

Calculating Churn Rates Data Analysis with SQL

Jose Luis Garcia Tucci
23/05/2022

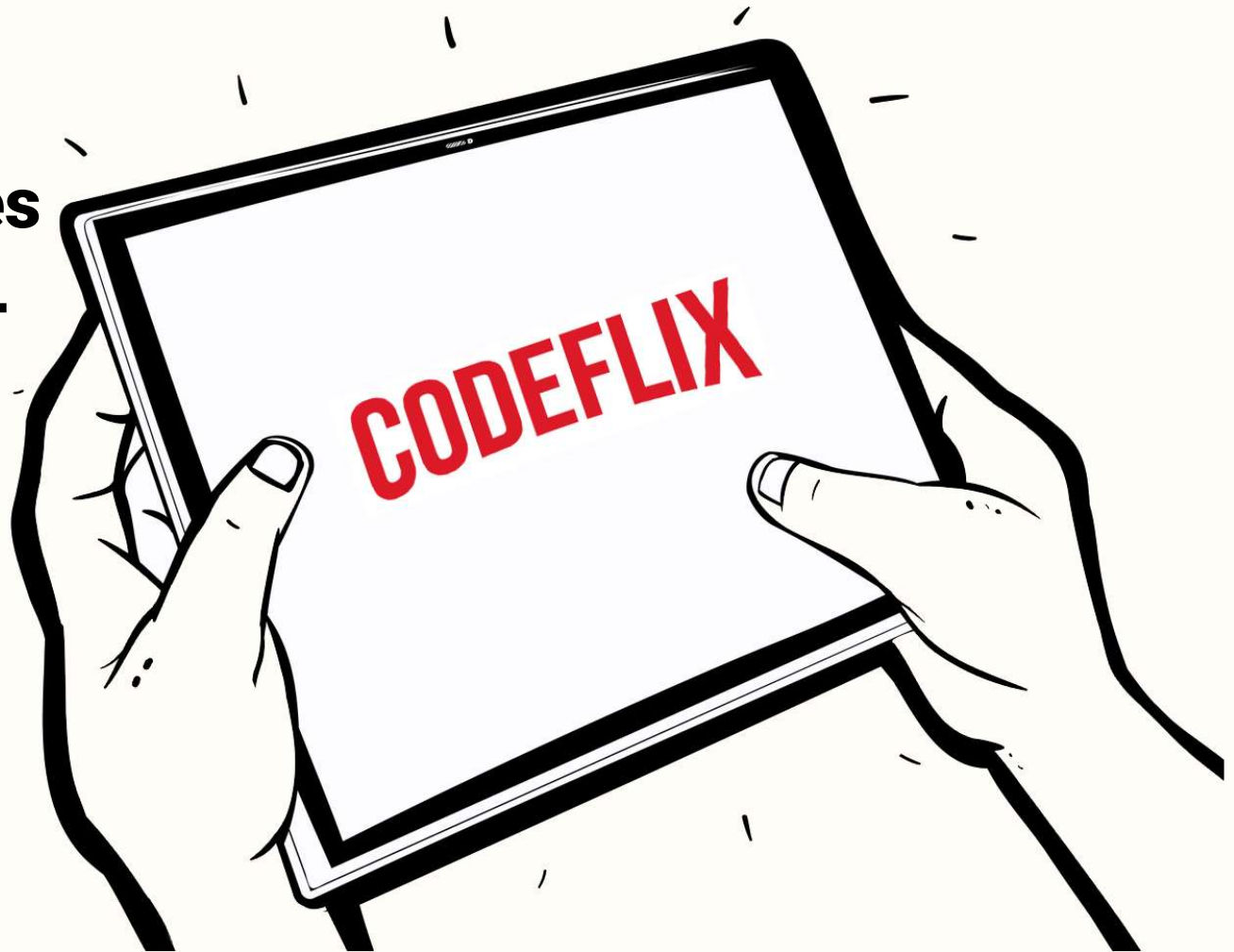


Table of Contents

1. Context
2. Codeflix Subscriber Data Sample and Queries
3. Calculate Churn Rates for each segment
4. Visualization for gained and lost subscribers per month

1. Context

Four months into launching Codeflix, management asks you to look into subscription churn rates. It's early on in the business and people are excited to know how the company is doing.

The marketing department is particularly interested in how the churn compares between two segments of users. They provide you with a dataset containing subscription data for users who were acquired through two distinct channels.

2. Codeflix Subscriber Data Sample and Queries

The dataset provided contains one SQL table, subscriptions. Within the table, there are 4 columns:

- *id* - the subscription id
- *subscription_start* - the start date of the subscription
- *subscription_end* - the end date of the subscription
- *segment* - this identifies which segment the subscription owner belongs to

Codeflix requires a minimum subscription length of 31 days, so a user can never start and end their subscription in the same month.

There are two segments, 87 and 30, which identify users acquired through two separate marketing channels.

Data in the table goes from 2016-12-01 up to 2017-03-30

Let's query first 15 lines from the subscription table:

```
-- SQL Query
SELECT *
FROM subscriptions
LIMIT 15;
```

Query results			
id	subscription_start	subscription_end	segment
1	2016-12-01	2017-02-01	87
2	2016-12-01	2017-01-24	87
3	2016-12-01	2017-03-07	87
4	2016-12-01	2017-02-12	87
5	2016-12-01	2017-03-09	87
6	2016-12-01	2017-01-19	87
7	2016-12-01	2017-02-03	87
8	2016-12-01	2017-03-02	87
9	2016-12-01	2017-02-17	87
10	2016-12-01	2017-01-01	87
11	2016-12-01	2017-01-17	87
12	2016-12-01	2017-02-07	87
13	2016-12-01		30
14	2016-12-01	2017-03-07	30
15	2016-12-01	2017-02-22	30

Database Schema	
subscription	
name	type
id	INTEGER
subscription_start	TEXT
subscription_end	TEXT
segment	INTEGER
Rows: 2000	

3. Calculate Churn Rates for each segment

For calculating the churn rates let's create some temporary tables

```
-- SQL Query
WITH months AS (
  SELECT
    '2017-01-01' AS first_day,
    '2017-01-31' AS last_day
  UNION
  SELECT
    '2017-02-01' AS first_day,
    '2017-02-28' AS last_day
  UNION
  SELECT
    '2017-03-01' AS first_day,
    '2017-03-31' AS last_day
),
```

1.- months temporary table

```
-- SQL Query
cross_join AS (
  SELECT *
  FROM subscriptions
  CROSS JOIN months
),
```

2.- cross_join temporary table

```
-- SQL Query
status AS (
  SELECT
    id,
    first_day AS month,
    CASE
      WHEN segment = 87
        AND (subscription_start < first_day)
        AND (
          subscription_end > first_day
          OR subscription_end IS NULL
        ) THEN 1
      ELSE 0
    END AS is_active_87,
    CASE
      WHEN segment = 30
        AND (subscription_start < first_day)
        AND (
          subscription_end > first_day
          OR subscription_end IS NULL
        ) THEN 1
      ELSE 0
    END AS is_active_30,
```

3.- status temporary table, showing active status by segment

```
-- SQL Query
status AS (
  SELECT
    id,
    first_day AS month,
    CASE
      WHEN segment = 87
        AND (subscription_start < first_day)
        AND (
          subscription_end > first_day
          OR subscription_end IS NULL
        ) THEN 1
      ELSE 0
    END AS is_active_87,
    CASE
      WHEN segment = 30
        AND (subscription_start < first_day)
        AND (
          subscription_end > first_day
          OR subscription_end IS NULL
        ) THEN 1
      ELSE 0
    END AS is_active_30,
    CASE
      WHEN segment = 87
        AND subscription_end BETWEEN first_day AND last_day THEN 1
      ELSE 0
    END AS is_canceled_87,
    CASE
      WHEN segment = 30
        AND subscription_end BETWEEN first_day AND last_day THEN 1
      ELSE 0
    END AS is_canceled_30
  FROM cross_join
```

4.- Adding canceled status for both segments to the status table

Churn Rates

```
-- SQL Query
status_aggregate AS (
  SELECT
    month,
    SUM(is_active_87) AS sum_active_87,
    SUM(is_active_30) AS sum_active_30,
    SUM(is_canceled_87) AS sum_canceled_87,
    SUM(is_canceled_30) AS sum_canceled_30
  FROM status
  GROUP BY month
)
SELECT
  month,
  100.0 * sum_canceled_30 / sum_active_30 AS churn_rate_30,
  100.0 * sum_canceled_87 / sum_active_87 AS churn_rate_87
FROM status_aggregate;
```

5.- Aggregating the sum of active and cancelled subscriptions by month, and calculation the churn rates in percentage multiplying by 100

Query results		
month	churn_rate_30	churn_rate_87
2017-01-01	7.56013745704467	25.1798561151079
2017-02-01	7.33590733590734	32.034632034632
2017-03-01	11.731843575419	48.5875706214689

Churn Rates		
Month	Segment 30	Segment 87
Jan 2017	7.56 %	25.18 %
Feb 2017	7.34 %	32.03 %
Mar 2017	11.73 %	48.59 %

Churn rates average for segment 30 is 9% whether for segment 87 is up to 35% making it's maximum in March 2017. Since segment 87 is losing subscriber 4 times faster as segment 30 something has to be done to reverse this trend.

3. Visualization for gained and lost subscribers per month

Let's get the total of subscribers gained or lost by month

```
-- SQL Query
SELECT
  SUBSTR(subscription_start,1,7) as month,
  COUNT(SUBSTR(subscription_start,1,7)) as gained_subscribers
FROM subscriptions
GROUP BY month;
```

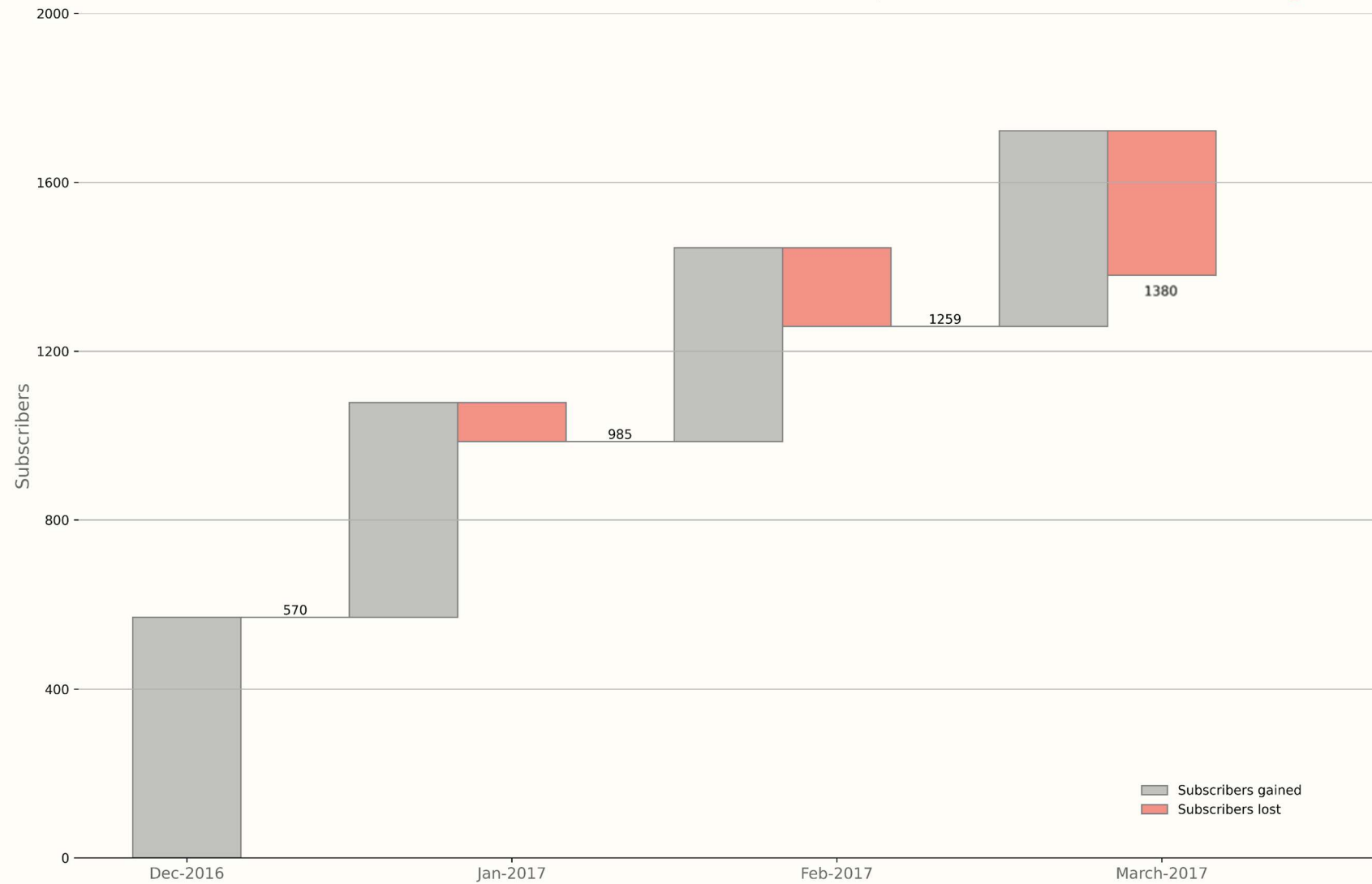
Let's get gained subscribers by month

```
-- SQL Query
SELECT
  SUBSTR(IFNULL(subscription_end,'2016-12'),1,7) as month,
  COUNT(SUBSTR(subscription_end,1,7)) as lost_subscribers
FROM subscriptions
GROUP BY month
```

Let's get lost subscribers by month

Query results	
month	gained_subscribers
2016-12	570
2017-01	507
2017-02	460
2017-03	463

Churn Rates	
Month	Segment 30
2016-12	0
2017-01	92
2017-02	186
2017-03	342



Visualization done in Python based on "The Churn Dashboard Explained" at www.datarevelations.com where the author Steve Wexler explains how he created the Churn Dashboard visualization in Tableau.