

Case Study: AB Testing

Observation of the table

```
SELECT *
FROM dsv1069.final_assignments_qa
LIMIT 100
```

index	item_id	test_a	test_b	test_c	test_d	test_e	test_f
0	2512.0	1.0	0.0	1.0	1.0	0.0	1.0
1	482.0	0.0	1.0	1.0	1.0	0.0	0.0
2	2446.0	0.0	1.0	1.0	0.0	1.0	0.0
3	1312.0	0.0	0.0	0.0	0.0	0.0	1.0
4	3556.0	1.0	1.0	0.0	1.0	0.0	0.0
...
95	3610.0	0.0	1.0	1.0	0.0	0.0	1.0
96	1205.0	0.0	0.0	0.0	1.0	1.0	1.0
97	1585.0	0.0	0.0	0.0	0.0	0.0	0.0
98	1315.0	0.0	1.0	1.0	0.0	1.0	0.0
99	1368.0	1.0	0.0	1.0	1.0	1.0	1.0

100 rows × 7 columns

Part 1: Reformating the 'final_assignments_qa' table for futher analysis. Table should have everything to compute metrics like 30-day view-binary

```
SELECT item_id,
       test_a AS test_assignment,
       (CASE
        WHEN test_a is not null THEN 'item_test_1'
        END) AS test_number,
       (CASE
        WHEN test_a is not null THEN '2013-01-05 00:00:00'
        END) AS test_start_date
FROM dsv1069.final_assignments_qa
UNION
SELECT item_id,
       test_b AS test_assignment,
       (CASE
        WHEN test_b is not null THEN 'item_test_2'
        END) AS test_number,
       (CASE
        WHEN test_b is not null THEN '2013-01-05 00:00:00'
        END) AS test_start_date
```

Case Study: AB Testing

```
END) AS test_start_date
FROM dsv1069.final_assignments_qa
UNION
SELECT item_id,
       test_c AS test_assignment,
       (CASE
         WHEN test_c is not null THEN 'item_test_3'
       END) AS test_number,
       (CASE
         WHEN test_c is not null THEN '2013-01-05 00:00:00'
       END) AS test_start_date
FROM dsv1069.final_assignments_qa
UNION
SELECT item_id,
       test_d AS test_assignment,
       (CASE
         WHEN test_d is not null THEN 'item_test_4'
       END) AS test_number,
       (CASE
         WHEN test_d is not null THEN '2013-01-05 00:00:00'
       END) AS test_start_date
FROM dsv1069.final_assignments_qa
UNION
SELECT item_id,
       test_d AS test_assignment,
       (CASE
         WHEN test_d is not null THEN 'item_test_5'
       END) AS test_number,
       (CASE
         WHEN test_d is not null THEN '2013-01-05 00:00:00'
       END) AS test_start_date
FROM dsv1069.final_assignments_qa
UNION
SELECT item_id,
       test_e AS test_assignment,
       (CASE
         WHEN test_e is not null THEN 'item_test_6'
       END) AS test_number,
       (CASE
         WHEN test_e is not null THEN '2013-01-05 00:00:00'
       END) AS test_start_date
FROM dsv1069.final_assignments_qa
UNION
SELECT item_id,
       test_f AS test_assignment,
       (CASE
         WHEN test_f is not null THEN 'item_test_7'
       END) AS test_number,
       (CASE
         WHEN test_f is not null THEN '2013-01-05 00:00:00'
       END) AS test_start_date
```

Case Study: AB Testing

```
FROM dsv1069.final_assignments_qa
ORDER BY test_number
```

index	item_id	test_assignment	test_number	test_start_date
0	210.0	0.0	item_test_1	2013-01-05 00:00:00
1	3825.0	0.0	item_test_1	2013-01-05 00:00:00
2	2827.0	1.0	item_test_1	2013-01-05 00:00:00
3	2542.0	1.0	item_test_1	2013-01-05 00:00:00
4	719.0	0.0	item_test_1	2013-01-05 00:00:00
...
95	2363.0	1.0	item_test_1	2013-01-05 00:00:00
96	290.0	1.0	item_test_1	2013-01-05 00:00:00
97	3965.0	0.0	item_test_1	2013-01-05 00:00:00
98	3673.0	1.0	item_test_1	2013-01-05 00:00:00
99	2590.0	0.0	item_test_1	2013-01-05 00:00:00

100 rows × 4 columns

Part 2: Calculate the order binary, and average views for the 30 day window after the test assignment for the 2 test

```
SELECT
  test_assignment,
  SUM(order_binary) AS orders_completed_30d,
  COUNT(DISTINCT item_id) AS items,
  SUM(orders)/COUNT(item_id) AS average_views_per_item
FROM
  (SELECT
    test_events.item_id,
    test_events.test_assignment,
    test_events.order_date,
    test_events.test_number,
    test_events.test_date,
    COUNT(invoice_id) AS orders,
    MAX(CASE WHEN (order_date > test_events.test_date AND DATE_PART('day', order_date - test_date) <= 30)
      THEN 1 ELSE 0 END) AS order_binary
  FROM
    (SELECT
      A.item_id AS item_id,
      test_assignment,
      test_number,
      test_start_date AS test_date,
      created_at AS order_date,
      invoice_id
    FROM
```

Case Study: AB Testing

```
dsv1069.final_assignments AS A
LEFT JOIN
  dsv1069.orders AS O
ON
  A.item_id = O.item_id
WHERE
  test_number = 'item_test_2'
) AS test_events
GROUP BY
  test_events.item_id,
  test_events.test_assignment,
  test_events.order_date,
  test_events.test_number,
  test_events.test_date
) AS order_binary
GROUP BY test_assignment
```

index	test_assignment	orders_completed_30d	items	average_views_per_item
0	0.0	399	1130	1.004173
1	1.0	381	1068	1.004659

Part 3: Calculate the view binary, and average views for the 30 day window after the test assignment for the 2 test

```
SELECT
  test_assignment,
  SUM(view_binary) AS view_binary_30d,
  COUNT(DISTINCT item_id) AS items,
  SUM(events)/COUNT(item_id) AS average_views_per_item
FROM
  (SELECT
    test_events.item_id,
    test_events.test_assignment,
    test_events.test_number,
    test_events.test_date,
    COUNT(event_id) AS events,
    MAX(CASE
      WHEN (event_time > test_events.test_date AND DATE_PART('day', event_time - test_date) <= 30)
      THEN 1 ELSE 0 END) AS view_binary
  FROM
    (SELECT
      A.item_id AS item_id,
      test_assignment,
      test_number,
      test_start_date AS test_date,
      event_time,
      event_id
    FROM
```

Case Study: AB Testing

```
dsv1069.final_assignments AS A
LEFT JOIN
(SELECT
  event_time,
  event_id,
  (CASE
    WHEN parameter_name = 'item_id' THEN CAST(parameter_value AS INT)
    ELSE null
  END) AS item_id
FROM
  dsv1069.events
WHERE
  event_name = 'view_item') AS views
ON
  A.item_id =views.item_id
WHERE
  test_number = 'item_test_2'
) AS test_events
GROUP BY
  test_events.item_id,
  test_events.test_assignment,
  test_events.test_number,
  test_events.test_date
) AS views_binary
GROUP BY
  test_assignment,
  test_date
```

index	test_assignment	view_binary_30d	items	average_views_per_item
0	0.0	918	1130	119.762832
1	1.0	890	1068	119.338951

Part 4: Compute the lifts in metrics and the p-values for the binary metrics using a interval 95% confidence.

I used Abba for interpreting the results of binomial experiment/test.

Results:

- *Order binary*
We have not collected enough samples to be able to detect statistically significant lift of 1%. The p-value is 0.86 and the true mean is likely to be between -10% and 12%. This result is not statistically significant. There is a no substantial difference in the number of orders within 30days of the assigned treatment date between the two treatments.
- *View Binary*
We have not collected enough samples to be able to detect statistically significant lift of 2,6%. The p-value is 0.2 and the true mean is likely to be between -1.4% to 6.5%. This result is statistically significant. However, there is still not a substantial difference in the number of views within 30days of the assigned treatment date between the two treatments.

Case Study: AB Testing