

```
In [2]: import numpy as np
import pandas as pd
```

```
In [3]: customer_transactions = pd.read_excel('customer_transaction_behavior.xlsx')
customer_transactions
```

2	2	2019-05-20	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9	170g	SMITHS	SIN
3	3	2018-08-17	2	2373	974	69	Smiths Chip S/Cream&Onion 175g	5	15.0	175g	SMITHS	SIN
4	4	2018-08-18	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	13.8	150g	KETTLE	SIN
...	...	...	...	...	...	...	...	...	...	...	...	...
264829	264829	2019-03-09	272	272319	270088	89	Kettle Sweet Chilli And Sour Cream 175g	2	10.8	175g	KETTLE	SIN
264830	264830	2018-08-13	272	272358	270154	74	Tostitos Splash Of Lime 175g	1	4.4	175g	TOSTITOS	SIN
...	...	...	...	...	...	...	...	...	...	...	...	...

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

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 264834 entries, 0 to 264833
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   Unnamed: 0            264834 non-null  int64   
1   DATE                  264834 non-null  datetime64[ns]
2   STORE_NBR            264834 non-null  int64   
3   LYLTY_CARD_NBR       264834 non-null  int64   
4   TXN_ID               264834 non-null  int64   
5   PROD_NBR             264834 non-null  int64   
6   PROD_NAME            264834 non-null  object   
7   PROD_QTY             264834 non-null  int64   
8   TOT_SALES            264834 non-null  float64  
9   PROD_SIZE            264834 non-null  object   
10  BRAND_NAME           264834 non-null  object   
11  LIFESTAGE            264834 non-null  object   
12  PREMIUM_CUSTOMER     264834 non-null  object   
dtypes: datetime64[ns](1), float64(1), int64(6), object(5)
memory usage: 26.3+ MB
```

```
In [5]: # Trial period if from February 2019 to April 2019
# https://stackoverflow.com/questions/2937057/select-dataframe-rows-between-two-dates
mask = (customer_transactions['DATE'] > '2019-02-01') & (customer_transactions['DATE'] <= '2019-04-30')
trial_dates = customer_transactions.loc[mask]
trial_dates
```

```
Out[5]:
```

Unnamed: 0	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	PROD_SIZE	BRAND_NAME	
205	205	2019-02-27	1	1081	93	96	WW Original Stacked Chips 160g	2	3.8	160g	WOOLWORTHS
206	206	2019-03-19	1	1081	94	8	Smiths Crinkle Cut Chips Original 170g	1	2.9	170g	SMITHS
210	210	2019-03-09	1	1307	347	54	CCs Original 175g	1	2.1	175g	CCS
212	212	2019-03-13	1	1348	391	48	Red Rock Deli Sp Salt & Truffle 150G	1	2.7	150g	RRD
213	213	2019-04-06	1	1395	453	61	Smiths Crinkle Cut Chips Chicken 170g	2	5.8	170g	SMITHS
...	...	...	...	...	...	...	...	...	...	...	

```
In [8]: trial_period = trial_dates[['DATE', 'STORE_NBR', 'LYLTY_CARD_NBR', 'PROD_QTY', 'TOT_SALES', 'PROD_SIZE', 'BRAND_NAME', 'LIFESTAGE', 'PREMIUM_CUSTOMER']]
trial_period
```

Out[8]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	PROD_QTY	TOT_SALES	PROD_SIZE	BRAND_NAME	LIFESTAGE	PREMIUM_CUSTOMER
205	2019-02-27	1	1081	2	3.8	160g	WOOLWORTHS	MIDAGE SINGLES/COUPLES	Budget
206	2019-03-19	1	1081	1	2.9	170g	SMITHS	MIDAGE SINGLES/COUPLES	Budget
210	2019-03-09	1	1307	1	2.1	175g	CCS	MIDAGE SINGLES/COUPLES	Budget
212	2019-03-13	1	1348	1	2.7	150g	RRD	MIDAGE SINGLES/COUPLES	Budget
213	2019-04-06	1	1395	2	5.8	170g	SMITHS	MIDAGE SINGLES/COUPLES	Budget
...	...	...	...	...	...	...	...	...	...
264821	2019-03-17	272	272156	2	8.4	135g	KETTLE	YOUNG SINGLES/COUPLES	Premium
264822	2019-03-13	272	272193	1	4.6	150g	KETTLE	YOUNG SINGLES/COUPLES	Premium
264824	2019-03-25	272	272194	2	7.6	110g	COBS	YOUNG SINGLES/COUPLES	Premium
264827	2019-03-16	272	272236	2	7.6	110g	INFUZIONI	YOUNG SINGLES/COUPLES	Premium
264829	2019-03-09	272	272319	2	10.8	175g	KETTLE	YOUNG SINGLES/COUPLES	Premium

64013 rows × 9 columns

```
In [52]: # Performance Stores
performance_stores = trial_period[(trial_period['STORE_NBR'] == 77) | (trial_period['STORE_NBR'] == 86) | (trial_period['STORE_NBR'] == 88)]
performance_stores.groupby(['STORE_NBR']).agg({'PROD_QTY': ['mean'], 'TOT_SALES': ['mean']})
```

Out[52]:

	PROD_QTY	TOT_SALES
	mean	mean
STORE_NBR		
77	1.575342	5.246575
86	1.997531	6.838519
88	2.000000	8.818333

```
In [58]: # Look for stores with similar performance in prod_qty and sales
# Store with sales more than 1.5 but less than 2 in mean
trial_dates[(trial_dates['TOT_SALES'] >= 8) & (trial_dates['PROD_QTY'] <= 9)]
```

266	266	2019-03-05	4	4074	2981	51	Doritos Mexicana 170g	2	8.8	170g	DORITOS	SINGLES/COUPLES
279	279	2019-02-12	4	4363	4283	65	Old El Paso Salsa Dip Chnky Tom Ht300g	2	10.2	300g	OLD EL PASO	SINGLES/COUPLES
...	...	...	...	...	...	...	...	...	...	...	...	...
264762	264762	2019-02-09	269	269133	265836	7	Smiths Crinkle Original 330g	2	11.4	330g	SMITHS	SINGLES/COUPLES
264791	264791	2019-04-28	270	270032	266817	71	Twisties Cheese Burger 250g	2	8.6	250g	TWISTIES	SINGLES/COUPLES
264806	264806	2019-03-07	271	271013	268303	70	Tyrrells Crisps Lightly Salted 165g	2	8.4	165g	TYRRELLS	SINGLES/COUPLES

```
In [65]: trial_dates.groupby(['STORE_NBR'])[['TOT_SALES']].mean().sort_values(by='TOT_SALES', ascending=False).to_string()
```

8.597101\n175	8.594048\n105	8.504553\n115	8.558556\n1119	8.545040\n1195	8.543309\n1179
8.535629\n240	8.534677\n68	8.528448\n80	8.443881\n7	8.420822\n216	8.407407\n245
8.355469\n34	8.339024\n175	8.314749\n215	8.291870\n17	8.288514\n143	8.277143\n272
8.171338\n154	8.165649\n108	7.967257\n19	7.939683\n166	7.932337\n258	7.771429\n145
7.755556\n87	7.744737\n193	7.600000\n221	7.420937\n132	7.346667\n243	7.292073\n259
7.288462\n213	7.242316\n271	7.216828\n164	7.193194\n191	7.186701\n168	7.186385\n33
7.180952\n196	7.176667\n152	7.166190\n45	7.154601\n178	7.143836\n70	7.142587\n115
7.142222\n28	7.129730\n100	7.124519\n93	7.119169\n10	7.106395\n30	7.106180\n269
7.105398\n116	7.078889\n63	7.078832\n180	7.073766\n32	7.068608\n129	7.067806\n109
7.065041\n262	7.064615\n83	7.056973\n23	7.051685\n112	7.050000\n24	7.046250\n43
7.045411\n71	7.043981\n137	7.043068\n69	7.042507\n15	7.033121\n105	7.027945\n247
7.026633\n230	7.024944\n128	7.024228\n147	7.023913\n156	7.017703\n56	7.014815\n106
7.012603\n104	7.011444\n110	7.007407\n257	7.005301\n236	7.003200\n184	7.002558\n229
6.996923\n118	6.990736\n208	6.990033\n101	6.988450\n157	6.986207\n113	6.985352\n158
6.980000\n270	6.979747\n13	6.971981\n162	6.970056\n78	6.967619\n62	6.965147\n209
6.962275\n55	6.954444\n122	6.945833\n67	6.945604\n227	6.942706\n179	6.935358\n172
6.931492\n133	6.928052\n160	6.928019\n212	6.923125\n241	6.915823\n232	6.915297\n222
6.914956\n155	6.913973\n190	6.910326\n183	6.909195\n48	6.908721\n207	6.904709\n94
6.904282\n47	6.893289\n102	6.873054\n148	6.856928\n86	6.838519\n97	6.836467\n219
6.836047\n114	6.834606\n5	6.825325\n144	6.821711\n223	6.794581\n39	6.755923\n107

```
In [68]: # For store 77 : Store 53
# For store 86 : Store 97
# For store 88 : Store 153
control_stores = trial_period[(trial_period['STORE_NBR'] == 18) | (trial_period['STORE_NBR'] == 97) | (trial_period['STORE_NBR']
control_stores.groupby(['STORE_NBR']).agg({'PROD_QTY': ['mean'], 'TOT_SALES': ['mean']})
```

Out[68]:

STORE_NBR	PROD_QTY		TOT_SALES	
	mean		mean	
18	1.557252		5.432824	
97	2.000000		6.836467	
153	1.980609		8.813573	

```
In [69]: performance_stores
```

Out[69]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	PROD_QTY	TOT_SALES	PROD_SIZE	BRAND_NAME	LIFESTAGE	PREMIUM_CUSTOMER
1438	2019-03-28	77	77000	1	3.3	190g	CHEETOS	MIDAGE SINGLES/COUPLES	Budget
1439	2019-04-13	77	77000	1	3.0	175g	SMITHS	MIDAGE SINGLES/COUPLES	Budget
1441	2019-03-03	77	77063	2	8.4	165g	TYRRELLS	MIDAGE SINGLES/COUPLES	Budget
1442	2019-02-20	77	77069	2	6.0	175g	NATURAL	MIDAGE SINGLES/COUPLES	Budget
1443	2019-03-08	77	77069	2	5.8	170g	SMITHS	MIDAGE SINGLES/COUPLES	Budget
...	...	...	...	...	...	...	...	...	...
260683	2019-04-26	88	88058	2	7.4	134g	PRINGLES	YOUNG SINGLES/COUPLES	Premium
260684	2019-04-28	88	88058	2	7.4	134g	PRINGLES	YOUNG SINGLES/COUPLES	Premium
260689	2019-04-15	88	88064	2	10.8	175g	KETTLE	YOUNG SINGLES/COUPLES	Premium
260692	2019-04-01	88	88106	2	7.6	110g	INFUZIONI	YOUNG SINGLES/COUPLES	Premium
260700	2019-03-02	88	88127	2	7.4	134g	PRINGLES	YOUNG SINGLES/COUPLES	Premium

1031 rows × 9 columns

In [76]:

```
performance_and_control = performance_stores.append(control_stores, ignore_index=True)
performance_and_control
```

C:\Program Files\Sublime Text\sublime\_text.exe\ipykernel\_12612\3705855491.py:1: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.  
performance\_and\_control = performance\_stores.append(control\_stores, ignore\_index=True)

Out[76]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	PROD_QTY	TOT_SALES	PROD_SIZE	BRAND_NAME	LIFESTAGE	PREMIUM_CUSTOMER
0	2019-03-28	77	77000	1	3.3	190g	CHEETOS	MIDAGE SINGLES/COUPLES	Budget
1	2019-04-13	77	77000	1	3.0	175g	SMITHS	MIDAGE SINGLES/COUPLES	Budget
2	2019-03-03	77	77063	2	8.4	165g	TYRRELLS	MIDAGE SINGLES/COUPLES	Budget
3	2019-02-20	77	77069	2	6.0	175g	NATURAL	MIDAGE SINGLES/COUPLES	Budget
4	2019-03-08	77	77069	2	5.8	170g	SMITHS	MIDAGE SINGLES/COUPLES	Budget
...	...	...	...	...	...	...	...	...	...

In [85]:

```
performance_and_control.groupby(['STORE_NBR']).agg({'LYLTY_CARD_NBR': 'count', 'PROD_QTY': 'mean', 'TOT_SALES': 'sum'}).rename(cc)
```

Out[85]:

	CUSTOMER COUNT	AVERAGE PRODUCT SOLD	TOTAL REVENUE
STORE_NBR			
18	131	1.557252	711.7
77	146	1.575342	766.0
86	405	1.997531	2769.6
88	480	2.000000	4232.8
97	351	2.000000	2399.6
153	361	1.980609	3181.7

In [83]:

```
performance_and_control.groupby(['STORE_NBR']).agg({'PROD_QTY': 'mean', 'TOT_SALES': 'sum'})
```

STORE_NBR	LYLTY_CARD_NBR		
18	18006	1.0	4.6
	18010	2.0	6.6
	18012	2.0	6.2
	18013	1.0	3.7
	18025	2.0	13.6
...	...	...	...
153	153363	2.0	32.4
	153366	2.0	27.0
	153367	2.0	7.8
	153368	2.0	19.6
	153370	2.0	9.2

1118 rows × 2 columns