GLOBOX (SQL)

July 4, 2023 7:10 PM

A : Control **B : Treatment**

Tableau Link

https://public.tableau.com/app/profile/sneha.tayde.nov27/viz/Project-GLOBOX/Project-GLOBOX

Extract the A/B Test Data

```
SELECT
  u.id AS user_id,
  u.country,
  u.gender,
  g.device AS device,
  g.group AS test_group,
  CASE
    WHEN a.uid IS NULL THEN 0
    ELSE 1
  END AS converted,
  COALESCE(SUM(a.spent), 0) AS total_spent
FROM
  users u
  LEFT JOIN groups g ON u.id = g.uid
  LEFT JOIN activity a ON u.id = a.uid
GROUP BY
  u.id,
  u.country,
  u.gender,
  g.device,
  g.group,
  a.uid
ORDER BY
  u.id;
```

Novelty Effects

```
WITH subquery AS
(
  SELECT g.join_dt, g.group, COUNT(DISTINCT u.id) AS user_count
  FROM groups g
  INNER JOIN users u ON g.uid = u.id
  GROUP BY g.join_dt, g.group
),
converted_users AS
(
  SELECT a.dt, g.group, COUNT(DISTINCT g.uid) AS converted_user_count
  FROM activity a
  INNER JOIN groups g ON a.uid = g.uid AND a.device = g.device
  GROUP BY a.dt, g.group
),
combined_data AS
(
  SELECT s.join_dt, s.group, s.user_count, cu.converted_user_count,
       SUM(s.user_count) OVER (PARTITION BY s.group ORDER BY s.join_dt) AS
cum_users,
       SUM(cu.converted_user_count) OVER (PARTITION BY cu.group ORDER BY
cu.dt) AS cum_converted_users
  FROM subquery s
  LEFT JOIN converted_users cu ON s.group = cu.group AND s.join_dt = cu.dt
)
SELECT cd.join_dt AS "DATE",
     SUM(CASE WHEN cd.group = 'A' THEN cd.user_count ELSE 0 END) AS
"A_COUNT",
     SUM(CASE WHEN cd.group = 'B' THEN cd.user_count ELSE 0 END) AS
"B_COUNT",
     SUM(CASE WHEN cd.group = 'A' THEN cd.converted_user_count ELSE 0 END)
AS "A_CONVERTED",
     SUM(CASE WHEN cd.group = 'B' THEN cd.converted_user_count ELSE 0 END)
AS "B_CONVERTED",
     ROUND((SUM(CASE WHEN cd.group = 'A' THEN cd.cum_converted_users ELSE
0 END) / SUM(CASE WHEN cd.group = 'A' THEN cd.cum_users ELSE 0 END)), 5) AS
"A_COMBINED_CONVERSION_RATE",
     ROUND((SUM(CASE WHEN cd.group = 'B' THEN cd.cum_converted_users ELSE
0 END) / SUM(CASE WHEN cd.group = 'B' THEN cd.cum_users ELSE 0 END)), 5) AS
"B_COMBINED_CONVERSION_RATE",
     ROUND(((SUM(CASE WHEN cd.group = 'B' THEN cd.cum_converted_users
ELSE 0 END) / SUM(CASE WHEN cd.group = 'B' THEN cd.cum_users ELSE 0 END)) -
     (SUM(CASE WHEN cd.group = 'A' THEN cd.cum_converted_users ELSE 0 END)
/ SUM(CASE WHEN cd.group = 'A' THEN cd.cum_users ELSE 0 END))), 5) AS
"COMBINED_DIFFERENCE"
FROM combined_data cd
GROUP BY cd.join_dt
ORDER BY cd.join_dt;
```

Q1 : Can a user show up more than once in the activity table? Yes or no, and why?

```
SELECT uid, COUNT(*) as count
FROM activity
GROUP BY uid
HAVING COUNT(*) > 1;
```

YES. Because user can make purchase on different days and each purchase is a new separate row.

Q2 : What type of join should we use to join the users table to the activity table?

```
SELECT *
FROM users u
LEFT JOIN activity a ON u.id = a.uid
ORDER BY u.id;
```

LEFT JOIN

Q7 : What was the conversion rate of all users?

```
SELECT
  COUNT(DISTINCT a.uid) AS converted_users,
  COUNT(DISTINCT u.id) AS total_users,
  ROUND(
    COUNT(DISTINCT a.uid) * 100.0 / COUNT(DISTINCT u.id),
    2
  ) AS conversion_rate
FROM
  USERS u
  LEFT JOIN ACTIVITY a ON u.id = a.uid;
```

total_users : 48943
converted_users : 2094
conversion_rate : 4.28%

Q3 : What SQL function can we use to fill in NULL values?

```
SELECT
  u.id AS user_id,
  u.country,
  u.gender,
  COALESCE(g.device, 'N/A') AS device,
  g.group AS test_group,
  CASE WHEN a.uid IS NULL THEN 0 ELSE 1 END AS converted,
  COALESCE(SUM(a.spent), 0) AS total_spent
FROM users u
LEFT JOIN groups g ON u.id = g.uid
LEFT JOIN activity a ON u.id = a.uid
GROUP BY u.id, u.country, u.gender, g.device, g.group, a.uid
ORDER BY user_id;
```

COALESCE

Q4 : What are the start and end dates of the experiment?

```
SELECT
  MIN(join_dt) AS start_date,
  MAX(join_dt) AS end_date
FROM groups;
```

Start Date : 2023-01-25
End Date : 2023-02-06

Q5 : How many total users were in the experiment?

```
SELECT COUNT(DISTINCT id) AS total_users
FROM USERS;
```

48943

Q6 : How many users were in the control and treatment groups?

```
SELECT "group", COUNT(DISTINCT uid) AS group_count
FROM GROUPS
GROUP BY "group";
```

A : 24343
B : 24600

Q8 : What is the user conversion rate for the control and treatment groups?

```
SELECT
  g.group,
  COUNT(DISTINCT a.uid) AS converted_users,
  COUNT(DISTINCT u.id) AS total_users,
  ROUND(
    COUNT(DISTINCT a.uid) * 100.0 / COUNT(DISTINCT u.id),
    2
  ) AS conversion_rate
FROM
  USERS u
  JOIN GROUPS g ON u.id = g.uid
  LEFT JOIN ACTIVITY a ON u.id = a.uid
GROUP BY
  g.group;
```

group	total_users	converted_users	converted_rate
A	24343	955	3.92%
B	24600	1139	4.63%

Q9 : What is the average amount spent per user for the control and treatment groups, including users who did not convert?

```
SELECT g.group,
    COUNT(DISTINCT g.uid) AS total_users,
    CONCAT('$', ROUND(AVG(COALESCE(a.spent::numeric, 0)), 2))
AS average_amount_spent
FROM GROUPS g
LEFT JOIN ACTIVITY a ON g.uid = a.uid
GROUP BY g.group;
```

group	total_users	average_amount_spent
A	24343	$3.37
B	24600	$3.38

Q10 : Why does it matter to include users who did not convert when calculating the average amount spent per user?

It provides a more accurate representation of the revenue generated per user and better understanding of potential conversion.

Join Curve

```
WITH all_dates AS (
  SELECT DISTINCT join_dt AS date
  FROM GROUPS
  UNION
  SELECT DISTINCT dt AS date
  FROM ACTIVITY
),
daily_users AS (
  SELECT
    all_dates.date,
    COUNT(DISTINCT GROUPS.uid) AS number_of_users
  FROM
    all_dates
  LEFT JOIN GROUPS ON all_dates.date = GROUPS.join_dt
  GROUP BY
    all_dates.date
  ORDER BY
    all_dates.date
),
cumulative_users AS (
  SELECT
    date,
    SUM(number_of_users) OVER (ORDER BY date) AS cumulative_users
  FROM
    daily_users
)
SELECT
  cumulative_users.date,
  cumulative_users.cumulative_users,
  COUNT(DISTINCT GROUPS.uid) AS number_of_users
FROM
```

```
  cumulative_users
LEFT JOIN GROUPS ON cumulative_users.date = GROUPS.join_dt
GROUP BY
  cumulative_users.date,
  cumulative_users.cumulative_users
ORDER BY
  cumulative_users.date


WITH all_dates AS (
  SELECT DISTINCT join_dt AS date
  FROM GROUPS
  UNION
  SELECT DISTINCT dt AS date
  FROM ACTIVITY
),
daily_users AS (
  SELECT
    all_dates.date,
    COUNT(DISTINCT GROUPS.uid) AS number_of_users
  FROM
    all_dates
  LEFT JOIN GROUPS ON all_dates.date = GROUPS.join_dt
  GROUP BY
    all_dates.date
  ORDER BY
    all_dates.date
),
cumulative_users AS (
  SELECT
    date,
    SUM(number_of_users) OVER (ORDER BY date) AS cumulative_users
  FROM
    daily_users
)
SELECT
  cumulative_users.date,
  cumulative_users.cumulative_users,
  COUNT(DISTINCT GROUPS.uid) AS number_of_users
FROM
  cumulative_users
LEFT JOIN GROUPS ON cumulative_users.date = GROUPS.join_dt
GROUP BY
  cumulative_users.date,
  cumulative_users.cumulative_users
ORDER BY
  cumulative_users.date
```