-- AB Testing Final Assignment

-- 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.

-- Compare this table to the assignment events we captured for user\_level\_testing.

-- Does this table have everything you need to compute metrics like 30-day view-binary?

-- Answer is No. The created\_at date is needed.

SELECT \*

FROM dsv1069.final\_assignments\_qa;

-- 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.

SELECT item\_id,

test\_a AS test\_assignment,

(CASE

WHEN test\_a IS NOT NULL then 'test\_a'

ELSE NULL

END) AS test\_number,

(CASE

WHEN test\_a IS NOT NULL then '2013-01-05 00:00:00'

ELSE NULL

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 'test\_b'

ELSE NULL

END) AS test\_number,

(CASE

WHEN test\_b IS NOT NULL then '2013-01-05 00:00:00'

ELSE NULL

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 'test\_c'

ELSE NULL

END) AS test\_number,

(CASE

WHEN test\_c IS NOT NULL then '2013-01-05 00:00:00'

ELSE NULL

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 'test\_d'

ELSE NULL

END) AS test\_number,

(CASE

WHEN test\_d IS NOT NULL then '2013-01-05 00:00:00'

ELSE NULL

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 'test\_e'

ELSE NULL

END) AS test\_number,

(CASE

WHEN test\_e IS NOT NULL then '2013-01-05 00:00:00'

ELSE NULL

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 'test\_f'

ELSE NULL

END) AS test\_number,

(CASE

WHEN test\_f IS NOT NULL then '2013-01-05 00:00:00'

ELSE NULL

END) AS test\_start\_date

FROM dsv1069.final\_assignments\_qa;

-- 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)

SELECT test\_assignment,

COUNT(DISTINCT item\_id) AS number\_of\_items,

SUM(order\_binary) AS items\_ordered\_30d

FROM

(SELECT item\_test\_2.item\_id,

item\_test\_2.test\_assignment,

item\_test\_2.test\_number,

item\_test\_2.test\_start\_date,

item\_test\_2.created\_at,

MAX(CASE

WHEN (created\_at > test\_start\_date

AND DATE\_PART('day', created\_at - test\_start\_date) <= 30) THEN 1

ELSE 0

END) AS order\_binary

FROM

(SELECT final\_assignments.\*,

DATE(orders.created\_at) AS created\_at

FROM dsv1069.final\_assignments AS final\_assignments

LEFT JOIN dsv1069.orders AS orders

ON final\_assignments.item\_id = orders.item\_id

WHERE test\_number = 'item\_test\_2') AS item\_test\_2

GROUP BY item\_test\_2.item\_id,

item\_test\_2.test\_assignment,

item\_test\_2.test\_number,

item\_test\_2.test\_start\_date,

item\_test\_2.created\_at) AS order\_binary

GROUP BY test\_assignment;

-- 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)

SELECT item\_test\_2.item\_id,

item\_test\_2.test\_assignment,

item\_test\_2.test\_number,

MAX(CASE

WHEN (view\_date > test\_start\_date

AND DATE\_PART('day', view\_date - test\_start\_date) <= 30) THEN 1

ELSE 0

END) AS view\_binary

FROM

(SELECT final\_assignments.\*,

DATE(events.event\_time) AS view\_date

FROM dsv1069.final\_assignments AS final\_assignments

LEFT JOIN

(SELECT event\_time,

CASE

WHEN parameter\_name = 'item\_id' THEN CAST(parameter\_value AS NUMERIC)

ELSE NULL

END AS item\_id

FROM dsv1069.events

WHERE event\_name = 'view\_item') AS events

ON final\_assignments.item\_id = events.item\_id

WHERE test\_number = 'item\_test\_2') AS item\_test\_2

GROUP BY item\_test\_2.item\_id,

item\_test\_2.test\_assignment,

item\_test\_2.test\_number

LIMIT 100;

-- 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.

SELECT test\_assignment,

test\_number,

COUNT(DISTINCT item) AS number\_of\_items,

SUM(view\_binary\_30d) AS view\_binary\_30d

FROM

(SELECT final\_assignments.item\_id AS item,

test\_assignment,

test\_number,

test\_start\_date,

MAX((CASE

WHEN date(event\_time) - date(test\_start\_date) BETWEEN 0 AND 30 THEN 1

ELSE 0

END)) AS view\_binary\_30d

FROM dsv1069.final\_assignments

LEFT JOIN dsv1069.view\_item\_events

ON final\_assignments.item\_id = view\_item\_events.item\_id

WHERE test\_number = 'item\_test\_2'

GROUP BY final\_assignments.item\_id,

test\_assignment,

test\_number,

test\_start\_date) AS view\_binary

GROUP BY test\_assignment,

test\_number,

test\_start\_date;