codecademy

Churn Rates with Codeflix

By jychoi910

Table of Contents

1. Get familiar with Codeflix

1.1 Get familiar with Codeflix

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

- 4 months from 2016-12-01 to 2017-03-31
- January, February, and March

```
/*
SELECT *
FROM users
LIMIT 10;
*/

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

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

1.2 Get familiar with Codeflix

What segment of users exist?

- Segment 30 and 87

```
SELECT *
FROM users
LIMIT 10;
```

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	Ø	30
14	2016-12-01	2017-03-07	30
15	2016-12-01	2017-02-22	30
16	2016-12-01	Ø	30
17	2016-12-01	Ø	30
18	2016-12-02	2017-01-29	87
19	2016-12-02	2017-01-13	87
20	2016-12-02	2017-01-15	87
21	2016-12-02	2017-01-15	87
22	2016-12-02	2017-01-24	87
23	2016-12-02	2017-01-14	87
24	2016-12-02	2017-01-18	87
25	2016-12-02	2017-02-24	87
26	2016-12-02	2017-01-18	87
27	2016-12-02	2017-01-11	87
28	2016-12-02	2017-03-30	30
29	2016-12-02	2017-02-11	30
30	2016-12-02	2017-01-20	30
31	2016-12-02	Ø	30
32	2016-12-02	2017-01-11	30
33	2016-12-02	Ø	30
34	2016-12-02	2017-02-06	30
35	2016-12-03	2017-02-17	87
36	2016-12-03	2017-03-06	87
37	2016-12-03	2017-03-08	87
38	2016-12-03	2017-02-28	87
39	2016-12-03	Ø	30
40	2016-12-03	Ø	30
41	2016-12-03	Ø	30
42	2016-12-03	2017-03-29	30
43	2016-12-03	Ø	30
44	2016-12-04	2017-03-11	87
45	2016-12-04	2017-02-02	87
46	2016-12-04	2017-02-18	87
47	2016-12-04	2017-02-06	87
48	2016-12-04	2017-03-12	87
49	2016-12-04	2017-03-06	87
50	2016-12-04	2017-02-15	87
51	2016-12-04	2017-01-06	87
52	2016-12-04	2017-02-21	87
53	2016-12-04	2017-01-31	87
54	2016-12-04	2017-03-08	87
55	2016-12-04	2017-02-21	87
56	2016-12-04	Ø	30
57	2016-12-04	Ø	30
58	2016-12-04	Ø	30
59	2016-12-04	Ø	30
60	2016-12-04	Ø	30
61	2016-12-04	Ø	30
62	2016-12-04	2017-03-11	30
63	2016-12-04	2017-01-14	30
64	2016-12-04	Ø	30
65	2016-12-04	Ø	30
66	2016-12-04	Ø	30
67	2016-12-04	Ø	30
68	2016-12-05	2017-01-13	87
69	2016-12-05	2017-02-15	87
70	2016-12-05	2017-03-12	87
71	2016-12-05	2017-01-13	87
72	2016-12-05	2017-01-29	87
73	2016-12-05	2017-01-20	87
74	2016-12-05	2017-01-09	87
75	2016-12-05	2017-02-25	87
76	2016-12-05	2017-01-28	87
77	2016-12-05	2017-02-09	87
78	2016-12-05	2017-01-23	87
79	2016-12-05	2017-01-27	87
80	2016-12-05	2017-01-11	87
81	2016-12-05	Ø	30
82	2016-12-05	Ø	30
83	2016-12-05	Ø	30
84	2016-12-05	Ø	30
85	2016-12-05	Ø	30
86	2016-12-05	2017-01-17	30
87	2016-12-05	Ø	30
88	2016-12-05	2017-03-26	30
89	2016-12-05	Ø	30
90	2016-12-06	2017-02-25	87
91	2016-12-06	2017-03-14	87
92	2016-12-06	2017-02-22	87
93	2016-12-06	2017-02-05	87
94	2016-12-06	2017-01-28	87
95	2016-12-06	2017-02-03	87
96	2016-12-06	2017-02-20	87
97	2016-12-06	2017-03-12	87
98	2016-12-06	2017-03-05	87
99	2016-12-06	Ø	30
100	2016-12-06	2017-03-11	30

2. What is the overall churn rate by month?

2.1 What is the overall churn rate by month?

What is the overall churn trend since the company started?

- The overall churn trend is rising, going from 16.17% for the month of January, to 18.98% for the month of February, and 27.43% for the month of March.

Query Results

month	churn_rate
2017-01-01	0.161687170474517
2017-02-01	0.189795918367347
2017-03-01	0.274258219727346

```
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,
  1.0 * canceled/active AS churn_rate
FROM status_aggregate;
```

3. Compare the churn rates by user segments

3.1 Compare the churn rates by user segments

Which segment of users should the company focus on expanding?

- The company should focus on users belonging to segment 87 since the churn rate is more than double than that of users from segment 30.

Query Results		
month	churn_rate_87	churn_rate_30
2017-01-01	0.251798561151079	0.0756013745704467
2017-02-01	0.32034632034632	0.0733590733590734
2017-03-01	0.485875706214689	0.11731843575419

3.2 Compare the churn rates by user segments

```
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 subscriptions.*, months.*
FROM subscriptions
CROSS JOIN months),
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),
status_aggregate AS
(SELECT
  month,
  SUM(is_active_87) as sum_active_87,
  SUM(is_canceled_87) as sum_canceled_87,
  SUM(is_active_30) as sum_active_30,
  SUM(is_canceled_30) as sum_canceled_30
FROM status
GROUP BY month)
SELECT
  month,
  1.0 * sum_canceled_87/sum_active_87 AS churn_rate_87,
  1.0 * sum_canceled_30/sum_active_30 AS churn_rate_30
FROM status_aggregate;
```