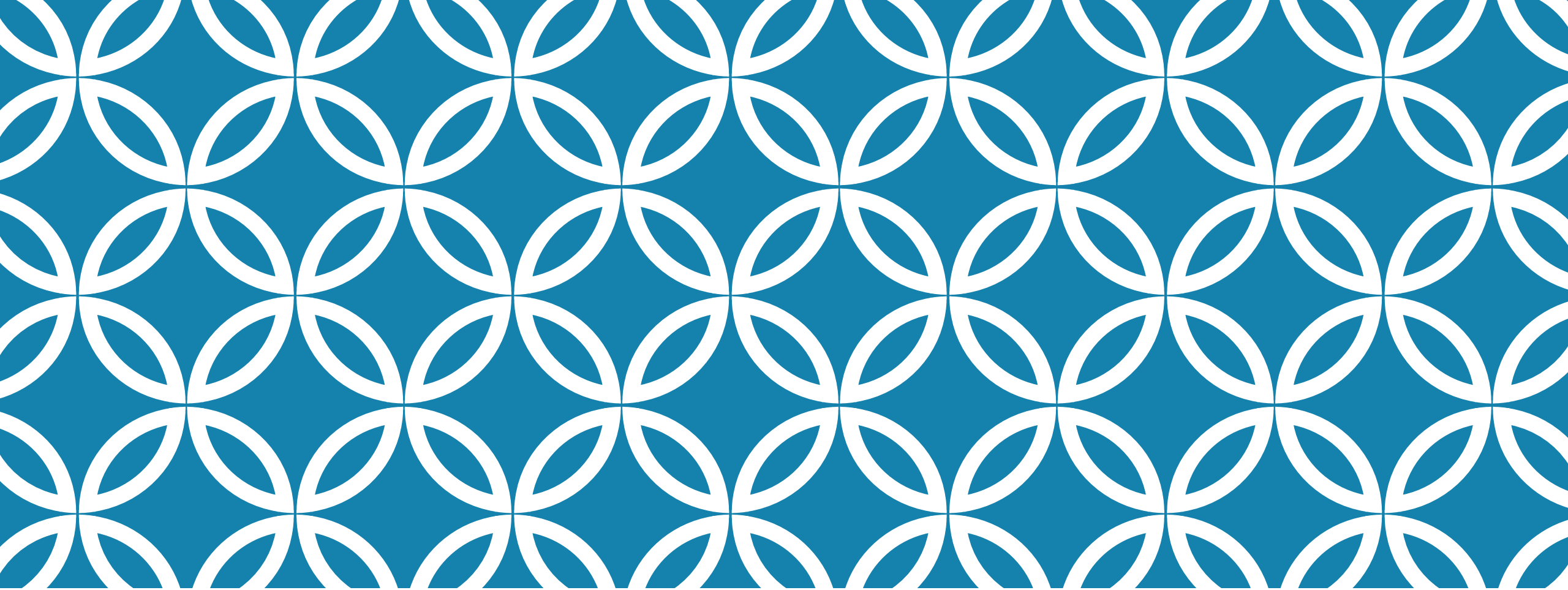


CODEFLIX

Churn Rate Project-LAURA CAMPERO

INTRODUCTION-CHURN RATE

Churn rate is the percent of subscribers that have canceled within a certain period, usually a month. For a user base to grow, the churn rate must be less than the new subscriber rate for the same period.

To calculate the churn rate, we only will be considering users who are subscribed at the beginning of the month. The churn rate is the number of these users who cancel during the month divided by the total number:

Churn Rate Formula: Cancellations/ Total Subscribers

PROJECT QUESTIONS -CODEFLIX

1. HOW MANY MONTHS HAS THE COMPANY BEEN OPERATING? WHICH MONTHS DO YOU HAVE ENOUGH INFORMATION TO CALCULATE A CHURN RATE?

2. WHAT SEGMENTS OF USERS EXIST?

3. WHAT IS THE OVERALL CHURN TREND SINCE THE COMPANY STARTED?

4. COMPARE THE CHURN RATES BETWEEN USER SEGMENTS.

PROJECT DESCRIPTION

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.

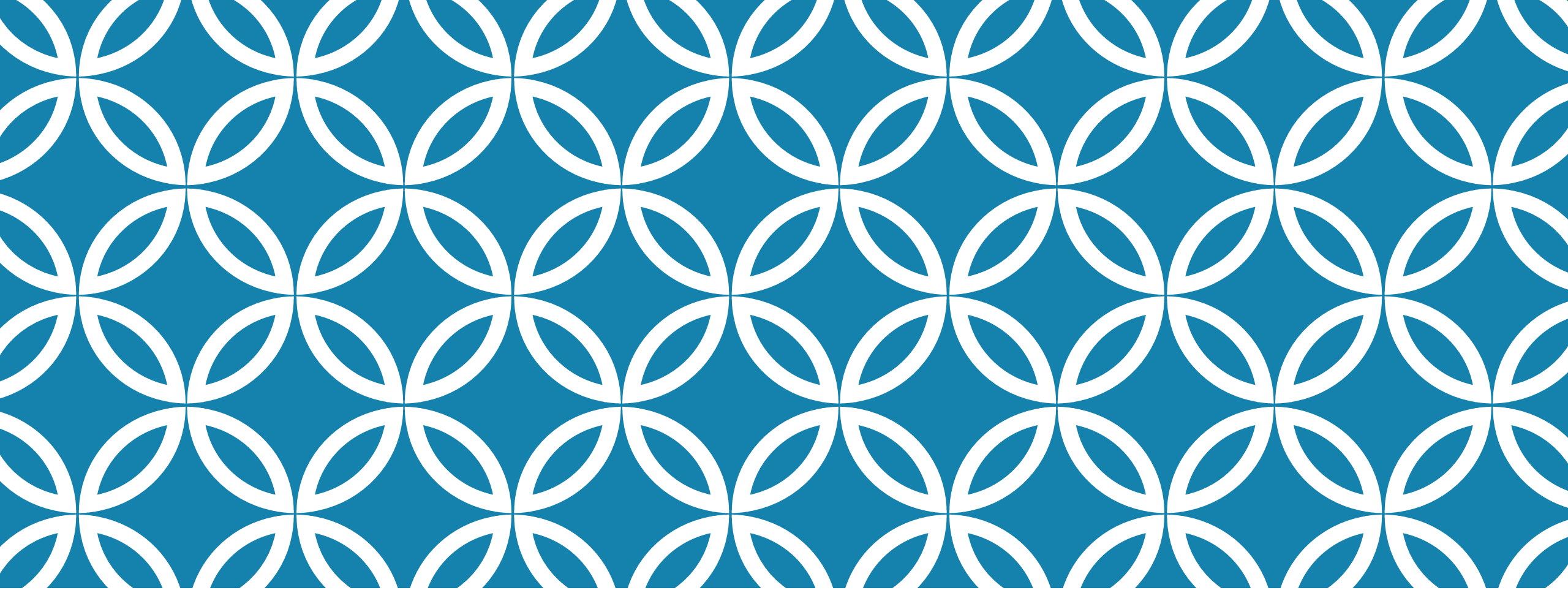
Dataset Provided: Table Subscription (columns below)

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



ANSWERS

Churn Rate Project-LAURA CAMPERO

1. HOW MANY MONTHS HAS THE COMPANY BEEN OPERATING? WHICH MONTHS DO YOU HAVE ENOUGH INFORMATION TO CALCULATE A CHURN RATE?

2. WHAT SEGMENTS OF USERS EXIST?

The Period that the company has been operating is **4 months** (FROM December 2016, to March 2017).

According to the provided data set we have information to calculate the **churn rate** for segments 87 and 30 **over the first 3 months of 2017**

(*) We can't calculate it for December, since there are no subscription_end values yet.

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

3. WHAT IS THE OVERALL CHURN TREND SINCE THE COMPANY STARTED?

4. COMPARE THE CHURN RATES BETWEEN USER SEGMENTS.

In order to Respond questions 3 and 4 we will follow below steps:

a) Create a **MONTHS** Temporary Table to indicate first day and last day of each month
b) Create a **CROSS_JOIN** Temporary table between Subscriptions and Month Tables in order to add last and first day of the month for each subscription
c) Create a **STATUS** Temporary Table to show status active and canceled subscription per month and segment (1= Active/canceled per respective column)
d) Create a **STATUS_AGREGATE** Temporary Table to add the columns active and canceled per segment per month
e) Calculate the **CHURN_RATE** by dividing Active/Canceled subscriptions by month by segment.

A) CREATE MONTHS TEMP.TABLE

```
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)
SELECT * FROM months;
```

first_day	last_day
2017-01-01	2017-01-31
2017-02-01	2017-02-28
2017-03-01	2017-03-31

B) CREATE CROSS_JOIN TEMP. TABLE

cross_join AS (SELECT * FROM subscriptions CROSS JOIN months),

id	subscription_start	subscription_end	segment	first_date	last_date
1	2016-12-01	2017-02-01	87	2017-01-01	2017-01-31
1	2016-12-01	2017-02-01	87	2017-02-01	2017-02-28
1	2016-12-01	2017-02-01	87	2017-03-01	2017-03-31
2	2016-12-01	2017-01-24	87	2017-01-01	2017-01-31
2	2016-12-01	2017-01-24	87	2017-02-01	2017-02-28
2	2016-12-01	2017-01-24	87	2017-03-01	2017-03-31
3	2016-12-01	2017-03-07	87	2017-01-01	2017-01-31
3	2016-12-01	2017-03-07	87	2017-02-01	2017-02-28
3	2016-12-01	2017-03-07	87	2017-03-01	2017-03-31

C) CREATE STATUS TEMP. TABLE

id	month	is_active_87	is_active_30	is_caceled_87	is_caceled_30
1	2017-01-01	1	0	0	0
1	2017-02-01	0	0	0	0
1	2017-03-01	0	0	0	0
2	2017-01-01	1	0	1	0
2	2017-02-01	0	0	0	0
2	2017-03-01	0	0	0	0
3	2017-01-01	1	0	0	0
3	2017-02-01	1	0	0	0
3	2017-03-01	1	0	1	0

```
Status AS (SELECT id ,first_day AS month,

CASE
WHEN (subscription_start < first_day) AND (segment = 87) AND
(subscription_end > first_day )OR subscription_end IS NULL THEN 1 ELSE 0 END
AS is_active_87,

CASE
WHEN subscription_start < first_day AND segment = 30 AND subscription_end >
first_day OR subscription_end IS NULL THEN 1 ELSE 0 END AS is_active_30 ,

CASE
WHEN subscription_end BETWEEN first_day AND last_day AND segment = 87
THEN 1 ELSE 0 END AS is_caceled_87,


CASE
WHEN subscription_end BETWEEN first_day AND last_day AND segment = 30
THEN 1 ELSE 0 END AS is_caceled_30 FROM cross_join)
```

D) CREATE STATUS_AGGREGATE TEMP.TABLE

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)

month	sum_active_87	sum_active_30	sum_canceled_87	sum_canceled_30
2017-01-01	1649	1469	70	22
2017-03-01	1628	1462	258	84

E) CHURN RATE

```
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  GROUP BY month;
```

3. WHAT IS THE OVERALL CHURN TREND SINCE THE COMPANY STARTED?
4. COMPARE THE CHURN RATES BETWEEN USER SEGMENTS.

The Overall CHURN trend shows an Increasing monthly Churn rate for both segments and it is higher for the segment 87 for every month.

month	churn_rate_87	churn_rate_30
2017-01-01	0.0424499696785931	0.0149761742682097
2017-02-01	0.0851063829787234	0.025554808338937 5
2017-03-01	0.158476658476658	0.057455540355677 2

COMMENTS ON BONUS TASK

(*) I couldn't submit the change in the code and . So was not sure if I understood the below question correctly.

How would you modify this code to support a large number of segments?

1- I modified the Status Temporary table to display 2 status (Active-Canceled)

2.

a) I created a temporary table only with the segments that are shown in subscription and Cross Join this table when calculating the Churn Rate

Status_aggregate AS (SELECT month, SUM (is_active) AS sum_active, SUM(is_canceled) AS sum_canceled FROM Status GROUP BY month),

Segments AS (SELECT segment FROM subscriptions)

SELECT Status_aggregate.month,Segments.Segment, 1.0 * sum_canceled/ sum_active AS churn_rate FROM status_aggregate CROSS JOIN Segments GROUP BY segment, month;

b) I calculated the Churn by eliminating the segments

c) Added segment column in the status table and then used Group By in the Churn Rate