codecademy

Calculating Churn Rates

By Anantya Sahney

Table of Contents

Explanation of terms

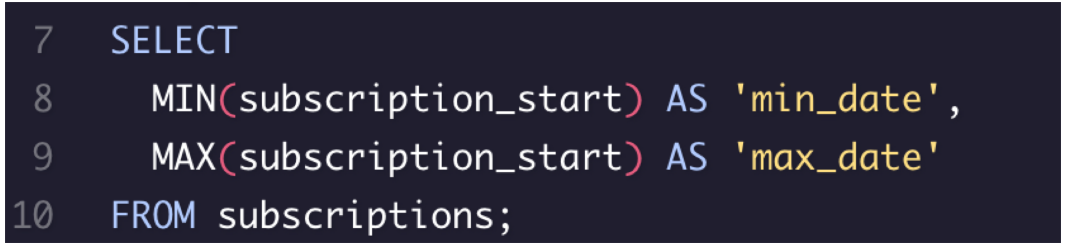
- The *churn rate* is the percentage of active users who cancelled their subscription within a given period, such as a month
- The formula used to calculate churn rate is: $\frac{no.of\ cancellations\ during\ period}{no.of\ active\ users\ at\ beginning\ of\ period}$
- In this context, *active users* is defined as users who are subscribed at the beginning of the month
- Calculating churn rates helps us understand how well a company retains its users

1.1 How many months has the company been operating? Which months do you have enough info to calculate a churn rate?

- The company has been operating for 4 months according to the dataset available
- We can only calculate churn rates for 3 months, from January through March.
- This is due to two factors:

1. No customers could have canceled in December as Codeflix has a minimum subscription length of 31 days
2. We can't calculate the number of active users at the beginning of December since 12-01 is the starting point of the dataset

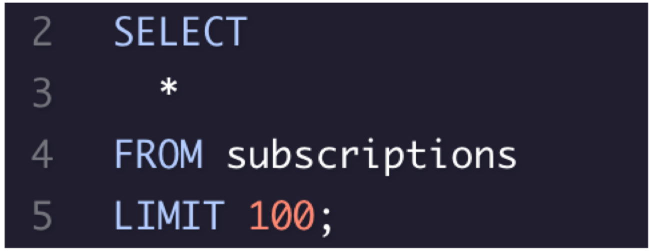
min_date	max_date
2016-12-01	2017-03-30



```sql
SELECT
  MIN(subscription_start) AS 'min_date',
  MAX(subscription_start) AS 'max_date'
FROM subscriptions;
```

1.2 What segments of users exist?

There are two segments of users: 87 and 30



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

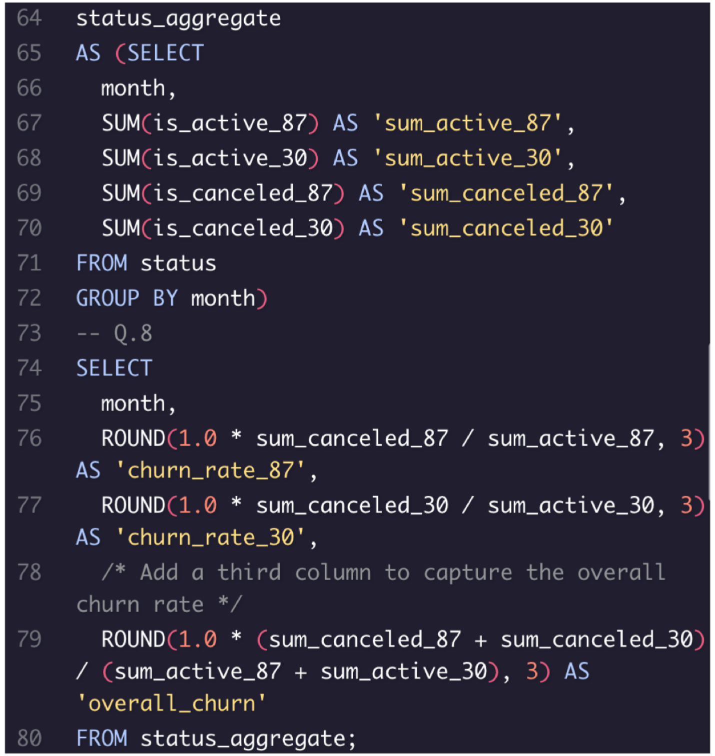
id	subscription_start	subscription_end	segment
1	2016-12-01	2017-02-01	87
2	2016-12-01	2017-01-24	87
99	2016-12-06		30
100	2016-12-06	2017-03-11	30

2. What is the overall churn trend since the company started? (1/2)

The overall churn trend is upward since the company started

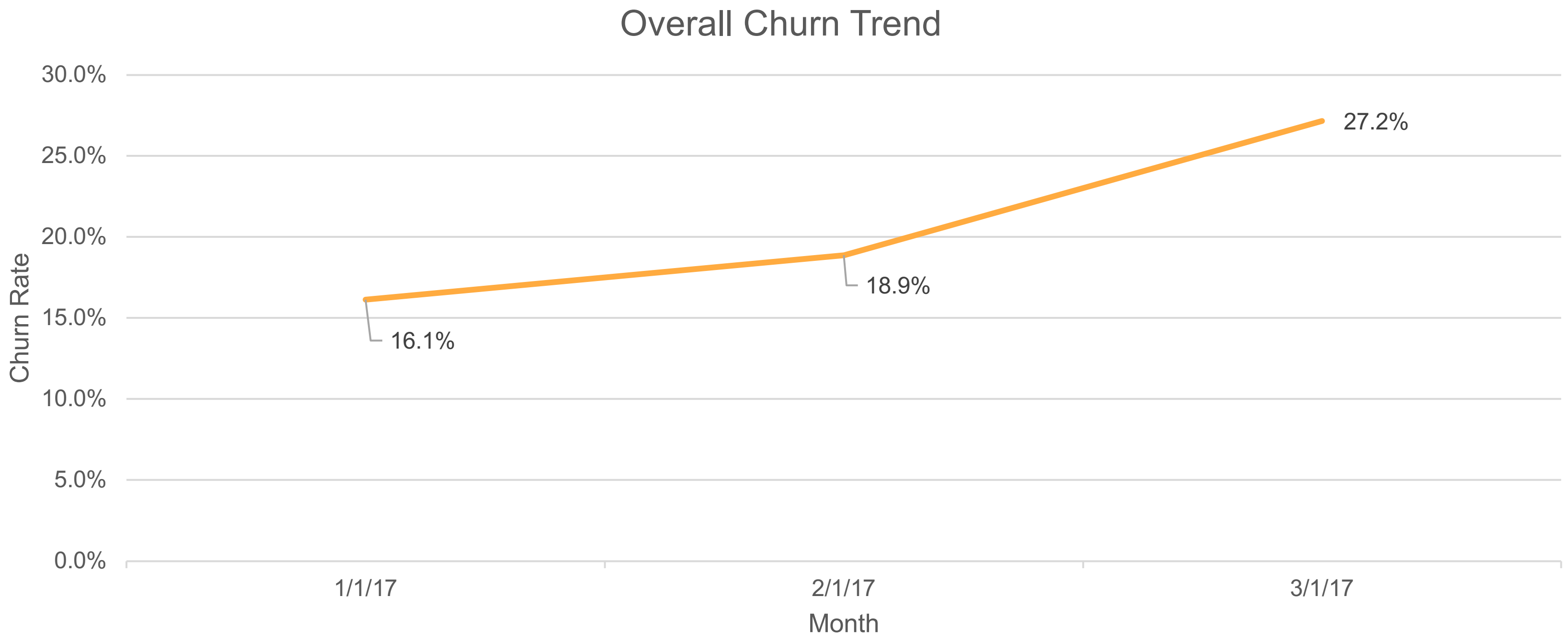
- The churn rate increased in each successive month
- Between January and March, the churn rate increased by 68%, from 16.1% to 27.2%

month	overall_churn
2017-01-01	16.1%
2017-02-01	18.9%
2017-03-01	27.2%



```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)
-- Q.8
SELECT
  month,
  ROUND(1.0 * sum_canceled_87 / sum_active_87, 3) AS 'churn_rate_87',
  ROUND(1.0 * sum_canceled_30 / sum_active_30, 3) AS 'churn_rate_30',
  /* Add a third column to capture the overall churn rate */
  ROUND(1.0 * (sum_canceled_87 + sum_canceled_30) / (sum_active_87 + sum_active_30), 3) AS 'overall_churn'
FROM status_aggregate;
```

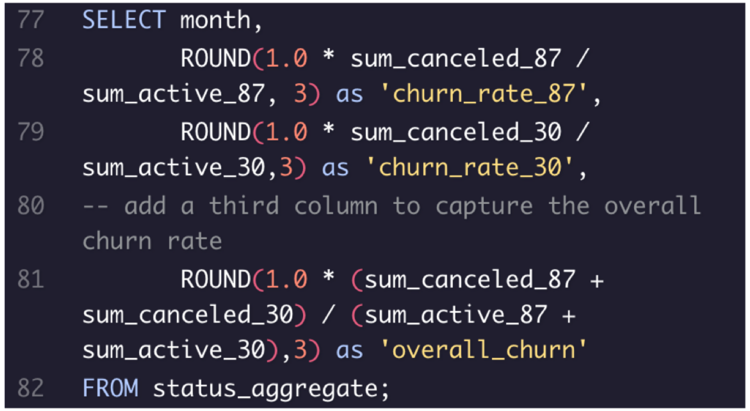
2. What is the overall churn trend since the company started? (2/2)



3.1 Compare the churn rates between user segments (1/2)

The churn rate of segment 87 is consistently higher than the churn rate of segment 30

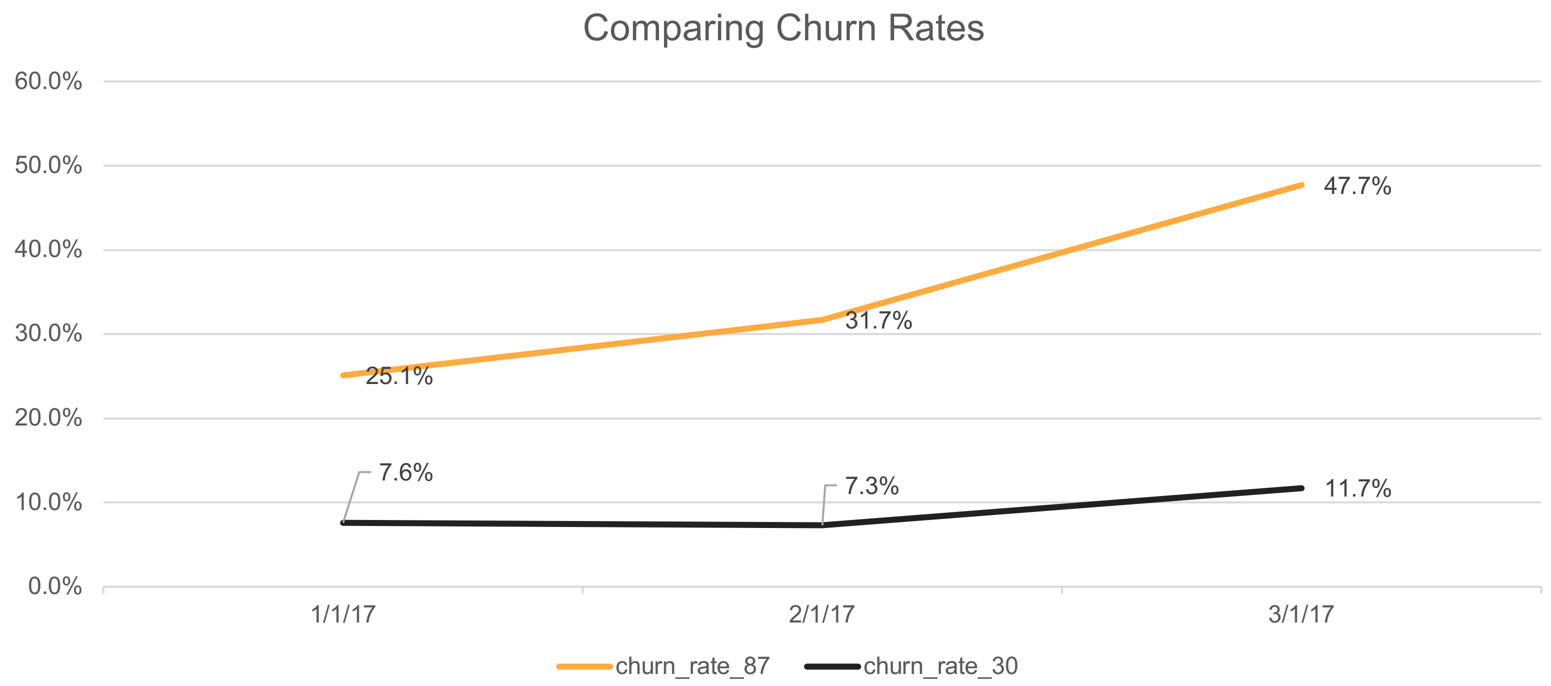
- The churn rate for segment 87 is between 3x – 4x that of segment 30 for the given data
- The churn rate for segment 30 actually decreased in February, whereas segment 87's churn rate increased
- Between January and March, the churn rate for segment 87 grew by 90% whereas the churn rate for segment 30 grew by 53%



```sql
SELECT month,
       ROUND(1.0 * sum_canceled_87 / sum_active_87, 3) as 'churn_rate_87',
       ROUND(1.0 * sum_canceled_30 / sum_active_30,3) as 'churn_rate_30',
-- add a third column to capture the overall churn rate
       ROUND(1.0 * (sum_canceled_87 + sum_canceled_30) / (sum_active_87 + sum_active_30),3) as 'overall_churn'
FROM status_aggregate;
```

month	churn_rate_87	churn_rate_30
1/1/17	25.10%	7.60%
2/1/17	31.70%	7.30%
3/1/17	47.70%	11.70%

3.1 Compare the churn rates between user segments (2/2)

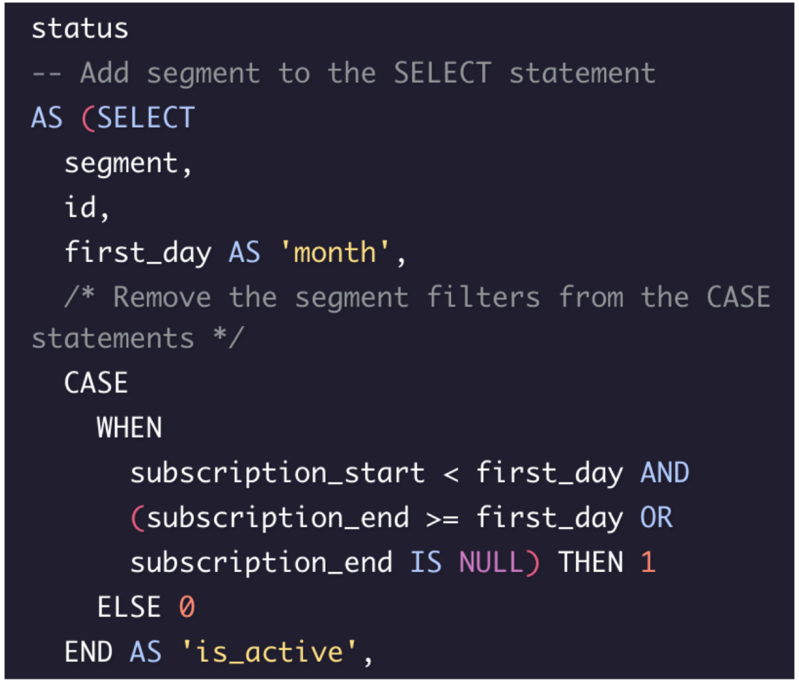


3.2 Which segment of users should the company focus on expanding?

- Due to the significant differences in churn rates between the two user segments, I would recommend that Codeflix focus on segment 30
- The customers in segment 30 are far less likely to cancel their subscriptions, as evidenced by the data on the previous slides
- It is largely the users in segment 87 that are driving the rapid increase in churn rate
- Users who who maintain their subscriptions are repeat customers and provide a consistent source of revenue
- All else held equal, money spent on acquiring users in segment 30 is likely to have a higher Return on Investment
- However, we would need to test this assertion by actually calculating ROI and other such metrics – this could be an interesting next step for this type of analysis

4. Bonus: How would you modify this code to support a large number of segments (1/2)

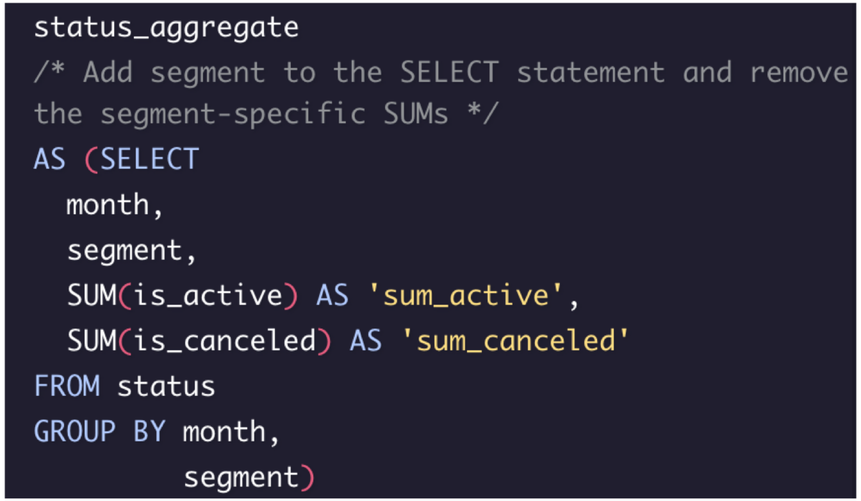
1. I would not hardcode the segment numbers in the ”status” temporary table (figure 1)
2. Instead, I would simply include segment in my SELECT statement and create two overall ”is_active” and “is_canceled” columns
3. I would then GROUP BY month *and* segment in the status_aggregate temporary table (figure 2)
4. Finally, I would include “segment” in my SELECT statement when calculating churn rates (figure 3)



```
status
-- Add segment to the SELECT statement
AS (SELECT
  segment,
  id,
  first_day AS 'month',
  /* Remove the segment filters from the CASE
statements */
  CASE
    WHEN
      subscription_start < first_day AND
      (subscription_end >= first_day OR
      subscription_end IS NULL) THEN 1
    ELSE 0
  END AS 'is_active',
```

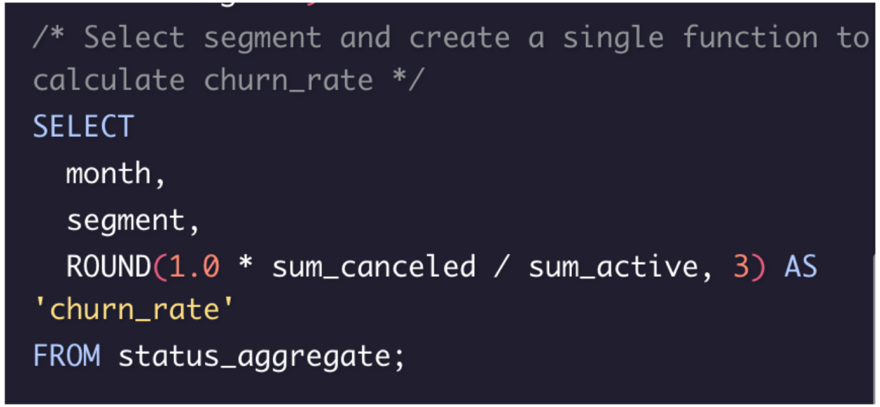
Figure 1

4. Bonus: How would you modify this code to support a large number of segments (2/2)



```sql
status_aggregate
/* Add segment to the SELECT statement and remove
the segment-specific SUMs */
AS (SELECT
  month,
  segment,
  SUM(is_active) AS 'sum_active',
  SUM(is_canceled) AS 'sum_canceled'
FROM status
GROUP BY month,
         segment)
```

Figure 2



```sql
/* Select segment and create a single function to
calculate churn_rate */
SELECT
  month,
  segment,
  ROUND(1.0 * sum_canceled / sum_active, 3) AS
'churn_rate'
FROM status_aggregate;
```

Figure 3