

## Analyzing Results

Exercise 1: Use the order\_binary metric from the previous exercise, count the number of users per treatment group for test\_id = 7, and count the number of users with orders (for test\_id 7).

```
1. select
2.   test_assignment,
3.   count(user_id) as users,
4.   sum(order_binary) as order_binary
5. from
6.   (
7.     select
8.       assignments.user_id,
9.       assignments.test_id,
10.      assignments.test_assignment,
11.      max(case when orders.created_at > assignments.event_time then 1 else 0 end) as order_binary
12.    from
13.      (
14.        select
15.          event_id,
16.          event_time,
17.          user_id,
18.          --platform,
19.          max(case when parameter_name = 'test_id'
20.                then cast(parameter_value as int)
21.                else null
22.                end) as test_id,
23.          max(case when parameter_name = 'test_assignment'
24.                then cast(parameter_value as int)
25.                else null
26.                end) as test_assignment
27.        from
28.          dsv1069.events_big
29.        where
30.          event_name = 'test_assignment'
31.        group by
32.          event_id,
33.          event_time,
34.          user_id
35.        order by event_id
36.      ) assignments
37.    left outer join
38.      dsv1069.orders_big orders
39.    on
40.      assignments.user_id = orders.user_id
41.    group by
42.      assignments.user_id,
43.      assignments.test_id,
44.      assignments.test_assignment
45.  ) user_level
46. where
47.   test_id = 7
48. group by
49.   test_assignment
```

Exercise 2: Create a new tem view binary metric. Count the number of users per treatment group, and count the number of users with views (for test\_id 7).

```
1. select
2.   test_assignment,
3.   count(user_id) as users,
4.   sum(view_binary) as views
5. from
```

```

6. (
7.   select
8.     assignments.user_id,
9.     assignments.test_id,
10.    assignments.test_assignment,
11.    max(case when views.event_time > assignments.event_time then 1 else 0 end) as view_binary
12.  from
13.    (
14.      select
15.        event_id,
16.        event_time,
17.        user_id,
18.        --platform,
19.        max(case when parameter_name = 'test_id'
20.              then cast(parameter_value as int)
21.              else null
22.            end) as test_id,
23.        max(case when parameter_name = 'test_assignment'
24.              then cast(parameter_value as int)
25.              else null
26.            end) as test_assignment
27.      from
28.        dsv1069.events_with_testing
29.      where
30.        event_name = 'test_assignment'
31.      group by
32.        event_id,
33.        event_time,
34.        user_id
35.      order by event_id
36.    ) assignments
37. left outer join
38.  (
39.    select *
40.    from
41.      dsv1069.events
42.    where
43.      event_name = 'view_item'
44.    and
45.      parameter_name = 'item_id'
46.    ) views
47. on
48.   assignments.user_id = views.user_id
49.
50. group by
51.   assignments.user_id,
52.   assignments.test_id,
53.   assignments.test_assignment
54. ) user_level
55. where
56.   test_id = 7
57. group by
58.   test_assignment

```

Exercise 3: Alter the result from EX 2, to compute the users who viewed an item WITHIN 30 days of their treatment event.

```

1. select
2.   test_assignment,
3.   count(user_id) as users,
4.   sum(view_binary) as views,
5.   sum(view_binary_30d) as views_30d
6. from
7.  (
8.    select
9.      assignments.user_id,
10.     assignments.test_id,
11.     assignments.test_assignment,

```

```

12.     max(case when views.event_time > assignments.event_time then 1 else 0 end) as view_binary,
13.
14.     max(case when (views.event_time > assignments.event_time
15.                    and date_part('day', views.event_time - assignments.event_time) <= 30)
16.                    then 1 else 0 end) as view_binary_30d
17. from
18.     (
19.     select
20.         event_id,
21.         event_time,
22.         user_id,
23.         --platform,
24.         max(case when parameter_name = 'test_id'
25.                  then cast(parameter_value as int)
26.                  else null
27.                  end) as test_id,
28.         max(case when parameter_name = 'test_assignment'
29.                  then cast(parameter_value as int)
30.                  else null
31.                  end) as test_assignment
32.     from
33.         dsv1069.events_with_testing
34.     where
35.         event_name = 'test_assignment'
36.     group by
37.         event_id,
38.         event_time,
39.         user_id
40.     order by event_id
41.     ) assignments
42. left outer join
43.     (
44.     select *
45.     from
46.         dsv1069.events
47.     where
48.         event_name = 'view_item'
49.     and
50.         parameter_name = 'item_id'
51.     ) views
52. on
53.     assignments.user_id = views.user_id
54. group by
55.     assignments.user_id,
56.     assignments.test_id,
57.     assignments.test_assignment
58. ) user_level
59. where
60.     test_id = 7
61. group by
62.     test_assignment

```

Exercise 4: Create the metric invoices (this is a mean metric, not a binary metric) and for test\_id = 7

----The count of users per treatment group

----The average value of the metric per treatment group

----The standard deviation of the metric per treatment group

```

1. select
2.     test_id,
3.     test_assignment,
4.     count(user_id) as users,
5.     avg(total_revenue) as metric_average,
6.     stddev(total_revenue) as metric_stddev
7. from
8.     (
9.     select

```

```

10. assignments.user_id,
11. assignments.test_id,
12. assignments.test_assignment,
13. count(distinct case when orders.created_at > assignments.event_time then orders.invoice_id else null end)
14. as invoices,
15. count(distinct case when orders.created_at > assignments.event_time then orders.line_item_id else null end)
16. as line_items,
17. coalesce(sum(case when orders.created_at > assignments.event_time then orders.price else 0 end), 0)

18. as total_revenue
19. from
20. (
21. select
22.     event_id,
23.     event_time,
24.     user_id,
25.     --platform,
26.     max(case when parameter_name = 'test_id'
27.           then cast(parameter_value as int)
28.           else null
29.         end) as test_id,
30.     max(case when parameter_name = 'test_assignment'
31.           then cast(parameter_value as int)
32.           else null
33.         end) as test_assignment
34. from
35.     dsv1069.events
36. where
37.     event_name = 'test_assignment'
38. group by
39.     event_id,
40.     event_time,
41.     user_id
42. order by event_id
43. ) assignments
44. left outer join
45.     dsv1069.orders
46. on
47.     assignments.user_id = orders.user_id
48. group by
49.     assignments.user_id,
50.     assignments.test_id,
51.     assignments.test_assignment
52. ) user_level
53. where test_id = 6
54. group by
55.     test_id,
56.     test_assignment
57. limit 100

```