_88 = qvi_2019[qvi_2019['STORE_NBR'] == 88]
qvi_203 = qvi_2019[qvi_2019['STORE_NBR'] == 203]
```

```
# Looking at the no. of products sold and total sales
qvi_88[['TOT_SALES','PROD_QTY']].sum()
```

```
TOT_SALES    4195.0
PROD_QTY      954.0
dtype: float64
```

```
qvi_203[['TOT_SALES','PROD_QTY']].sum()
```

```
TOT_SALES    3178.8
PROD_QTY      724.0
dtype: float64
```

```
# looking for repeated customers
qvi_88[['LYLTY_CARD_NBR']].count()
```

```
LYLTY_CARD_NBR    477
dtype: int64
```

```
qvi_88[['LYLTY_CARD_NBR']].value_counts().head(140)
# 139 repeated customers
```

```
LYLTY_CARD_NBR
88313             6
88231             5
88259             4
88105             4
88111             4
                 ..
88010             2
88011             2
88129             2
88128             2
88036             1
Length: 140, dtype: int64
```

```
qvi_203[['LYLTY_CARD_NBR']].count()
```

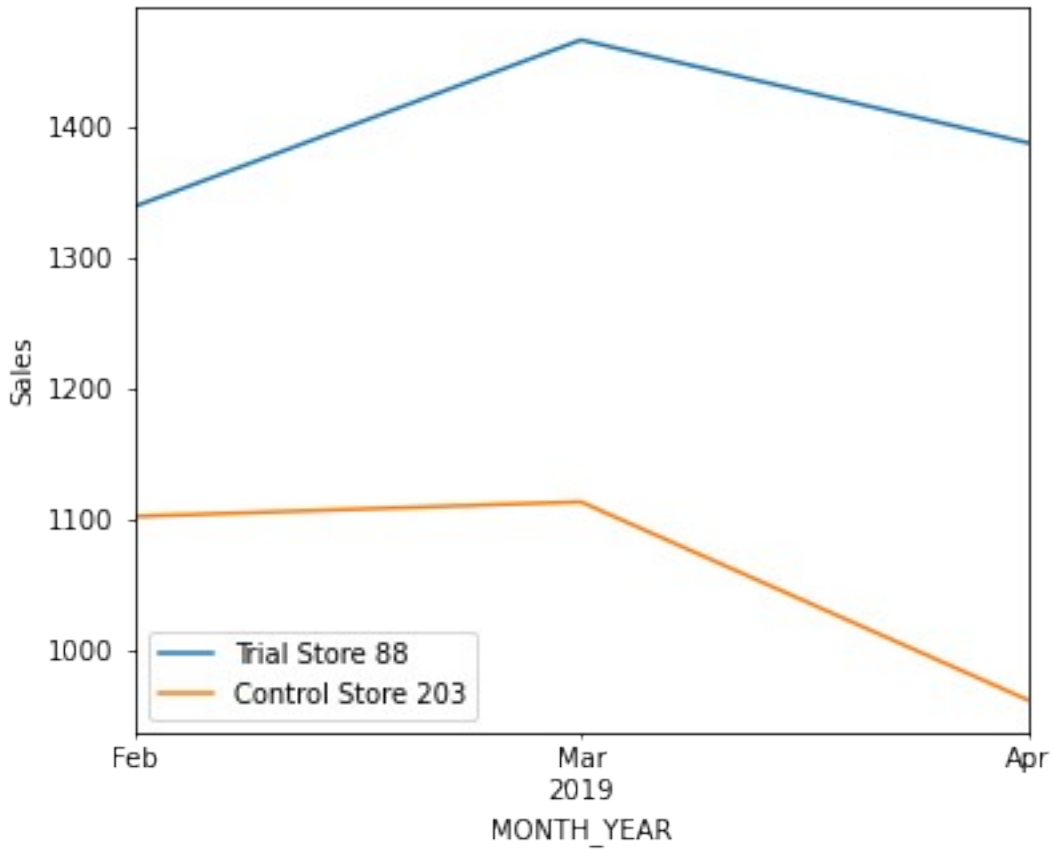
```
LYLTY_CARD_NBR    362
dtype: int64
```

```
qvi_203[['LYLTY_CARD_NBR']].value_counts().head(90)
# 89 repeated customers
```

```
LYLTY_CARD_NBR
203210            5
203021            4
203065            4
203371            4
203067            4
                 ..
203352            2
```

```
203074            2
203251            2
203005            2
203272            1
Length: 90, dtype: int64
```

```
plt.figure(figsize = [6,5])
qvi_88.groupby('MONTH_YEAR')['TOT_SALES'].sum().plot(label = 'Trial
Store 88')
qvi_203.groupby('MONTH_YEAR')['TOT_SALES'].sum().plot(label = 'Control
Store 203')
plt.ylabel('Sales')
plt.legend()
plt.show()
```



Summary

Store 88 : -- No. of customers between feb and april = 477

-- No. of repeated customers = 139

-- TOT_SALES = 4195.0

-- PROD_QTY = 954.0

Store 203 :

-- No. of customers between feb and april = 362

-- No. of repeated customers = 89

-- TOT_SALES = 3178.8

-- PROD_QTY = 724.0

Lets see how they stack up with average transactions per customer

deriving a bar chart to check the total sales between the control and trial stores taken

```
qvi_88['LYLTY_CARD_NBR'].value_counts().mean()
```

```
1.8346153846153845
```

```
qvi_203['LYLTY_CARD_NBR'].value_counts().mean()
```

```
1.465587044534413
```

```
qvi_86['LYLTY_CARD_NBR'].value_counts().mean()
```

```
1.8790697674418604
```

```
qvi_109['LYLTY_CARD_NBR'].value_counts().mean()
```

```
1.9839572192513368
```

```
qvi_77['LYLTY_CARD_NBR'].value_counts().mean()
```

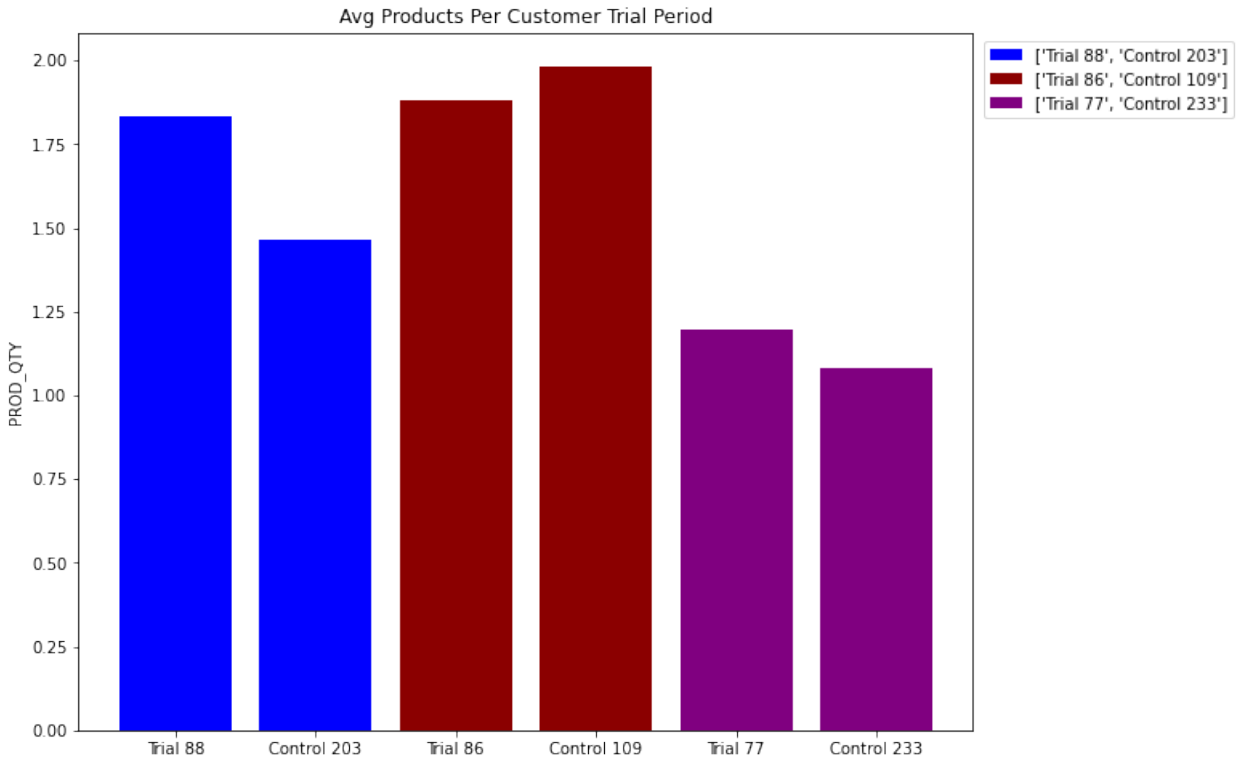
```
1.1967213114754098
```

```
qvi_233['LYLTY_CARD_NBR'].value_counts().mean()
```

```
1.0825688073394495
```

```
plt.figure(figsize = [10,8])
Group1 = ['Trial 88', 'Control 203']
Group2 = ['Trial 86', 'Control 109']
Group3 = ['Trial 77', 'Control 233']

values_group1 = [1.834, 1.465]
values_group2 = [1.879, 1.983]
values_group3 = [1.196, 1.082]

plt.bar(Group1, values_group1, label = Group1, color = 'blue')
plt.bar(Group2, values_group2, label = Group2, color = 'darkred')
plt.bar(Group3, values_group3, label = Group3, color = 'purple')
plt.ylabel('PROD_QTY')
plt.legend(loc = 'upper right', bbox_to_anchor = (1.3, 1))
```

```
plt.title('Avg Products Per Customer Trial Period')
plt.show()
```



Summary

- As we can see the avg transactions were slightly higher for two of the 3 trial stores
- I believe the new layout is working to increase sales
- comparing the trial stores to the control stores:
  - Sales, no. of products sold, no. of repeated customers, all show signs that the trial stores are outperforming the control stores in that period
  - My recommendation would be to increase the number of trial stores and run another analysis in a few months to see if the increased sales stay true & stabilize at a higher point

```
qvi_merged.to_csv('qvi_merged.csv')
```

Thank You