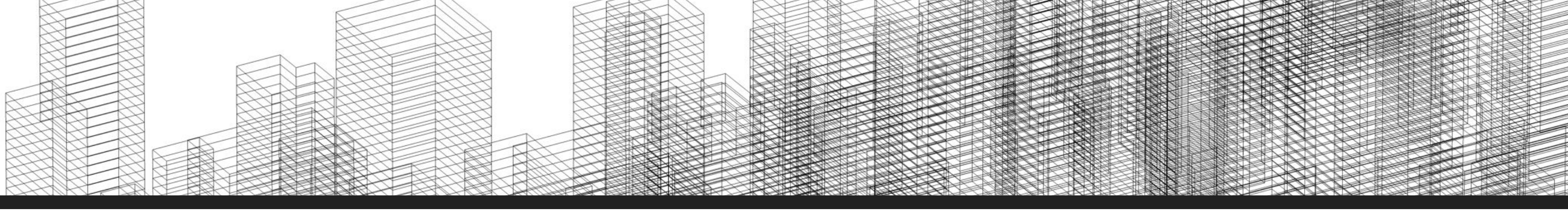
codecademy

Calculating Churn Rates

At Codeflix

Learn SQL from Scratch

Matthew Farrell - Capstone Project



Contents

1. Getting familiar with Codeflix / Scoping the data out
2. Calculating the overall churn rate for Codeflix by month
3. Comparing the churn rates between user segments
4. Making a business decision based on data evidence

Codeflix Churn Rates - Table Discovery

Getting familiar with the company data.

- First we select all columns and rows from the table that we've been provided, 'subscriptions'. This allows us to get a glimpse on how the table is structured and what sort of data we have access to.
- We limit our query to 100 results so that we don't pull everything in the table. 100 results should be plenty to see what the data will look like**.
- With the primary key being 'id', we know each value in this column will be unique for each row.
- 'subscription_start' and 'subscription_end' are straightforward in this table
- 'segment' appears to have one of two types of values, 87 or 30.

** For presentation purposes, rows 6-11 have been omitted to show variance in the table

```
SELECT *
FROM subscriptions
LIMIT 100;
```

table_discovery.sql

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
~	~	~	~
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
16	2016-12-01		30
17	2016-12-01		30

Codeflix Churn Rates - Scope

Question 1: How many months has Codflix been operating?

- Now that we know what data is available to us, we start by running a query on the minimum (MIN) start date for any ‘subscription_start’ in the ‘subscriptions’ table. Here we retrieve the earliest date of any subscription in our dataset.
- Conversely, by adding a column for the maximum (MAX) ‘subscription_end’ we’re able to retrieve the last and most recent date a subscription with the service has ended.
- Codeflix enrolled members beginning 12/1/2016 with the last subscription ending on 3/31/2017.

Answer: Therefore, Codeflix has operated for a total of **4 months.**

```
SELECT
MIN(subscription_start),
MAX(subscription_end)
FROM subscriptions;
```

scope_months_op.sql

MIN(subscription_start)	MAX(subscription_end)
2016-12-01	2017-03-31

Codeflix Churn Rates - Scope

Question 2: Which months do we have enough information to calculate a churn rate for?

- To calculate churn rate we simply need to know the total amount of membership *cancellations* for a given period versus total memberships
- To get an idea on what months of information we can calculate for, we use MIN and MAX to examine the range of ‘subscription_start’.
- Since we’re calculating churn rate on a per-month interval, and members have a minimum subscription length of 31 days, we can begin our churn rate calculation on the second month of Codeflix’s service offering. We can confirm this by querying the MIN and MAX (subscription_end) date as seen below, which give gives us 3 months (the first three months of 2017) for the top half of our equation.

Answer: We have enough information to calculate the churn rate for **3 months**

```
SELECT
MIN(subscription_start),
MAX(subscription_start)
FROM subscriptions;
```

MIN(subscription_start)	MAX(subscription_start)
2016-12-01	2017-03-30

scope_months_for_churn.sql

```
SELECT
MIN(subscription_end),
MAX(subscription_end)
FROM subscriptions;
```

MIN(subscription_end)	MAX(subscription_end)
2017-01-01	2017-03-31

scope_months_for_churn_extra.sql

Codeflix Churn Rates - User Segments

Question 3: What segments of users exist in our dataset?

- Looking back at our original “Table Discovery” query, we found a column that had multiple values in the ‘segment’ column.
- For the purposes of this research, the marketing department was interested in the churn rate between two segments of users that were acquired through two distinct channels.
- Without explicitly having the two segments defined, we can assume the table only has data for these two segments. We can confirm that by querying our ‘subscriptions’ table’s segment column. We should only have two results.

Answer: 2 segments exist, **87 and 30**

segment
87
30

```
SELECT DISTINCT segment
FROM subscriptions;
```

distinct_segments_extra.sql

Codeflix Churn Rates - Overall Churn Trend

Question 4: What is the overall churn trend since the company started?

- By creating a temporary table we can structure our data into months by using the dataset columns ‘first_day’ and ‘last_day’.
- Using a CASE statement we can calculate the sum of active subscriptions for each month, and thus those that are also cancelled - giving us the recipes for a churn rate calculation (1.0 * canceled / active).
- The results show a trend of each month having a higher churn rate than the previous.

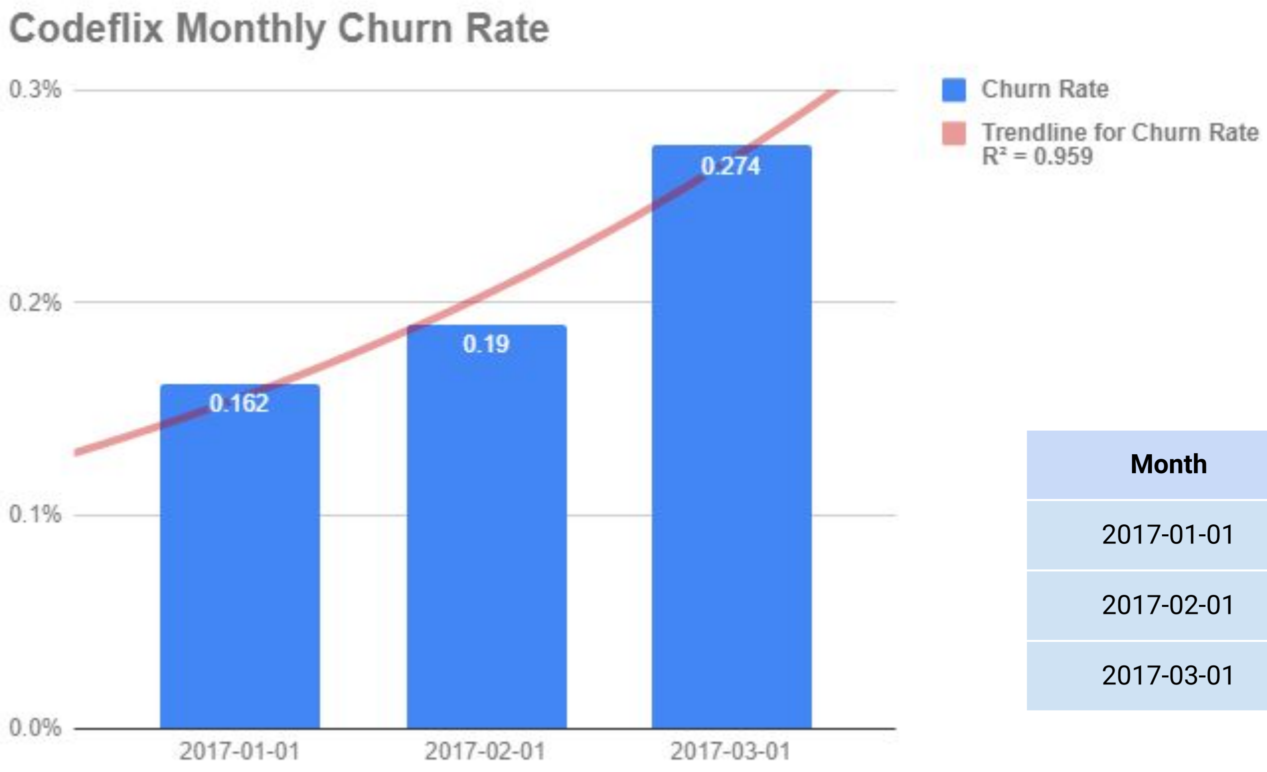
Answer: Overall trend is that the monthly **churn rate is increasing month over month**

month	churn_rate
2017-01-01	0.162
2017-02-01	0.19
2017-03-01	0.274

```
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
),
cross_join AS (
  SELECT *
  FROM subscriptions
  CROSS JOIN months
),
status AS (
  SELECT
    id,
    first_day AS month,
    CASE
      WHEN (subscription_start < first_day)
        AND (
          subscription_end > first_day
          OR subscription_end IS NULL
        ) THEN 1
      ELSE 0
    END AS is_active,
    CASE
      WHEN subscription_end BETWEEN first_day AND last_day THEN 1
      ELSE 0
    END AS is_canceled
  FROM cross_join
),
status_aggregate AS (
  SELECT
    month,
    SUM(is_active) AS active,
    SUM(is_canceled) AS canceled
  FROM status
  GROUP BY month
)
SELECT
  month,
  ROUND(1.0 * canceled / active, 3) AS churn_rate
FROM status_aggregate;
```

overall_churn.sql

Codeflix Churn Rates - Overall Churn Trend

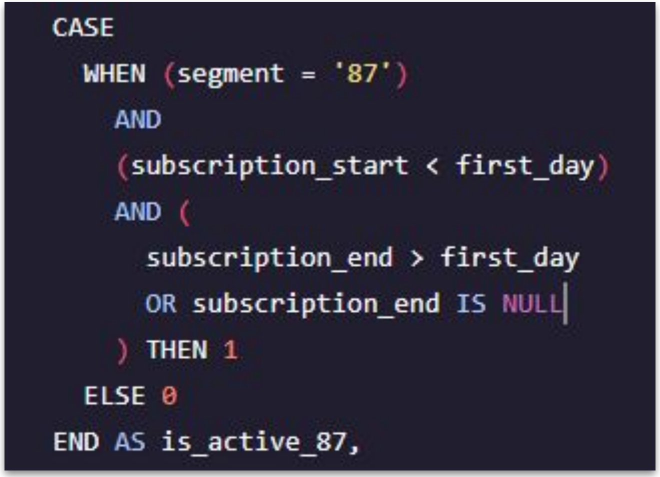


Month	Churn Rate %
2017-01-01	0.162 %
2017-02-01	0.19 %
2017-03-01	0.274 %

Codeflix Churn Rates - Churn Between User Segments

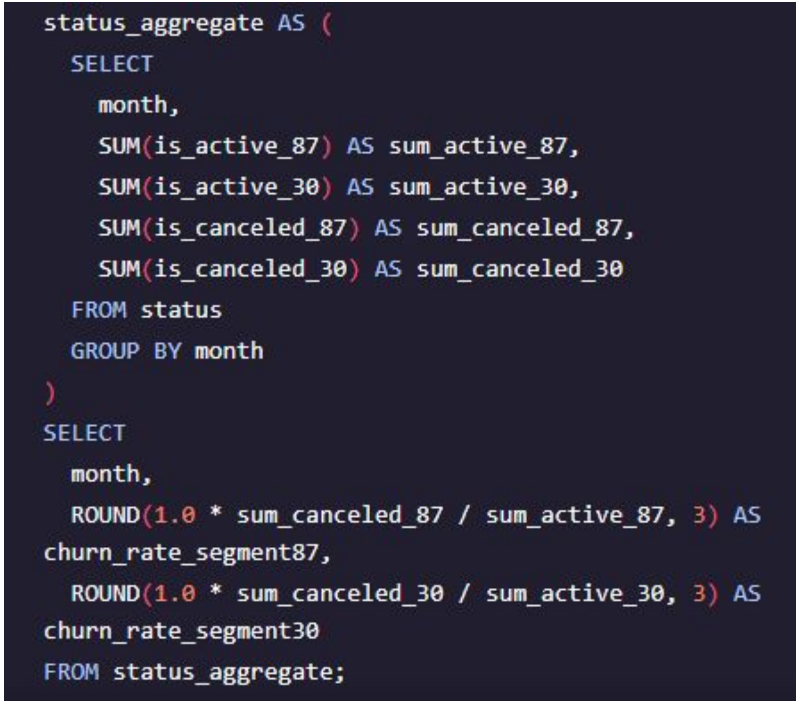
Question 5: How do the churn rates between user segments compare?

- By adding another CASE statement we can split our churn rate analysis into both of our different user segments.
- This allows us to determine which segment has a higher rate of churn.



```
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,
```

Note: Due to length, please view full query in: churn_rate_between_segments.sql



```
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,
  ROUND(1.0 * sum_canceled_87 / sum_active_87, 3) AS
churn_rate_segment87,
  ROUND(1.0 * sum_canceled_30 / sum_active_30, 3) AS
churn_rate_segment30
FROM status_aggregate;
```

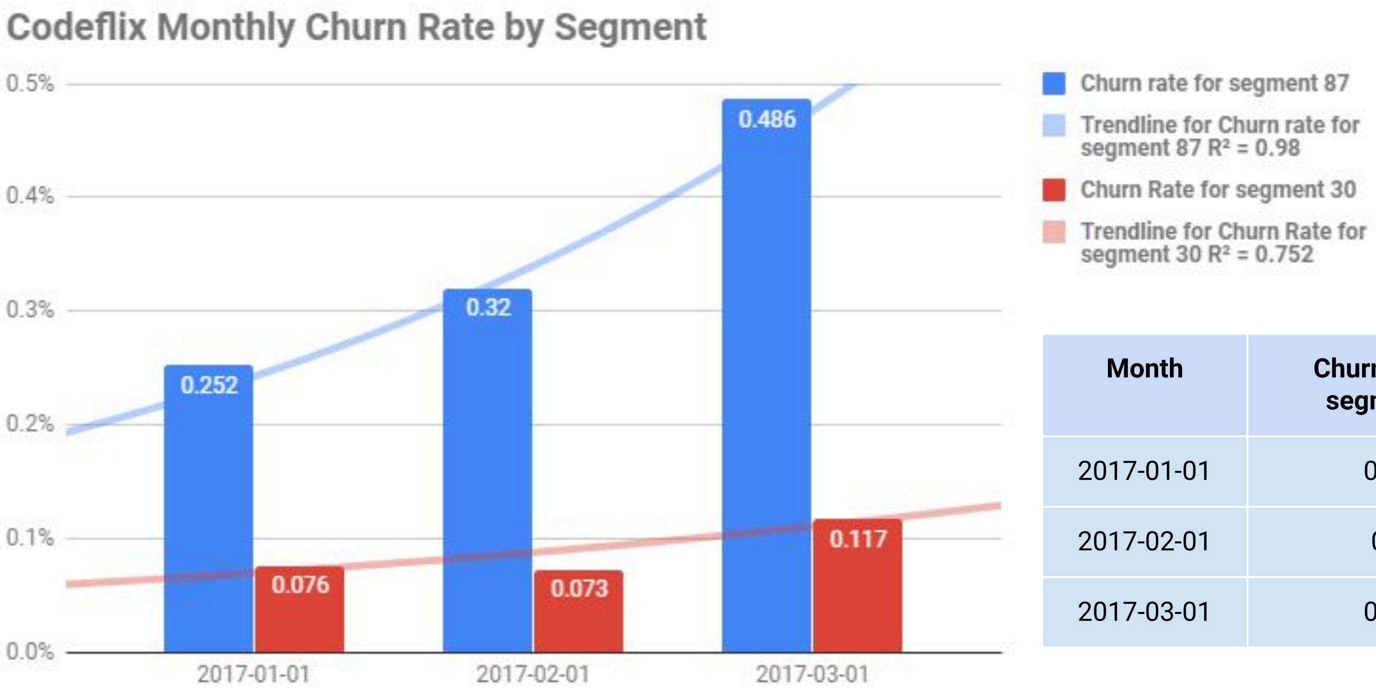
Codeflix Churn Rates - Churn Between User Segments

Question 5: How do the churn rates between user segments compare?

What we find is that **segment 87 has a much higher rate of churn** versus segment 30

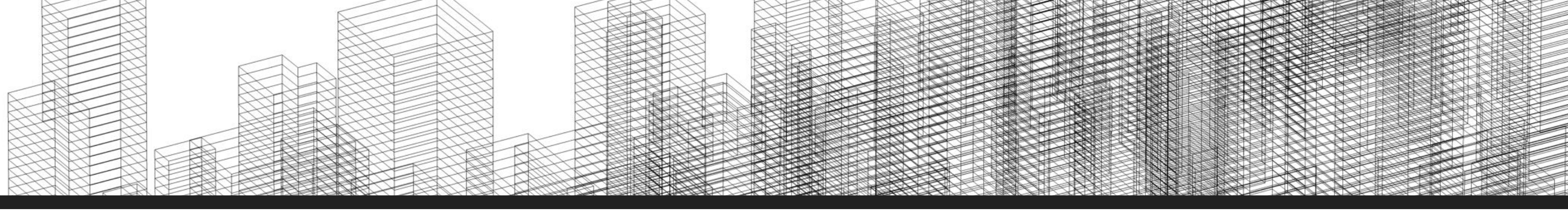
month	churn_rate_segment87	churn_rate_segment30
2017-01-01	0.252 %	0.076 %
2017-02-01	0.32 %	0.073 %
2017-03-01	0.486 %	0.117 %

Codeflix Churn Rates - Overall Churn Trend



Month	Churn rate for segment 87	Churn Rate for segment 30
2017-01-01	0.252	0.076
2017-02-01	0.32	0.073
2017-03-01	0.486	0.117

What we do with this data



Codeflix Churn Rates - Data-Backed Business Decision

With the use of SQL, we’ve been able to determine the following about Codeflix’s business:

- The total months of business operation
- Which of these months that we have sufficient data to be able to calculate the membership churn rate
- The individual marketing channel user segments that Codeflix uses to acquire new members
- The overall subscription churn trend since the company started
- The comparison of churn rates between each of the user segments

With these findings within the data, we can report to marketing and management that the acquisition channel for user segment 30 completely outperforms the channel for user segment 87. In addition, while user segment 30’s churn rate remains around the .1% mark, user segment 87’s churn rate is over 400% higher and increasing at a slightly exponential rate.

The data suggests that membership acquisition should focus primarily on the marketing channel that the members of user segment 30 were first introduced through in order to keep membership churn low.