Segmentation Operator tasks

Linh Tran

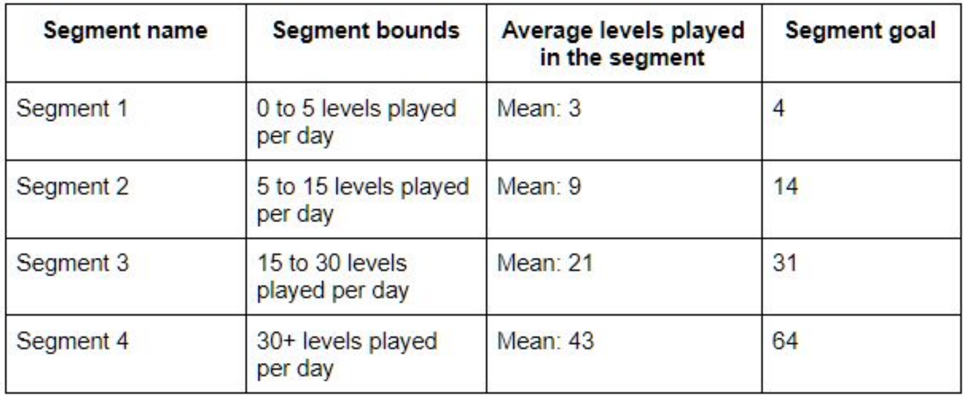
Context for the project

I am designing an event that is meant to be equally difficult for most players; the event will run on Friday. The goal is to encourage players to play more levels by introducing a personalised level win target (hard, but not impossible). To estimate event difficulty, I can use a measure we call “Activity lift” - required target compared to segment’s average levels played (mean or median, whichever I feel suits better); the project requires the activity lift of around 50% higher than segment’s average.

Tasks

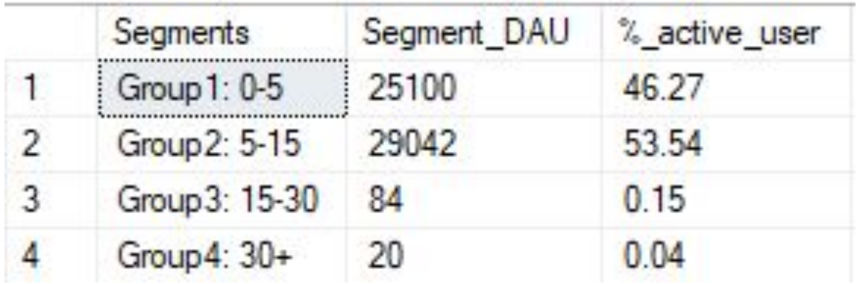
- Group users into segments based on levels played while keeping in mind that it is recommended to have 3 or 5 segments
- Set a target per segment. In other words, in each segment of users, set a number of levels I want users to play to increase their activity
- Calculate % of active users will fall into each segment
- Based on levels players typically played (could be calculated from the data), calculate target completion rate for each segment as well as calculate how many users would reach the 50% of the target (assume no lift from the event itself to make things simpler)

Segments and Goals for the event

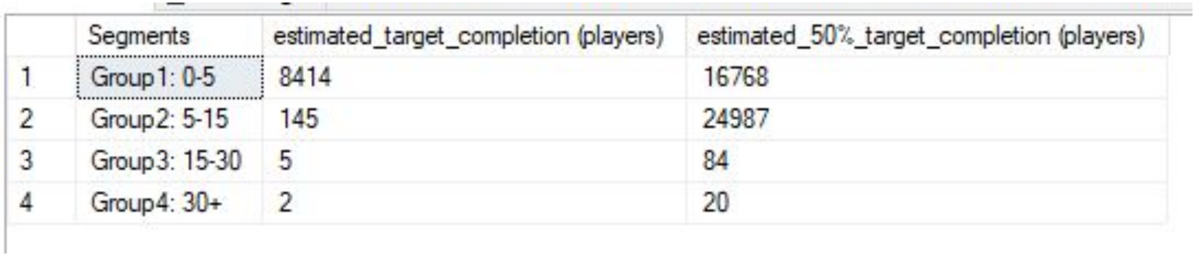


Segment name	Segment bounds	Average levels played in the segment	Segment goal
Segment 1	0 to 5 levels played per day	Mean: 3	4
Segment 2	5 to 15 levels played per day	Mean: 9	14
Segment 3	15 to 30 levels played per day	Mean: 21	31
Segment 4	30+ levels played per day	Mean: 43	64

Segments’ active users and predicted target completion

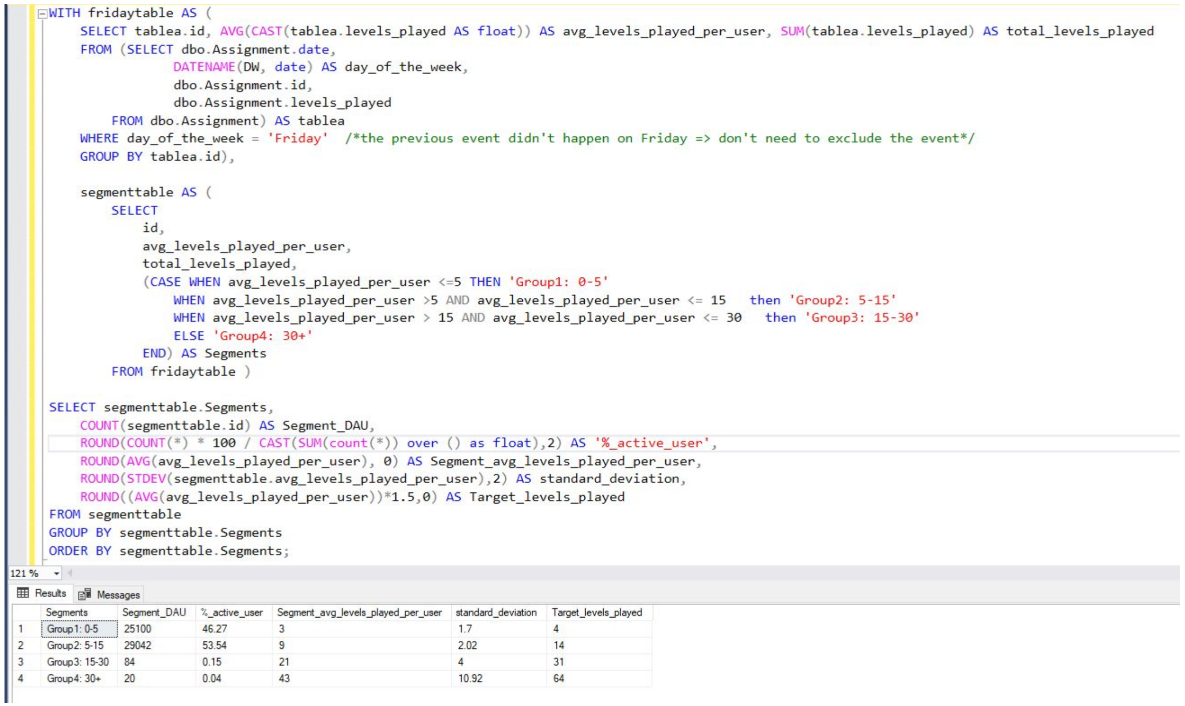


	Segments	Segment_DAU	%_active_user
1	Group1: 0-5	25100	46.27
2	Group2: 5-15	29042	53.54
3	Group3: 15-30	84	0.15
4	Group4: 30+	20	0.04



	Segments	estimated_target_completion (players)	estimated_50%_target_completion (players)
1	Group1: 0-5	8414	16768
2	Group2: 5-15	145	24987
3	Group3: 15-30	5	84
4	Group4: 30+	2	20

How segments' target and % active user were calculated



```sql
WITH fridaytable AS (
    SELECT tablea.id, AVG(CAST(tablea.levels_played AS float)) AS avg_levels_played_per_user, SUM(tablea.levels_played) AS total_levels_played
    FROM (SELECT dbo.Assignment.date,
                DATENAME(DW, date) AS day_of_the_week,
                dbo.Assignment.id,
                dbo.Assignment.levels_played
        FROM dbo.Assignment) AS tablea
    WHERE day_of_the_week = 'Friday'  /*the previous event didn't happen on Friday => don't need to exclude the event*/
    GROUP BY tablea.id),

    segmenttable AS (
        SELECT
            id,
            avg_levels_played_per_user,
            total_levels_played,
            (CASE WHEN avg_levels_played_per_user <=5 THEN 'Group1: 0-5'
                WHEN avg_levels_played_per_user >5 AND avg_levels_played_per_user <= 15   then 'Group2: 5-15'
                WHEN avg_levels_played_per_user > 15 AND avg_levels_played_per_user <= 30   then 'Group3: 15-30'
                ELSE 'Group4: 30+'
            END) AS Segments
        FROM fridaytable )

SELECT segmenttable.Segments,
    COUNT(segmenttable.id) AS Segment_DAU,
    ROUND(COUNT(*) * 100 / CAST(SUM(count(*)) over () as float),2) AS '%_active_user',
    ROUND(AVG(avg_levels_played_per_user), 0) AS Segment_avg_levels_played_per_user,
    ROUND(STDEV(segmenttable.avg_levels_played_per_user),2) AS standard_deviation,
    ROUND((AVG(avg_levels_played_per_user))*1.5,0) AS Target_levels_played
FROM segmenttable
GROUP BY segmenttable.Segments
ORDER BY segmenttable.Segments;
```

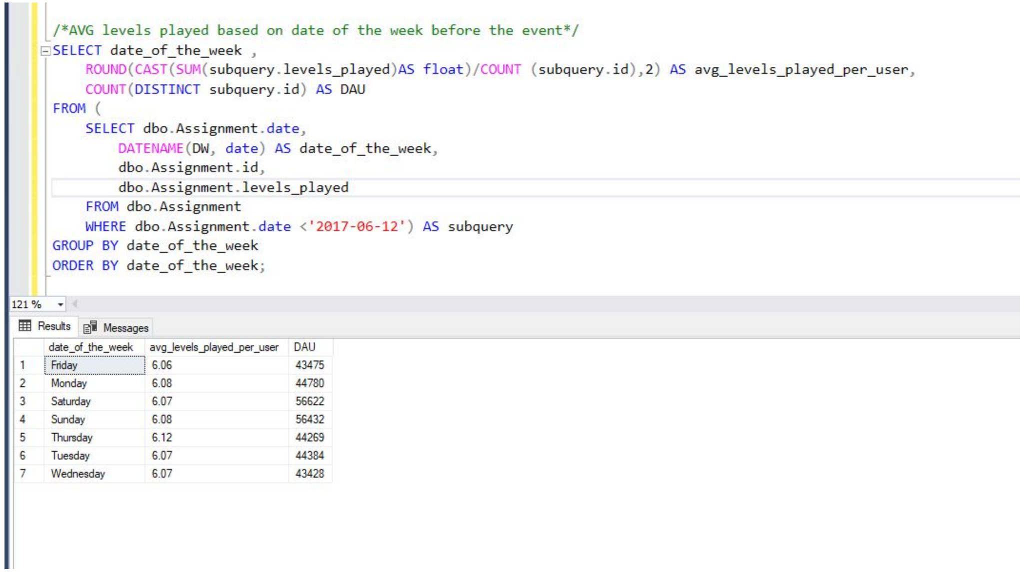
	Segments	Segment_DAU	%_active_user	Segment_avg_levels_played_per_user	standard_deviation	Target_levels_played
1	Group1: 0-5	25100	46.27	3	1.7	4
2	Group2: 5-15	29042	53.54	9	2.02	14
3	Group3: 15-30	84	0.15	21	4	31
4	Group4: 30+	20	0.04	43	10.92	64

This rather long query is used to create segment category and calculate active users of each category, % of active user, average levels played per user and target levels played for the event.

Note: Since the upcoming event will happen on Friday, I retrieved data of users only on Friday (this does not exclude the old event in the data since the old event didn't happen on Friday)

The target was (Activity lift) calculated by adding up 50% from the segment's average levels played per user.

How target completion was estimated (1)



```sql
/*AVG levels played based on date of the week before the event*/
SELECT date_of_the_week ,
    ROUND(CAST(SUM(subquery.levels_played)AS float)/COUNT (subquery.id),2) AS avg_levels_played_per_user,
    COUNT(DISTINCT subquery.id) AS DAU
FROM (
    SELECT dbo.Assignment.date,
        DATENAME(DW, date) AS date_of_the_week,
        dbo.Assignment.id,
        dbo.Assignment.levels_played
    FROM dbo.Assignment
    WHERE dbo.Assignment.date <'2017-06-12') AS subquery
GROUP BY date_of_the_week
ORDER BY date_of_the_week;
```

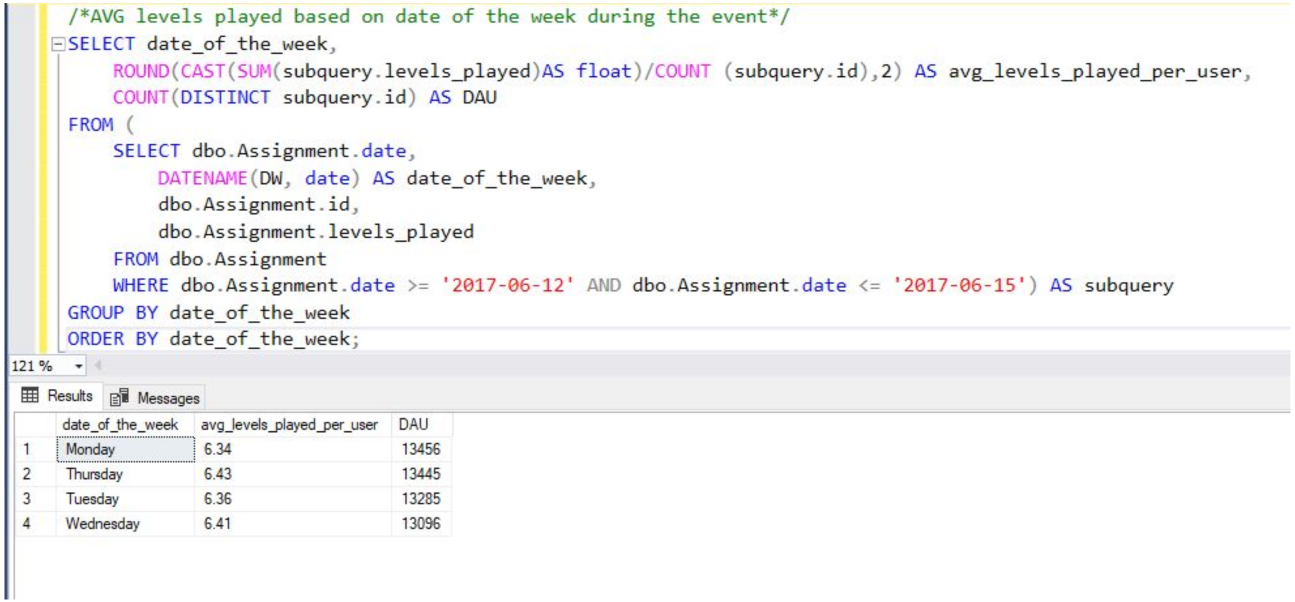
121 %

Results | Messages

	date_of_the_week	avg_levels_played_per_user	DAU
1	Friday	6.06	43475
2	Monday	6.08	44780
3	Saturday	6.07	56622
4	Sunday	6.08	56432
5	Thursday	6.12	44269
6	Tuesday	6.07	44384
7	Wednesday	6.07	43428

First step: I used this query to calculate the average levels played per user distributed on date of the week **before the old event** in the data

How target completion was estimated (2)



```sql
/*AVG levels played based on date of the week during the event*/
SELECT date_of_the_week,
    ROUND(CAST(SUM(subquery.levels_played)AS float)/COUNT (subquery.id),2) AS avg_levels_played_per_user,
    COUNT(DISTINCT subquery.id) AS DAU
FROM (
    SELECT dbo.Assignment.date,
        DATENAME(DW, date) AS date_of_the_week,
        dbo.Assignment.id,
        dbo.Assignment.levels_played
    FROM dbo.Assignment
    WHERE dbo.Assignment.date >= '2017-06-12' AND dbo.Assignment.date <= '2017-06-15') AS subquery
GROUP BY date_of_the_week
ORDER BY date_of_the_week;
```

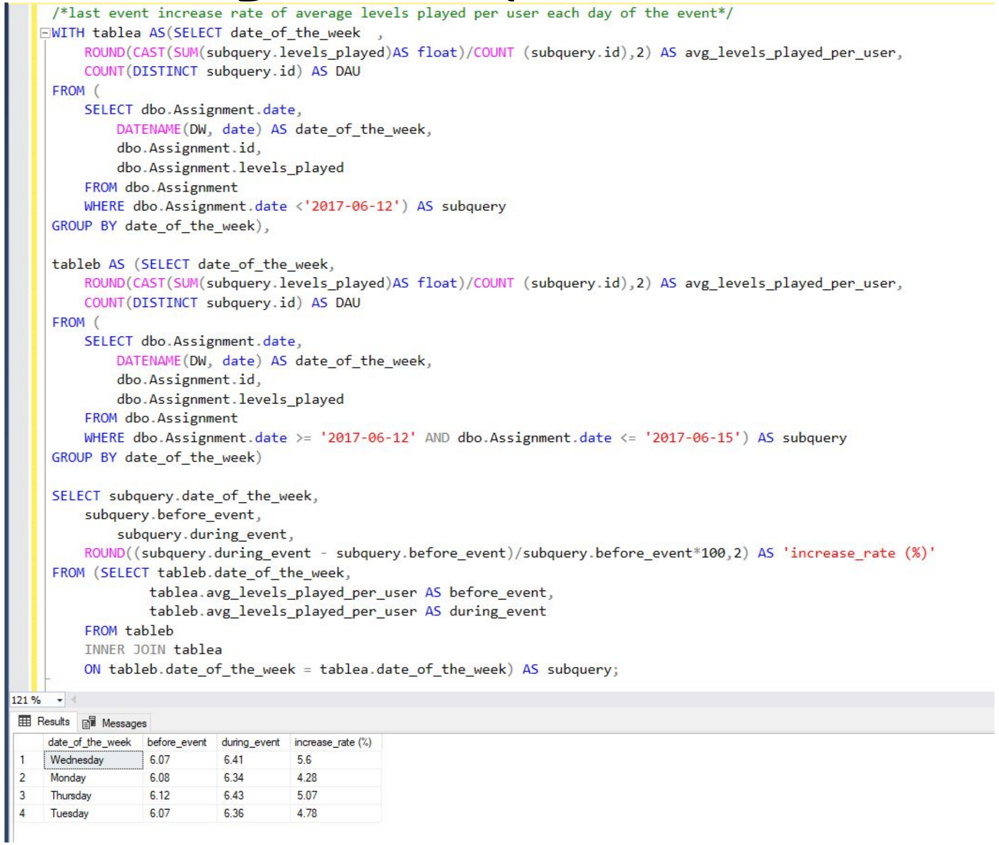
121 %

Results | Messages

	date_of_the_week	avg_levels_played_per_user	DAU
1	Monday	6.34	13456
2	Thursday	6.43	13445
3	Tuesday	6.36	13285
4	Wednesday	6.41	13096

Next step: I calculated the average levels played per user **during the 4 days of the old event** (also clarify the date of the week of 4 days of the old event)

How target completion was estimated (2)



```sql
/*last event increase rate of average levels played per user each day of the event*/
WITH tablea AS(SELECT date_of_the_week  ,
    ROUND(CAST(SUM(subquery.levels_played)AS float)/COUNT (subquery.id),2) AS avg_levels_played_per_user,
    COUNT(DISTINCT subquery.id) AS DAU
FROM (
    SELECT dbo.Assignment.date,
        DATENAME(DW, date) AS date_of_the_week,
        dbo.Assignment.id,
        dbo.Assignment.levels_played
    FROM dbo.Assignment
    WHERE dbo.Assignment.date <'2017-06-12') AS subquery
GROUP BY date_of_the_week),

tableb AS (SELECT date_of_the_week,
    ROUND(CAST(SUM(subquery.levels_played)AS float)/COUNT (subquery.id),2) AS avg_levels_played_per_user,
    COUNT(DISTINCT subquery.id) AS DAU
FROM (
    SELECT dbo.Assignment.date,
        DATENAME(DW, date) AS date_of_the_week,
        dbo.Assignment.id,
        dbo.Assignment.levels_played
    FROM dbo.Assignment
    WHERE dbo.Assignment.date >= '2017-06-12' AND dbo.Assignment.date <= '2017-06-15') AS subquery
GROUP BY date_of_the_week)

SELECT subquery.date_of_the_week,
    subquery.before_event,
        subquery.during_event,
    ROUND((subquery.during_event - subquery.before_event)/subquery.before_event*100,2) AS 'increase_rate (%)'
FROM (SELECT tableb.date_of_the_week,
            tablea.avg_levels_played_per_user AS before_event,
            tableb.avg_levels_played_per_user AS during_event
    FROM tableb
    INNER JOIN tablea
    ON tableb.date_of_the_week = tablea.date_of_the_week) AS subquery;
```

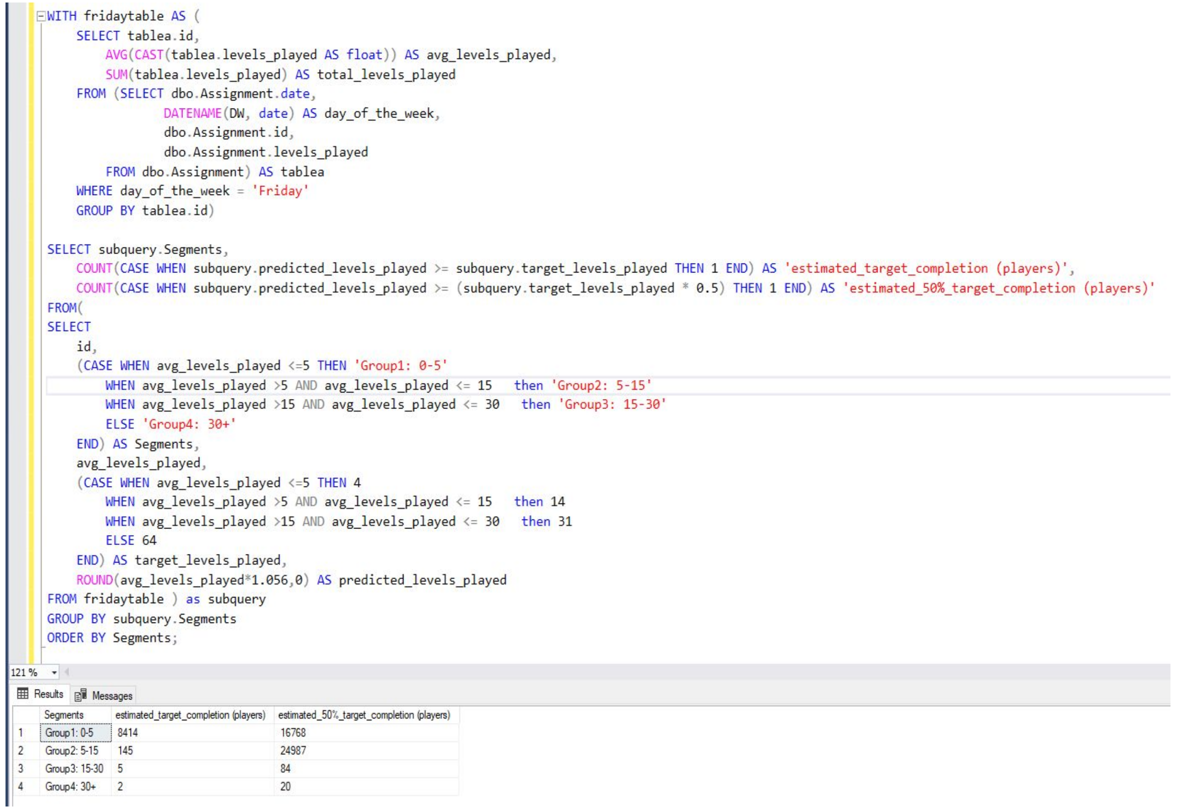
121 %

Results | Messages

	date_of_the_week	before_event	during_event	increase_rate (%)
1	Wednesday	6.07	6.41	5.6
2	Monday	6.08	6.34	4.28
3	Thursday	6.12	6.43	5.07
4	Tuesday	6.07	6.36	4.78

By combining the queries of 2 previous steps and using Inner Join, I wrote this long query to calculate the increase of average levels played per user between during the old event and the same date of the week before it.

How target completion was estimated (3)



```sql
WITH fridaytable AS (
    SELECT tablea.id,
        AVG(CAST(tablea.levels_played AS float)) AS avg_levels_played,
        SUM(tablea.levels_played) AS total_levels_played
    FROM (SELECT dbo.Assignment.date,
                DATENAME(DW, date) AS day_of_the_week,
                dbo.Assignment.id,
                dbo.Assignment.levels_played
        FROM dbo.Assignment) AS tablea
    WHERE day_of_the_week = 'Friday'
    GROUP BY tablea.id)

SELECT subquery.Segments,
    COUNT(CASE WHEN subquery.predicted_levels_played >= subquery.target_levels_played THEN 1 END) AS 'estimated_target_completion (players)',
    COUNT(CASE WHEN subquery.predicted_levels_played >= (subquery.target_levels_played * 0.5) THEN 1 END) AS 'estimated_50%_target_completion (players)'
FROM(
SELECT
    id,
    (CASE WHEN avg_levels_played <=5 THEN 'Group1: 0-5'
        WHEN avg_levels_played >5 AND avg_levels_played <= 15   then 'Group2: 5-15'
        WHEN avg_levels_played >15 AND avg_levels_played <= 30   then 'Group3: 15-30'
        ELSE 'Group4: 30+'
    END) AS Segments,
    avg_levels_played,
    (CASE WHEN avg_levels_played <=5 THEN 4
        WHEN avg_levels_played >5 AND avg_levels_played <= 15   then 14
        WHEN avg_levels_played >15 AND avg_levels_played <= 30   then 31
        ELSE 64
    END) AS target_levels_played,
    ROUND(avg_levels_played*1.056,0) AS predicted_levels_played
FROM fridaytable ) as subquery
GROUP BY subquery.Segments
ORDER BY Segments;
```

121 %

Results | Messages

	Segments	estimated_target_completion (players)	estimated_50%_target_completion (players)
1	Group1: 0-5	8414	16768
2	Group2: 5-15	145	24987
3	Group3: 15-30	5	84
4	Group4: 30+	2	20

So I had the rate of increase in the average of levels played per user of the old event. Since I'm optimistic, I decided to choose the highest increase rate in 4 days of the old event, which is 5.6%.

I wrote this query to calculate prediction of how many levels each player would play during the upcoming event (using 5.6% increase rate) and compare it with the goal of each segment. Each user, who has predicted levels played bigger than the goal of his/her group, will be counted as target completed. Each user, who has predicted levels bigger than 50% of the goal, will be counted as 50% target completed.