TASK 2
Experimentation and uplift testing

In [529…

```
#imports
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [530…

```
from plotly.offline import init_notebook_mode, iplot
init_notebook_mode(connected=True)
import plotly.offline as offline
offline.init_notebook_mode()
import cufflinks as cf
cf.go_offline()
```

In [531…

```
#reading data
data=pd.read_csv("QVI_data.csv");
data.head(2)
```

Out[531…

	LYLTY_CARD_NBR	DATE	STORE_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY
0	1000	2018-10-17	1	1	5	Natural Chip Compny SeaSalt175g	2
1	1002	2018-09-16	1	2	58	Red Rock Deli Chikn&Garlic Aioli 150g	1

In [532…

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

In [533…

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 264834 entries, 0 to 264833
Data columns (total 12 columns):
LYLTY_CARD_NBR      264834 non-null int64
DATE                264834 non-null datetime64[ns]
STORE_NBR           264834 non-null int64
TXN_ID              264834 non-null int64
PROD_NBR            264834 non-null int64
PROD_NAME           264834 non-null object
PROD_QTY            264834 non-null int64
TOT_SALES           264834 non-null float64
PACK_SIZE           264834 non-null int64
BRAND               264834 non-null object
LIFESTAGE           264834 non-null object
PREMIUM_CUSTOMER    264834 non-null object
dtypes: datetime64[ns](1), float64(1), int64(6), object(4)
memory usage: 24.2+ MB
```

In [534…

```
data['YEARMONTH']=[s.year*100+s.month for s in data['DATE']]
```

In [535…

```
data
```

Out[535…

	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 × 13 columns

METRICS UNDER CONSIDERATION:

- Monthly overall sales revenue

- Monthly number of customers

- Monthly number of transactions per customer

In [536…

```
metrics=data.groupby(['STORE_NBR','YEARMONTH']).agg({'TOT_SALES':'sum','LYLTY_C
metrics['PRICE_PER_UNIT']=metrics['TOT_SALES']/metrics['PROD_QTY']
metrics['CHIP_PER_TXN']=metrics['PROD_QTY']/metrics['TXN_ID']
metrics=metrics.rename(columns={'LYLTY_CARD_NBR':'CUSTOMERS'})
metrics['TXN_PER_CUST']=metrics['TXN_ID']/metrics['CUSTOMERS']
metrics.drop(['TXN_ID'],axis=1,inplace=True)
```

In [537…

```
full=metrics.copy()
```

In [538…

```
#taking data before 2019-02 into consideration
trial=[]
for i in metrics.index:
    if(i[1]>=201902):
        if(i[1]<=201904):
            trial.append(metrics.loc[i])
        metrics.drop(i,inplace=True)
trial=pd.DataFrame(trial)
```

In [539…

```
#taking data after 2019-02 into trial dataframe
trial.index.name=('IDX')
k=0
trial['STORE_NBR']=0
trial['MONTHYEAR']=0
for (i,j) in trial.reset_index()['IDX']:
    trial['STORE_NBR'].iloc[k]=i
    trial['MONTHYEAR'][k]=j
    k=k+1
trial=trial.set_index(['STORE_NBR','MONTHYEAR'])
```

In [540…

```
metrics
```

Out[540…

		TOT_SALES	CUSTOMERS	PROD_QTY	PRICE_PER_UNIT	CHII
STORE_NBR	**YEARMONTH**					
1	201807	206.9	49	62	3.337097	
	201808	176.1	42	54	3.261111	
	201809	278.8	59	75	3.717333	
	201810	188.1	44	58	3.243103	
	201811	192.6	46	57	3.378947	
...	...	...	...	...	...	
272	201809	304.7	32	71	4.291549	
	201810	430.6	44	99	4.349495	
	201811	376.2	41	87	4.324138	
	201812	403.9	47	89	4.538202	
	201901	423.0	46	96	4.406250	

1848 rows × 6 columns

Funtions to find correlation and magnitude of any store wih another store

In [541…

```
def calcCorr(store):
    '''
    input=store number which is to be compared
    output=dataframe with corelation coefficient values
    '''
    a=[]
    metrix=metrics[['TOT_SALES','CUSTOMERS']]#add metrics as required e.g. ,'TX
    for i in metrix.index:
        a.append(metrix.loc[store].corrwith(metrix.loc[i[0]]))
    df= pd.DataFrame(a)
    df.index=metrix.index
    df=df.drop_duplicates()
    df.index=[s[0] for s in df.index]
    df.index.name="STORE_NBR"
    return df
```

In [542…

```
def standardizer(df):
    '''
    input=dataframe with metrics
    output=dataframe with mean of the metrics in a new column
    '''
    df=df.abs()
    df['MAGNITUDE']=df.mean(axis=1)
    return df
```

Store 77

Finding stores corelated to store 77

In [543…

```
corr77=calcCorr(77)
```

In [544…

```
corr77.head(3)
```

Out[544…

STORE_NBR	TOT_SALES	CUSTOMERS
1	0.075218	0.322168
2	-0.263079	-0.572051
3	0.806644	0.834207

In [545…

```
corr77=standardizer(corr77)
corr77
```

Out[545…

STORE_NBR	TOT_SALES	CUSTOMERS	MAGNITUDE
1	0.075218	0.322168	0.198693

2	0.263079	0.572051	0.417565
3	0.806644	0.834207	0.820426
4	0.263300	0.295639	0.279469
5	0.110652	0.370659	0.240655
...	...	...	...
268	0.344757	0.369517	0.357137
269	0.315730	0.474293	0.395011
270	0.315430	0.131259	0.223345
271	0.355487	0.019629	0.187558
272	0.117622	0.223217	0.170420

266 rows × 3 columns

In [546…

```
corr77=corr77.sort_values(['MAGNITUDE'],ascending=False).dropna()
```

In [547…

```
corr77
```

Out[547…

	TOT_SALES	CUSTOMERS	MAGNITUDE
STORE_NBR			
77	1.000000	1.000000	1.000000
233	0.903774	0.990358	0.947066
119	0.867664	0.983267	0.925466
71	0.914106	0.754817	0.834461
3	0.806644	0.834207	0.820426
...	...	...	...
256	0.014245	0.047863	0.031054
159	0.001655	0.054404	0.028030
260	0.016618	0.027446	0.022032
194	0.010182	0.032053	0.021117
166	0.005875	0.012896	0.009386

263 rows × 3 columns

**shows that stores 233,119,71 are the most correlated to store 77

Selecting 233 as control store as it has max correlation

Visualizing ...

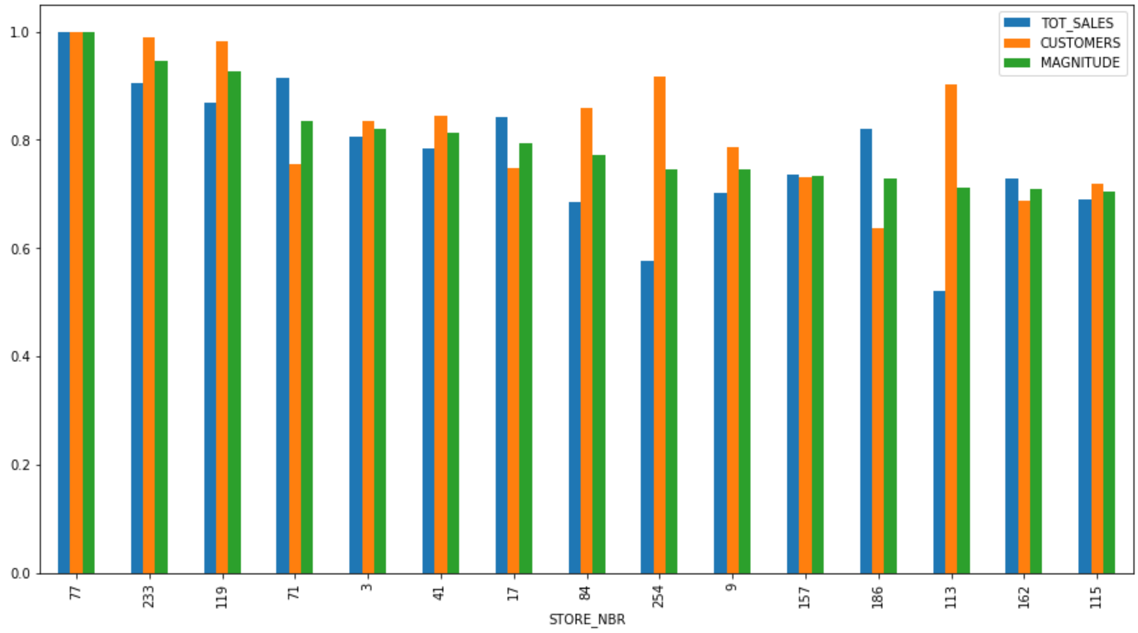
In [173…

In [173…

```
#Taking 0.7 as threshold corelation
corr77[(corr77.MAGNITUDE.abs()>0.7)].plot(kind='bar',figsize=(15,8))
```

Out[173…

<AxesSubplot:xlabel='STORE_NBR'>

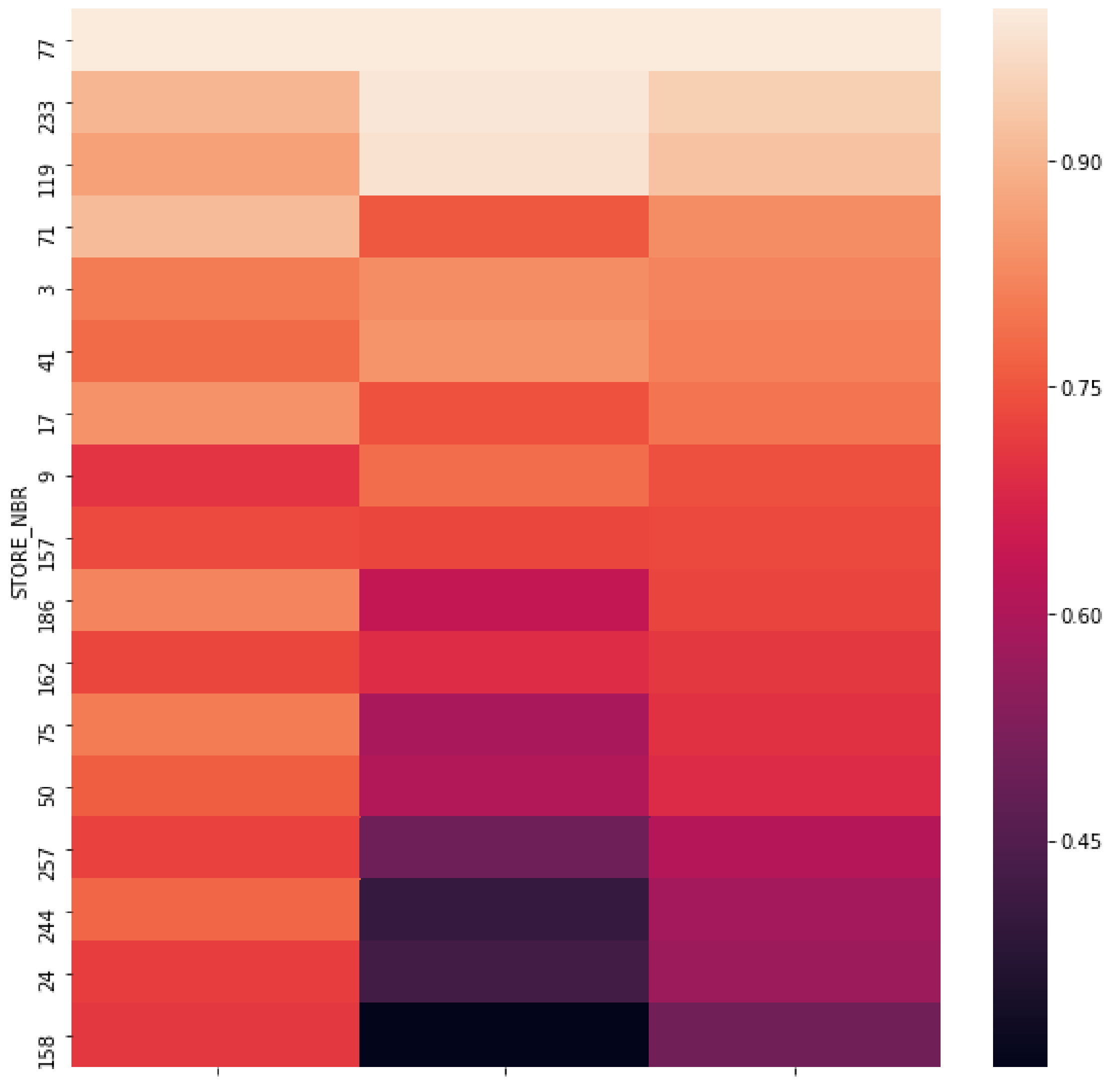


In [174…

```
plt.figure(figsize=(10,10))
sns.heatmap(corr77[corr77.TOT_SALES.abs()>0.7])
```

Out[174…

<AxesSubplot:ylabel='STORE_NBR'>



TOT_SALES	CUSTOMERS	MAGNITUDE

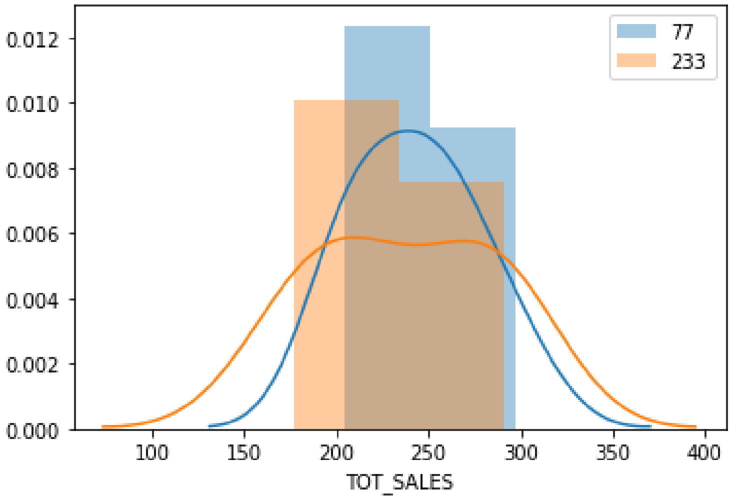
Taking the store 233 into consideration plotting different measure against those of store 77

In [175…

```
sns.distplot(metrics.loc[77]['TOT_SALES'])
sns.distplot(metrics.loc[233]['TOT_SALES'])
plt.legend(labels=['77','233'])
```

Out[175…

```
<matplotlib.legend.Legend at 0x27644cae908>
```

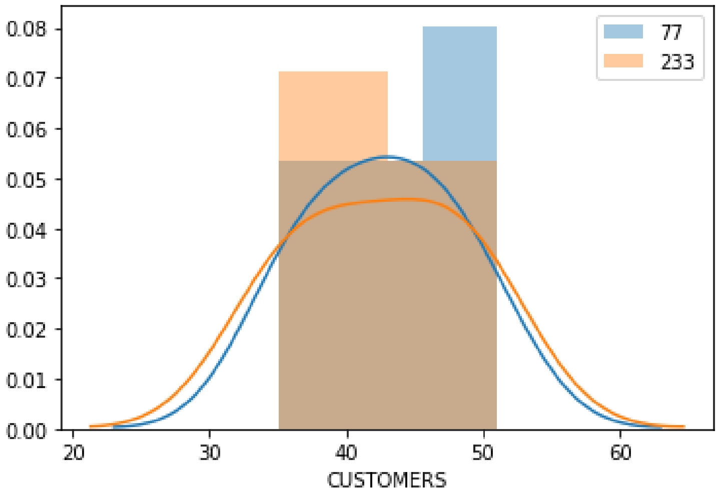


In [176…

```
sns.distplot(metrics.loc[77]['CUSTOMERS'])
sns.distplot(metrics.loc[233]['CUSTOMERS'])
plt.legend(labels=['77','233'])
```

Out[176…

```
<matplotlib.legend.Legend at 0x27645037bc8>
```

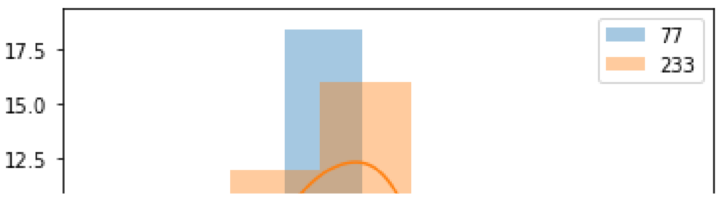


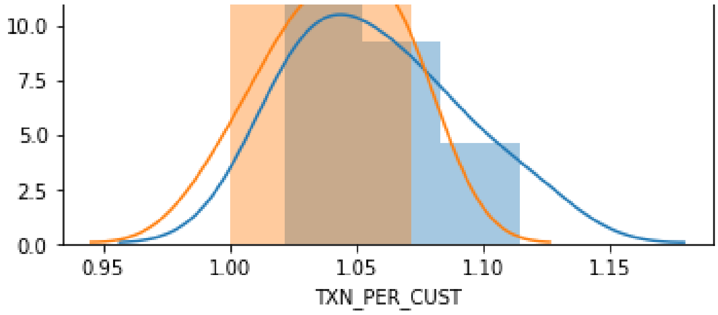
In [177…

```
sns.distplot(metrics.loc[77]['TXN_PER_CUST'])
sns.distplot(metrics.loc[233]['TXN_PER_CUST'])
plt.legend(labels=['77','233'])
```

Out[177…

```
<matplotlib.legend.Legend at 0x27644ea9308>
```





Since distributions of store 233 are similar to that of store 77, selecting store 233 as control store with max similarities to store 77

Calculating difference between scaled control sales and trial sales

Let null hypothesis be that both stores 77 ans 233 have no difference

In [189…

```
from scipy.stats import ks_2samp,ttest_ind,t
```

In [548…

```
# difference between control and trial sales
a=[]
for x in metrics.columns:
    a.append(ks_2samp(metrics.loc[77][x], metrics.loc[233][x]))
a=pd.DataFrame(a,index=metrics.columns)
```

In [549…

```
a
```

Out[549…

	statistic	pvalue
TOT_SALES	0.285714	0.962704
CUSTOMERS	0.142857	1.000000
PROD_QTY	0.285714	0.962704
PRICE_PER_UNIT	0.285714	0.962704
CHIP_PER_TXN	0.285714	0.962704
TXN_PER_CUST	0.428571	0.575175

For pre trial period, since all of the p-values are high (say more than 0.05), we can't reject the null hypothesis

Assessment of trial

The trial period goes from the start of February 2019 to April 2019. We now want to see if there has been an uplift in overall chip sales.

Sampling march and april from the 3 months

In [550

In [550...

```
b=[]
for x in trial.columns:
    b.append(ttest_ind(trial.loc[77][x].tail(2), trial.loc[233][x].tail(2)))
b=pd.DataFrame(b,index=metrics.columns)
```

In [551...

```
b
```

Out[551...

	statistic	pvalue
TOT_SALES	4.267336	0.050769
CUSTOMERS	2.586131	0.122618
PROD_QTY	4.043680	0.056063
PRICE_PER_UNIT	-0.634173	0.590828
CHIP_PER_TXN	1.785126	0.216165
TXN_PER_CUST	0.332434	0.771171

In [552...

```
#critical value
t.ppf(0.95,df=7)
```

Out[552...

```
1.894578605061305
```

Since all of the p-values are high (say more than 0.05), we reject the null hypothesis i.e. there means are significantly different.
We can observe that the t-value is much larger than the 95th percentile value of the t-distribution for March and April - i.e. the increase in sales in the trial store in March and April is statistically greater than in the control store.

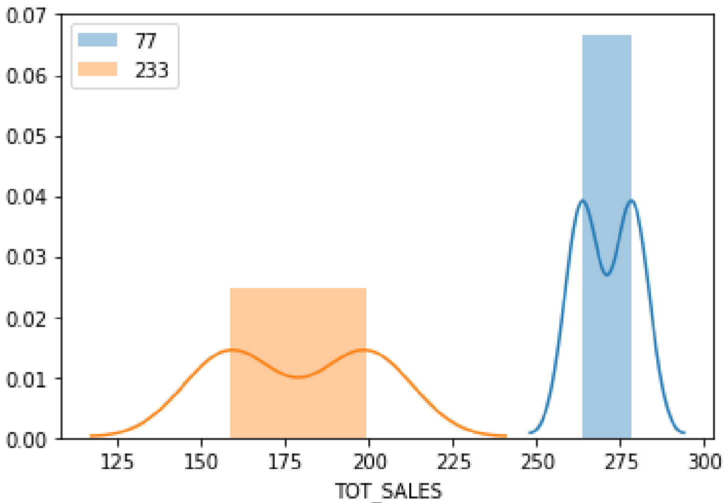
Vizualizing means

In [553...

```
sns.distplot(trial.loc[77]['TOT_SALES'].tail(2))
sns.distplot(trial.loc[233]['TOT_SALES'].tail(2))
plt.legend(labels=['77','233'])
```

Out[553...

```
<matplotlib.legend.Legend at 0x27660b0e408>
```



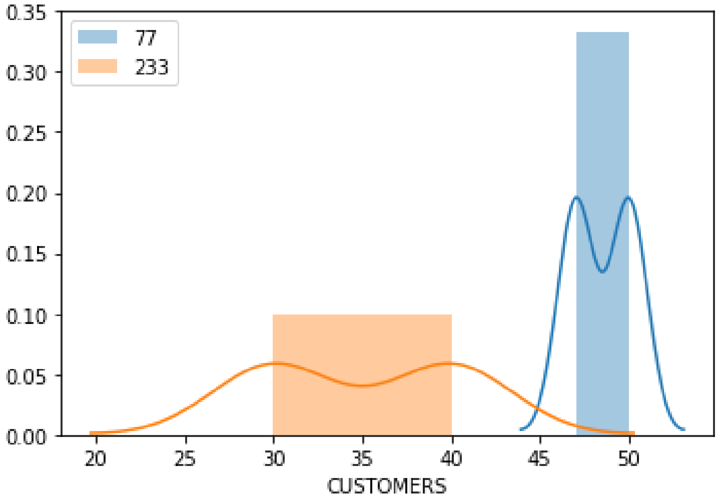
In [554...

```
sns.distplot(trial.loc[77]['CUSTOMERS'].tail(2))
sns.distplot(trial.loc[233]['CUSTOMERS'].tail(2))
```

```
plt.legend(labels=['77','233'])
```

Out[554…

```
<matplotlib.legend.Legend at 0x276602204c8>
```



It can be visualized that the is a significant difference in the means, so trial store behavior(77) is different from control store (233).

The results show that the trial in store 77 is significantly different to its control store in the trial period as the trial store performance lies outside the 5% to 95% confidence interval of the control store in two of the three trial months.

Store 86

Repeating same process for trial store 86

In [555…

```
corr86=calcCorr(86)
```

In [556…

```
corr86.head(3)
```

Out[556…

	TOT_SALES	CUSTOMERS
STORE_NBR		
1	0.445632	0.485831
2	-0.403835	-0.086161
3	-0.261284	-0.353786

In [557…

```
corr86=standardizer(corr86)
corr86
```

Out[557…

	TOT_SALES	CUSTOMERS	MAGNITUDE
STORE_NBR			
1	0.445632	0.485831	0.465731
2	0.403835	0.086161	0.244998

3	0.261284	0.353786	0.307535
4	0.039035	0.169608	0.104322
5	0.235159	0.253229	0.244194
...	...	...	...
268	0.452182	0.034273	0.243228
269	0.697055	0.098587	0.397821
270	0.730679	0.767267	0.748973
271	0.527637	0.267393	0.397515
272	0.004926	0.353815	0.179371

266 rows × 3 columns

In [335…

```
corr86=corr86.sort_values(['MAGNITUDE'],ascending=False).dropna()
```

In [336…

```
corr86
```

Out[336…

	TOT_SALES	CUSTOMERS	MAGNITUDE
STORE_NBR			
86	1.000000	1.000000	1.000000
155	0.877882	0.942876	0.910379
23	0.784698	0.943559	0.864128
120	0.872693	0.815097	0.843895
114	0.734415	0.855339	0.794877
...	...	...	...
91	0.019027	0.041271	0.030149
17	0.029793	0.030039	0.029916
131	0.028487	0.031142	0.029815
219	0.046653	0.004999	0.025826
234	0.010509	0.040306	0.025407

263 rows × 3 columns

**shows that stores 155,23,120 are the most correlated to store 86

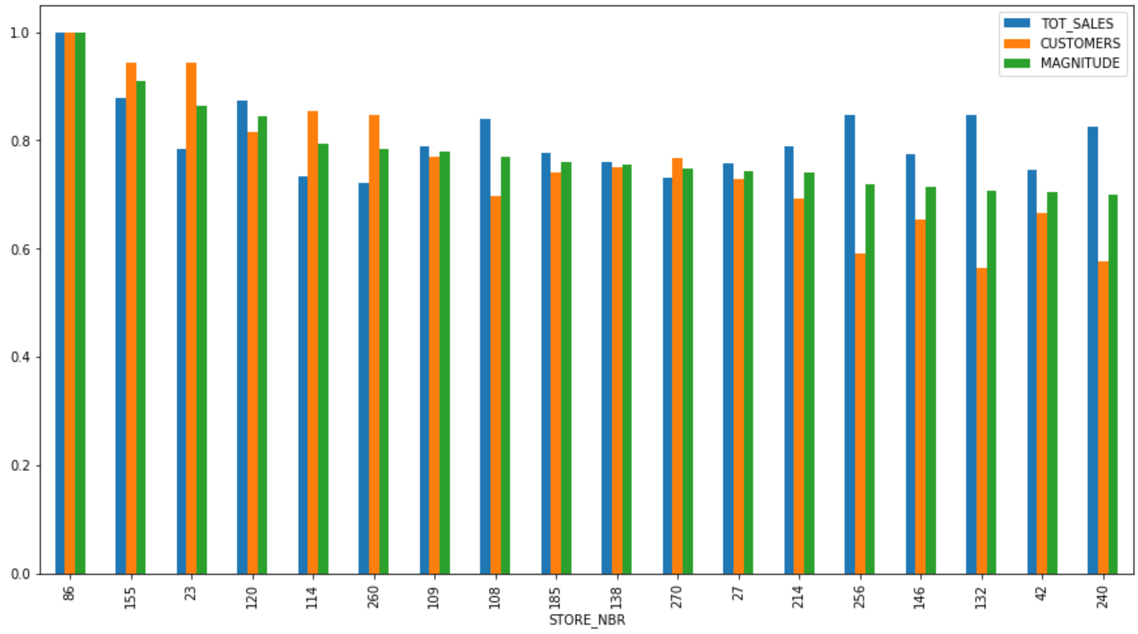
Selecting 155 as control store as it has max correlation

Visualizing ...

In [337…

```
#Taking 0.7 as threshold corelation
corr86[(corr86.MAGNITUDE.abs()>0.7)].plot(kind='bar',figsize=(15,8))
```

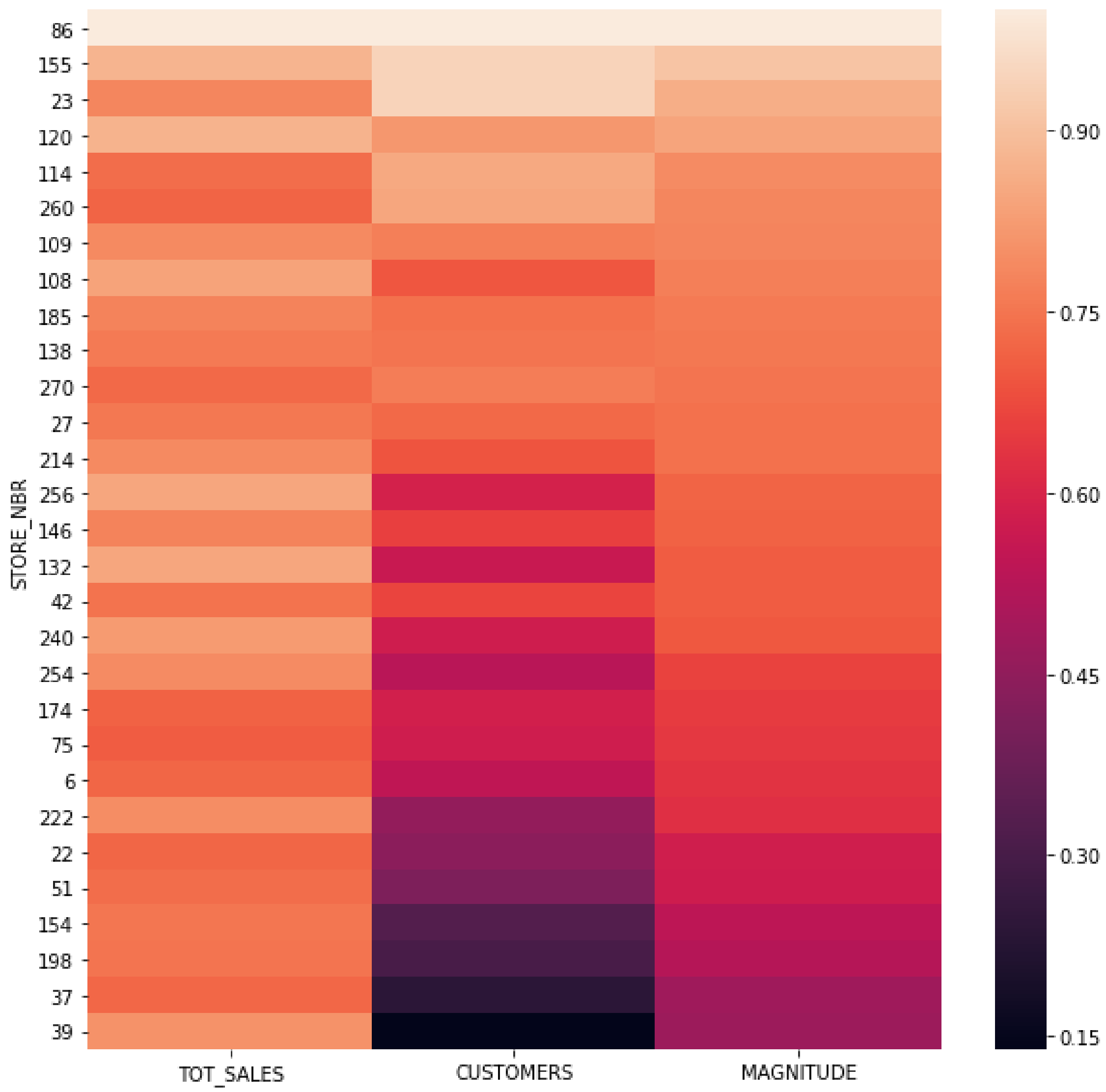
Out[337… <AxesSubplot:xlabel='STORE_NBR'>



In [338…

```
plt.figure(figsize=(10,10))
sns.heatmap(corr86[corr86.TOT_SALES.abs()>0.7])
```

Out[338… <AxesSubplot:ylabel='STORE_NBR'>



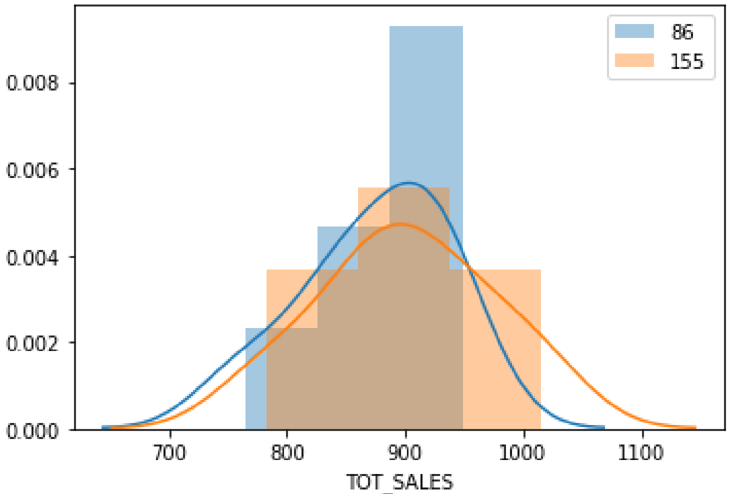
Taking the store 155 into consideration plotting different measure against those of store 86

In [339…

```
sns.distplot(metrics.loc[86]['TOT_SALES'])
sns.distplot(metrics.loc[155]['TOT_SALES'])
plt.legend(labels=['86','155'])
```

Out[339…

```
<matplotlib.legend.Legend at 0x27654e85c48>
```

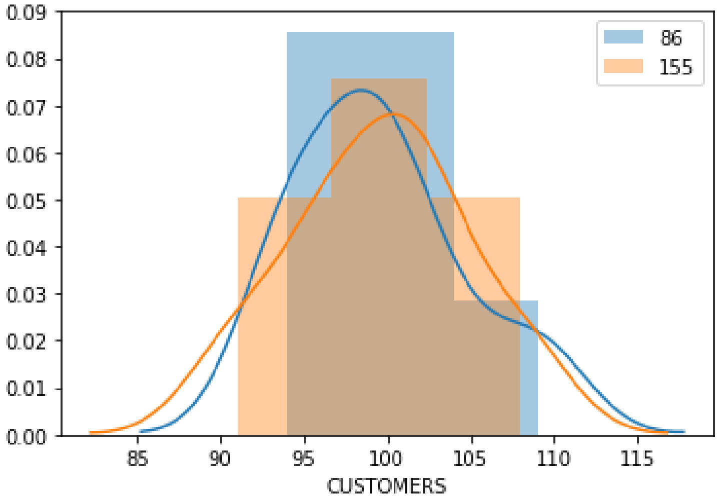


In [340…

```
sns.distplot(metrics.loc[86]['CUSTOMERS'])
sns.distplot(metrics.loc[155]['CUSTOMERS'])
plt.legend(labels=['86','155'])
```

Out[340…

```
<matplotlib.legend.Legend at 0x276569f5b48>
```

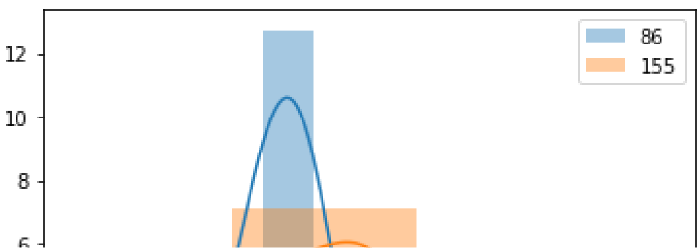


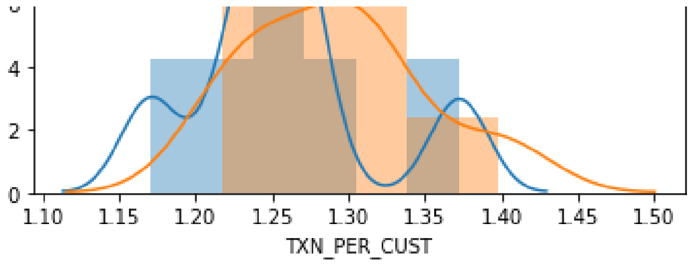
In [341…

```
sns.distplot(metrics.loc[86]['TXN_PER_CUST'])
sns.distplot(metrics.loc[155]['TXN_PER_CUST'])
plt.legend(labels=['86','155'])
```

Out[341…

```
<matplotlib.legend.Legend at 0x276569f5988>
```



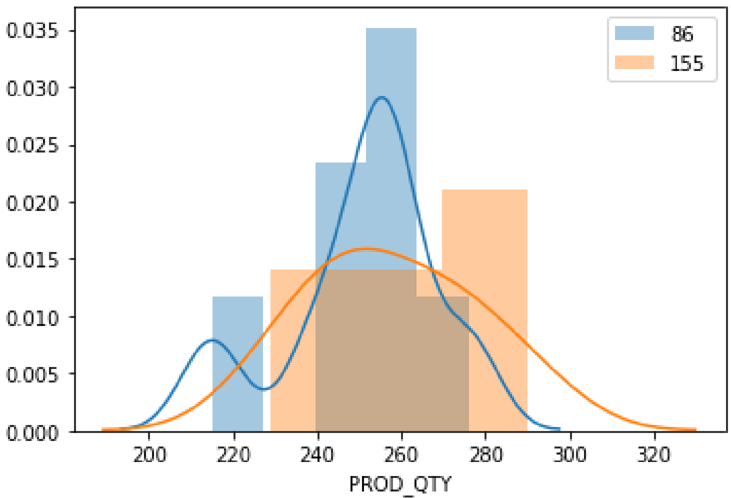


In [342...

```
sns.distplot(metrics.loc[86]['PROD_QTY'])
sns.distplot(metrics.loc[155]['PROD_QTY'])
plt.legend(labels=['86','155'])
```

Out[342...

```
<matplotlib.legend.Legend at 0x27651d82f08>
```



Since distributions of store 155 are similar to that of store 86, selecting store 155 as control store with max similarities to store 86

Calculating difference between scaled control sales and trial sales

Let null hypothesis be that both stores 77 ans 233 have no difference

In [343...

```
from scipy.stats import ks_2samp,ttest_ind,ttest_rel,t
```

In [344...

```
# difference between control and trial sales
a=[]
for x in metrics.columns:
    a.append(ks_2samp(metrics.loc[86][x], metrics.loc[155][x]))
a=pd.DataFrame(a,index=metrics.columns)
```

In [345...

```
a
```

Out[345...

	statistic	pvalue
TOT_SALES	0.285714	0.962704
CUSTOMERS	0.285714	0.962704
PROD_QTY	0.285714	0.962704

PROD_QTY	0.285714	0.962704
PRICE_PER_UNIT	0.428571	0.575175
CHIP_PER_TXN	0.428571	0.575175
TXN_PER_CUST	0.428571	0.575175

For pre trial period, since p-values for TOT_SALES, CUSTOMERS and PROD_QTY are high (say more than 0.95), we can't reject the null hypothesis

Assessment of trial

The trial period goes from the start of February 2019 to April 2019. We now want to see if there has been an uplift in overall chip sales.

In [353...

```
b=[]
for x in trial.columns:
    b.append(ttest_ind(trial.loc[86][x].tail(2), trial.loc[155][x].tail(2)))
b=pd.DataFrame(b,index=metrics.columns)
```

In [354...

```
b
```

Out[354...

	statistic	pvalue
TOT_SALES	1.234512	0.342378
CUSTOMERS	2.414953	0.137076
PROD_QTY	1.862532	0.203568
PRICE_PER_UNIT	0.366214	0.749316
CHIP_PER_TXN	-0.285938	0.801822
TXN_PER_CUST	-1.074767	0.394929

In [355...

```
#critical value
t.ppf(0.95,df=7)
```

Out[355...

```
1.894578605061305
```

Since all of the p-values are high (say more than 0.05), we reject the null hypothesis i.e. there means are significantly different.
We can observe that the t-value is much larger than the 95th percentile value of the t-distribution for March and April - i.e. the increase in sales in the trial store in March and April is statistically greater than in the control store.

The results show that the trial in store 88 is significantly different to its control store in the trial period as the trial store performance lies outside of the 5% to 95% confidence interval of the control store in two of the three trial months.

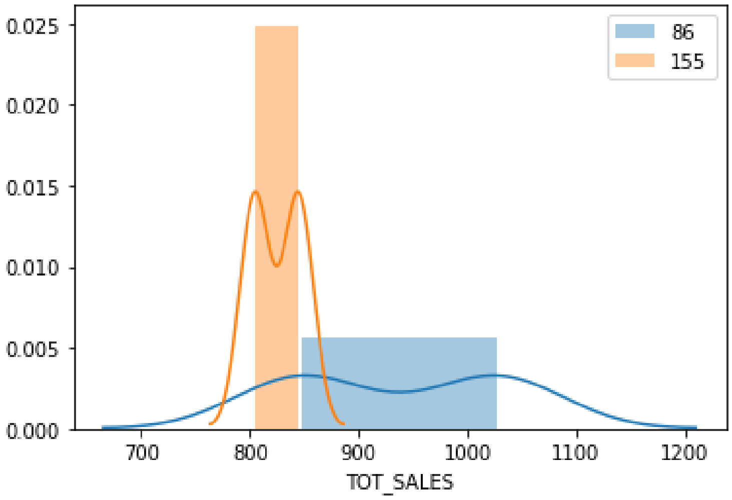
Vizualizing means

In [357...

```
sns.distplot(trial.loc[86]['TOT_SALES'].tail(2))
sns.distplot(trial.loc[155]['TOT_SALES'].tail(2))
plt.legend(labels=['86','155'])
```

Out[357…

```
<matplotlib.legend.Legend at 0x27651e6f608>
```

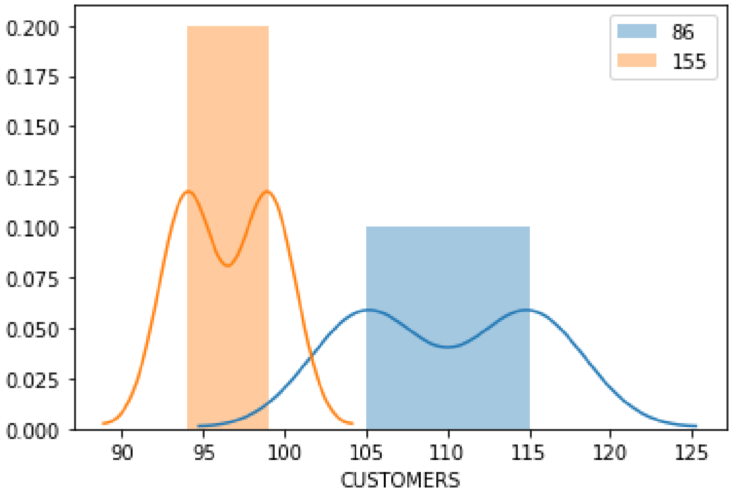


In [358…

```
sns.distplot(trial.loc[86]['CUSTOMERS'].tail(2))
sns.distplot(trial.loc[155]['CUSTOMERS'].tail(2))
plt.legend(labels=['86','155'])
```

Out[358…

```
<matplotlib.legend.Legend at 0x2765323f388>
```



It can be visualized that the is a significant difference in the means, so trial store behavior(86) is different from control store (155).

It looks like the number of customers is significantly higher in all of the three months. This seems to suggest that the trial had a significant impact on increasingthe number of customers in trial store 86 but as we saw, sales were not significantly higher. We should check with the Category Manager if there were special deals in the trial store that were may have resulted in lower prices, impacting the results.

Store 88

Finding stores corelated to store 88

In [359…

```
corr88=calcCorr(88)
```

In [360…

```
corr88.head(3)
```

Out[360…

STORE_NBR	TOT_SALES	CUSTOMERS
1	0.813636	0.305334
2	-0.067927	-0.452379
3	-0.507847	0.522884

In [361…

```
corr88=standardizer(corr88)
corr88
```

Out[361…

STORE_NBR	TOT_SALES	CUSTOMERS	MAGNITUDE
1	0.813636	0.305334	0.559485
2	0.067927	0.452379	0.260153
3	0.507847	0.522884	0.515365
4	0.745566	0.361503	0.553534
5	0.190330	0.025320	0.107825
...	...	...	...
268	0.021429	0.672672	0.347050
269	0.172578	0.274781	0.223679
270	0.723272	0.103032	0.413152
271	0.103037	0.018831	0.060934
272	0.772772	0.026909	0.399841

266 rows × 3 columns

In [385…

```
corr88=corr88.sort_values(['MAGNITUDE'],ascending=False).dropna()
```

In [388…

```
corr88.head(15)
```

Out[388…

STORE_NBR	TOT_SALES	CUSTOMERS	MAGNITUDE
88	1.000000	1.000000	1.000000
178	0.731857	0.939466	0.835661
14	0.698557	0.942976	0.820767
133	0.735407	0.835426	0.785417

204	0.885774	0.550263	0.718018
134	0.864293	0.508880	0.686587
113	0.495763	0.862632	0.679198
253	0.811838	0.500962	0.656400
239	0.642329	0.660672	0.651501
230	0.908883	0.373350	0.641117
187	0.599076	0.671264	0.635170
227	0.537448	0.729943	0.633695
69	0.450029	0.815792	0.632910
237	0.308479	0.947326	0.627903
141	0.690590	0.547399	0.618994

**shows that stores 178,14,133 are the most correlated to store 88

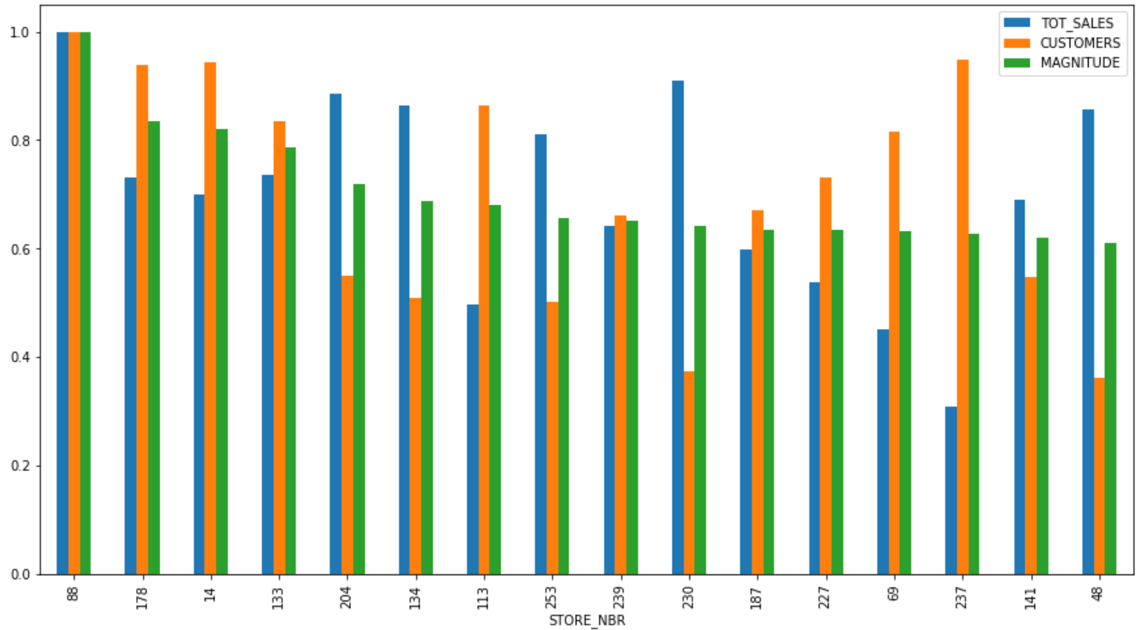
Visualizing ...

In [389...

```
#Taking 0.6 as threshold corelation
corr88[(corr88.MAGNITUDE.abs()>0.6)].plot(kind='bar',figsize=(15,8))
```

Out[389...

```
<AxesSubplot:xlabel='STORE_NBR'>
```



In [391...

```
plt.figure(figsize=(10,10))
sns.heatmap(corr88[corr88.MAGNITUDE.abs()>0.6])
```

Out[391...

```
<AxesSubplot:ylabel='STORE_NBR'>
```