

Final Project: A/B Testing of an e-Commerce Site

We are running an experiment at an item-level, which means all users who visit will see the same page, but the layout of different item pages may differ. Please follow the steps below and good luck!

Task 1: Compare the `final_assignments_qa` table to the assignment events we captured for `user_level_testing`. Write an answer to the following question: Does this table have everything you need to compute metrics like 30-day view-binary?

```
1. SELECT *
2. FROM dsv1069.final_assignments_qa LIMIT 5;
3.
4. SELECT item_id,
5.        test_a AS test_assignment,
6.        (CASE
7.          WHEN test_a is NOT NULL
8.          THEN 'test_a'
9.          ELSE NULL
10.         END) AS test_number
11. FROM dsv1069.final_assignments_qa
12. UNION
13. SELECT item_id,
14.        test_b AS test_assignment,
15.        (CASE
16.          WHEN test_b is NOT NULL
17.          THEN 'test_b'
18.          ELSE NULL
19.         END) AS test_number
20. FROM dsv1069.final_assignments_qa
21. UNION
22. SELECT item_id,
23.        test_c AS test_assignment,
24.        (CASE
25.          WHEN test_c is NOT NULL
26.          THEN 'test_c'
27.          ELSE NULL
28.         END) AS test_number
29. FROM dsv1069.final_assignments_qa
30. UNION
31. SELECT item_id,
32.        test_d AS test_assignment,
33.        (CASE
34.          WHEN test_d is NOT NULL
35.          THEN 'test_d'
36.          ELSE NULL
37.         END) AS test_number
38. FROM dsv1069.final_assignments_qa
39. UNION
40. SELECT item_id,
41.        test_e AS test_assignment,
42.        (CASE
43.          WHEN test_e is NOT NULL
44.          THEN 'test_e'
45.          ELSE NULL
46.         END) AS test_number
47. FROM dsv1069.final_assignments_qa
48. UNION
49. SELECT item_id,
50.        test_f AS test_assignment,
51.        (CASE
52.          WHEN test_f is NOT NULL
53.          THEN 'test_f'
54.          ELSE NULL
55.         END) AS test_number
56. FROM dsv1069.final_assignments_qa
```

```
57. ORDER BY item_id DESC
58. LIMIT 5;
```

	item_id	test_assignment	test_number
1	3997	0	test_b
2	3997	1	test_f
3	3997	1	test_c
4	3997	0	test_d
5	3997	1	test_e

Comment: *We aren't given any kind of dates or time-frame to compute the 30-day metrics.*

Task 2: Write a query and table creation statement to make final_assignments_qa look like the final_assignments table. If you discovered something missing in part 1, you may fill in the value with a place holder of the appropriate data type.

```
1. SELECT item_id,
2.     test_a AS test_assignment,
3.     (CASE
4.         WHEN test_a is NOT NULL
5.         THEN 'test_a'
6.         ELSE NULL
7.     END) AS test_number,
8.     (CASE
9.         WHEN test_a is NOT NULL
10.        THEN '2013-01-05 00:00:00'
11.        ELSE NULL
12.    END) AS test_start_date
13. FROM dsv1069.final_assignments_qa
14. UNION
15. SELECT item_id,
16.     test_b AS test_assignment,
17.     (CASE
18.         WHEN test_b is NOT NULL
19.         THEN 'test_b'
20.         ELSE NULL
21.     END) AS test_number,
22.     (CASE
23.         WHEN test_b is NOT NULL
24.         THEN '2016-01-07 00:00:00'
25.         ELSE NULL
26.     END) AS test_start_date
27. FROM dsv1069.final_assignments_qa
28. UNION
29. SELECT item_id,
30.     test_c AS test_assignment,
31.     (CASE
32.         WHEN test_c is NOT NULL
33.         THEN 'test_c'
34.         ELSE NULL
35.     END) AS test_number,
36.     (CASE
37.         WHEN test_c is NOT NULL
38.         THEN '2015-03-14 00:00:00'
39.         ELSE NULL
40.     END) AS test_start_date
41. FROM dsv1069.final_assignments_qa
42. UNION
43. SELECT item_id,
44.     test_d AS test_assignment,
45.     (CASE
```

```

46.         WHEN test_d is NOT NULL
47.         THEN 'test_d'
48.         ELSE NULL
49.     END) AS test_number,
50.     (CASE
51.         WHEN test_d is NOT NULL
52.         THEN '2013-01-06 00:00:00'
53.         ELSE NULL
54.     END) AS test_start_date
55. FROM dsv1069.final_assignments_qa
56. UNION
57. SELECT item_id,
58.        test_e AS test_assignment,
59.        (CASE
60.            WHEN test_e is NOT NULL
61.            THEN 'test_e'
62.            ELSE NULL
63.        END) AS test_number,
64.        (CASE
65.            WHEN test_e is NOT NULL
66.            THEN '2016-01-08 00:00:00'
67.            ELSE NULL
68.        END) AS test_start_date
69. FROM dsv1069.final_assignments_qa
70. UNION
71. SELECT item_id,
72.        test_f AS test_assignment,
73.        (CASE
74.            WHEN test_f is NOT NULL
75.            THEN 'test_f'
76.            ELSE NULL
77.        END) AS test_number,
78.        (CASE
79.            WHEN test_f is NOT NULL
80.            THEN '2015-03-15 00:00:00'
81.            ELSE NULL
82.        END) AS test_start_date
83. FROM dsv1069.final_assignments_qa
84. ORDER BY test_number
85. LIMIT 5;

```

	item_id	test_assignment	test_number	test_start_date
1	406	1 test_a		2013-01-05 00:00:00
2	2700	0 test_a		2013-01-05 00:00:00
3	7	0 test_a		2013-01-05 00:00:00
4	2133	0 test_a		2013-01-05 00:00:00
5	3520	1 test_a		2013-01-05 00:00:00

Task 3: Use the final_assignments table to calculate the order binary for the 30 day window after the test assignment for item_test_2 (You may include the day the test started)

Part 1:

```

1. SELECT final_assignments.item_id AS item_id,
2.        final_assignments.test_assignment,
3.        (CASE
4.            WHEN (DATE(created_at) > DATE(test_start_date)
5.            AND (DATE(created_at) - DATE(test_start_date)) <= 30)
6.            THEN DATE(created_at)
7.        END) AS order_date,
8.        MAX(CASE

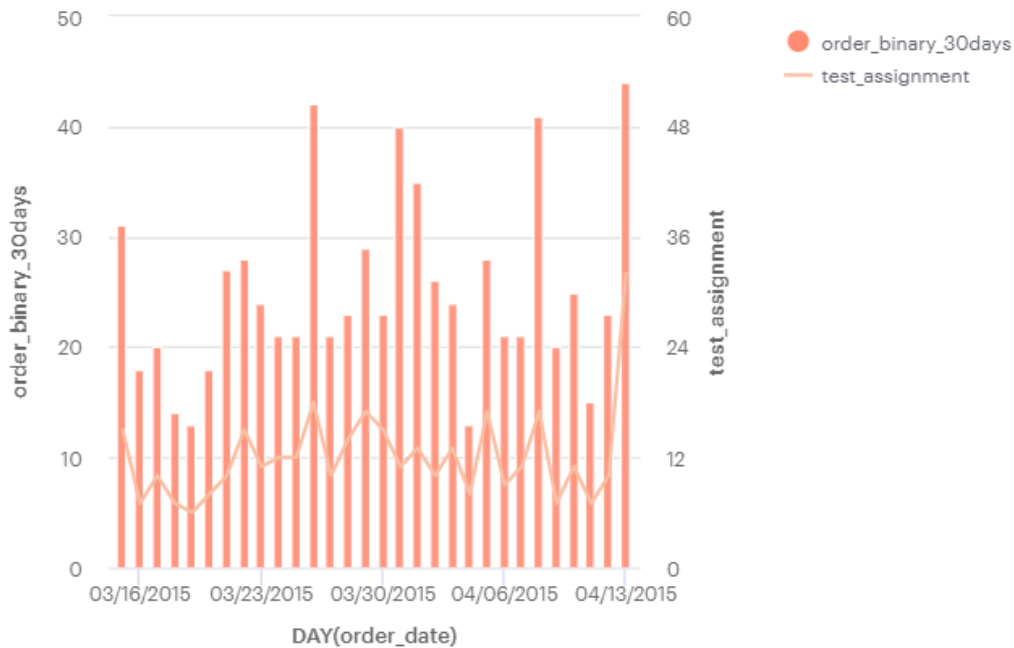
```

```

9.          WHEN (DATE(created_at) > DATE(test_start_date)
10.            AND (DATE(created_at) - DATE(test_start_date)) <= 30)
11.            THEN 1
12.            ELSE 0
13.        END) AS order_binary_30days
14. FROM dsv1069.final_assignments
15. LEFT JOIN dsv1069.orders AS orders
16. ON final_assignments.item_id = orders.item_id
17. WHERE test_number = 'item_test_2'
18. GROUP BY 1, 2, 3;

```

Total Orders & Total Treatments by Date (2015)



Part 2:

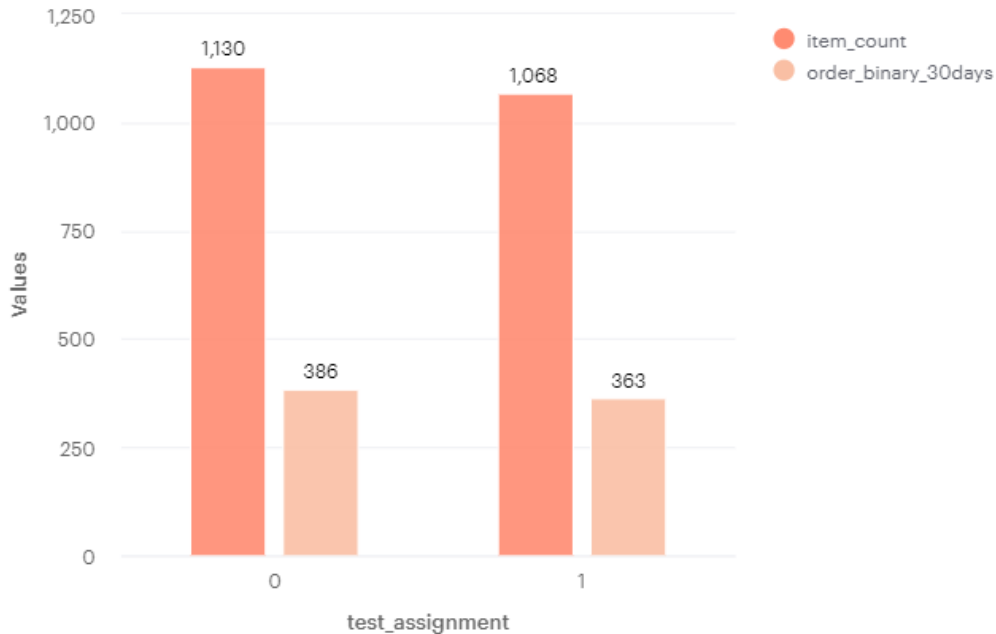
```

1. SELECT test_assignment,
2.        COUNT(distinct item_id) AS item_count,
3.        SUM(order_binary_30days) AS order_binary_30days
4. FROM(
5.     SELECT final_assignments.item_id AS item_id,
6.            final_assignments.test_assignment,
7.            (CASE
8.              WHEN (DATE(created_at) > DATE(test_start_date)
9.                AND (DATE(created_at) - DATE(test_start_date)) <= 30)
10.             THEN DATE(created_at)
11.            END) AS order_date,
12.            MAX(CASE
13.              WHEN (DATE(created_at) > DATE(test_start_date)
14.                AND (DATE(created_at) - DATE(test_start_date)) <= 30)
15.             THEN 1
16.             ELSE 0
17.            END) AS order_binary_30days
18.     FROM dsv1069.final_assignments
19.     LEFT JOIN dsv1069.orders AS orders
20.     ON final_assignments.item_id = orders.item_id
21.     WHERE test_number = 'item_test_2'
22.     GROUP BY 1, 2, 3
23. ) AS order_binary
24. GROUP BY 1;

```

	test_assignment	item_count	order_binary_30days
1	0	1130	386
2	1	1068	363

Item Count & Order Binary for Control (0) and Treatment (1)



Task 4: Use the final_assignments table to calculate the view binary, and average views for the 30 day window after the test assignment for item_test_2. (You may include the day the test started)

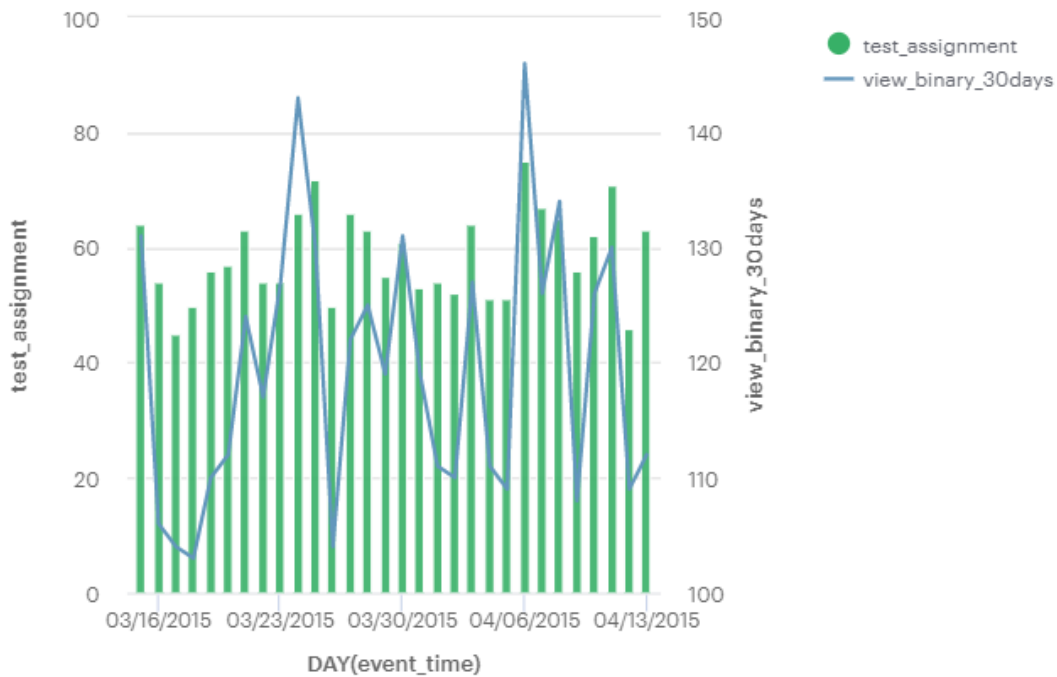
Part 1:

```

1. SELECT final_assignments.item_id AS item_id,
2.        final_assignments.test_assignment,
3.        (CASE
4.          WHEN (DATE(event_time) > DATE(test_start_date)
5.               AND (DATE(event_time) - DATE(test_start_date)) <= 30)
6.          THEN DATE(event_time)
7.          ELSE NULL
8.        END) AS event_time,
9.        MAX(CASE
10.           WHEN (DATE(event_time) > DATE(test_start_date)
11.                AND (DATE(event_time) - DATE(test_start_date)) <= 30)
12.           THEN 1
13.           ELSE 0
14.         END) AS view_binary_30days
15. FROM dsv1069.final_assignments
16. LEFT JOIN dsv1069.view_item_events
17. ON final_assignments.item_id = view_item_events.item_id
18. WHERE test_number = 'item_test_2'
19. GROUP BY 1, 2, 3;

```

Total Views & Total Treatments by Date



Part 2:

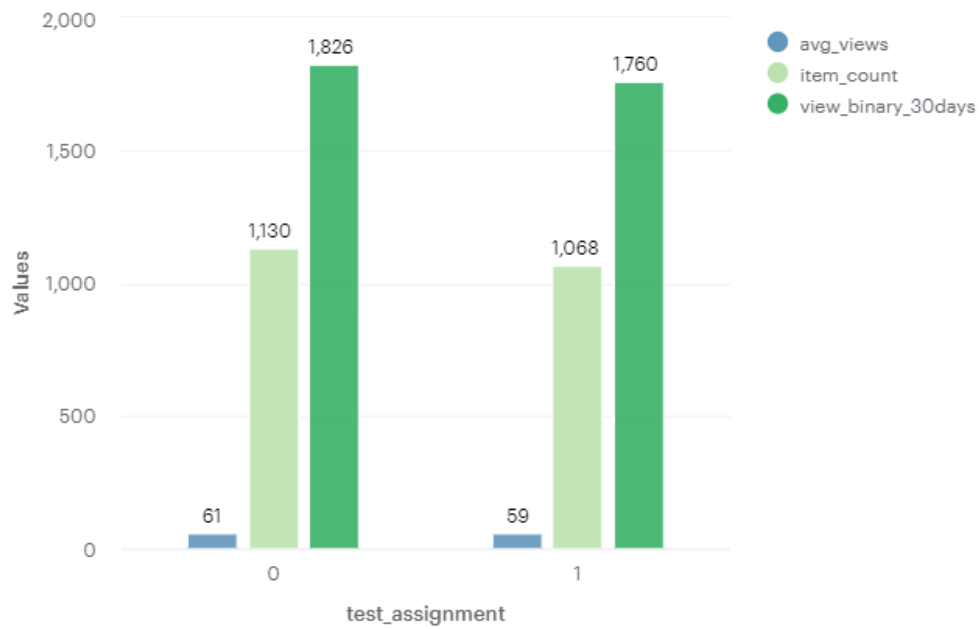
```

1. SELECT test_assignment,
2.         COUNT(distinct item_id) AS item_count,
3.         SUM(view_binary_30days) AS view_binary_30days,
4.         ROUND(SUM(view_binary_30days)/30.) AS avg_views
5. FROM(
6.     SELECT final_assignments.item_id AS item_id,
7.            final_assignments.test_assignment,
8.            (CASE
9.                WHEN (DATE(event_time) > DATE(test_start_date)
10.                 AND (DATE(event_time) - DATE(test_start_date)) <= 30)
11.                 THEN DATE(event_time)
12.                 ELSE NULL
13.            END) AS event_time,
14.            MAX(CASE
15.                WHEN (DATE(event_time) > DATE(test_start_date)
16.                 AND (DATE(event_time) - DATE(test_start_date)) <= 30)
17.                 THEN 1
18.                 ELSE 0
19.            END) AS view_binary_30days
20.     FROM dsv1069.final_assignments
21.    LEFT JOIN dsv1069.view_item_events
22.    ON final_assignments.item_id = view_item_events.item_id
23.   WHERE test_number = 'item_test_2'
24.  GROUP BY 1, 2, 3
25. ) AS view_binary
26. GROUP BY 1;


```

	test_assignment	item_count	view_binary_30days	avg_views
1	0	1130	1826	61
2	1	1068	1760	59

Item Count & View Binary for Control (0) and Treatment (1)



Task 5: Use the <https://thumbtack.github.io/abba/demo/abba.html> to compute the lifts in metrics and the p-values for the binary metrics (30 day order binary and 30 day view binary) using a interval 95% confidence.



A/B testing statistics

Label	Number of successes	Number of trials	
Control (order_binary)	386	1130	Remove
Treatment (order_binary)	363	1068	Remove

Interval confidence level:

Use multiple testing correction: ☒

[Compute](#) [Add another group](#)

	Successes	Total	Success Rate		p-value	Improvement
Control (order_binary)	386	1,130	31% – 37% (34%)	<div></div>	—	—
Treatment (order_binary)	363	1,068	31% – 37% (34%)	<div></div>	0.93	-12% – 11% (-0.5%)



A/B testing statistics

Label	Number of successes	Number of trials	
Control (view_binary)	1826	1130	Remove
Treatment (view_binary)	1760	1068	Remove

Interval confidence level:

Use multiple testing correction: ☒

[Compute](#) [Add another group](#)

	Successes	Total	Success Rate		p-value	Improvement
Control (view_binary)	1,826	1,130	N/A – N/A (162%)		—	—
Treatment (view_binary)	1,760	1,068	N/A – N/A (165%)		< 0.0001	N/A – N/A (2%)

Task 6: Use Mode's Report builder feature to write up the test. Your write-up should include a title, a graph for each of the two binary metrics you've calculated. The lift and p-value (from the AB test calculator) for each of the two metrics, and a complete sentence to interpret the significance of each of the results.

Order binary: No improvement is observed. p-value = 0.94 which is very high. The lift value is negative 0.5%. The number of orders during this 30 day period didn't change much and apparently didn't bring any improvement.

View binary: There is a 2% improvement and the p_value is extremely small; $p < .0001$. The number of views during the 30 days after test assignment didn't change significantly.

Mode Report Link: <https://app.mode.com/sum14/reports/1d46c9d7ec8a>