fit

Ina Litso

July 2019

Problem with docker

```
(base) C:\Users\ina.litso\data_analyst_case>docker-compose up
Starting data_analyst_case_postgres_1 ... done
Attaching to data_analyst_case_postgres_1
postgres_1  | 2019-07-14 10:39:17.236 UTC [1] LOG:  listening on IPv4 address "0.0.0.0", port 5432
postgres_1  | 2019-07-14 10:39:17.236 UTC [1] LOG:  listening on IPv6 address "::", port 5432
postgres_1  | 2019-07-14 10:39:17.248 UTC [1] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
postgres_1  | 2019-07-14 10:39:17.290 UTC [23] LOG:  database system was shut down at 2019-07-14 10:36:44 UTC
postgres_1  | 2019-07-14 10:39:17.327 UTC [1] LOG:  database system is ready to accept connections
```

I was not able to complete the set up after this step.
I cloned the repo, used Python for the SQL questions, wrote some pseudo-queries to show the logic and did the analysis with Python and Excel.

Data Structure

Subscriptions

```
Data columns (total 8 columns):
signup_country_code     29809 non-null object
marketing_channel_id    25908 non-null float64
signup_platform         29818 non-null object
gender                  29823 non-null object
is_paid_channel         29823 non-null bool
subscription_date       29823 non-null object
net_revenue             29823 non-null float64
subscription_count      29823 non-null int64
```

2983 lines

- 14 lines have null country
- 3915 lines have null channel id
- 5 lines have null platform

Spendings

```
Data columns (total 6 columns):
report_date             59450 non-null object
country_code            59339 non-null object
platform                59450 non-null object
marketing_channel_id    59450 non-null int64
is_paid_channel         59450 non-null bool
spendings               59450 non-null float64
```

59450 lines

- 111 lines have null country

How much did we spent per channel in December?

marketing_channel_id	
0	0.000000
1	0.000000
2	89519.720000
3	8501.770000
4	35628.319968
6	16128.329970
7	0.000000
9	1358.800019
10	27.410000
11	0.000000
12	0.000000
13	0.000000
14	0.000000
16	0.000000
18	51299.700017

```python
#First question:How much did we spent per channel in December?
spendings_dec=spendings.loc[spendings['report_date'].str.contains('2016-12')]
spendings_dec.groupby(['marketing_channel_id'])['spendings'].sum()
```

```sql
SELECT
  marketing_channel_id
  , SUM(spendings)  AS Spendings
FROM spendings
WHERE report_date BETWEEN '2016-12-01' and '2016-12-31'
GROUP BY marketing_channel_id
```

What is the average cost of acquisition for a subscription per country?

Results in csv*

```python
#Second question:What is the average cost of acquisition for a subscription per country?
subscription_per_country=subscriptions.groupby(['signup_country_code']).agg({'net_revenue':'sum','subscription_count':'sum'}).reset_index()
spendings_per_country=spendings.groupby(['country_code'])['spendings'].sum().reset_index()
merge_data_country = pd.merge(spendings_per_country, subscription_per_country,  how='outer', left_on=['country_code'], right_on = ['signup_country_code'])
merge_data_country['country_code'].isnull().sum()
merge_data_country['signup_country_code'].isnull().sum()

average=round(merge_data_country.groupby(['country_code'])['spendings'].sum()/merge_data_country.groupby(['country_code'])['subscription_count'].sum(),1).sort_values(ascending=False)
average.to_csv('average_cac.csv', sep = ',')
```

```sql
WITH spendings AS
(
SELECT
  country_code, SUM(spendings)  AS Spendings
FROM spendings
WHERE report_date BETWEEN '2016-10-01' and '2017-01-31'
GROUP BY country_code
)
, subscriptions AS
(
  SELECT
    signup_country_code
    , SUM(subscription_count)  AS subscriptions
  FROM spendings
  WHERE signup_date BETWEEN '2016-10-01' and '2017-01-31'
  GROUP BY signup_country_code
)
```

```sql
SELECT
  spendings.country_code
  , SUM(spendings/subscriptions) average_cac
FROM spendings
LEFT JOIN subscriptions ON subscriptions.signup_country_code=spendings.country_code
GROUP BY spendings.country_code
```

*We have countries with marketing costs but without subscriptions where we are not able to calculate the cac.

What is the average cost of acquisition for a subscription per country?

```
In [307]:
subscriptions['subscription_date'].min()
Out[307]: '2016-10-01'
```

```
In [309]:
subscriptions['subscription_date'].max()
Out[309]: '2017-01-31'
```

```
In [310]: spendings['report_date'].min()
Out[310]: '2016-10-01'

In [311]: spendings['report_date'].max()
Out[311]: '2017-01-31'
```

Ensure that both metrics have similar timeframe.
Although, we need to emphasize that some channels may need longer time window for conversions.
So it would have been better to have data of +X days after the latest spent

What is our average revenue and spending per day of the week?

spend_weekDay	net_revenue	spendings
Friday	5915.127328	4364.969987
Monday	9581.819217	6613.345011
Saturday	7191.827951	5614.789988
Sunday	11454.516941	9449.899991
Thursday	6710.256042	4953.550000
Tuesday	8063.870676	5996.179993
Wednesday	6200.152790	4757.129988

What is our average revenue and spending per day of the week?

```
#Third Question: What is our average revenue and spending per day of the week (Monday, Tuesday...)?

subscription_per_day=subscriptions.groupby(['subscription_date']).agg({'net_revenue':'sum','subscription_count':'sum'}).reset_index()
spendings_per_day=spendings.groupby(['report_date'])['spendings'].sum().to_frame().reset_index()
merge_data_per_day = pd.merge(spendings_per_day, subscription_per_day,  how='left', left_on=['report_date'], right_on = ['subscription_date'])

merge_data_per_day['report_date']=pd.to_datetime(merge_data_per_day['report_date'])
merge_data_per_day['subscription_date']=pd.to_datetime(merge_data_per_day['subscription_date'])
merge_data_per_day.info()
merge_data_per_day['spend_weekDay']=merge_data_per_day['report_date'].dt.day_name()

merge_data_per_day['subscription_weekDay']=merge_data_per_day['subscription_date'].dt.day_name()

merge_data_per_day.groupby(['spend_weekDay']).agg({'net_revenue':'median','subscription_count':'median'})
```

What is our average revenue and spending per day of the week?

```sql
WITH spendings AS
(
SELECT
   SUM(spendings)  AS Spendings
  , report_date
FROM spendings
WHERE report_date BETWEEN '2016-10-01' and '2017-01-31'
GROUP BY report_date
)
, subscriptions AS
(
  SELECT
    SUM(net_revenue)  AS subscriptions
    , signup_date
  FROM spendings
  WHERE signup_date BETWEEN '2016-10-01' and '2017-01-31'
  GROUP BY signup_date
)
SELECT
  DATENAME(weekday, report_date) weekday
  , AVG(spendings) average_revenue
  , AVG(Spending) average_spendings
FROM spendings
LEFT JOIN subscriptions ON subscriptions.subscription_date=spendings.report_date
GROUP BY DATENAME(weekday, report_date)
```

Analysis

CAC & Average Revenue per OS

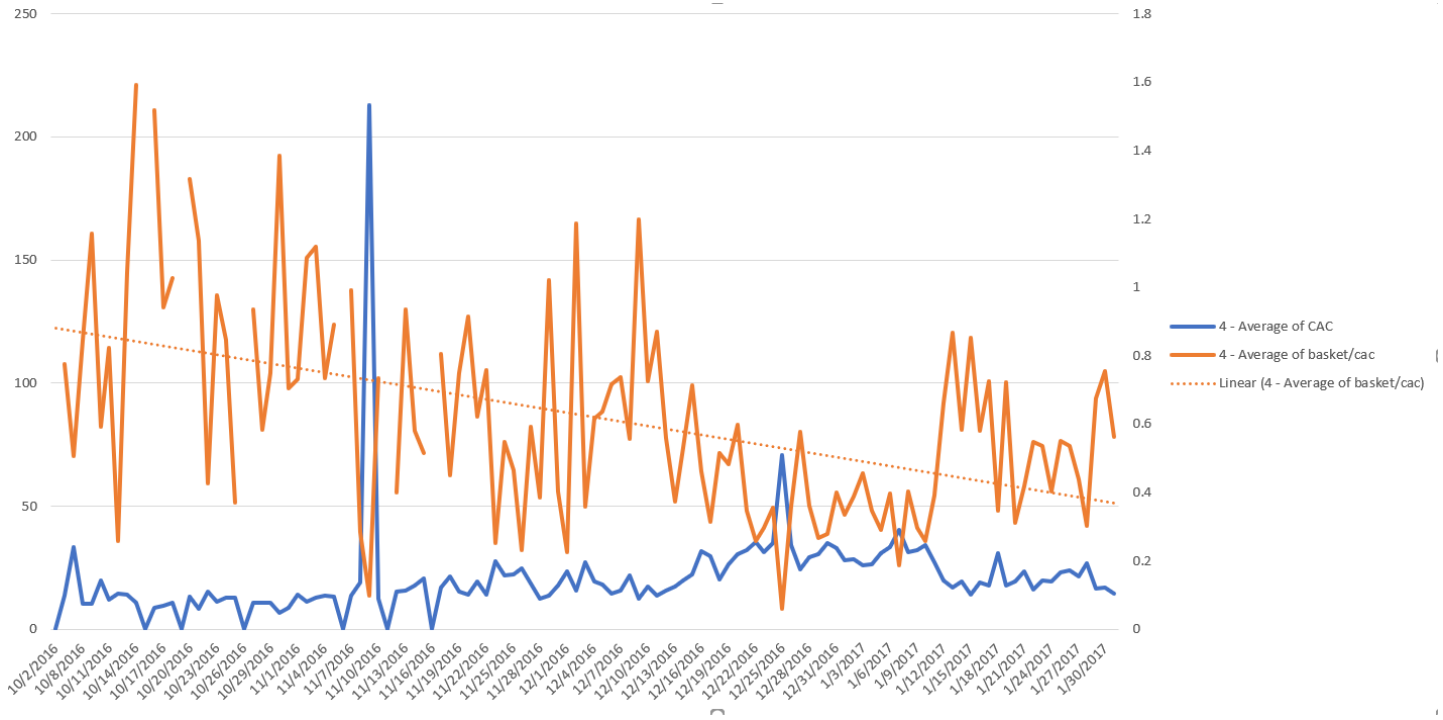
```
In [20]: merge_data
Out[20]:
  platform      spendings       ...             CAC  basket_value
0  android  390595.839949       ...       11.609328     11.582887
1      ios  862532.093725       ...       11.252131     12.425933
```

Basket_value/CAC : Shows if a cohort is profitable or no after their first purchase.

So, we can assume that ios platform has higher quality customers that Android.

*I assum that the net revenue is the amount that a customer generate in his activations.
We need to have also the CLV for better understanding of the market.

Channel 4 Performance



In this graph we can see the average CAC and net revenue per subscription for channel 4.

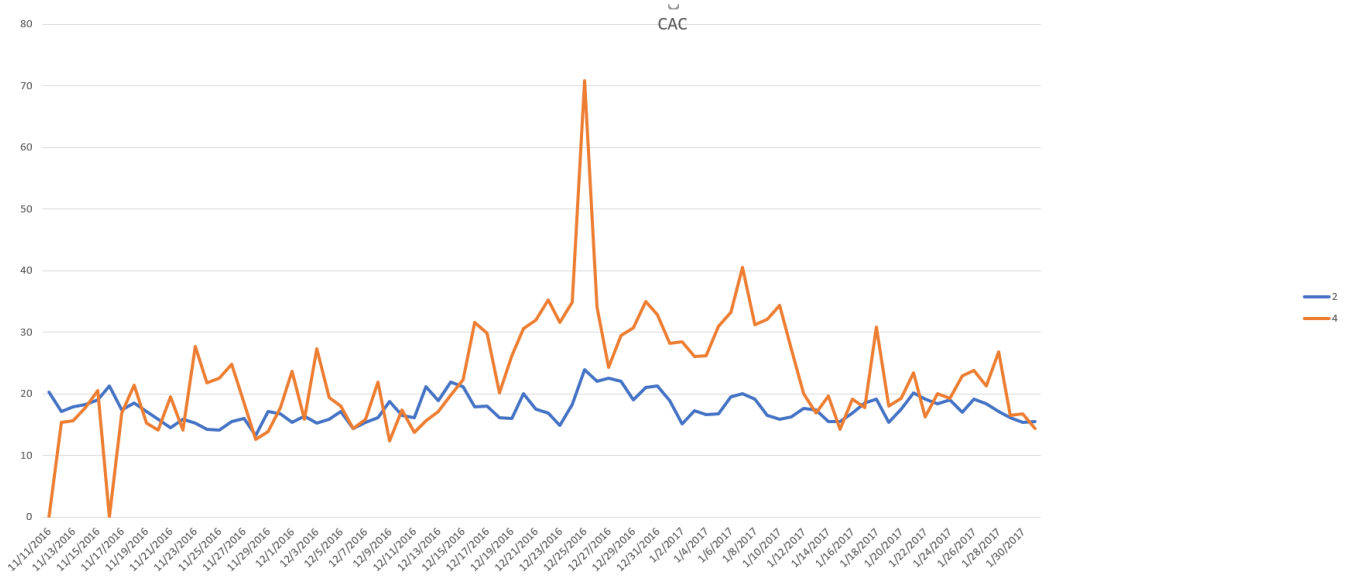
It is true that since 12/1 the CAC is going down and the average basket value has been increased but looking historically the performance of the channel has not improved.

If the channel manager has made changes in campaign settings, then yes we can recommend to invest more on this channel.

But, the beginning of a year is the right moment to start a healthier lifestyle so the company needs to be careful with the marketing budget and seasonality.

Channel Performance

Row Labels	Sum of spendings	Sum of net_revenue	Sum of subscription_count
2	50.93%	60.10%	59.12%
3	7.54%	4.04%	4.30%
4	9.77%	8.37%	8.77%
6	8.92%	8.25%	5.30%
9	1.98%	1.26%	1.63%
10	0.21%	0.02%	0.01%
16	0.04%	0.00%	0.00%
18	20.62%	17.96%	20.85%
Grand Total	**100.00%**	**100.00%**	**100.00%**



It seems that Channel 2 is the best performing paid channel and we can use it as point of reference for the other channels.

In this graph we can see the CAC over time for Channel 2 and Channel 4.

Channel Performance

Tracking issue: 11,426 (~10% of subscriptions and revenue) of the subscriptions do not have a channel id.

Row Labels	Sum of net_revenue	Sum of subscription_count
0	0.00%	0.00%
1	36.22%	31.98%
2	31.26%	33.09%
3	2.12%	2.41%
4	4.50%	5.07%
6	4.29%	2.97%
7	0.04%	0.03%
9	0.66%	0.92%
10	0.11%	0.05%
11	0.35%	0.27%
12	0.40%	0.50%
13	0.54%	0.68%
14	0.02%	0.02%
16	0.00%	0.00%
18	9.34%	11.67%
(blank)	10.15%	10.34%

Country Performance

country_code	spendings	net_revenue	subscription_count	% revenue	% spends	ROI
US	555,926	545,541	43,965	40.5%	44.3%	-2%
CA	117,899	102,881	8,548	7.6%	9.4%	-13%
CH	58,369	73,090	4,237	5.4%	4.7%	25%
AU	60,105	70,181	5,065	5.2%	4.8%	17%
FR	43,176	63,040	4,874	4.7%	3.4%	46%
ES	60,488	56,684	5,770	4.2%	4.8%	-6%
GB	38,620	48,696	5,418	3.6%	3.1%	26%
MX	44,408	45,118	5,131	3.3%	3.5%	2%
AR	21,296	40,262	2,693	3.0%	1.7%	89%
CL	19,529	32,505	2,117	2.4%	1.6%	66%
DE	27,894	31,516	3,285	2.3%	2.2%	13%

GB is an interesting market considering that the 3.1% of costs generates the 3.6% of total revenue. Also the ROI is **26%.**

But based on this logic, **FR** seems to be more promising. it holds slightly higher share of the total cost (3.4%) and generates 4.7% of total revenue. Also the France ROI is one of the highest (**46%**)