

Faire Direct Analytics Case

Problem: The number of new retailers that place their first confirmed order has gone down.



Goal: Investigate cause of this decline

Questions

- What are some of your hypotheses about the underlying cause of this decline?
- What are some product ideas you have about how to bring this number back up?

Approach: Retailers come to the Brand's Faire Direct Page to stockup or add new line of items to store inventory.

I have designed a map for 'Retailer Journey on Faire Direct Acquisition Strategy' to understand Retailer Behaviour (Refer Journey Map)

Hypothesis - What are the possible causes

Awareness Stage Issue

- The overall new leads(retailers) clicks have declined

Consideration Stage Issue

- New Retailers might be facing difficulty in deciding if they want to try new line items

Acquisition Stage Issue

- The overall conversion rate on Brands Faire Direct Sign-up Page page might have declined
- Order confirmation rates might have declined

You don't have enough leads coming at the top of your current funnel. Your website is getting a lot of visitors, but many of them aren't converting into customers. You have plenty of users who sign up for free trials, but many of them aren't converting into paying customers.

1) Importing Data

In [1]:

```
# Importing necessary Libraries
import numpy as np
import pandas as pd
import pandasql as ps
import datetime as dt
```

```
from datetime import datetime, timedelta
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
```

In [2]:

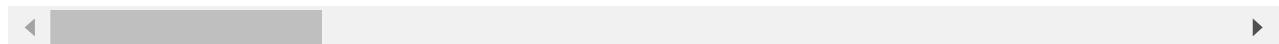
```
# reading dataset
df = pd.read_csv('elevate_analytics_case_data.csv')
df.head()
```

C:\Users\sushm\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3146: DtypeWarning: Columns (6) have mixed types.Specify dtype option on import or set low_memory=False.

Out[2]:

	brand_id	retailer_id	brand_relationship_id	brand_relationship_created_at	brand_relationship_confirmed_at
0	20940	NaN	262526	2019-06-16 23:59:30	
1	20940	NaN	262525	2019-06-16 23:58:26	
2	19207	NaN	262524	2019-06-16 23:58:08	
3	20940	NaN	262523	2019-06-16 23:47:47	
4	20940	NaN	262522	2019-06-16 23:46:54	

5 rows × 21 columns



2) Transforming data

In [3]:

```
# converting data types

# from object to date

df[['brand_relationship_confirmed_at',
     'brand_relationship_created_at',
     'retailer_signed_up_at',
     'retailer_placed_first_order_at',
     'retailer_placed_first_confirmed_order_at',
     'brand_first_active_at',
     'email_sent_at',
     'brand_adopted_elevate_at']] = df[['brand_relationship_confirmed_at',
                                         'brand_relationship_created_at',
                                         'retailer_signed_up_at',
                                         'retailer_placed_first_order_at',
                                         'retailer_placed_first_confirmed_order_at',
                                         'brand_first_active_at',
                                         'email_sent_at',
                                         'brand_adopted_elevate_at']].apply(pd.t

# from datetime to date

df[['brand_relationship_confirmed_at',
     'brand_relationship_created_at',
     'retailer_signed_up_at',
     'retailer_placed_first_order_at',
```

```
'retailer_placed_first_confirmed_order_at',
'brand_first_active_at',
'email_sent_at',
'brand_adopted_elevate_at']] = df[['brand_relationship_confirmed_at',
                                    'brand_relationship_created_at',
                                    'retailer_signed_up_at',
                                    'retailer_placed_first_order_at',
                                    'retailer_placed_first_confirmed_order',
                                    'brand_first_active_at',
                                    'email_sent_at',
                                    'brand_adopted_elevate_at']].apply(lambda
```

In [4]: #Power maker
df['power_retailer_flag'] = np.where(df['power_retailer_converted_at'].isnull() == False, 1, 0)

In [5]: df['power_retailer_flag'].value_counts()

Out[5]: no 215167
yes 3371
Name: power_retailer_flag, dtype: int64

In [6]: df.retailer_id.nunique()

Out[6]: 32873

In [7]: # get only unique retailer id
df1 = df.query("power_retailer_flag == 'yes'")
df1.shape[0]

Out[7]: 3371

In [8]: df.retailer_id.nunique()

Out[8]: 32873

In [9]: df1.retailer_id.nunique()

Out[9]: 3363

In [10]: df1.retailer_id.nunique()/df.retailer_id.nunique() * 100

Out[10]: 10.230280169135765

**Out of all the total retailers(32873), only 10% (3363) comprise of power retailers.
(GTM: we need to increase this number)****

In [11]: #email sent
#internal / external

df['email_flag'] = np.where(df['outgoing_email_id'].isnull() == False, 'internal', 'external')
df['email_flag'].value_counts()

Out[11]: external 159380
internal 59158
Name: email_flag, dtype: int64

In [12]: df.head()

```
#"Enterprise Maker" are accounts managed by the Enterprise team and have at Least >500
#"Maker Success" are managed by our success team and focus on brands with >50 stockists
#"Control" is a control group for Maker Success.
#"Small maker or other" are all of the other brands that are not actively managed by on
```

Out[12]:

	brand_id	retailer_id	brand_relationship_id	brand_relationship_created_at	brand_relationship_confirmation_reason
0	20940	NaN	262526	2019-06-16	SIGNED_UP
1	20940	NaN	262525	2019-06-16	SIGNED_UP
2	19207	NaN	262524	2019-06-16	SIGNED_UP
3	20940	NaN	262523	2019-06-16	SIGNED_UP
4	20940	NaN	262522	2019-06-16	SIGNED_UP

5 rows × 23 columns

3) Reusable Functions

In [13]:

```
# Reusable function to get week start date
def week_start_date(datetime):
    return datetime.date() - timedelta(days=datetime.date().weekday())

# direction to use: faire_direct['brand_adopted_elevate_at'].apply(week_start_date)
```

I am looking at only those new retailer leads where the confirmation reason for joining Faire is through Faire Direct

In [30]:

```
# filtering
faire_direct = df.loc[(df['confirmation_reason'] == 'SIGNED_UP') | (df['confirmation_reason'] == 'FIRST_ORDER_FOR_BRAND')]
faire_direct['confirmation_reason'].unique()
```

Out[30]:

```
array(['SIGNED_UP', 'FIRST_ORDER_FOR_BRAND'], dtype=object)
```

In [31]:

```
faire_direct.retailer_id.nunique()
```

Out[31]:

```
32873
```

In [32]:

```
faire_direct.head()
```

Out[32]:

	brand_id	retailer_id	brand_relationship_id	brand_relationship_created_at	brand_relationship_confirmation_reason
14	5663	145941.0	262510	2019-06-16	SIGNED_UP
28	26821	145936.0	262495	2019-06-16	SIGNED_UP
31	17454	145934.0	262492	2019-06-16	SIGNED_UP
45	5072	145929.0	262478	2019-06-16	SIGNED_UP
63	13695	145923.0	262457	2019-06-16	SIGNED_UP

5 rows × 23 columns

4) Acquisition Funnel

From the prompt, we can see that Faire Direct Acquisition strategy is loosing new retailers, so now we need to identify blockage points in Acquisition Funnel

```
In [72]: df = pd.read_csv("elevate_analytics_case_data.csv")
df.head()
```

C:\Users\sushm\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3146: DtypeWarning: Columns (6) have mixed types. Specify dtype option on import or set low_memory=False.

```
has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

	brand_id	retailer_id	brand_relationship_id	brand_relationship_created_at	brand_relationship_confirmed_at
0	20940	NaN	262526	2019-06-16 23:59:30	
1	20940	NaN	262525	2019-06-16 23:58:26	
2	19207	NaN	262524	2019-06-16 23:58:08	
3	20940	NaN	262523	2019-06-16 23:47:47	
4	20940	NaN	262522	2019-06-16 23:46:54	

5 rows × 21 columns

```
In [73]: df_req = df[['brand_id', 'retailer_id', 'brand_relationship_id', 'brand_relationship_cr
                     'normalized_referer', 'retailer_signed_up_at', 'retailer_placed_first_order_
                     'retailer_placed_first_confirmed_order_at', 'email_sent_at', 'power_retailer_
                     'brand_first_active_at', 'brand_adopted_elevate_at', 'account_owner']]
```

```
In [74]: # converting data types
```

```
# from object to date
```

```
df_req[['brand_relationship_confirmed_at',
         'brand_relationship_created_at',
         'retailer_signed_up_at',
         'retailer_placed_first_order_at',
         'retailer_placed_first_confirmed_order_at',
         'email_sent_at',
         'power_retailerConverted_at',
         'brandFirstActive_at',
         'brandAdoptedElevate_at']] = df_req[['brand_relationship_confirmed_at',
                                                'brand_relationship_created_at',
                                                'retailer_signed_up_at',
                                                'retailer_placed_first_order_at',
                                                'retailer_placed_first_confirmed_order_
                                                'email_sent_at',
                                                'power_retailerConverted_at',
                                                'brandFirstActive_at',
                                                'brandAdoptedElevate_at']].apply(pd.t
```

```
# from datetime to date

df_req[['brand_relationship_confirmed_at',
        'brand_relationship_created_at',
        'retailer_signed_up_at',
        'retailer_placed_first_order_at',
        'retailer_placed_first_confirmed_order_at',
        'email_sent_at',
        'power_retailer_CONVERTED_AT',
        'brand_first_active_at',
        'brand_adopted_elevate_at']] = df_req[['brand_relationship_confirmed_at',
                                                'brand_relationship_created_at',
                                                'retailer_signed_up_at',
                                                'retailer_placed_first_order_at',
                                                'retailer_placed_first_confirmed_order_at',
                                                'email_sent_at',
                                                'power_retailer_CONVERTED_AT',
                                                'brand_first_active_at',
                                                'brand_adopted_elevate_at']].apply(lambda
```

C:\Users\sushm\anaconda3\lib\site-packages\pandas\core\frame.py:3065: 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

```
self[k1] = value[k2]
```

Journey from order placed to order confirmed

```
In [75]: df_acquired_retailers = df_req.query('confirmation_reason == "SIGNED_UP" or confirmation_reason == "REREFERRED_BY_FRIEND"')
df_acquired_retailers = df_acquired_retailers.loc[(df_acquired_retailers['retailer_placed_first_order_at'] < '2019-06-17') &
                                                    (df_acquired_retailers['retailer_placed_first_confirmed_order_at'] >= '2019-06-17')]
```

```
In [76]: df_acquired_retailers.retailer_id.nunique()
```

```
Out[76]: 7378
```

Journey from signing up to placing order

```
In [77]: df_placed_orders = pd.merge(df_acquired_retailers, df_req, on='retailer_placed_first_order_at')
```

```
In [78]: df_placed_orders.retailer_placed_first_order_at_y.min(), df_placed_orders.retailer_placed_first_order_at_x.max()
```

```
Out[78]: (Timestamp('2018-08-16 00:00:00'), Timestamp('2019-06-16 00:00:00'))
```

```
In [79]: df_placed_orders_acq = df_req.query('confirmation_reason == "SIGNED_UP" or confirmation_reason == "REREFERRED_BY_FRIEND"')
df_placed_orders_acq = df_placed_orders_acq.loc[(df_placed_orders_acq['retailer_placed_first_order_at'] < '2019-06-17') &
                                                    (df_placed_orders_acq['retailer_placed_first_order_at'] >= '2018-08-16')]
```

```
In [80]: df_placed_orders_acq.retailer_id.nunique()
```

```
Out[80]: 15389
```

Journey from relationship created (click) to sign - up

```
In [81]: df_signups = pd.merge(df_acquired_retailers, df_req, on='retailer_placed_first_confirmed')

In [82]: df_signups.brand_relationship_confirmed_at_y.min(), df_signups.brand_relationship_confirmed_at_x.max()

Out[82]: (Timestamp('2018-04-27 00:00:00'), Timestamp('2019-06-16 00:00:00'))

In [83]: df_signups_acq = df_req.query('confirmation_reason == "SIGNED_UP" or confirmation_reason == "NOT_SENT"')
df_signups_acq = df_signups_acq.loc[(df_signups_acq['brand_relationship_confirmed_at'] < '2019-06-17') &
                                      (df_signups_acq['brand_relationship_confirmed_at'] >= '2018-04-27')]

In [84]: df_signups_acq.retailer_id.nunique()

Out[84]: 32826
```

Journey at top of the funnel - incoming clicks

```
In [85]: df_clicks = pd.merge(df_acquired_retailers, df_req, on='retailer_placed_first_confirmed')

In [86]: df_clicks.brand_relationship_created_at_y.min(), df_clicks.brand_relationship_created_at_x.max()

Out[86]: (Timestamp('2018-04-27 00:00:00'), Timestamp('2019-06-16 00:00:00'))

In [87]: df_clicks_acq = df_req.query('confirmation_reason == "SIGNED_UP" or confirmation_reason == "NOT_SENT"')
df_clicks_acq = df_clicks_acq.loc[(df_clicks_acq['brand_relationship_created_at'] < '2019-06-17') &
                                      (df_clicks_acq['brand_relationship_created_at'] >= '2018-04-27')]

In [88]: df_clicks_acq.retailer_id.nunique()

Out[88]: 32850
```

4.1) Funnel Analysis

```
In [93]: # creating email flags
df_acquired_retailers['email_flag'] = np.where(df_acquired_retailers['email_sent_at'].isnull(), 'not sent', 'sent')
df_acquired_retailers['order_confirmed_week_start'] = df_acquired_retailers['retailer_purchased_at'].dt.week

In [103...]: tmp1 = df_acquired_retailers.groupby(['order_confirmed_week_start', 'email_flag'], as_index=False)
# visualizing the output
plt.figure(figsize=(13,6.5))
sns.set(style='darkgrid', font_scale=1.2, rc={"lines.linewidth":1.4});
x1 =tmp1.query("email_flag == 'sent'")['order_confirmed_week_start']
y1 =tmp1.query("email_flag == 'sent'")['No of Retailers Acquired']
g = sns.barplot(x1, y1, color= 'green', label = 'sent', alpha = 0.5)

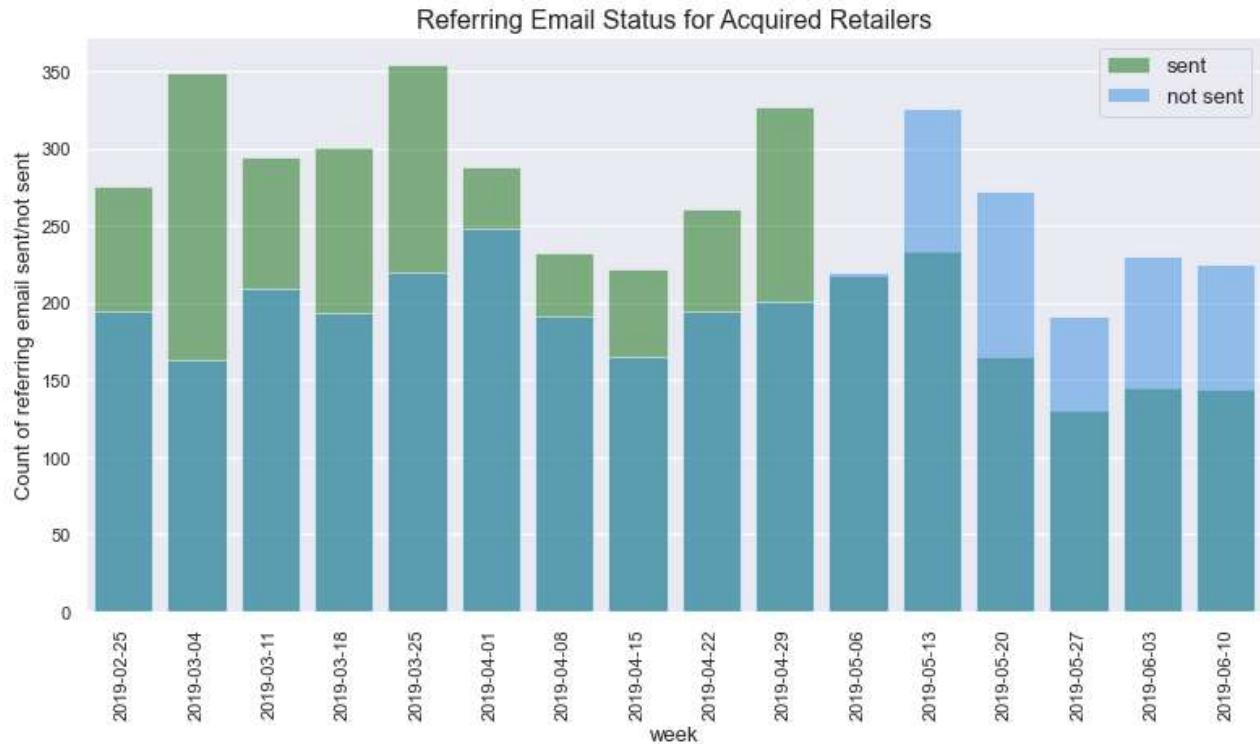
x2 =tmp1.query("email_flag == 'not sent'")['order_confirmed_week_start']
y2 =tmp1.query("email_flag == 'not sent'")['No of Retailers Acquired']
g = sns.barplot(x2, y2, color= 'dodgerblue',label = 'not sent', alpha = 0.5)

g.set_title("Referring Email Status for Acquired Retailers", fontsize = 16)
```

```

g.set_xlabel("week", fontsize = 13)
g.set_ylabel("Count of referring email sent/not sent", fontsize = 13)
plt.xticks(rotation=90, fontsize=11)
plt.yticks(fontsize=11)
g.legend();

```



```
In [127...]: # check difference in brand_relationship_confirmed_at_y & retailer_placed_first_order_at
df_signups_acq['placed_flag'] = np.where(df_signups_acq['retailer_placed_first_order_at']
```

```
In [129...]: df_signups_acq['signup_week_start'] = df_signups_acq['brand_relationship_confirmed_at']
```

```
In [130...]: tmp1 = df_signups_acq.loc[(df_signups_acq['brand_relationship_created_at'] < '2019-06-17') &
                               (df_signups_acq['brand_relationship_created_at'] >= '2019-04-08')]
tmp1 = tmp1.groupby(['signup_week_start', 'placed_flag'], as_index=False)[['retailer_id']]
tmp1.tail(20)
```

```
Out[130...]: signup_week_start  placed_flag  No of Retailers
```

12	2019-04-08	not_placed	363
13	2019-04-08	placed	429
14	2019-04-15	not_placed	515
15	2019-04-15	placed	453
16	2019-04-22	not_placed	612
17	2019-04-22	placed	566
18	2019-04-29	not_placed	820
19	2019-04-29	placed	770
20	2019-05-06	not_placed	585

	signup_week_start	placed_flag	No of Retailers
21	2019-05-06	placed	509
22	2019-05-13	not_placed	850
23	2019-05-13	placed	775
24	2019-05-20	not_placed	543
25	2019-05-20	placed	458
26	2019-05-27	not_placed	341
27	2019-05-27	placed	319
28	2019-06-03	not_placed	410
29	2019-06-03	placed	348
30	2019-06-10	not_placed	477
31	2019-06-10	placed	343

In [138...]

```
'''# visualizing the output
plt.figure(figsize=(13,6.5))

sns.set(style='darkgrid', font_scale=1.2, rc={"lines.linewidth":1.4});

x1 =tmp1.query("placed_flag == 'not placed'")['signup_week_start']
y1 =tmp1.query("placed_flag == 'not placed'")['No of Retailers']
g = sns.barplot(x2, y2, color= 'white',label = 'not placed', alpha = 1)

x2 =tmp1.query("placed_flag == 'placed'")['signup_week_start']
y2 =tmp1.query("placed_flag == 'placed'")['No of Retailers']
g = sns.barplot(x2, y2, color= 'black',label = 'placed', alpha = 0.25)

g.set_title("Trend Analysis - Signup vs orders placed", fontsize = 16)
g.set_xlabel("week", fontsize = 13)
g.set_ylabel("Number of Retailers", fontsize = 13)
plt.xticks(rotation=90, fontsize=11)
plt.yticks(fontsize=11)
g.legend();'''
```

Out[138...]

```
'# visualizing the output\nplt.figure(figsize=(13,6.5))\n\nsns.set(style='darkgrid', font_scale=1.2, rc={"lines.linewidth":1.4});\n\nx1 =tmp1.query("placed_flag == \'not placed\'")['signup_week_start']\ny1 =tmp1.query("placed_flag == \'not placed\'")['No of Retailers']\ng = sns.barplot(x2, y2, color= 'white',label = 'not placed', alpha = 1)\n\nx2 =tmp1.query("placed_flag == \'placed\'")['signup_week_start']\ny2 =tmp1.query("placed_flag == \'placed\'")['No of Retailers']\ng = sns.barplot(x2, y2, color= 'black',label = 'placed', alpha = 0.25)\n\nng.set_title("Trend Analysis - Signup vs orders placed", fontsize = 16)\nng.set_xlabel("week", fontsize = 13)\nng.set_ylabel("Number of Retailers", fontsize = 13)\nplt.xticks(rotation=90, fontsize=11)\nplt.yticks(fontsize=11)\nng.legend();'
```

There are lot of retailers who signed up in 2018 but placed their order in 2019. So to understand the decline trend in 2019, we will focus only on the signups from 2019. Below are some of the insights:

- In start of 2019, there is 30% decline from 18 Feb and then sudden spike in April as compared to last week of March 2019 and first week of April 2019.

- After that the numbers fluctuated.
- May also observed dropping numbers (56% decline for middle of May to 10th of June)

```
In [140...]: # orders placed to order confirmation analysis
df_placed_orders_acq['order_confirmation_flag'] = np.where(df_placed_orders_acq['retail
.isnull() == False, "confirmed", "not confirmed"]

In [141...]: df_placed_orders_acq['order_placed_week_start'] = df_placed_orders_acq['retailer_placed
week_start']

In [143...]: df_placed_orders_acq.to_excel("placed_orders_confirm.xlsx")

In [142...]: '''tmp2 = df_placed_orders_acq.loc[(df_placed_orders_acq['brand_relationship_created_at
< '2019-06-17') &
(df_placed_orders_acq['brand_relationship_created_at'] >=
tmp2 = df_placed_orders_acq.groupby(['order_placed_week_start', 'order_confirmation_flag
tmp2.tail(20)'''
```

Out[142...]:

	order_placed_week_start	order_confirmation_flag	No of Retailers
66	2019-04-08	confirmed	420
67	2019-04-08	not confirmed	79
68	2019-04-15	confirmed	383
69	2019-04-15	not confirmed	110
70	2019-04-22	confirmed	457
71	2019-04-22	not confirmed	150
72	2019-04-29	confirmed	529
73	2019-04-29	not confirmed	272
74	2019-05-06	confirmed	431
75	2019-05-06	not confirmed	129
76	2019-05-13	confirmed	557
77	2019-05-13	not confirmed	302
78	2019-05-20	confirmed	435
79	2019-05-20	not confirmed	94
80	2019-05-27	confirmed	321
81	2019-05-27	not confirmed	89
82	2019-06-03	confirmed	372
83	2019-06-03	not confirmed	86
84	2019-06-10	confirmed	361
85	2019-06-10	not confirmed	124

In [148...]: # visualizing the output

```

plt.figure(figsize=(13,6.5))

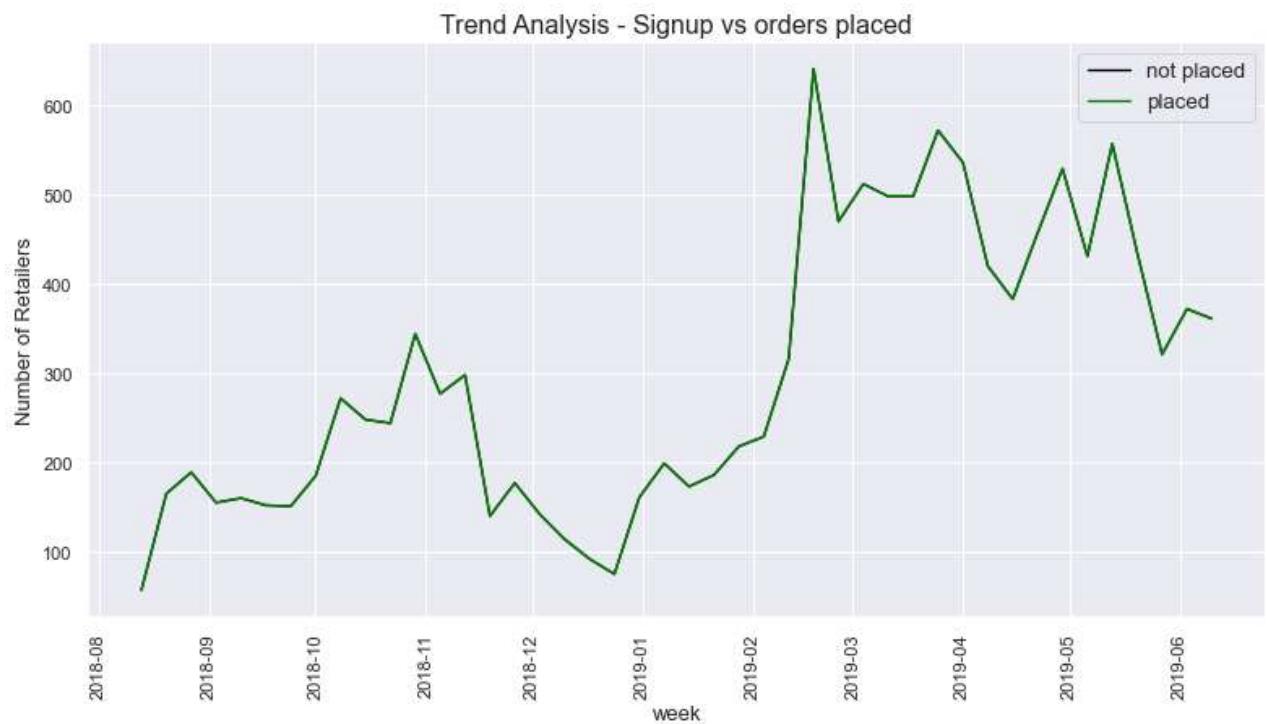
sns.set(style='darkgrid', font_scale=1.2, rc={"lines.linewidth":1.4});

x1 =tmp2.query("order_confirmation_flag == 'not confirmed'")['order_placed_week_start']
y1 =tmp2.query("order_confirmation_flag == 'not confirmed'")['No of Retailers']
g = sns.lineplot(x2, y2, color= 'black',label = 'not placed', alpha = 1)

x2 =tmp2.query("order_confirmation_flag == 'confirmed'")['order_placed_week_start']
y2 =tmp2.query("order_confirmation_flag == 'confirmed'")['No of Retailers']
g = sns.lineplot(x2, y2, color= 'green',label = 'placed', alpha = 1)

g.set_title("Trend Analysis - Signup vs orders placed", fontsize = 16)
g.set_xlabel("week", fontsize = 13)
g.set_ylabel("Number of Retailers", fontsize = 13)
plt.xticks(rotation=90, fontsize=11)
plt.yticks(fontsize=11)
g.legend();

```



In []:

In []:

Bottom of the funnel: What is the weekly trend for the number of new retailers that place their first confirmed order that was confirmed by Faire

In [34]: `#Filter data between date range '2019-06-17' and '2019-02-25' date range
faire_direct = faire_direct.loc[(faire_direct['retailer_placed_first_confirmed_order_at`

In [35]: `# get week start date for retailer_placed_first_confirmed_order_at
faire_direct['confirmed_first_order_at_week'] = faire_direct['retailer_placed_first_con
grouping
stage1= faire_direct.groupby('confirmed_first_order_at_week',as_index=False)[['retailer_
calculating percent change
stage1['week_over_week_variance_percent'] = stage1['No of Retailers Acquired'].pct Chan`

In [36]: stage1

Out[36]:

	confirmed_first_order_at_week	No of Retailers Acquired	week_over_week_variance_percent
0	2019-02-25	469	NaN
1	2019-03-04	512	0.091684
2	2019-03-11	503	-0.017578
3	2019-03-18	494	-0.017893
4	2019-03-25	574	0.161943
5	2019-04-01	536	-0.066202
6	2019-04-08	423	-0.210821
7	2019-04-15	387	-0.085106
8	2019-04-22	455	0.175711
9	2019-04-29	528	0.160440
10	2019-05-06	438	-0.170455
11	2019-05-13	559	0.276256
12	2019-05-20	437	-0.218247
13	2019-05-27	321	-0.265446
14	2019-06-03	375	0.168224
15	2019-06-10	369	-0.016000

In [37]:

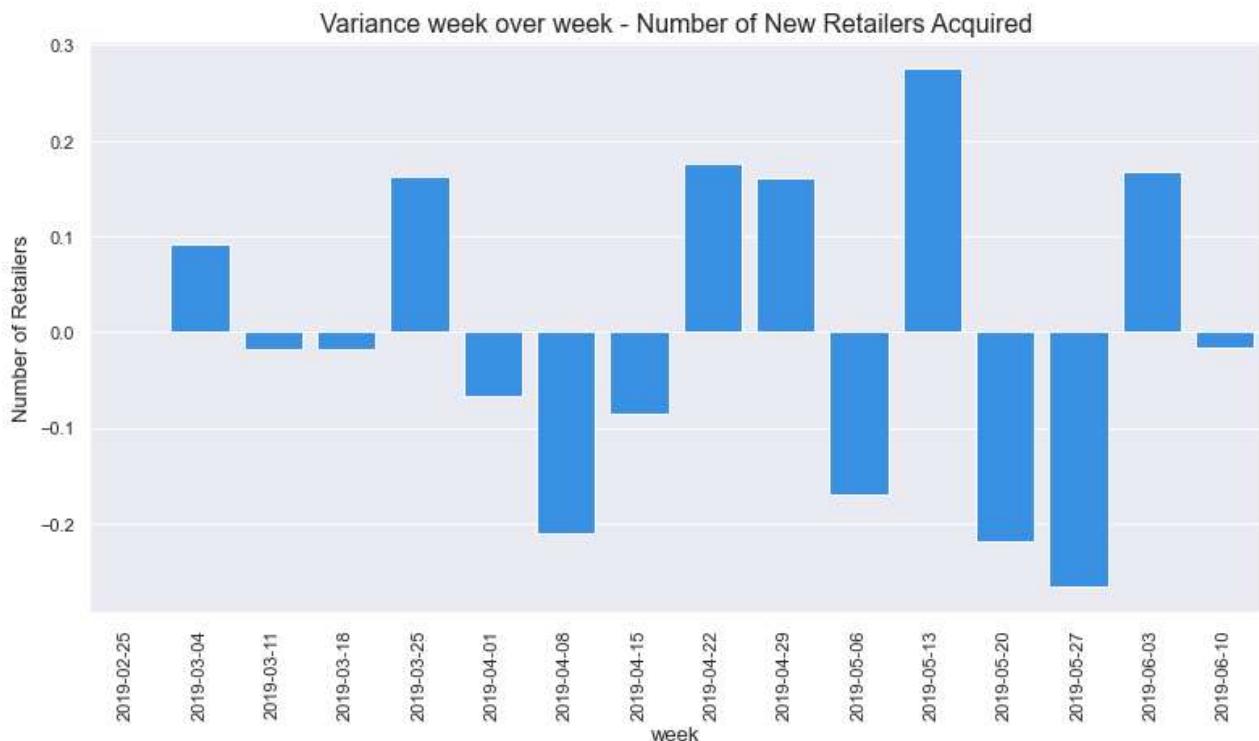
```
# visualizing the output
plt.figure(figsize=(13,6.5))
sns.set(style='darkgrid', font_scale=1.2)
x = x=stage1['confirmed_first_order_at_week']
y = y=stage1['No of Retailers Acquired']
g = sns.barplot(x, y, color= 'lightgreen')
g.set_title("# New Faire Direct Retailers Acquired", fontsize = 16)
g.set_xlabel("week", fontsize = 13)
g.set_ylabel("Number of Retailers", fontsize = 13)
plt.xticks(rotation=90, fontsize=11)
plt.yticks(fontsize=11);
```

New Faire Direct Retailers Acquired



In [38]:

```
# visualizing the output
plt.figure(figsize=(13,6.5))
sns.set(style='darkgrid', font_scale=1.2)
x = x=stage1['confirmed_first_order_at_week']
y = y=stage1['week_over_week_variance_percent']
g = sns.barplot(x, y, color= 'dodgerblue')
g.set_title("Variance week over week - Number of New Retailers Acquired", fontsize = 16)
g.set_xlabel("week", fontsize = 13)
g.set_ylabel("Number of Retailers", fontsize = 13)
plt.xticks(rotation=90, fontsize=11)
plt.yticks(fontsize=11);
```



Analyst notes:

- From the acquisition trend, we can see that the number of retailers acquired have dropped.
- After March 2019, there is a significant drop in first 3 weeks of April 2019
- Although there is a significant rise (28%) in these numbers in second week of May 2019, the numbers again start dropping by 22% and 27% in the last two weeks of May 2019 respectively
- There are high chances that the drop initiated from 1st three weeks of April 2019 and then 3rd week of May 2019

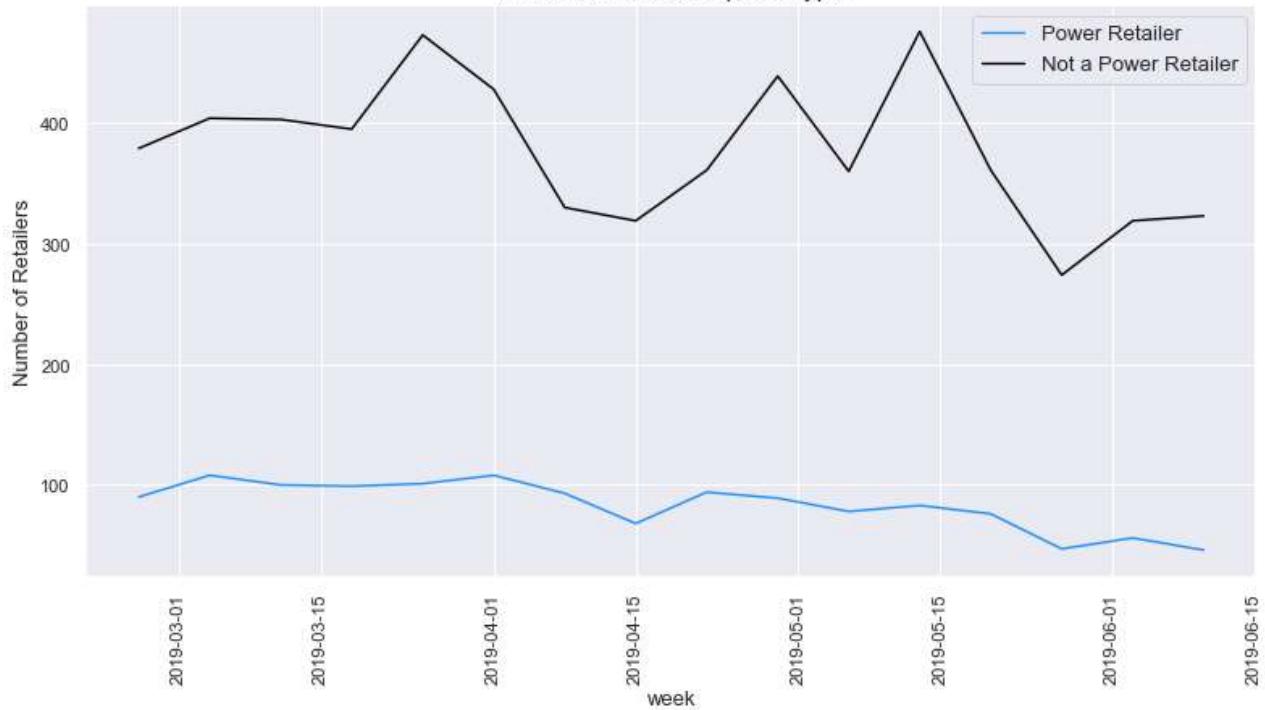
```
In [39]: # Slicing by retailer and email and team type
# email_flag, account_owner, power_retailer_flag
a = faire_direct.groupby(['confirmed_first_order_at_week','power_retailer_flag'],as_index=True)
```

```
In [40]: # visualizing the output
plt.figure(figsize=(13,6.5))
sns.set(style='darkgrid', font_scale=1.2, rc={"lines.linewidth":1.4});
x1 = a.query("power_retailer_flag == 'yes'")['confirmed_first_order_at_week']
y1 = a.query("power_retailer_flag == 'yes'")['No of Retailers Acquired']
g = sns.lineplot(x1, y1, color= 'dodgerblue', label = 'Power Retailer')

x2 = a.query("power_retailer_flag == 'no'")['confirmed_first_order_at_week']
y2 = a.query("power_retailer_flag == 'no'")['No of Retailers Acquired']
g = sns.lineplot(x2, y2, color= 'black',label = 'Not a Power Retailer')

g.set_title("# New Retailers Acquired type", fontsize = 16)
g.set_xlabel("week", fontsize = 13)
g.set_ylabel("Number of Retailers", fontsize = 13)
plt.xticks(rotation=90, fontsize=11)
plt.yticks(fontsize=11)
g.legend();
```

New Retailers Acquired type



Lets go one step back to see if number of orders placed have similar heavy drop in April2019 and May2019

```
In [41]: # filtering date range
faire_direct = faire_direct.loc[(faire_direct['retailer_placed_first_order_at'] < '2019-05-15') & (faire_direct['retailer_placed_first_order_at'] > '2019-03-15')]
```

```
In [42]: # get week start date
faire_direct['order_placed_at_week'] = faire_direct['retailer_placed_first_order_at'].apply(lambda x: x.strftime('%Y-%m-%d')[:10])
# grouping
stage2 = faire_direct.groupby('order_placed_at_week', as_index=False)[['brand_id']].count()
# calculating percent change
stage2['week_over_week_variance_percent'] = stage2['No of Retailers'].pct_change();
```

```
In [43]: # convert to date datatype
#faire_direct['order_placed_at_week'] = faire_direct['order_placed_at_week'].apply(pd.to_datetime)
```

```
In [44]: stage2.head()
```

```
Out[44]:
```

	order_placed_at_week	No of Retailers	week_over_week_variance_percent
0	2019-02-25	470	NaN
1	2019-03-04	512	0.089362
2	2019-03-11	498	-0.027344
3	2019-03-18	498	0.000000
4	2019-03-25	572	0.148594

```
In [45]: stage2['No of Retailers'].sum()
```

```
Out[45]: 7346
```

```
In [46]: # visualizing the output
plt.figure(figsize=(13,6.5))
sns.set(style='darkgrid', font_scale=1.2)
x = x=stage2['order_placed_at_week']
y = y=stage2['week_over_week_variance_percent']
g = sns.barplot(x, y, color= 'dodgerblue')
g.set_title("# Orders Placed", fontsize = 16)
g.set_xlabel("week", fontsize = 13)
g.set_ylabel("Number of Retailers", fontsize = 13)
plt.xticks(rotation=90, fontsize=11)
plt.yticks(fontsize=11);
```



Lets go one step back to see if signups have similar heavy drop in April 2019 and May 2019

```
In [47]: # filtering date range
faire_direct = faire_direct.loc[(faire_direct['brand_relationship_confirmed_at'] < '201
```

```
In [48]: # get week start date
faire_direct['sign_up_at_week'] = faire_direct['brand_relationship_confirmed_at'].apply
# grouping
stage3 = faire_direct.groupby('sign_up_at_week', as_index=False)[['brand_id']].count().re
# calculating percent change
stage3['week_over_week_variance_percent'] = stage3['No of Retailers'].pct_change();
```

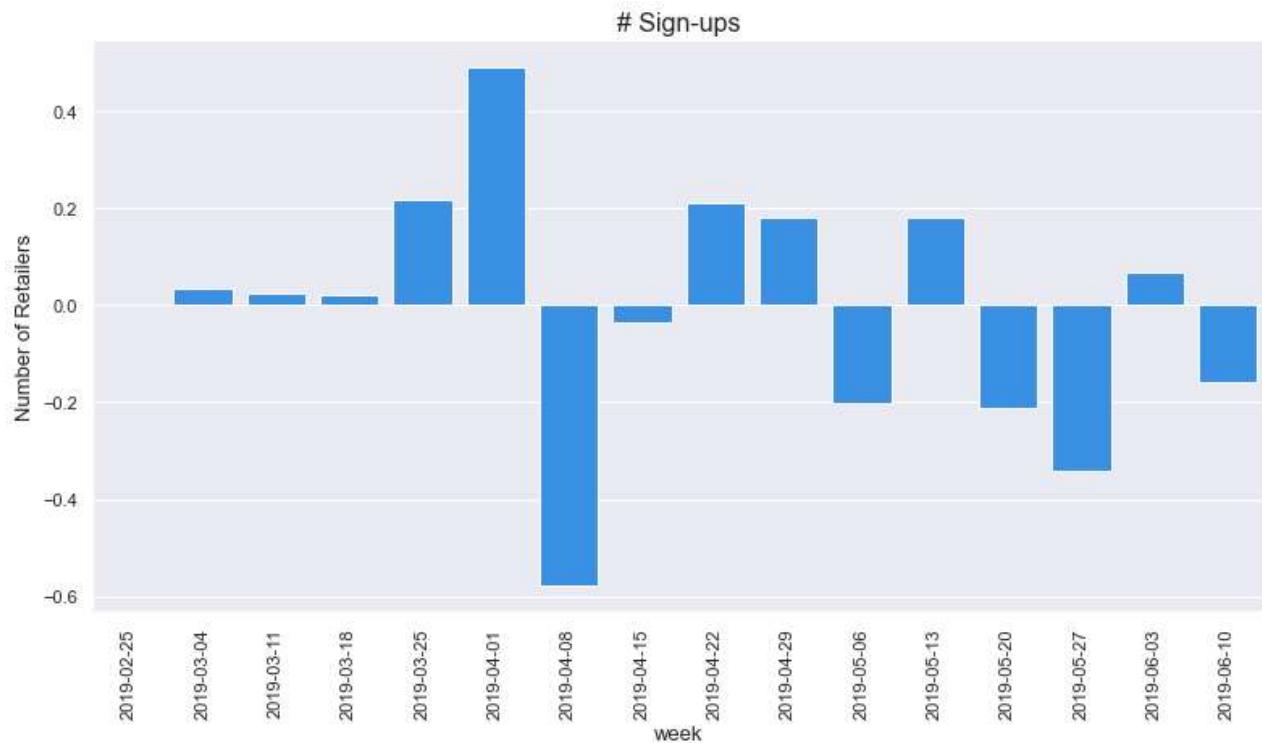
```
In [49]: stage3
```

	sign_up_at_week	No of Retailers	week_over_week_variance_percent
0	2019-02-25	427	NaN
1	2019-03-04	441	0.032787
2	2019-03-11	452	0.024943

	sign_up_at_week	No of Retailers	week_over_week_variance_percent
3	2019-03-18	461	0.019912
4	2019-03-25	561	0.216920
5	2019-04-01	837	0.491979
6	2019-04-08	354	-0.577061
7	2019-04-15	341	-0.036723
8	2019-04-22	413	0.211144
9	2019-04-29	488	0.181598
10	2019-05-06	390	-0.200820
11	2019-05-13	461	0.182051
12	2019-05-20	364	-0.210412
13	2019-05-27	240	-0.340659
14	2019-06-03	256	0.066667
15	2019-06-10	215	-0.160156

In [50]:

```
# visualizing the output
plt.figure(figsize=(13,6.5))
sns.set(style='darkgrid', font_scale=1.2)
x = stage3['sign_up_at_week']
y = stage3['week_over_week_variance_percent']
g = sns.barplot(x, y, color= 'dodgerblue')
g.set_title("# Sign-ups", fontsize = 16)
g.set_xlabel("week", fontsize = 13)
g.set_ylabel("Number of Retailers", fontsize = 13)
plt.xticks(rotation=90, fontsize=11)
plt.yticks(fontsize=11);
```



```
In [51]: stage3['No of Retailers'].sum()
```

```
Out[51]: 6701
```

Lets go one step back to see if new visits 7378 (Retailer) clicks have similar heavy drop in April2019 and May2019

```
In [52]: # filtering date range
faire_direct = faire_direct.loc[(faire_direct['brand_relationship_created_at'] < '2019-
```

```
In [53]: faire_direct.retailer_id.nunique()
```

```
Out[53]: 6630
```

```
In [54]: # get week start date
faire_direct['clicks_at_week'] = faire_direct['brand_relationship_created_at'].apply(we
# grouping
stage4 = faire_direct.groupby('clicks_at_week',as_index=False)[['retailer_id']].count()
# calculating percent change
stage4['week_over_week_variance_percent'] = stage4['No of Retailers'].pct_change();
```

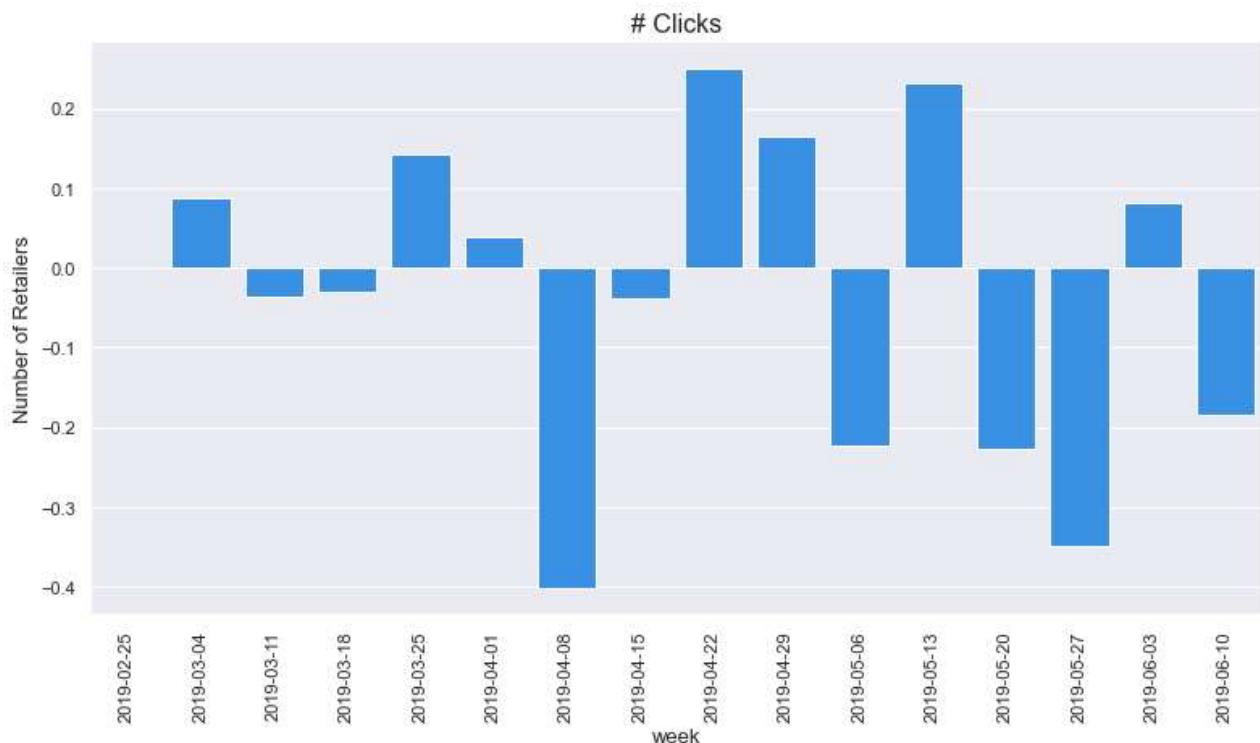
```
In [55]: stage4
```

	clicks_at_week	No of Retailers	week_over_week_variance_percent
0	2019-02-25	481	NaN
1	2019-03-04	523	0.087318
2	2019-03-11	504	-0.036329
3	2019-03-18	489	-0.029762
4	2019-03-25	559	0.143149

	clicks_at_week	No of Retailers	week_over_week_variance_percent
5	2019-04-01	581	0.039356
6	2019-04-08	348	-0.401033
7	2019-04-15	335	-0.037356
8	2019-04-22	419	0.250746
9	2019-04-29	488	0.164678
10	2019-05-06	379	-0.223361
11	2019-05-13	467	0.232190
12	2019-05-20	361	-0.226981
13	2019-05-27	235	-0.349030
14	2019-06-03	254	0.080851
15	2019-06-10	207	-0.185039

In [56]:

```
# visualizing the output
plt.figure(figsize=(13,6.5))
sns.set(style='darkgrid', font_scale=1.2)
x = stage4['clicks_at_week']
y = stage4['week_over_week_variance_percent']
g = sns.barplot(x, y, color= 'dodgerblue')
g.set_title("# Clicks", fontsize = 16)
g.set_xlabel("week", fontsize = 13)
g.set_ylabel("Number of Retailers", fontsize = 13)
plt.xticks(rotation=90, fontsize=11)
plt.yticks(fontsize=11);
```



In [57]:

```
# visualizing the output
plt.figure(figsize=(13,6.5))
```

```

sns.set(style='darkgrid', font_scale=1.2, rc={"lines.linewidth":1.4});
x1 =stage1['confirmed_first_order_at_week']
y1 =stage1['week_over_week_variance_percent']
g = sns.lineplot(x1, y1, color= 'dodgerblue', label = 'order confirm')

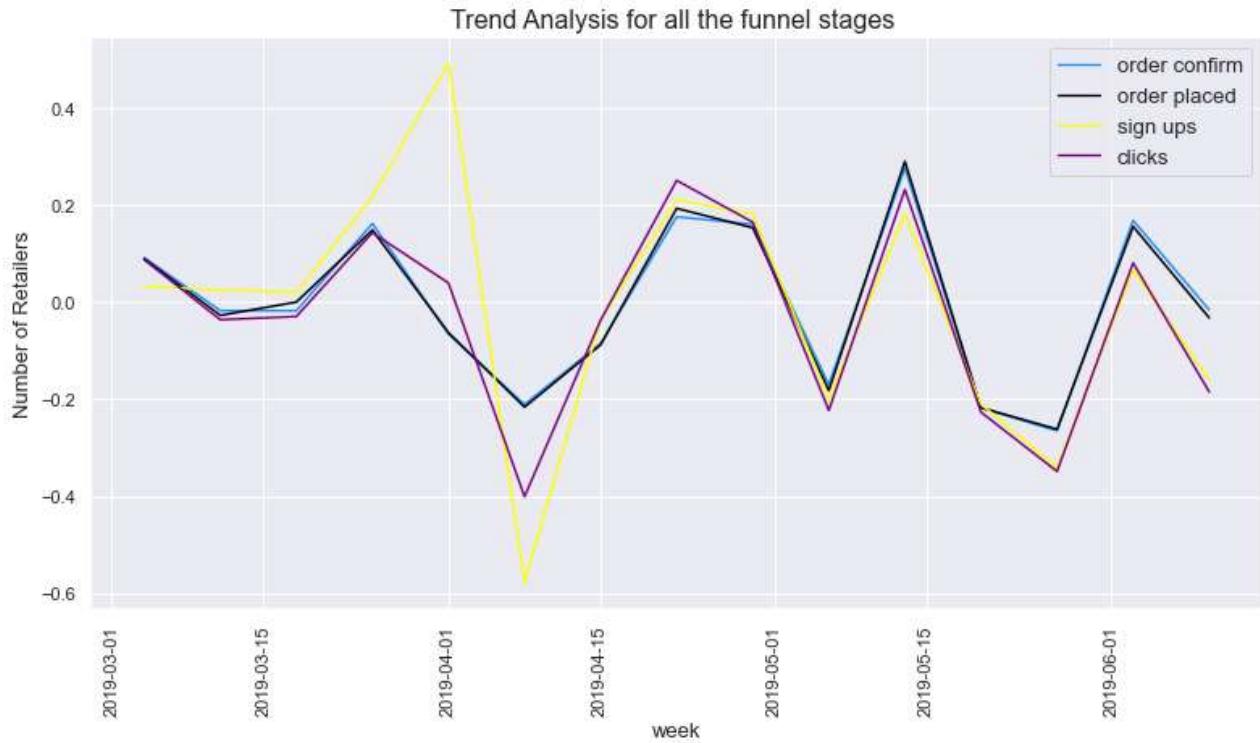
x2 =stage2['order_placed_at_week']
y2 =stage2['week_over_week_variance_percent']
g = sns.lineplot(x2, y2, color= 'black',label = 'order placed')

x3 =stage3['sign_up_at_week']
y3 =stage3['week_over_week_variance_percent']
g = sns.lineplot(x3, y3, color= 'yellow',label = 'sign ups')

x4 =stage4['clicks_at_week']
y4 =stage4['week_over_week_variance_percent']
g = sns.lineplot(x4, y4, color= 'purple',label = 'clicks')

g.set_title("Trend Analysis for all the funnel stages ", fontsize = 16)
g.set_xlabel("week", fontsize = 13)
g.set_ylabel("Number of Retailers", fontsize = 13)
plt.xticks(rotation=90, fontsize=11)
plt.yticks(fontsize=11)
g.legend();

```



Analyst notes: The overall New Retailers by the Faire Direct Strategy have declined

- From the awareness trend, we can see that the number of visitors on all Brand's Faire Direct Page have dropped.
- Second week of April 2019 observed a heavy 66% drop & similarly last 2 weeks of May 2019 brought low number of New Retailers
- Although there is a significant rise (28%) in these numbers in second week of May 2019, the numbers again start dropping by 22% and 27% in the last two weeks of May 2019 respectively

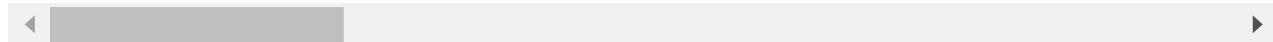
- There are high chances that the drop initiated from 1st three weeks of April 2019 and then 3rd week of May 2019

Determine where does high quality visitors come from?

In [58]: faire_direct.head()

	brand_id	retailer_id	brand_relationship_id	brand_relationship_created_at	brand_relationship_confirm
45	5072	145929.0	262478	2019-06-16	2019
71	9390	145919.0	262446	2019-06-16	2019
151	17742	145897.0	262362	2019-06-16	2019
353	14379	145824.0	262153	2019-06-16	2019
398	28795	145797.0	262090	2019-06-16	2019

5 rows × 27 columns



In [59]: faire_direct_email = faire_direct.loc[(faire_direct['retailer_placed_first_confirmed_order_at'] <= faire_direct['retailer_placed_first_order_at'])]

In [60]: faire_direct_email.retailer_id.nunique()

Out[60]: 6630

In []:

In [61]: # brand ke page pe kaise aa sakte hai?
#faire_direct[['brand_relationship_id']]
sabse pehle email bheja aur uske baad referer
faire_direct.query('email_flag == "internal"').normalized_referer.unique()

Out[61]: array(['Other', nan, 'brand_website'], dtype=object)

So basically, the retailers who clicked on the link

In [62]: # Lets only see channels where most of new retailers came
stage4.query('week_over_week_variance_percent > 0 ')

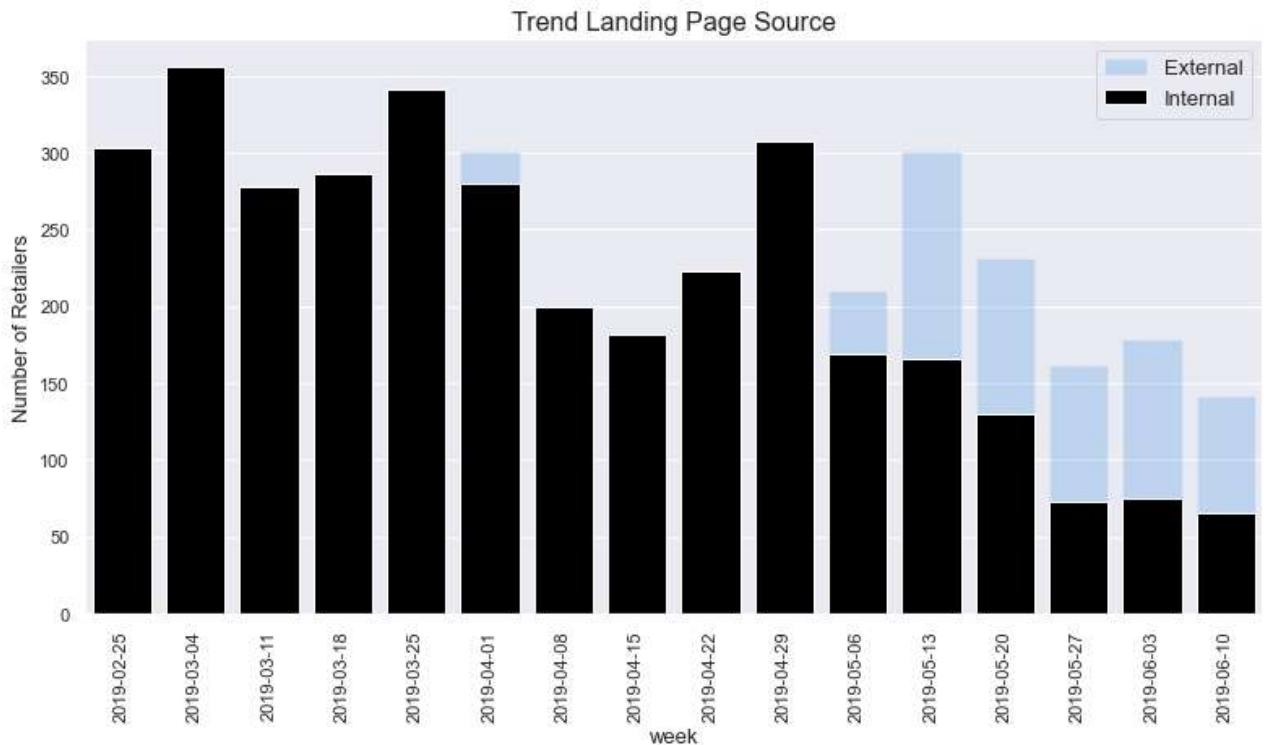
	clicks_at_week	No of Retailers	week_over_week_variance_percent
1	2019-03-04	523	0.087318
4	2019-03-25	559	0.143149
5	2019-04-01	581	0.039356
8	2019-04-22	419	0.250746
9	2019-04-29	488	0.164678
11	2019-05-13	467	0.232190

	clicks_at_week	No of Retailers	week_over_week_variance_percent
14	2019-06-03	254	0.080851

```
In [63]: tmp1 = pd.merge(stage4,faire_direct, on =['clicks_at_week'], how='inner')
tmp1 = tmp1.groupby(['clicks_at_week','email_flag'],as_index=False)[['brand_id']].count()
# visualizing the output
plt.figure(figsize=(13,6.5))
sns.set(style='darkgrid', font_scale=1.2, rc={"lines.linewidth":1.4});
x1 =tmp1.query("email_flag == 'external'")['clicks_at_week']
y1 =tmp1.query("email_flag == 'external'")['No of Retailers Acquired']
g = sns.barplot(x1, y1, color= 'dodgerblue', label = 'External', alpha = 0.25)

x2 =tmp1.query("email_flag == 'internal'")['clicks_at_week']
y2 =tmp1.query("email_flag == 'internal'")['No of Retailers Acquired']
g = sns.barplot(x2, y2, color= 'black',label = 'Internal')

g.set_title("Trend Landing Page Source", fontsize = 16)
g.set_xlabel("week", fontsize = 13)
g.set_ylabel("Number of Retailers", fontsize = 13)
plt.xticks(rotation=90, fontsize=11)
plt.yticks(fontsize=11)
g.legend();
```



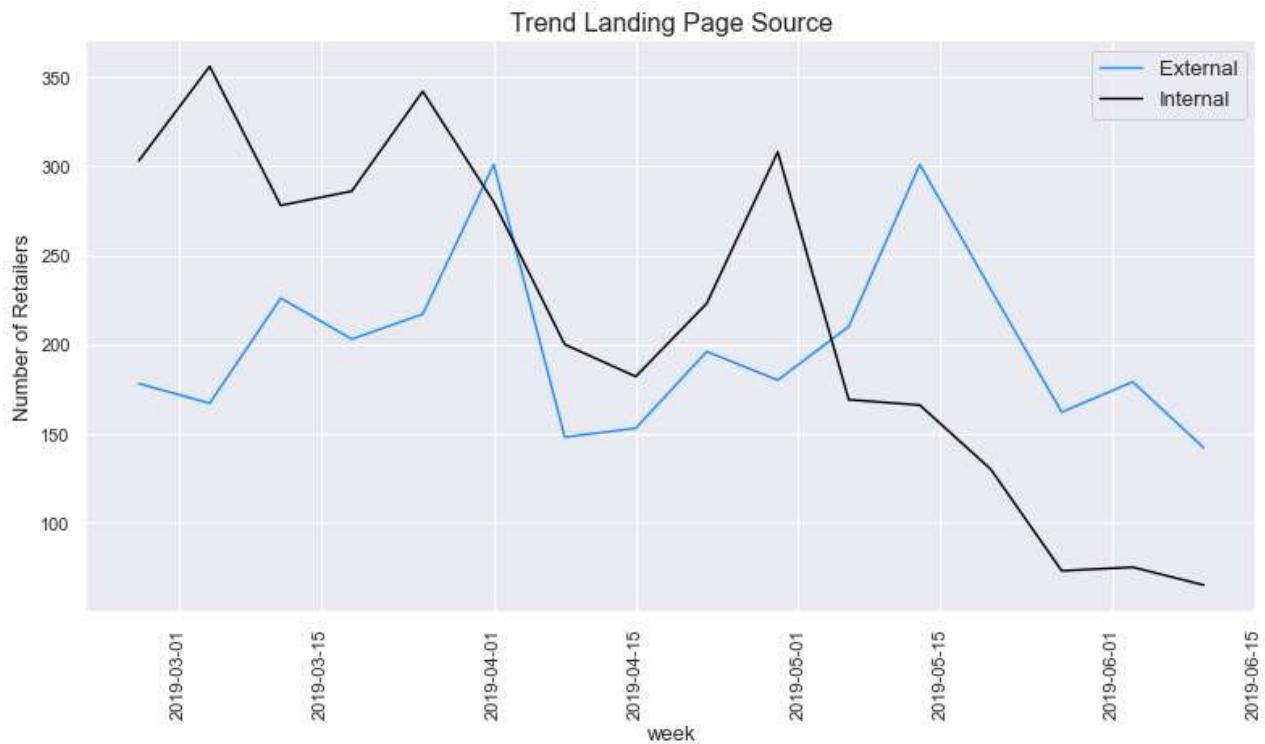
```
In [64]: # visualizing the output
plt.figure(figsize=(13,6.5))
sns.set(style='darkgrid', font_scale=1.2, rc={"lines.linewidth":1.4});
x1 =tmp1.query("email_flag == 'external'")['clicks_at_week']
y1 =tmp1.query("email_flag == 'external'")['No of Retailers Acquired']
g = sns.lineplot(x1, y1, color= 'dodgerblue', label = 'External')

x2 =tmp1.query("email_flag == 'internal'")['clicks_at_week']
y2 =tmp1.query("email_flag == 'internal'")['No of Retailers Acquired']
g = sns.lineplot(x2, y2, color= 'black',label = 'Internal')
```

```

g.set_title("Trend Landing Page Source", fontsize = 16)
g.set_xlabel("week", fontsize = 13)
g.set_ylabel("Number of Retailers", fontsize = 13)
plt.xticks(rotation=90, fontsize=11)
plt.yticks(fontsize=11)
g.legend();

```



Which teams are affected?

```
In [195... team = df_acquired_retailers.groupby(['order_confirmed_week_start','account_owner'])['r
team.rename(columns = {'retailer_id':'No of Retailers Acquired'}, inplace=True)
```

```
In [201... team.head()
```

	order_confirmed_week_start	account_owner	No of Retailers Acquired
0	2019-02-25	Control	11
1	2019-02-25	Enterprise Maker	13
2	2019-02-25	Maker Success	403
3	2019-02-25	Small maker or other	42
4	2019-03-04	Control	17

```
In [200... team.account_owner.unique()
```

```
Out[200... array(['Control', 'Enterprise Maker', 'Maker Success',
 'Small maker or other'], dtype=object)
```

```
In [204... # visualizing the output
plt.figure(figsize=(13,6.5))
```

```

sns.set(style='darkgrid', font_scale=1.2, rc={"lines.linewidth":1.4});
x1 =team.query("account_owner == 'Control'")['order_confirmed_week_start']
y1 =team.query("account_owner == 'Control'")['No of Retailers Acquired']
g = sns.lineplot(x1, y1, color= 'dodgerblue', label = 'Control')

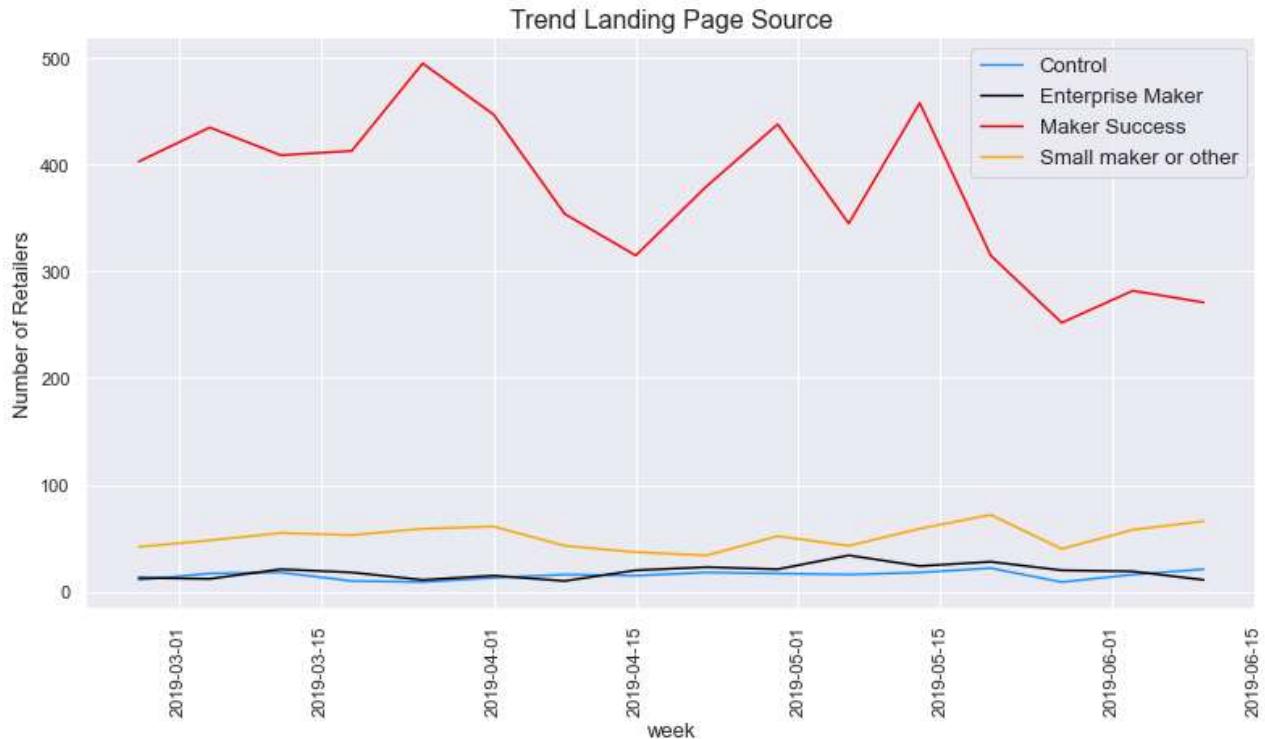
x2 =team.query("account_owner == 'Enterprise Maker'")['order_confirmed_week_start']
y2 =team.query("account_owner == 'Enterprise Maker'")['No of Retailers Acquired']
g = sns.lineplot(x2, y2, color= 'black',label = 'Enterprise Maker')

x3 =team.query("account_owner == 'Maker Success'")['order_confirmed_week_start']
y3 =team.query("account_owner == 'Maker Success'")['No of Retailers Acquired']
g = sns.lineplot(x3, y3, color= 'red', label = 'Maker Success')

x4 =team.query("account_owner == 'Small maker or other'")['order_confirmed_week_start']
y4 =team.query("account_owner == 'Small maker or other'")['No of Retailers Acquired']
g = sns.lineplot(x4, y4, color= 'orange',label = 'Small maker or other')

g.set_title("Trend Landing Page Source", fontsize = 16)
g.set_xlabel("week", fontsize = 13)
g.set_ylabel("Number of Retailers", fontsize = 13)
plt.xticks(rotation=90, fontsize=11)
plt.yticks(fontsize=11)
g.legend();

```



In []: