```
and will raise in a future version of pandas. Select only the columns
that can be aggregated.

tab=dfn.pivot_table(index='PROD_QTY',columns=['STORE_NBR'],values=['TX
N_ID','TOT_SALES'],aggfunc='mean')

          TOT_SALES
\
STORE_NBR       1          2         3          4          5
6
PROD_QTY

1.0        3.479866   3.520575  4.435841   4.161111   4.143750
3.437037
2.0        6.616260   7.417308  8.857941   8.842506   7.008472
7.111304
3.0        6.300000  13.800000  9.150000  11.775000  13.200000
7.500000
4.0        9.000000        NaN  8.200000        NaN  10.600000
NaN
5.0             NaN  15.000000  8.566667        NaN  15.000000
13.500000

                                                      ...
\
STORE_NBR        7         8          9          10   ...   263
264
PROD_QTY                                              ...

1.0         4.438667  4.583333   3.522642   3.480645  ...  4.80
4.706562
2.0         8.852413  8.802834   6.852419   7.064006  ...  5.15
4.317901
3.0        12.300000       NaN  13.200000  12.800000  ...   NaN
NaN
4.0        15.200000       NaN   7.600000  11.866667  ...   NaN
2.700000
5.0              NaN       NaN        NaN  16.500000  ...   NaN
2.950000


\
STORE_NBR       265       266       267       268       269       270

PROD_QTY

1.0        4.569553  3.667793  4.274576  4.650681  6.547500  7.000000

2.0        4.579762  3.475000  3.314286  4.813081  6.957198  6.898953
```

```
3.0        3.000000       NaN       NaN  4.350000  9.200000  8.125000

4.0        3.750000  3.000000       NaN  5.100000  6.000000  7.400000

5.0             NaN       NaN       NaN  5.700000  7.640000  7.900000
```

```
STORE_NBR       271       272
PROD_QTY
1.0        6.760000  7.869231
2.0        7.023626  8.289617
3.0        8.266667       NaN
4.0        6.600000  9.200000
5.0             NaN       NaN

[5 rows x 272 columns]
```

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
```

```
dfn['YEARMONTH']=dfn['DATE'].dt.year*100+dfn['DATE'].dt.month
```

```
<ipython-input-46-702d0116f1ab>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  dfn['YEARMONTH']=dfn['DATE'].dt.year*100+dfn['DATE'].dt.month
```

```
t1=dfn.groupby(['STORE_NBR','YEARMONTH'])
['TOT_SALES'].sum().reset_index() #Total sales revenue
t2=dfn.groupby(['STORE_NBR','YEARMONTH'])
['LYLTY_CARD_NBR'].nunique().reset_index()#total n customers
t3=(dfn.groupby(['STORE_NBR','YEARMONTH'])['LYLTY_CARD_NBR'].sum()/
dfn.groupby(['STORE_NBR','YEARMONTH'])
['TXN_ID'].nunique()).reset_index()# avg trans/customer
t4=(dfn.groupby(['STORE_NBR','YEARMONTH'])['PROD_QTY'].sum()/dfn.group
by(['STORE_NBR','YEARMONTH'])['TXN_ID'].nunique()).reset_index()
#nchips/txn
t5=(dfn.groupby(['STORE_NBR','YEARMONTH'])['TOT_SALES'].sum()/dfn.grou
```

```
pby(['STORE_NBR','YEARMONTH'])['PROD_QTY'].sum()).reset_index()# avg
price per unit

m= pd.merge(t1, t2, on=['STORE_NBR', 'YEARMONTH'])
m = pd.merge(m, t3, on=['STORE_NBR', 'YEARMONTH'])
m = pd.merge(m, t4, on=['STORE_NBR', 'YEARMONTH'])
m = pd.merge(m, t5, on=['STORE_NBR', 'YEARMONTH'])
m.columns=['STORE_NBR','YEARMONTH','TOT_SALES','LYLTY_CARD_NBR','AVGTX
NCUS','NCHIPERSTXN','AVGPRICEUNIT']
m
```

	STORE_NBR	YEARMONTH	TOT_SALES	LYLTY_CARD_NBR	AVGTXNCUS
0	1	201807	206.9	49	1267.365385
1	1	201808	176.1	42	1267.953488
2	1	201809	278.8	59	1235.403226
3	1	201810	188.1	44	1264.666667
4	1	201811	192.6	46	1278.787234
...	...	...	...	...	...
3164	272	201902	395.5	45	272189.666667
3165	272	201903	442.3	50	272189.264151
3166	272	201904	445.1	54	277136.581818
3167	272	201905	314.6	34	272222.175
3168	272	201906	312.1	34	272182.135135

	NCHIPERSTXN	AVGPRICEUNIT
0	1.192308	3.337097
1	1.255814	3.261111
2	1.209677	3.717333
3	1.288889	3.243103
4	1.212766	3.378947
...	...	...
3164	1.895833	4.346154
3165	1.924528	4.336275
3166	1.963636	4.121296
3167	1.925000	4.085714
3168	1.810811	4.658209

```
[3169 rows x 7 columns]
```

```
fulobstr=m.groupby('STORE_NBR')['YEARMONTH'].size().reset_index()
fulobstr=fulobstr[fulobstr['YEARMONTH']==12]['STORE_NBR'].unique()
fulobstr#unique store numbers contains records of entire 12 months
```

```
array([  1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  12,  13,
14,
        15,  16,  17,  18,  19,  20,  21,  22,  23,  24,  25,  26,
27,
        28,  29,  30,  32,  33,  34,  35,  36,  37,  38,  39,  40,
41,
        42,  43,  45,  46,  47,  48,  49,  50,  51,  52,  53,  54,
55,
        56,  57,  58,  59,  60,  61,  62,  63,  64,  65,  66,  67,
68,
        69,  70,  71,  72,  73,  74,  75,  77,  78,  79,  80,  81,
82,
        83,  84,  86,  87,  88,  89,  90,  91,  93,  94,  95,  96,
97,
        98,  99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109,
110,
       111, 112, 113, 114, 115, 116, 118, 119, 120, 121, 122, 123,
124,
       125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136,
137,
       138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149,
150,
       151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162,
163,
       164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175,
176,
       177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188,
189,
       190, 191, 192, 194, 195, 196, 197, 198, 199, 200, 201, 202,
203,
       204, 205, 207, 208, 209, 210, 212, 213, 214, 215, 216, 217,
219,
       220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231,
232,
       233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244,
245,
       246, 247, 248, 249, 250, 251, 253, 254, 255, 256, 257, 258,
259,
       260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271,
272])
```

```
#filtering the pretrial periods with the full observations stores
fpt=m[(m['YEARMONTH']<201902) & m['STORE_NBR'].isin(fulobstr)]
fpt
```

```
      STORE_NBR  YEARMONTH  TOT_SALES  LYLTY_CARD_NBR
AVGTXNCUS  \
0             1     201807      206.9              49    1267.365385

1             1     201808      176.1              42    1267.953488

2             1     201809      278.8              59    1235.403226

3             1     201810      188.1              44    1264.666667

4             1     201811      192.6              46    1278.787234

...         ...        ...        ...             ...            ...

3159        272     201809      304.7              32  272224.444444

3160        272     201810      430.6              44      277656.12

3161        272     201811      376.2              41  272217.222222

3162        272     201812      403.9              47  272209.617021

3163        272     201901      423.0              46      272203.46


      NCHIPERSTXN  AVGPRICEUNIT
0        1.192308      3.337097
1        1.255814      3.261111
2        1.209677      3.717333
3        1.288889      3.243103
4        1.212766      3.378947
...           ...           ...
3159     1.888889      4.480882
3160     1.920000      4.485417
3161     1.888889      4.425882
3162     1.914894      4.487778
3163     1.800000      4.700000

[1820 rows x 7 columns]
```

```
fpt['AVGTXNCUS'] = fpt['AVGTXNCUS'].astype(float)
fpt.dtypes
```

```
<ipython-input-125-d777299ccdba>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
```

```
returning-a-view-versus-a-copy
  fpt['AVGTXNCUS'] = fpt['AVGTXNCUS'].astype(float)
```

```
STORE_NBR           int64
YEARMONTH           int64
TOT_SALES         float64
LYLTY_CARD_NBR      int64
AVGTXNCUS         float64
NCHIPERSTXN       float64
AVGPRICEUNIT      float64
dtype: object
```

COMPARISON: CONTROL STORES vs TRIAL STORES

- Control Stores:These kind of stores are usually not fall under any criteria or conditions to follow when takend under the study or experiments based on its performances.
- Trial Stores:These stroes are ones need to staisfy the certain conditions and everything will be monitored based on its real time performances and it will receives some extra treaments when compare to control stores.

  *Control Stores : We can choose it on our own.I choose(100,160,200)

  Metric To be used: Pearson Correlation

PEARSON CORRELATION:

- Pearson's correlation coefficient is the test statistics that measures the statistical relationship, or association, between two continuous variables.
- It is known as the best method of measuring the association between variables of interest because it is based on the method of covariance.
- It gives information about the magnitude of the association, or correlation, as well as the direction of the relationship.

ASSUMPTIONS:

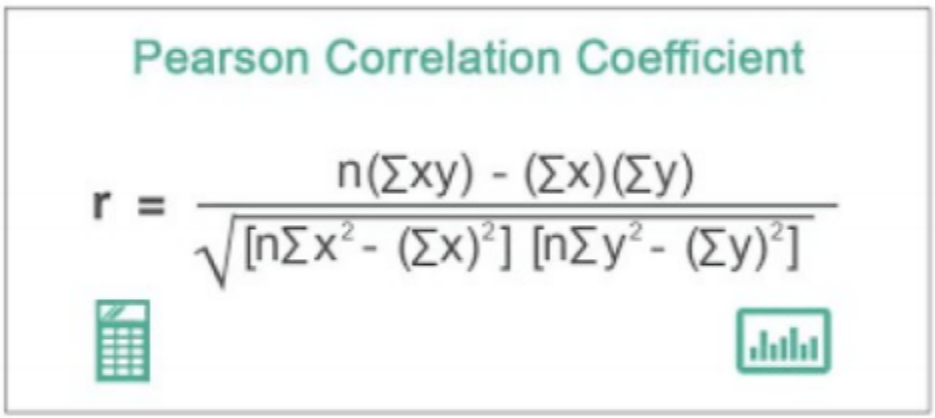
- Data pts should be independent to each other.
- Two variable which we are comparing its should be posses linear in relationship(visulalized with scatter plot).
- Residuals should satisfy the homoscedasticity(homogeneous variances)

#HYPOTHESIS STATEMENTS:

H0: There is significant relationship between the Trial store and Control stores in terms of Total Sales(ie., r=0)

H0: There is a significant relationship between the Trial store and Control stores in terms of Total Sales(ie., r≠0)

```
from PIL import Image

a= Image.open('/content/drive/MyDrive/01/1.png')
plt.imshow(a)
plt.axis('off')
plt.show()
```



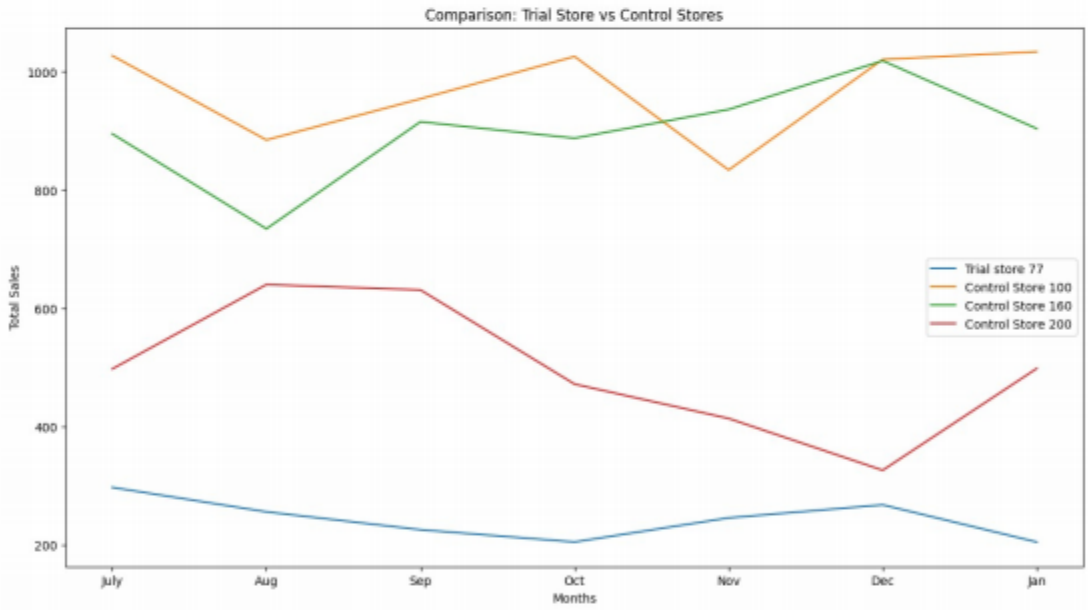

```
result=[]
def co(measure):


  trial_str=int(input('Enter the trial store number to compare:'  ))
  control_str=[]
  for i in range(3):
    c=int(input("Enter the control store number to compare : "))
    control_str.append(c)
  print(control_str)

  plt.figure(figsize=(15,8))

  # trial store part
  triald=fpt[fpt['STORE_NBR']==trial_str][measure].values
  plt.plot(range(len(triald)),triald,label=f"Trial store {trial_str}")


  #control store part
  for store in control_str:
    controld=fpt[fpt['STORE_NBR']==store][measure].values
```

```
    plt.plot(range(len(controld)),controld,label=f'Control Store
{store}')

    #correlation part
    cor,p=r(triald,controld)
    result.append(f"Trial Store: {trial_str} vs Control Store:
{store}\n Pearson Corr Coeff: {cor} and Pvalue: {p}")

  #visualization
  months=['July','Aug',"Sep","Oct","Nov","Dec","Jan"]
  plt.xticks(range(0,len(triald)),months)
  plt.xlabel('Months')
  plt.ylabel('Total Sales')
  plt.title("Comparison: Trial Store vs Control Stores")
  plt.legend()
  plt.show()
```

TOTAL SALES

```
co('TOT_SALES')
```

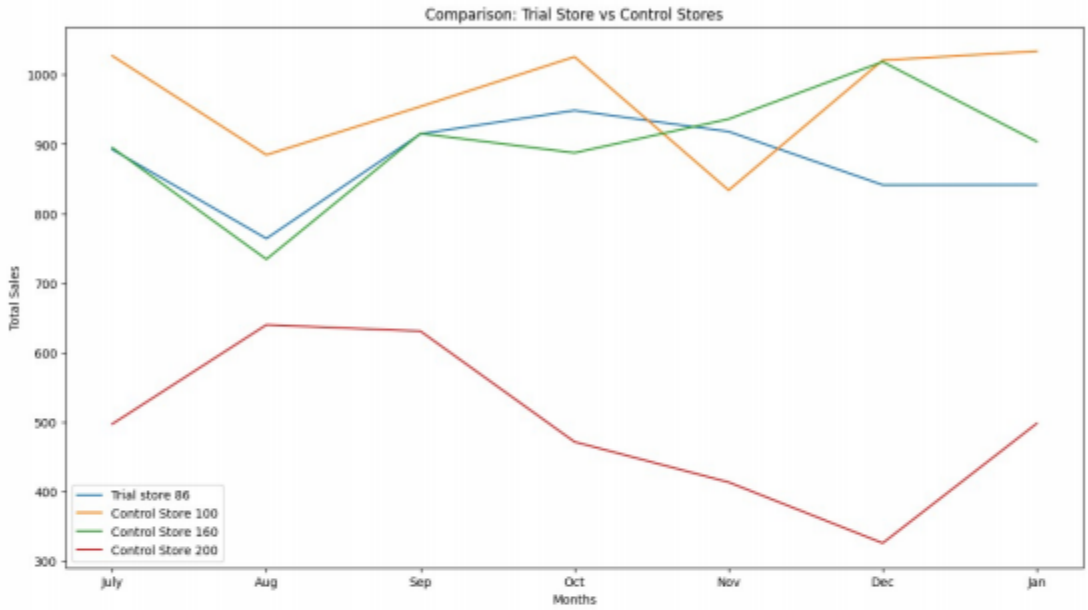
```
Enter the trial store number to compare:77
Enter the control store number to compare : 100
Enter the control store number to compare : 160
Enter the control store number to compare : 200
[100, 160, 200]
```



Note: Thus from the above visualization one can easily infer that the Control store 100 and 160 performed well when comapring with trial store 77.

```
co('TOT_SALES')
```

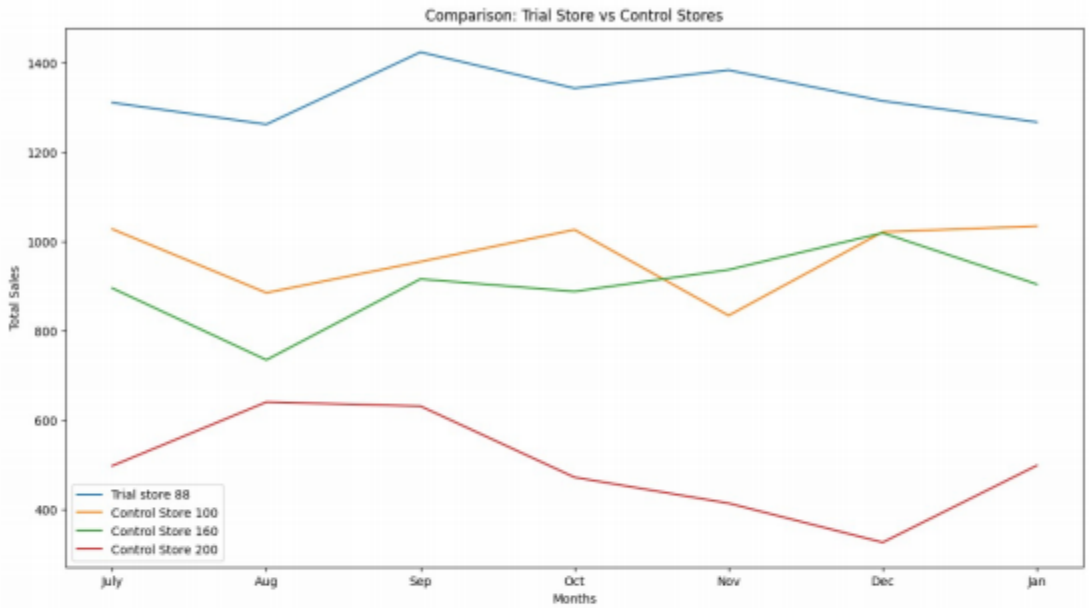
```
Enter the trial store number to compare:86
Enter the control store number to compare : 100
Enter the control store number to compare : 160
Enter the control store number to compare : 200
[100, 160, 200]
```



Note: Thus from the above visualization one can easily infer that the control store 100 and 160 performed well than trial store 86 in Total sales

```
co('TOT_SALES')
```

```
Enter the trial store number to compare:88
Enter the control store number to compare : 100
Enter the control store number to compare : 160
Enter the control store number to compare : 200
[100, 160, 200]
```



Note: Thus from the above visualization one can easily infer that the trial store 88 performed well than the Control stores 100 and 160 in Total sales.

#RESULTS

```
for i in result: print(i)
```

```
Trial Store: 77 vs Control Store: 100
 Pearson Corr Coeff: -0.08509299640777934 and Pvalue:
0.856064072466849
Trial Store: 77 vs Control Store: 160
 Pearson Corr Coeff: 0.03929376972427545 and Pvalue:
0.9333443107219689
Trial Store: 77 vs Control Store: 200
 Pearson Corr Coeff: -0.17483822339321067 and Pvalue:
0.7077011097876388
Trial Store: 86 vs Control Store: 100
 Pearson Corr Coeff: 0.13815335794044958 and Pvalue:
0.7676953748591607
Trial Store: 86 vs Control Store: 160
 Pearson Corr Coeff: 0.4814263901366128 and Pvalue:
0.27402753632610627
Trial Store: 86 vs Control Store: 200
 Pearson Corr Coeff: -0.24322192253487876 and Pvalue:
0.5991976206057374
Trial Store: 88 vs Control Store: 100
 Pearson Corr Coeff: -0.284847926108035 and Pvalue: 0.5358041609783513
Trial Store: 88 vs Control Store: 160
 Pearson Corr Coeff: 0.41428840929454946 and Pvalue:
0.35545172635588856
Trial Store: 88 vs Control Store: 200
```

```
 Pearson Corr Coeff: 0.01553886691174368 and Pvalue:
0.9736235848695028
```

RESULTS:

🛩STORE NO: 77

Since all the Control Stores(100,160,200) pvalues > 0.05, so There is no significant corelation between control store and trail store based on the Total Sales.

🛩STORE NO: 86

Since all the Control Stores(100,160,200) pvalues > 0.05, so There is no significant correlation between control store and trail store based on the Total Sales.

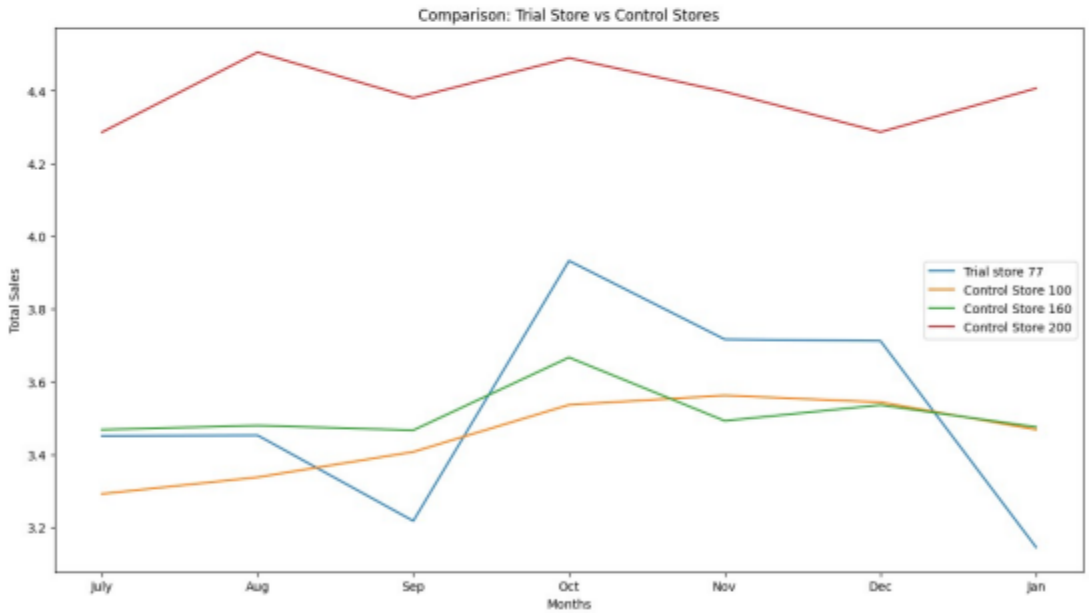
🛩STORE NO: 88

Since all the Control Stores(100,160,200) pvalues > 0.05, so There is no significant correlation between control store and trail store based on the Total Sales.

AVG PRICE PER UNIT

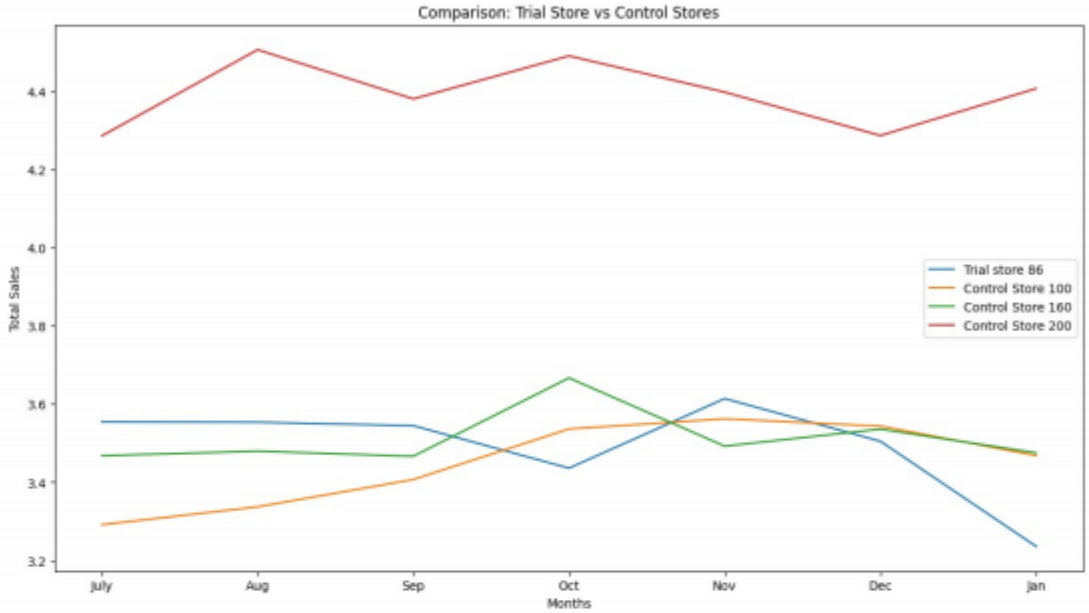
```
co('AVGPRICEUNIT')
```

```
Enter the trial store number to compare:77
Enter the control store number to compare : 100
Enter the control store number to compare : 160
Enter the control store number to compare : 200
[100, 160, 200]
```



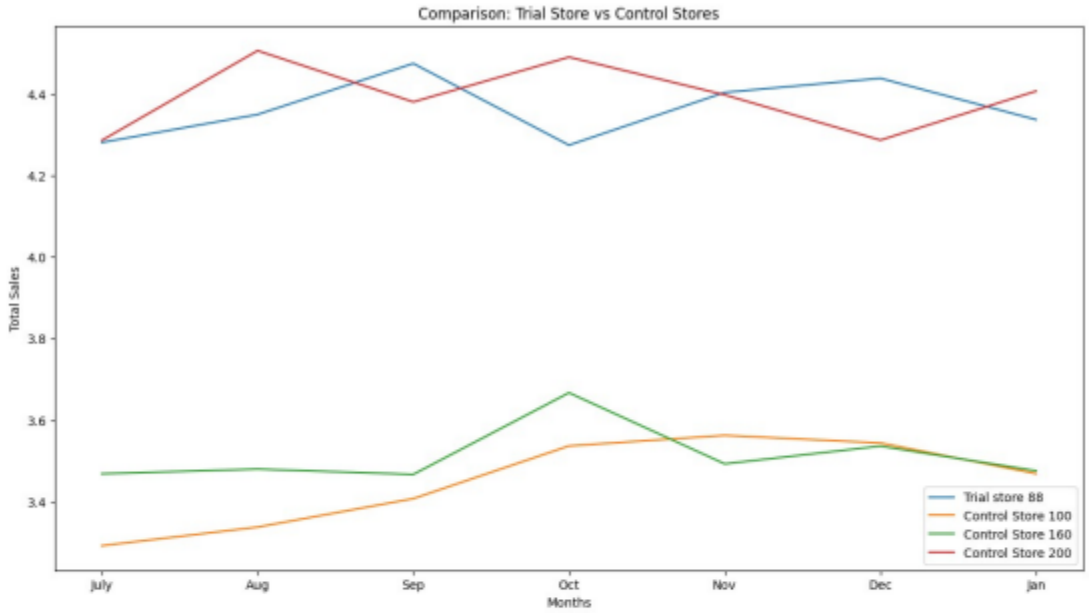

```
co('AVGPRICEUNIT')
```

```
Enter the trial store number to compare:86
Enter the control store number to compare : 100
Enter the control store number to compare : 160
Enter the control store number to compare : 200
[100, 160, 200]
```




```
co('AVGPRICEUNIT')
```

```
Enter the trial store number to compare:88
Enter the control store number to compare : 100
Enter the control store number to compare : 160
Enter the control store number to compare : 200
[100, 160, 200]
```




```
for i in result: print(i)
```

```
Trial Store: 77 vs Control Store: 100
 Pearson Corr Coeff: 0.5477703449850323 and Pvalue:
0.20306315364859653
Trial Store: 77 vs Control Store: 160
 Pearson Corr Coeff: 0.7880146574139497 and Pvalue:
0.03532061938604388
Trial Store: 77 vs Control Store: 200
 Pearson Corr Coeff: 0.1274350314290931 and Pvalue: 0.7854119344840905
Trial Store: 86 vs Control Store: 100
 Pearson Corr Coeff: -0.1704881384732178 and Pvalue: 0.714758254556404
Trial Store: 86 vs Control Store: 160
 Pearson Corr Coeff: -0.15604808593329636 and Pvalue:
0.7382982043765167
Trial Store: 86 vs Control Store: 200
 Pearson Corr Coeff: -0.15553597826057863 and Pvalue:
0.7391361268494586
Trial Store: 88 vs Control Store: 100
 Pearson Corr Coeff: 0.26848751544490634 and Pvalue:
0.5604504882624801
Trial Store: 88 vs Control Store: 160
 Pearson Corr Coeff: -0.3848144935444825 and Pvalue:
0.39399408058542834
Trial Store: 88 vs Control Store: 200
 Pearson Corr Coeff: -0.2743616736558129 and Pvalue:
0.5515596992208659
```

RESULTS:

✈STORE NO: 77

Since all the Control Stores(100 and 200) pvalues > 0.05, so There is no significant correlation between control store and trail store based on the Avg price/unit.

But Control Store: 160 pvalue(0.035) < 0.05, There is a significant Strong Positive correlation(uphill trend) found between the Control Store 160 and Trial store 77 based on Avg price/unit.

✈STORE NO: 86

Since all the Control Stores(100,160,200) pvalues > 0.05, so There is no significant relation between control store and trail store based on the Avg price/unit.

✈STORE NO: 88

Since all the Control Stores(100,160,200) pvalues > 0.05, so There is no significant relation between control store and trail store based on the Avg price/unit.

INDEPENDENT SAMPLE t-TEST

- Independent sample t-test is a statistical technique that is used to analyze the mean comparison of two independent groups.
- In independent samples t-test, when we take two samples from the same population, then the mean of the two samples may be identical.
- But when samples are taken from two different populations, then the mean of the sample may differ.
- In this case, it is used to draw conclusions about the means of two populations, and used to tell whether or not they are simila

```
b = Image.open("/content/drive/MyDrive/01/t.webp")
plt.imshow(b)
plt.axis('off')
plt.show()
```

$$t = \frac{\overline{x}_1 - \overline{x}_2}{\sqrt{\left(s^2\left(\frac{1}{n_1} + \frac{1}{n_2}\right)\right)}}$$

#HYPOTHESIS:

#H0: There is no significant difference between the means of Control stores and Trial store in terms of the (measure). ie., μi = μj

#H1: There is no significant difference between the means of Control stores and Trial store in terms of the (measure). ie., μi ≠ μj

```
#t test

def tindtest(measure):
  t_results=[]
  # inputs of Control and trial stores
  trial_str=int(input("Enter the Trial Store no for t-Test : "))
  control_str=[]

  for i in range(3):
    control=int(input("Enter the Control Store no for t-Test : "))
    control_str.append(control)
  print(control_str)

  #data for control and Trial stores
  triald=fpt[fpt['STORE_NBR']==trial_str][measure].values

  for i in control_str:
    controld=fpt[fpt['STORE_NBR']==i][measure].values
    tstat,pvalue=tin(triald,controld)
    t_results.append(f"Trial Store {trial_str} vs Control Store {i} \
n" + f" t-Statistic : {tstat} and pvalue : {pvalue}")

  print(*t_results,sep='\n')
```

TOTAL SALES

```
tindtest('TOT_SALES')
```

```
Enter the Trial Store no for t-Test : 77
Enter the Control Store no for t-Test : 100
Enter the Control Store no for t-Test : 160
Enter the Control Store no for t-Test : 200
[100, 160, 200]
Trial Store 77 vs Control Store 100
 t-Statistic : -21.915609403967654 and pvalue : 4.7823256357852965e-11
Trial Store 77 vs Control Store 160
 t-Statistic : -18.97778171047652 and pvalue : 2.571205180150007e-10
Trial Store 77 vs Control Store 200
 t-Statistic : -5.739937055818855 and pvalue : 9.309304070787809e-05
```

RESULTS:

🎯Trial Store: 77

Since the all the p values < 0.05. So, that there is a significant differences among the means of the Control Stores and Trial stores in terms of Total Sales.

```
tindtest('TOT_SALES')
```

```
Enter the Trial Store no for t-Test : 78
Enter the Control Store no for t-Test : 100
Enter the Control Store no for t-Test : 160
Enter the Control Store no for t-Test : 200
[100, 160, 200]
Trial Store 78 vs Control Store 100
 t-Statistic : -5.253536078711854 and pvalue : 0.00020321639269872007
Trial Store 78 vs Control Store 160
 t-Statistic : -3.2024572540693304 and pvalue : 0.0075978126211862515
Trial Store 78 vs Control Store 200
 t-Statistic : 6.054428989648165 and pvalue : 5.719795520260562e-05
```

RESULTS:

- Since the P value < 0.05 of Control Store:100 and 200. So, There is a significant differences among the means of the Control Store(100 and 200) and Trial store in terms of Total Sales.
- But the pvalue > 0.05 for Control Store : 160. So , There is no significant differences among the means of the Control store: 160 and Trial store: 78 in terms of Total Sales.

```
tindtest('TOT_SALES')
```

```
Enter the Trial Store no for t-Test : 80
Enter the Control Store no for t-Test : 100
Enter the Control Store no for t-Test : 160
Enter the Control Store no for t-Test : 200
[100, 160, 200]
Trial Store 80 vs Control Store 100
 t-Statistic : 0.4992964759893807 and pvalue : 0.6265983777122568
Trial Store 80 vs Control Store 160
 t-Statistic : 2.0152844591712897 and pvalue : 0.06683672903360273
Trial Store 80 vs Control Store 200
 t-Statistic : 9.245162885994075 and pvalue : 8.30528423278519e-07
```

RESULTS:

- Since the P value < 0.05 of Control Store:200. So, There is a significant differences among the means of the Control Store: 200 and Trial store: 80 in terms of Total Sales.
- But the pvalue > 0.05 for Control Store : 100 and 160. So , There is no significant differences among the means of the Control store:100 and 160 vs Trial store: 80 in terms of Total Sales.

AVG TRANSACTION PER CUSTOMER

```
tindtest('AVGTXNCUS')
```

```
Enter the Trial Store no for t-Test : 77
Enter the Control Store no for t-Test : 100
Enter the Control Store no for t-Test : 160
Enter the Control Store no for t-Test : 200
[100, 160, 200]
Trial Store 77 vs Control Store 100
 t-Statistic : 1.4693145773945786 and pvalue : 0.1674710184745898
Trial Store 77 vs Control Store 160
 t-Statistic : 0.3327595732329309 and pvalue : 0.7450570331277895
Trial Store 77 vs Control Store 200
 t-Statistic : -0.3945687984171769 and pvalue : 0.7000839230007138
```

RESULTS:

- Since the pvalue > 0.05 for Control Store : 100,160 and 200. So , There is no significant differences among the means of the Control store:100 and 160 vs Trial store: 77 in terms of Average Transaction per customer.

```
tindtest('AVGTXNCUS')
```

```
Enter the Trial Store no for t-Test : 86
Enter the Control Store no for t-Test : 100
Enter the Control Store no for t-Test : 160
Enter the Control Store no for t-Test : 200
[100, 160, 200]
Trial Store 86 vs Control Store 100
 t-Statistic : -14.348459936202042 and pvalue : 6.450968746863597e-09
Trial Store 86 vs Control Store 160
 t-Statistic : -64.15604633098889 and pvalue : 1.3630262323020899e-16
Trial Store 86 vs Control Store 200
 t-Statistic : -92.45871069281459 and pvalue : 1.7125029392549285e-18
```

RESULTS:

- Since the P value < 0.05 of Control Store:200. So, There is a significant differences among the means of the Control Store: 200 and Trial store: 86 in terms of Average Transaction per customer.

```
tindtest('AVGTXNCUS')
```

```
Enter the Trial Store no for t-Test : 88
Enter the Control Store no for t-Test : 100
Enter the Control Store no for t-Test : 160
Enter the Control Store no for t-Test : 200
[100, 160, 200]
Trial Store 88 vs Control Store 100
 t-Statistic : 0.7873423846152146 and pvalue : 0.44635422150793147
Trial Store 88 vs Control Store 160
 t-Statistic : -5.758993586162026 and pvalue : 9.035001795764042e-05
```