In [1]:
```
import pandas as pd
```

In [3]:
```
trans_data = pd.read_excel(r"C:\\Users\\LORD\\OneDrive\\Desktop\\genuies world\\QVI_transaction_data.xlsx")
```

In [4]:
```
purchase_data = pd.read_csv(r"C:\\Users\\LORD\\OneDrive\\Desktop\\genuies world\\QVI_purchase_behaviour.csv")
```

In [5]:
```
trans_data
```

Out[5]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT
0	43390	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	
1	43599	1	1307	348	66	CCs Nacho Cheese 175g	3	
2	43605	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	
3	43329	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	
4	43330	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	
...	...	...	...	...	...	...	...	
264831	43533	272	272319	270088	89	Kettle Sweet Chilli And Sour Cream 175g	2	
264832	43325	272	272358	270154	74	Tostitos Splash Of Lime 175g	1	
264833	43410	272	272379	270187	51	Doritos Mexicana 170g	2	
264834	43461	272	272379	270188	42	Doritos Corn Chip Mexican Jalapeno 150g	2	
264835	43365	272	272380	270189	74	Tostitos Splash Of Lime 175g	2	

264836 rows × 8 columns

In [6]:
```
trans_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 264836 entries, 0 to 264835
Data columns (total 8 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   DATE            264836 non-null  int64  
 1   STORE_NBR       264836 non-null  int64  
 2   LYLTY_CARD_NBR  264836 non-null  int64  
 3   TXN_ID          264836 non-null  int64  
 4   PROD_NBR        264836 non-null  int64  
 5   PROD_NAME       264836 non-null  object 
 6   PROD_QTY        264836 non-null  int64  
 7   TOT_SALES       264836 non-null  float64
dtypes: float64(1), int64(6), object(1)
memory usage: 16.2+ MB
```

In [7]:
```
trans_data['new_date'] = pd.to_datetime(trans_data['DATE'] , unit = 'd' , origin = '1899-12-30' )
del trans_data['DATE']
trans_data.head(3)
```

Out[7]:

	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	new_da
0	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0	2018-1
1	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3	2019-0
2	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9	2019-0

In [8]:
```
trans_cleaned = trans_data[trans_data['PROD_NAME'].str.contains(r'Chip|Chps', na = False)]
trans_cleaned['TOT_SALES'].max()
```

Out[8]:
```
29.5
```

In [9]:
```
trans_cleaned[trans_cleaned['TOT_SALES'].isin(['NAN','',0])]
trans_cleaned[trans_cleaned['TOT_SALES'] <= 0]
```

Out[9]:

	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	new_dat

In [10]:
```
trans_cleaned.head(4)
```

Out[10]:

	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	new_
0	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0	2018
2	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9	2019
3	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	15.0	2018
4	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	13.8	2018

In [11]:
```
purchase_data.drop_duplicates(subset ='LYLTY_CARD_NBR')
```

Out[11]:

	LYLTY_CARD_NBR	LIFESTAGE	PREMIUM_CUSTOMER
0	1000	YOUNG SINGLES/COUPLES	Premium
1	1002	YOUNG SINGLES/COUPLES	Mainstream
2	1003	YOUNG FAMILIES	Budget
3	1004	OLDER SINGLES/COUPLES	Mainstream
4	1005	MIDAGE SINGLES/COUPLES	Mainstream
...	...	...	...
72632	2370651	MIDAGE SINGLES/COUPLES	Mainstream
72633	2370701	YOUNG FAMILIES	Mainstream
72634	2370751	YOUNG FAMILIES	Premium
72635	2370961	OLDER FAMILIES	Budget
72636	2373711	YOUNG SINGLES/COUPLES	Mainstream

72637 rows × 3 columns

In [12]:
```
full_data = pd.merge(purchase_data , trans_cleaned , on =
'LYLTY_CARD_NBR')
```

In [13]:
```
full_data[['pack_name' ,'pack_size']] =
full_data['PROD_NAME'].str.split(r'(?=\d)' , n = 1 , expand = True)
full_data['company_name'] = full_data['pack_name'].str.split().str[0]
full_data.head(4)
```

Out[13]:

	LYLTY_CARD_NBR	LIFESTAGE	PREMIUM_CUSTOMER	STORE_NBR	TXN_ID	PROD_NBR	PRO
0	1000	YOUNG SINGLES/COUPLES	Premium	1	1	5	Nat Sea
1	1003	YOUNG FAMILIES	Budget	1	4	106	Chip Ch
2	1004	OLDER SINGLES/COUPLES	Mainstream	1	5	96	WW Stack
3	1011	OLDER SINGLES/COUPLES	Mainstream	1	15	1	Cr

In [25]:
```
sum_of_sales_per_company_name = full_data.groupby('company_name' ,
as_index = False)['TOT_SALES'].sum().sort_values(by = 'TOT_SALES' ,
ascending = False)
sum_of_sales_per_company_name
```

Out[25]:

	company_name	TOT_SALES
5	Smiths	129237.8
1	Doritos	127597.3
6	Thins	88852.5
3	Kettle	84419.2
0	Cobs	70569.8
4	Natural	34272.0
8	WW	26655.1
7	Tostitos	26474.8
2	French	7929.0

In [24]:
```
Total_sales = round(full_data['TOT_SALES'].sum(),2)
Total_sales
```

Out[24]:
```
596007.5
```

In [23]:
```
slaes_per_store_nmbr = full_data.groupby('STORE_NBR',as_index = False)
['TOT_SALES'].sum().sort_values(by = 'TOT_SALES' , ascending = False)
```

```
slaes_per_store_nmbr
```

Out[23]:

	STORE_NBR	TOT_SALES
221	226	5456.9
162	165	4920.9
3	4	4882.1
57	58	4840.5
85	88	4790.4
...	...	...
114	117	51.7
89	92	9.2
10	11	6.7
30	31	6.0
202	206	4.6

267 rows × 2 columns

In [22]:

```
count_of_txnid_per_customer = full_data.groupby('LYLTY_CARD_NBR' , as_index = False)['TXN_ID'].count().sort_values(by = 'TXN_ID' , ascending = False)
count_of_txnid_per_customer
```

Out[22]:

	LYLTY_CARD_NBR	TXN_ID
36350	212185	10
18884	107167	9
46231	270001	9
31878	184013	9
5268	32060	9
...	...	...
20677	118050	1
20674	118047	1
20669	118042	1
20668	118040	1
46810	2373711	1

46811 rows × 2 columns

In [21]:
```
sum_of_sales_per_customer = full_data.groupby('LYLTY_CARD_NBR' , as_index = False)['TOT_SALES'].sum().sort_values(by = 'TOT_SALES' , ascending = False)
sum_of_sales_per_customer
```

Out[21]:

	LYLTY_CARD_NBR	TOT_SALES
11883	69154	79.6
5268	32060	68.6
36350	212185	67.8
14928	86059	67.6
31878	184013	67.6
...	...	...
15566	89138	1.9
45616	265183	1.9
33533	195352	1.9
45615	265181	1.9
45939	268287	1.9

46811 rows × 2 columns

In [20]:
```
sum_of_sales_per_life_stage = full_data.groupby('LIFESTAGE' , as_index = False)['TOT_SALES'].sum().sort_values(by = 'TOT_SALES' , ascending = False)
sum_of_sales_per_life_stage
```

Out[20]:

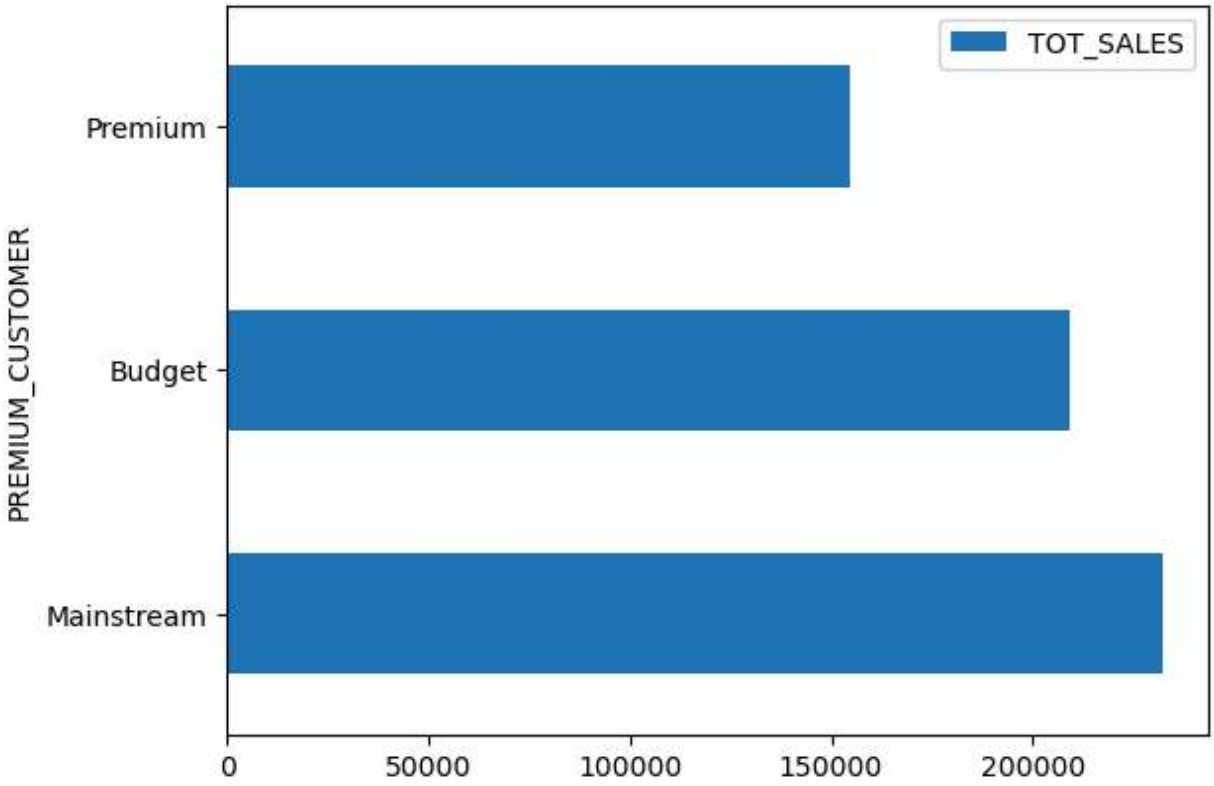
	LIFESTAGE	TOT_SALES
3	OLDER SINGLES/COUPLES	124004.6
4	RETIREES	112026.7
2	OLDER FAMILIES	108687.6
5	YOUNG FAMILIES	98444.9
6	YOUNG SINGLES/COUPLES	80121.2
0	MIDAGE SINGLES/COUPLES	57353.7
1	NEW FAMILIES	15368.8

In [30]:
```
sum_of_sales_per_premium_customer = full_data.groupby('PREMIUM_CUSTOMER' , as_index = False)['TOT_SALES'].sum().sort_values(by = 'TOT_SALES' ,
```

```
ascending = False)
sum_of_sales_per_premium_customer.plot.barh(x = 'PREMIUM_CUSTOMER' )
```

Out[30]:

```
<Axes: ylabel='PREMIUM_CUSTOMER'>
```



In [18]:

```
sum_of_sales_per_product = full_data.groupby('PROD_NAME' , as_index =
False)['TOT_SALES'].sum().sort_values(by = 'TOT_SALES' , ascending =
False)
sum_of_sales_per_product
```

Out[18]:

	PROD_NAME	TOT_SALES
24	Smiths Crnkle Chip Orgnl Big Bag 380g	36367.6
19	Smiths Crinkle Chips Salt & Vinegar 330g	34804.2
11	Kettle Tortilla ChpsHny&Jlpno Chili 150g	29021.4
9	Kettle Tortilla ChpsBtroot&Ricotta 150g	27770.2
10	Kettle Tortilla ChpsFeta&Garlic 150g	27627.6
5	Doritos Corn Chips Cheese Supreme 170g	27183.2
6	Doritos Corn Chips Nacho Cheese 170g	26562.8
30	Tostitos Smoked Chipotle 175g	26474.8
7	Doritos Corn Chips Original 170g	26228.4
3	Doritos Corn Chip Mexican Jalapeno 150g	23887.5
0	Cobs Popd Sea Salt Chips 110g	23852.6
2	Cobs Popd Swt/Chlli &Sr/Cream Chips 110g	23772.8
4	Doritos Corn Chip Southern Chicken 150g	23735.4
1	Cobs Popd Sour Crm &Chives Chips 110g	22944.4
29	Thins Potato Chips Hot & Spicy 175g	20410.5
26	Thins Chips Light& Tangy 175g	20113.5
28	Thins Chips Seasonedchicken 175g	19753.8
27	Thins Chips Salt & Vinegar 175g	19575.6
16	Smiths Chip Thinly Cut Original 175g	9135.0
25	Thins Chips Originl saltd 175g	8999.1
13	Natural Chip Co Tmato Hrb&Spce 175g	8934.0
15	Natural ChipCo Sea Salt & Vinegr 175g	8733.0
12	Natural Chip Compny SeaSalt175g	8331.0
18	Smiths Chip Thinly S/Cream&Onion 175g	8313.0
14	Natural ChipCo Hony Soy Chckn175g	8274.0
17	Smiths Chip Thinly CutSalt/Vinegr175g	8196.0
21	Smiths Crinkle Cut Chips Chicken 170g	8183.8
20	Smiths Crinkle Cut Chips Barbecue 170g	8125.8
22	Smiths Crinkle Cut Chips Chs&Onion170g	8111.3
23	Smiths Crinkle Cut Chips Original 170g	8001.1
8	French Fries Potato Chips 175g	7929.0
35	WW Supreme Cheese Corn Chips 200g	5390.3
32	WW Original Corn Chips 200g	5367.5
33	WW Original Stacked Chips 160g	5323.8

	PROD_NAME	TOT_SALES
34	WW Sour Cream &OnionStacked Chips 160g	5323.8
31	WW D/Style Chip Sea Salt 200g	5249.7

In [17]:
```
sum_of_sales_per_pack_size = full_data.groupby('pack_size' , as_index = False)['TOT_SALES'].sum().sort_values(by = 'TOT_SALES' , ascending = False)
sum_of_sales_per_pack_size
```

Out[17]:

	pack_size	TOT_SALES
4	175g	183172.3
1	150g	132042.1
3	170g	112396.4
0	110g	70569.8
7	380g	36367.6
6	330g	34804.2
5	200g	16007.5
2	160g	10647.6

In [14]:
```
full_data.head(4)
```

Out[14]:

	LYLTY_CARD_NBR	LIFESTAGE	PREMIUM_CUSTOMER	STORE_NBR	TXN_ID	PROD_NBR	PRO
0	1000	YOUNG SINGLES/COUPLES	Premium	1	1	5	Nat Sea
1	1003	YOUNG FAMILIES	Budget	1	4	106	Chip Ch
2	1004	OLDER SINGLES/COUPLES	Mainstream	1	5	96	WW Stack
3	1011	OLDER SINGLES/COUPLES	Mainstream	1	15	1	C

In [15]:
```
trial_data = pd.read_csv(r"C:\\Users\\LORD\\Downloads\\QVI_data.csv")
trial_data['DATE'] = pd.to_datetime(trial_data['DATE'], format = '%Y-%m-
```

```
%d',errors = 'coerce')
trial_data['month'] = trial_data['DATE'].dt.to_period('M')
trial_data.head(4)
trial_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 264834 entries, 0 to 264833
Data columns (total 13 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   LYLTY_CARD_NBR    264834 non-null  int64
 1   DATE              264834 non-null  datetime64[ns]
 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
 12  month             264834 non-null  period[M]
dtypes: datetime64[ns](1), float64(1), int64(6), object(4), period[M](1)
memory usage: 26.3+ MB
```

In [16]:

```
trial_data.groupby(['STORE_NBR' ,'month'])
['TOT_SALES'].sum().reset_index()
```

Out[16]:

	STORE_NBR	month	TOT_SALES
0	1	2018-07	206.9
1	1	2018-08	176.1
2	1	2018-09	278.8
3	1	2018-10	188.1
4	1	2018-11	192.6
...	...	...	...
3164	272	2019-02	395.5
3165	272	2019-03	442.3
3166	272	2019-04	445.1
3167	272	2019-05	314.6
3168	272	2019-06	312.1

3169 rows × 3 columns

In [17]:
```
trial_stores_all_the_time = trial_data[trial_data['STORE_NBR'].isin([77 ,
86 , 88])]

other_stores = trial_data[~trial_data['STORE_NBR'].isin([77 , 86 , 88])]
other_stores

other_monthly_sales = other_stores.groupby(['STORE_NBR' ,'month'])
['TOT_SALES'].sum().reset_index()


trial_monthly_sales = trial_stores_all_the_time.groupby(['STORE_NBR'
,'month'])['TOT_SALES'].sum().reset_index()


other_monthly_customers = other_stores.groupby(['STORE_NBR' ,'month'])
['LYLTY_CARD_NBR'].count().reset_index()

trial_monthly_customers = trial_stores_all_the_time.groupby(['STORE_NBR'
,'month'])['LYLTY_CARD_NBR'].count().reset_index()

# trial
trial_pivot_sales = trial_monthly_sales.pivot(index='month',
columns='STORE_NBR', values='TOT_SALES')
trial_pivot_customers = trial_monthly_customers.pivot(index='month',
columns='STORE_NBR', values='LYLTY_CARD_NBR')

# other
other_pivot_sales = other_monthly_sales.pivot(index='month',
columns='STORE_NBR', values='TOT_SALES')
other_pivot_customers = other_monthly_customers.pivot(index='month',
columns='STORE_NBR', values='LYLTY_CARD_NBR')
```

In [18]:
```
trial_monthly_sales =
trial_stores_all_the_time.groupby(['STORE_NBR','month'])
['TOT_SALES'].sum().reset_index()
other_monthly_sales = other_stores.groupby(['STORE_NBR','month'])
['TOT_SALES'].sum().reset_index()
```

```
trial_monthly_customers =
trial_stores_all_the_time.groupby(['STORE_NBR','month'])
['LYLTY_CARD_NBR'].count().reset_index()
other_monthly_customers = other_stores.groupby(['STORE_NBR','month'])
['LYLTY_CARD_NBR'].count().reset_index()


trial_pivot_sales = trial_monthly_sales.pivot(index='month',
columns='STORE_NBR', values='TOT_SALES')
trial_pivot_customers = trial_monthly_customers.pivot(index='month',
columns='STORE_NBR', values='LYLTY_CARD_NBR')

# other
other_pivot_sales = other_monthly_sales.pivot(index='month',
columns='STORE_NBR', values='TOT_SALES')
other_pivot_customers = other_monthly_customers.pivot(index='month',
columns='STORE_NBR', values='LYLTY_CARD_NBR')

best_controls = {}
for trial_store in trial_pivot_sales.columns:
    best_corr = -1
    best_store = None

    for other_store in other_pivot_sales.columns:
        corr_sales =
trial_pivot_sales[trial_store].corr(other_pivot_sales[other_store])
        corr_customers =
trial_pivot_customers[trial_store].corr(other_pivot_customers[other_store])


        corr = (corr_sales + corr_customers)/2

        if corr > best_corr:
            best_corr = corr
            best_store = other_store

    best_controls[trial_store] = (best_corr, best_store)
```

```
best_controls_df = pd.DataFrame(best_controls).T
best_controls_df.columns = ['correlation', 'best_control_store']
best_controls_df
```

```
C:\Users\LORD\anaconda35\Lib\site-packages\numpy\lib\function_base.py:2846: RuntimeWa
rning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
C:\Users\LORD\anaconda35\Lib\site-packages\numpy\lib\function_base.py:2705: RuntimeWa
rning: divide by zero encountered in divide
  c *= np.true_divide(1, fact)
```

Out[18]:

	correlation	best_control_store
77	0.755553	35.0
86	0.656690	147.0
88	0.715785	159.0

The best control markets are store 35 to store 77 , store 147 to store 86 and store 159 to store 88

In [19]:

```
trial_period_of_time = trial_data[(trial_data['DATE'] >= '2019-02-01') &
(trial_data['DATE'] <= '2019-04-30')]
trial_period_of_time

other_period_of_time = trial_data[(trial_data['DATE'] >= '2019-02-01') &
(trial_data['DATE'] <= '2019-04-30')]
```

In [21]:

```
trial_stores_trial =
trial_period_of_time[trial_period_of_time['STORE_NBR'].isin([77 , 86 ,
88])]
trial_stores_trial.head(4)

control_stores =
other_period_of_time[trial_period_of_time['STORE_NBR'].isin([35 , 147 ,
159])]
control_stores.head(4)

control_stores.to_csv(r"C:\\Users\\LORD\\Downloads\\control_stores_exp.csv")
trial_stores_trial.to_csv(r"C:\\Users\\LORD\\Downloads\\trial_stores_exp.csv
```

In [71]:
```
sum_of_sales_trail_stores =
trial_stores_trial.groupby('STORE_NBR',as_index = False)
['TOT_SALES'].sum().sort_values(by = 'TOT_SALES' , ascending = False)

sum_of_sales_control_stores = control_stores.groupby('STORE_NBR',as_index
= False)['TOT_SALES'].sum().sort_values(by ='TOT_SALES' ,ascending =
False)

print(sum_of_sales_trail_stores)
print(sum_of_sales_control_stores)
```

```
   STORE_NBR  TOT_SALES
2         88     4286.8
1         86     2788.2
0         77      777.0
   STORE_NBR  TOT_SALES
1        147     2629.6
0         35      417.2
2        159      116.1
```

The trail stores are making more money than the control stores that means this experiment is workong

Now you can do it in the real world

In [75]:
```
trail_stores_count_of_cutomers =
trial_stores_trial.groupby('STORE_NBR',as_index = False)
['TXN_ID'].count().sort_values(by ='TXN_ID' ,ascending = False)

control_stores_count_of_cutomers =
control_stores.groupby('STORE_NBR',as_index = False)
['TXN_ID'].count().sort_values(by ='TXN_ID' ,ascending = False)

print(trail_stores_count_of_cutomers)
print(control_stores_count_of_cutomers)
```

```
   STORE_NBR  TXN_ID
2         88     486
1         86     408
0         77     148
   STORE_NBR  TXN_ID
1        147     375
0         35     111
2        159      21
```

In [48]:
```
trial_stores.groupby('STORE_NBR',as_index = False)
['LYLTY_CARD_NBR'].nunique().sort_values(by ='LYLTY_CARD_NBR' ,ascending =
False)
```

Out[48]:

	STORE_NBR	LYLTY_CARD_NBR
2	88	209
1	86	169
0	77	91

In [25]:
```
final_data_sim_job_2 = pd.concat([trial_stores_trial , control_stores]
,ignore_index = True )
final_data_sim_job_2.to_csv(r"C:\\Users\\LORD\\Downloads\\final_data_sim_job
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: