

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import scipy as sp
from statistics import stdev
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
```

```
In [2]: df = pd.read_csv('QVI_data.csv')
```

```
In [3]: df.head()
```

```
Out[3]:
```

	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	
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	

```
In [4]: df.info()

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

```
In [5]: df.describe()
```

Out [5]:

	LYLTY_CARD_NBR	STORE_NBR	TXN_ID	PROD_NBR	PROD_QTY	TO
<b>count</b>	2.648340e+05	264834.000000	2.648340e+05	264834.000000	264834.000000	264834.000000
<b>mean</b>	1.355488e+05	135.079423	1.351576e+05	56.583554	1.905813	1.905813
<b>std</b>	8.057990e+04	76.784063	7.813292e+04	32.826444	0.343436	0.343436
<b>min</b>	1.000000e+03	1.000000	1.000000e+00	1.000000	1.000000	1.000000
<b>25%</b>	7.002100e+04	70.000000	6.760050e+04	28.000000	2.000000	2.000000
<b>50%</b>	1.303570e+05	130.000000	1.351365e+05	56.000000	2.000000	2.000000
<b>75%</b>	2.030940e+05	203.000000	2.026998e+05	85.000000	2.000000	2.000000
<b>max</b>	2.373711e+06	272.000000	2.415841e+06	114.000000	5.000000	5.000000

In [6]: `df['YEAR_MONTH'] = pd.to_numeric([''.join(i.split('-')[:2]) for i in df.DATE])`

In [7]: `df.head()`

Out [7]:

	LYLTY_CARD_NBR	DATE	STORE_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY
<b>0</b>	1000	2018-10-17	1	1	5	Natural Chip Compny SeaSalt175g	
<b>1</b>	1002	2018-09-16	1	2	58	Red Rock Deli Chikn&Garlic Aioli 150g	
<b>2</b>	1003	2019-03-07	1	3	52	Grain Waves Sour Cream&Chives 210G	
<b>3</b>	1003	2019-03-08	1	4	106	Natural ChipCo Hony Soy Chckn175g	
<b>4</b>	1004	2018-11-02	1	5	96	WW Original Stacked Chips 160g	

In [8]: `missing_mnths = pd.pivot_table(df, index='STORE_NBR', columns='YEAR_MONTH', values='TOT_SALES')`

In [9]: `rem_stores = missing_mnths[missing_mnths.isnull().any(axis=1)].index.tolist()`

In [10]: `sales = df.groupby(['STORE_NBR', 'YEAR_MONTH'])['TOT_SALES'].sum()`

In [11]: `avgprice = df.groupby(['STORE_NBR', 'YEAR_MONTH'])['TOT_SALES'].sum() / df`

In [12]: `count_custs = df.groupby(['STORE_NBR', 'YEAR_MONTH'])['LYLTY_CARD_NBR'].nunique()`

In [13]: `txn_cust = pd.DataFrame(df.groupby(['STORE_NBR', 'YEAR_MONTH'])['TXN_ID'].nunique()  
# txn_cust.reset_index(inplace=True)`

In [14]: `count_chips = df.groupby(['STORE_NBR', 'YEAR_MONTH'])['PROD_QTY'].sum() / count_chips`

In [15]: `df_rollup = [sales, avgprice, count_custs, txn_cust, count_chips]`

```
In [25]: tc_sales.head()
```

Out [25]:

		Correlation	Score	fin_score
TrialStore	ControlStore			
77.0	1.0	0.075218	0.408163	0.241691
	2.0	-0.263079	0.590119	0.163520
	3.0	0.806644	0.522914	0.664779
	4.0	-0.263300	0.644934	0.190817
	5.0	-0.110652	0.516320	0.202834

```
In [26]: tc_custs = control_store(t_store, 'N_CUSTOMERS', 0.5, pre_trial_df)
```

```
In [27]: tc_custs.head()
```

Out [27]:

		Correlation	Score	fin_score
TrialStore	ControlStore			
77.0	1.0	0.322168	0.663866	0.493017
	2.0	-0.572051	0.471429	-0.050311
	3.0	0.834207	0.489796	0.662002
	4.0	-0.295639	0.498258	0.101310
	5.0	0.370659	0.512605	0.441632

```
In [28]: tc_avgprice = control_store(t_store, 'AVG_PRICE_PER_UNIT', 0.5, pre_trial_df)
```

```
In [29]: tc_avgprice.head()
```

Out [29]:

		Correlation	Score	fin_score
TrialStore	ControlStore			
77.0	1.0	-0.842303	0.585197	-0.128553
	2.0	0.250020	0.627857	0.438938
	3.0	-0.002904	0.518489	0.257793
	4.0	-0.407973	0.531141	0.061584
	5.0	0.267377	0.510465	0.388921

```
In [30]: control_store_df = pd.concat([tc_sales['fin_score'], tc_custs['fin_score'],  
control_store_df.columns = ['sales_score', 'custs_score', 'avgprice_score']
```

```
In [31]: control_store_df.head()
```

Out [31]:

		sales_score	custs_score	avgprice_score
TrialStore	ControlStore			
77.0	1.0	0.241691	0.493017	-0.128553
	2.0	0.163520	-0.050311	0.438938
	3.0	0.664779	0.662002	0.257793
	4.0	0.190817	0.101310	0.061584
	5.0	0.202834	0.441632	0.388921

```
In [32]: control_store_df['fin_score'] = (control_store_df['sales_score']+control_store_df['custs_score']+control_store_df['avgprice_score'])
control_store_df.sort_values(by='fin_score',ascending=False,inplace=True)
```

```
In [33]: control_store_df.head()
```

Out [33]:

		sales_score	custs_score	avgprice_score	fin_score
TrialStore	ControlStore				
77.0	157.0	0.580429	0.651573	0.751482	0.661161
	167.0	0.590409	0.680385	0.527865	0.599553
	162.0	0.610291	0.626244	0.537008	0.591181
	84.0	0.656972	0.715000	0.341074	0.571015
	71.0	0.789497	0.663123	0.257262	0.569961

```
In [34]: control_store_df.index[0][1]
```

Out [34]: 157.0

```
In [35]: pre_trial_df.head(2)
```

```
Out [35]:
```

	STORE_NBR	YEAR_MONTH	TOT_SALES	AVG_PRICE_PER_UNIT	N_CUSTOMERS	N_CHIEF
0	1	201807	206.9	3.337097	49	
1	1	201808	176.1	3.261111	42	

```
In [37]: viz_df = pre_trial_df.loc[pre_trial_df[(pre_trial_df['STORE_NBR'] == t_store) & (pre_trial_df['YEAR_MONTH'] == t_year_month)]]
```

```
In [38]: viz_df['YEAR_MONTH'] = pd.to_datetime(viz_df['YEAR_MONTH'].astype(str), format='%Y%m')
```

```
In [39]: viz_df['store_type'] = np.where(viz_df['STORE_NBR'] == t_store, 'Trial Store', 'Control Store')
```

```
In [40]: viz_df
```

Out [40]:

	STORE_NBR	YEAR_MONTH	TOT_SALES	AVG_PRICE_PER_UNIT	N_CUSTOMERS	N_C
880	77	2018-07-01	296.8	3.533333	51	
881	77	2018-08-01	255.5	3.452703	47	
882	77	2018-09-01	225.2	3.217143	42	
883	77	2018-10-01	204.5	3.932692	37	
884	77	2018-11-01	245.3	3.661194	41	
885	77	2018-12-01	267.3	3.712500	46	
886	77	2019-01-01	204.4	3.144615	35	
1817	157	2018-07-01	1045.4	3.416340	124	
1818	157	2018-08-01	904.1	3.504264	103	
1819	157	2018-09-01	887.6	3.387786	104	
1820	157	2018-10-01	961.4	3.587313	102	
1821	157	2018-11-01	954.0	3.559701	105	
1822	157	2018-12-01	978.4	3.494286	104	
1823	157	2019-01-01	806.8	3.306557	100	

```
In [41]: def trial_control_plot(data,metric):
          control = data.loc[data['store_type'] == 'Control Store',['YEAR_MONTH',metric]]
          control.set_index('YEAR_MONTH',inplace=True)
          control.rename(columns={metric:'Control'},inplace=True)
          trial = data.loc[data['store_type'] == 'Trial Store',['YEAR_MONTH',metric]]
          trial.set_index('YEAR_MONTH',inplace=True)
          trial.rename(columns={metric:'Trial'},inplace=True)
          df_fin = pd.concat([control,trial],axis=1)
          return df_fin
```

```
In [42]: df_viz = trial_control_plot(viz_df,'TOT_SALES')
```

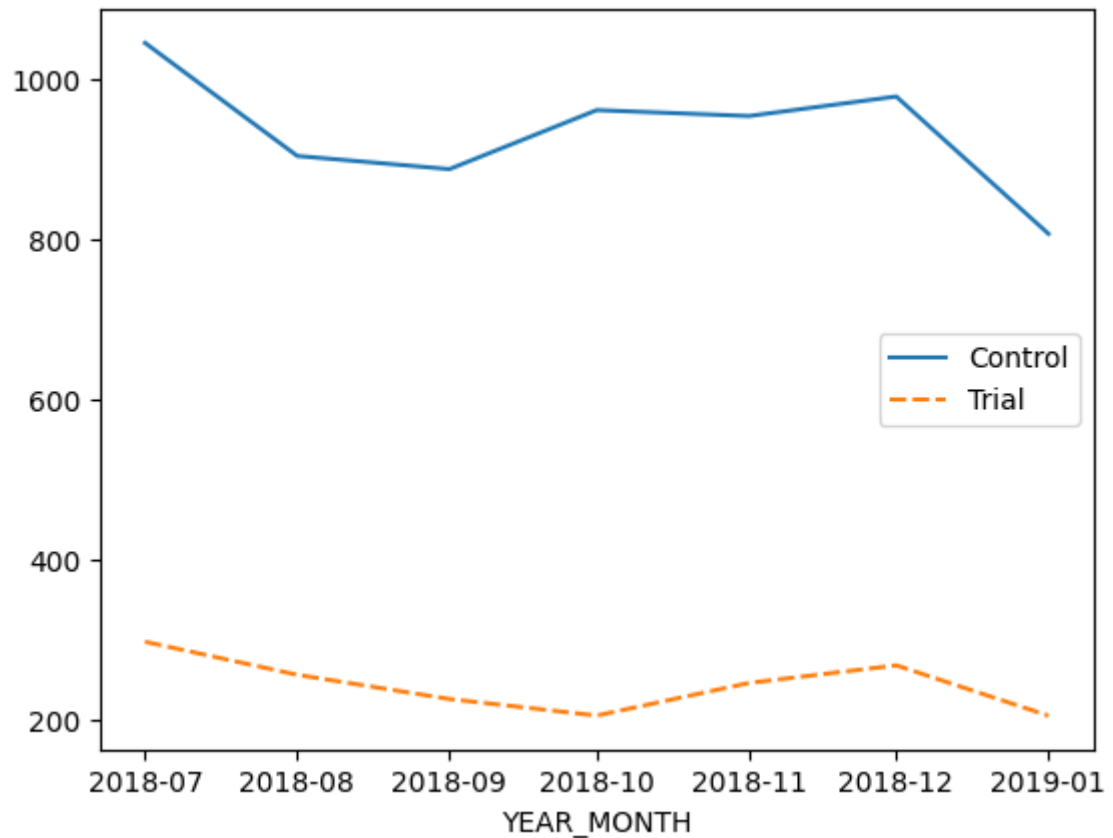
```
In [43]: df_viz.head(2)
```

Out [43]:

	Control	Trial
YEAR_MONTH		
2018-07-01	1045.4	296.8
2018-08-01	904.1	255.5

```
In [44]: sns.lineplot(data=df_viz)
```

Out [44]: <AxesSubplot: xlabel='YEAR\_MONTH'>



### Not using avg price

```
In [66]: control_store_df = pd.concat([tc_sales['fin_score'], tc_custs['fin_score']]
control_store_df.columns = ['sales_score', 'custs_score']
```

```
In [67]: control_store_df['fin_score'] = (control_store_df['sales_score'] + control_
control_store_df.sort_values(by='fin_score', ascending=False, inplace=True)
```

```
In [68]: control_store_df.head()
```

```
Out [68]:
```

		sales_score	custs_score	fin_score	
TrialStore	ControlStore				
	77.0	233.0	0.697290	0.816607	0.756949
		71.0	0.789497	0.663123	0.726310
		84.0	0.656972	0.715000	0.685986
		119.0	0.636046	0.729729	0.682887
		115.0	0.708347	0.645155	0.676751

```
In [73]: viz_df = pre_trial_df.loc[pre_trial_df[(pre_trial_df['STORE_NBR'] == t_store)
```

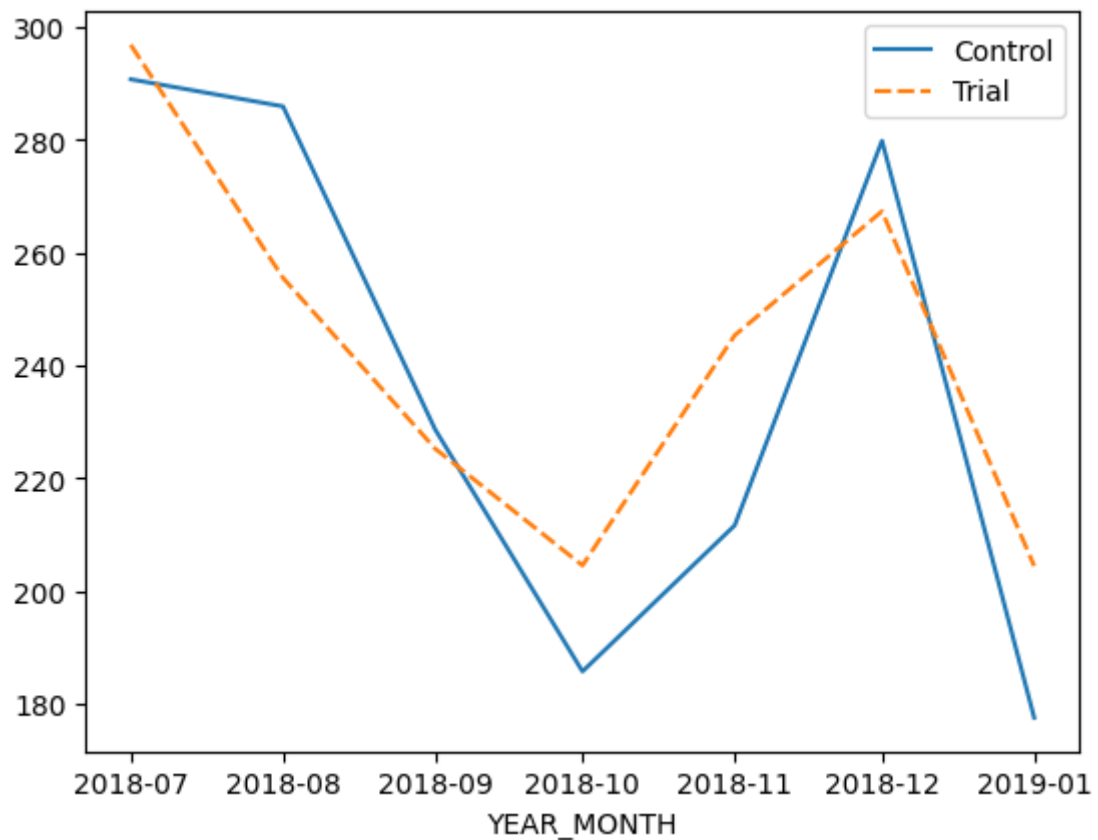
```
In [74]: viz_df['YEAR_MONTH'] = pd.to_datetime(viz_df['YEAR_MONTH'].astype(str), for
```

```
In [75]: viz_df['store_type'] = np.where(viz_df['STORE_NBR'] == t_store, 'Trial Sto
```

```
In [76]: df_viz = trial_control_plot(viz_df, 'TOT_SALES')
```

```
In [77]: sns.lineplot(data=df_viz)
```

Out [77]: <AxesSubplot: xlabel='YEAR\_MONTH'>



## Assessing the trials for Store 77

In [78]: viz\_df.head(1)

Out [78]:

	STORE_NBR	YEAR_MONTH	TOT_SALES	AVG_PRICE_PER_UNIT	N_CUSTOMERS	N_C
880	77	2018-07-01	296.8	3.533333	51	

```
In [58]: trial_data = viz_df.loc[viz_df['store_type'] == 'Trial Store', 'TOT_SALES']
control_data = viz_df.loc[viz_df['store_type'] == 'Control Store', 'TOT_SALES']
assess_factor = trial_data / control_data
```

In [60]: assess\_factor

Out [60]: 1.023617303289553

```
In [107... control_sales = pd.concat(df_rollup, join='outer', axis=1)
control_sales.reset_index(inplace=True)
control_sales['YEAR_MONTH'] = pd.to_datetime(control_sales['YEAR_MONTH']).dt.strftime('%Y-%m-%d')
control_sales = control_sales.loc[control_sales['STORE_NBR'] == 233]
```

In [108... control\_sales.head()



```
Out [108]:
```

	STORE_NBR	YEAR_MONTH	TOT_SALES	0	LYLTY_CARD_NBR	0
<b>2699</b>	233	2018-07-01	290.7	3.303409	51	1.058824 1.6
<b>2700</b>	233	2018-08-01	285.9	3.573750	48	1.041667 1.6
<b>2701</b>	233	2018-09-01	228.6	3.265714	42	1.071429 1.5
<b>2702</b>	233	2018-10-01	185.7	3.316071	35	1.028571 1.5
<b>2703</b>	233	2018-11-01	211.6	3.412903	40	1.025000 1.5

```
In [109...] control_sales['factored_sales'] = control_sales['TOT_SALES'] * assess_fact
```

```
In [173...] diff_df = control_sales[['YEAR_MONTH', 'factored_sales']]
diff_df.reset_index(drop=True, inplace=True)
```

```
In [174...] diff_df.head()
```

```
Out [174]:
```

	YEAR_MONTH	factored_sales
<b>0</b>	2018-07-01	297.565550
<b>1</b>	2018-08-01	292.652187
<b>2</b>	2018-09-01	233.998916
<b>3</b>	2018-10-01	190.085733
<b>4</b>	2018-11-01	216.597421

```
In [175...] fin_df.head(2)
```

```
Out [175]:
```

	STORE_NBR	YEAR_MONTH	TOT_SALES	AVG_PRICE_PER_UNIT	N_CUSTOMERS	N_CH
<b>0</b>	1	201807	206.9	3.337097	49	
<b>1</b>	1	201808	176.1	3.261111	42	

```
In [151...] trial_sales = fin_df.loc[fin_df['STORE_NBR'] == t_store, ['YEAR_MONTH', 'TOT_SALES']]
```

```
In [152...] trial_sales = pd.DataFrame(trial_sales)
trial_sales.reset_index(drop=True, inplace=True)
trial_sales['YEAR_MONTH'] = pd.to_datetime(trial_sales['YEAR_MONTH'].astype(str) + '-01')
```

```
In [176...] diff_df = pd.concat([diff_df, trial_sales], axis=1)
```

```
In [177...] diff_df
```

```
Out [177]:
```

	YEAR_MONTH	factored_sales	YEAR_MONTH	TOT_SALES
0	2018-07-01	297.565550	2018-07-01	296.8
1	2018-08-01	292.652187	2018-08-01	255.5
2	2018-09-01	233.998916	2018-09-01	225.2
3	2018-10-01	190.085733	2018-10-01	204.5
4	2018-11-01	216.597421	2018-11-01	245.3
5	2018-12-01	286.408121	2018-12-01	267.3
6	2019-01-01	181.692071	2019-01-01	204.4
7	2019-02-01	249.762622	2019-02-01	235.0
8	2019-03-01	203.802205	2019-03-01	278.5
9	2019-04-01	162.345704	2019-04-01	263.5
10	2019-05-01	352.533799	2019-05-01	299.3
11	2019-06-01	226.219424	2019-06-01	264.7

```
In [178... diff_df.columns = ['YEAR_MONTH', 'factored_sales', 'REM', 'TOT_SALES']
diff_df.drop(columns='REM', inplace=True)
diff_df.rename(columns={'TOT_SALES': 'trial_sales'}, inplace=True)
```

```
In [179... diff_df.head()
```

```
Out [179]:
```

	YEAR_MONTH	factored_sales	trial_sales
0	2018-07-01	297.565550	296.8
1	2018-08-01	292.652187	255.5
2	2018-09-01	233.998916	225.2
3	2018-10-01	190.085733	204.5
4	2018-11-01	216.597421	245.3

```
In [180... diff_df['%diff'] = abs(diff_df.factored_sales - diff_df.trial_sales) / dif
```

```
In [181... diff_df['YEAR_MONTH'] = diff_df['YEAR_MONTH'].dt.strftime('%Y%m')
```

```
In [182... diff_df['YEAR_MONTH'] = pd.to_numeric(diff_df['YEAR_MONTH'])
```

```
In [186... sd = stdev(diff_df.loc[diff_df['YEAR_MONTH'] < 201902, '%diff'])
```

```
In [187... sd
```

```
Out[187]: 0.049940762641425364
```

```
In [188... diff_df['t_val'] = (diff_df['%diff'] - 0) / sd
```

```
In [190... diff_df.loc[(diff_df['YEAR_MONTH'] > 201901) & (diff_df['YEAR_MONTH'] < 20
```

```
Out [190]:
```

	YEAR_MONTH	factored_sales	trial_sales	%diff	t_val
7	201902	249.762622	235.0	0.059107	1.183534
8	201903	203.802205	278.5	0.366521	7.339116
9	201904	162.345704	263.5	0.623080	12.476373

```
In [189... sp.stats.t.isf(0.05,7)
```

```
Out [189]: 1.8945786050613054
```

```
In [192... control_sales.columns = ['STORE_NBR', 'YEAR_MONTH', 'TOT_SALES', 'AVG_PRICE_P
```

```
In [193... control_sales.head()
```

```
Out [193]:
```

	STORE_NBR	YEAR_MONTH	TOT_SALES	AVG_PRICE_PER_UNIT	N_CUSTOMERS	N
2699	233	2018-07-01	290.7	3.303409	51	
2700	233	2018-08-01	285.9	3.573750	48	
2701	233	2018-09-01	228.6	3.265714	42	
2702	233	2018-10-01	185.7	3.316071	35	
2703	233	2018-11-01	211.6	3.412903	40	

```
In [194... c_sales = control_sales.loc[:, ['YEAR_MONTH', 'factored_sales']]
c_sales.set_index('YEAR_MONTH', inplace=True)
```

```
In [195... c_sales.head()
```

```
Out [195]:
```

factored_sales	
YEAR_MONTH	
2018-07-01	297.565550
2018-08-01	292.652187
2018-09-01	233.998916
2018-10-01	190.085733
2018-11-01	216.597421

```
In [213... c_sales['Control 95% C.I'] = c_sales['factored_sales'] * (1-sd*2)
c_sales['Control 5% C.I'] = c_sales['factored_sales'] * (1+sd)
```

```
In [214... t_sales = pd.concat(df_rollup, join='outer', axis=1)
t_sales.reset_index(inplace=True)
t_sales['YEAR_MONTH'] = pd.to_datetime(t_sales['YEAR_MONTH'].astype(str),
```

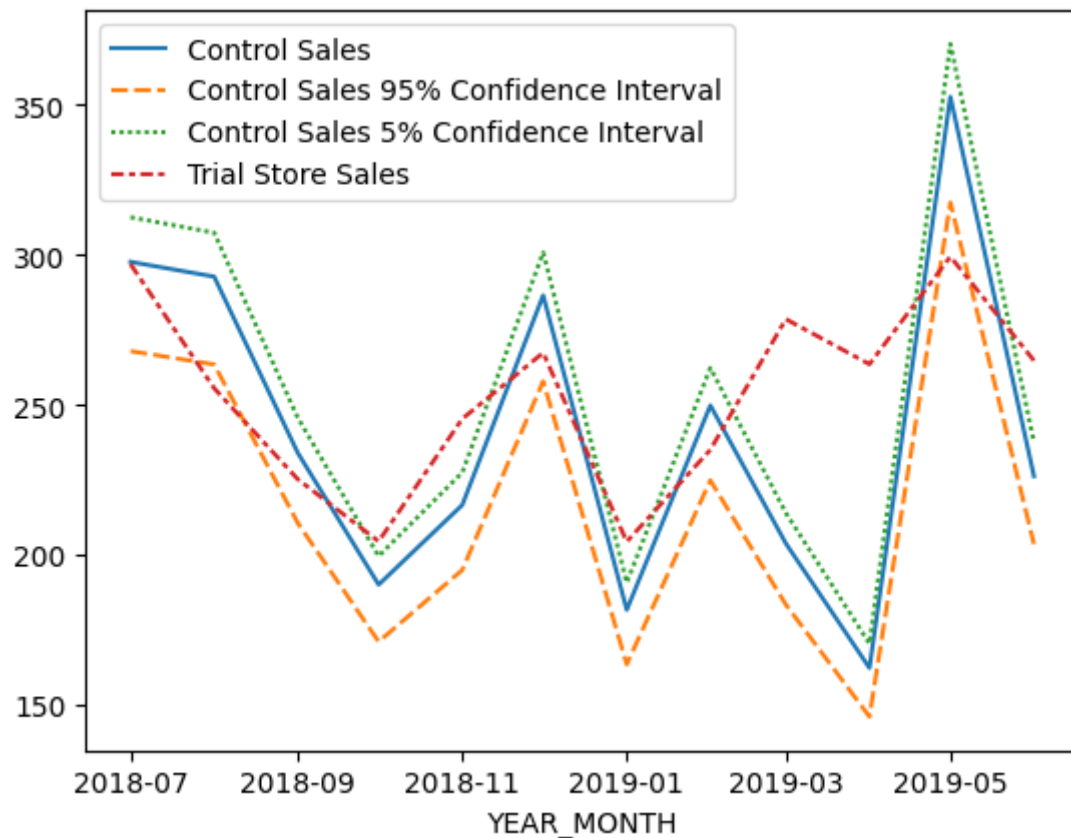
```
In [215... t_sales = t_sales.loc[t_sales['STORE_NBR'] == t_store, ['YEAR_MONTH', 'TOT_S
t_sales.set_index('YEAR_MONTH', inplace=True)
```

```
In [216... fin_sales = pd.merge(c_sales, t_sales, left_index=True, right_index=True)
```

```
In [218... fin_sales.columns = ['Control Sales', 'Control Sales 95% Confidence Interv
```

```
In [219]: sns.lineplot(data=fin_sales)
```

```
Out[219]: <AxesSubplot: xlabel='YEAR_MONTH'>
```



```
In [226]: cust_data = pre_trial_df
cust_data = cust_data.loc[cust_data['STORE_NBR'] == t_store] |
cust_data['store_type'] = np.where(cust_data['STORE_NBR'] == t_store, 'Trial Store', 'Control Store')
cust_data['YEAR_MONTH'] = pd.to_numeric(cust_data['YEAR_MONTH'])
```

```
In [239]: pre_trial_df.head(1)
```

```
Out[239]:
```

	STORE_NBR	YEAR_MONTH	TOT_SALES	AVG_PRICE_PER_UNIT	N_CUSTOMERS	N_CH
0	1	201807	206.9	3.337097	49	

```
In [235]: c_custs = cust_data.loc[cust_data['store_type'] == 'Control Store', ['YEAR_MONTH', 'TOT_SALES', 'AVG_PRICE_PER_UNIT', 'N_CUSTOMERS']]
c_custs['YEAR_MONTH'] = pd.to_datetime(c_custs['YEAR_MONTH'], format='%Y%m')
c_custs.rename(columns={'N_CUSTOMERS': 'Num_Customers_Control_Store'}, inplace=True)
c_custs.set_index('YEAR_MONTH', inplace=True)
t_custs = cust_data.loc[cust_data['store_type'] == 'Trial Store', ['YEAR_MONTH', 'TOT_SALES', 'AVG_PRICE_PER_UNIT', 'N_CUSTOMERS']]
t_custs['YEAR_MONTH'] = pd.to_datetime(t_custs['YEAR_MONTH'], format='%Y%m')
t_custs.rename(columns={'N_CUSTOMERS': 'Num_Customers_Trial_Store'}, inplace=True)
t_custs.set_index('YEAR_MONTH', inplace=True)
```

```
In [238]: merged_custs = pd.concat([c_custs, t_custs], axis=1)
```

```
In [241]: tc = pre_trial_df.loc[pre_trial_df['STORE_NBR'] == t_store, 'N_CUSTOMERS']
cc = pre_trial_df.loc[pre_trial_df['STORE_NBR'] == control_store_df.index[0], 'N_CUSTOMERS']
assess_factor = tc/cc
assess_factor
```

```
Out[241]: 1.0033557046979866
```

```
In [243... c_custs['control customers'] = c_custs['Num_Customers_Control_Store'] * as
```

```
In [256... c_custs = pd.concat(df_rollup,join='outer',axis=1)
c_custs.reset_index(inplace=True)
c_custs['YEAR_MONTH'] = pd.to_datetime(c_custs['YEAR_MONTH'].astype(str),
c_custs = c_custs.loc[c_custs['STORE_NBR'] == 233]
c_custs = c_custs[['YEAR_MONTH', 'LYLTY_CARD_NBR']]
c_custs.columns = ['YEAR_MONTH', 'Control Customers']
c_custs.reset_index(drop=True,inplace=True)
```

```
In [257... t_custs = pd.concat(df_rollup,join='outer',axis=1)
t_custs.reset_index(inplace=True)
t_custs['YEAR_MONTH'] = pd.to_datetime(t_custs['YEAR_MONTH'].astype(str),
t_custs = t_custs.loc[t_custs['STORE_NBR'] == 77]
t_custs = t_custs[['YEAR_MONTH', 'LYLTY_CARD_NBR']]
t_custs.columns = ['YEAR_MONTH', 'Trial Customers']
t_custs.reset_index(drop=True,inplace=True)
```

```
In [266... merged_custs = pd.concat([c_custs,t_custs],axis=1)
merged_custs.columns = ['YEAR_MONTH', 'Control_Customers', 'rem', 'Trial_Cust
merged_custs.drop(columns='rem',inplace=True)
```

```
In [267... merged_custs['Control_Customers'] = merged_custs['Control_Customers'] * as
```

```
In [268... merged_custs['%diff'] = abs(merged_custs.Control_Customers - merged_custs
```

```
In [271... merged_custs['YEAR_MONTH'] = merged_custs['YEAR_MONTH'].dt.strftime('%Y%m
merged_custs['YEAR_MONTH'] = pd.to_numeric(merged_custs['YEAR_MONTH'])
```

```
In [272... sd = stdev(merged_custs.loc[merged_custs['YEAR_MONTH'] < 201902,'%diff'])
```

```
In [273... sd
```

```
Out[273]: 0.018240748558243945
```

```
In [274... merged_custs['t_val'] = (merged_custs['%diff'] - 0)/sd
```

```
In [276... merged_custs.loc[(merged_custs['YEAR_MONTH'] > 201901) & (merged_custs['YI
```

```
Out[276]: 7      0.183352
8      13.476388
9      30.778725
Name: t_val, dtype: float64
```

```
In [277... sp.stats.t.isf(0.05,7)
```

```
Out[277]: 1.8945786050613054
```

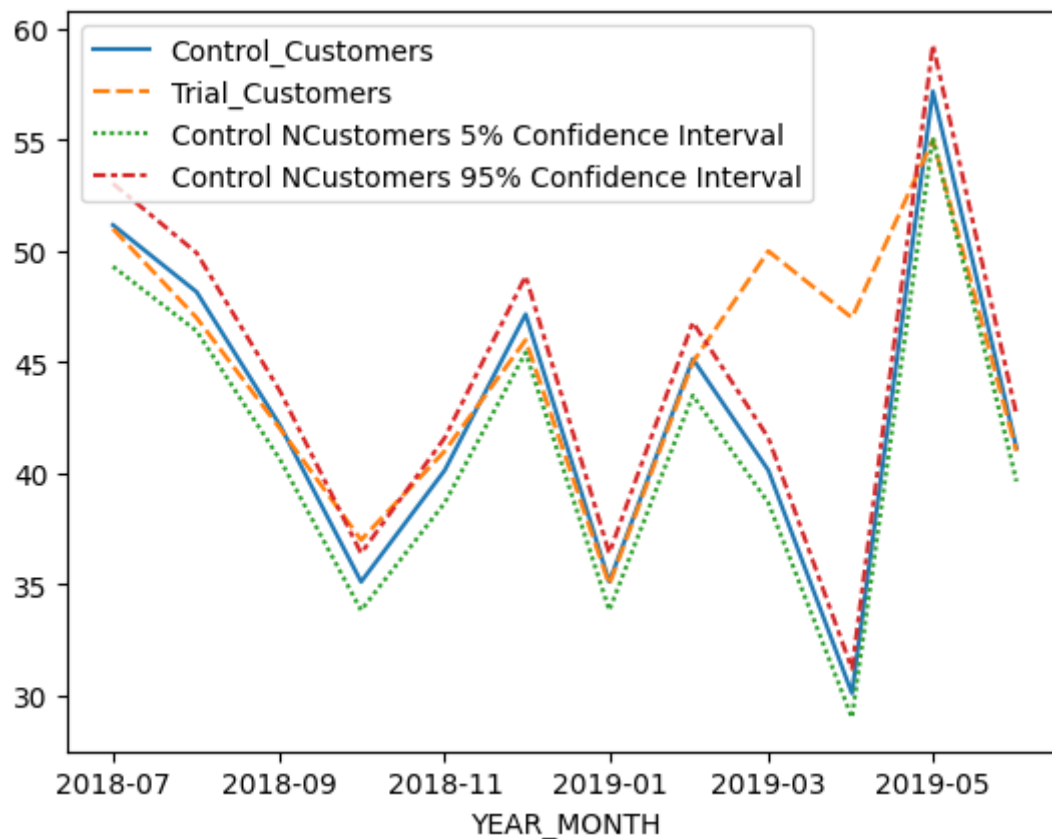
```
In [ ]: controlCustomers['Control 5% Confidence Interval'] = controlCustomers['Co
controlCustomers['Control 95% Confidence Interval'] = controlCustomers['Co
controlCustomers
```

```
In [281... merged_custs = merged_custs[['YEAR_MONTH', 'Control_Customers', 'Trial_Custo
merged_custs.loc[:, 'Control NCustomers 5% Confidence Interval'] = merged_c
merged_custs.loc[:, 'Control NCustomers 95% Confidence Interval'] = merged_c
```

```
In [286... merged_custs['YEAR_MONTH'] = pd.to_datetime(merged_custs['YEAR_MONTH'], fo
merged_custs.set_index('YEAR_MONTH', inplace=True)
```

```
In [287... sns.lineplot(data=merged_custs)
```

```
Out[287]: <AxesSubplot: xlabel='YEAR_MONTH'>
```



## Conclusion

### Trial store 77

- The sales of store 77 lies outside the 5% and in 2 of 3 trial months.

### Trial store 86

- Performance lies within the 5% and 95% CI, not significantly different to its control store.
- The trial had a significant impact on increasing the number of customers, as the # of customers is significantly higher in all trial months. However sales trend doesn't match this. We must check if there were sales in the trial store that may have caused this.

### Trial store 88

- Sales performance is significantly different to control store.
- The # of customers for the trial store in trial period is significantly higher than the control store for 2 of the 3 months period.