LALAMOVE ASSESSMENT

SAMPLE DATA

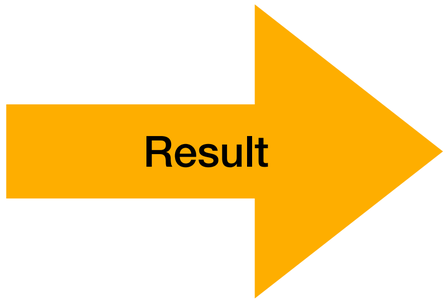
Created by Kelvin, Chan Ka Fung

Part 1

Use one SQL statement for finding each of the following questions.

a) For hours with orders, how many orders are there each hour based on order time?

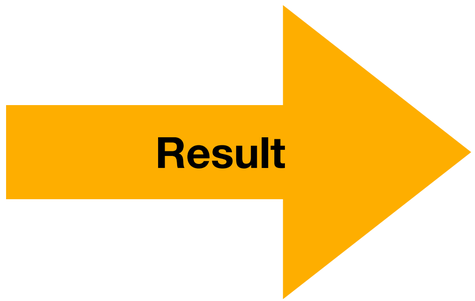
```sql
-- a) For hours with orders, how many orders are there each hour based on order time?
SELECT CONCAT(DATE_FORMAT(order_datetime,'%Y-%m-%d %H:00:00')," - ", CONCAT(TIME(SUBTIME(DATE_ADD(DATE_FORMAT(order_datetime,'%Y-%m-%d %H:00:00'), INTERVAL 1 HOUR),'00:00:01')))) AS Time_interval,
    COUNT(idvanOrder) AS number_per_interval
FROM vanorder
GROUP BY Time_interval
ORDER BY Time_interval;
```



Time_interval	number_per_interval
2017-04-18 07:00:00 - 07:59:59.000000	2
2017-04-18 08:00:00 - 08:59:59.000000	2
2017-04-18 09:00:00 - 09:59:59.000000	5
2017-04-18 10:00:00 - 10:59:59.000000	14
2017-04-18 11:00:00 - 11:59:59.000000	28
2017-04-18 12:00:00 - 12:59:59.000000	29
2017-04-18 13:00:00 - 13:59:59.000000	30
2017-04-18 14:00:00 - 14:59:59.000000	26
2017-04-18 15:00:00 - 15:59:59.000000	30
2017-04-18 16:00:00 - 16:59:59.000000	41
2017-04-18 17:00:00 - 17:59:59.000000	24
2017-04-18 18:00:00 - 18:59:59.000000	13
2017-04-18 19:00:00 - 19:59:59.000000	14
2017-04-18 20:00:00 - 20:59:59.000000	10
2017-04-18 21:00:00 - 21:59:59.000000	4
2017-04-18 22:00:00 - 22:59:59.000000	3
2017-04-19 04:00:00 - 04:59:59.000000	1
2017-04-19 07:00:00 - 07:59:59.000000	4
2017-04-19 08:00:00 - 08:59:59.000000	3
2017-04-19 09:00:00 - 09:59:59.000000	5
2017-04-19 10:00:00 - 10:59:59.000000	8
2017-04-19 11:00:00 - 11:59:59.000000	1
2017-04-19 16:00:00 - 16:59:59.000000	1
2017-04-19 20:00:00 - 20:59:59.000000	1
2017-04-21 07:00:00 - 07:59:59.000000	1

b) What is the percentage of money spent for each of the following group of clients?
- Clients who completed 1 order
- Clients who completed more than 1 order

```sql
SELECT
Count_group,
SUM(SUM_TOTAL)/(SELECT SUM(total_price) FROM vanorder WHERE order_status = 2) * 100 AS Percentage
    FROM
        (SELECT
            (CASE WHEN COUNT(requestor_client_id) = 1 THEN '1' ELSE '>1' END) AS Count_group,
            SUM(total_price) AS SUM_TOTAL
        FROM vanorder
        WHERE order_status = 2
        GROUP BY requestor_client_id) AS sub_tab
    GROUP BY Count_group;
```



	Count_group	Percentage
▶	1	71.3907
	>1	28.6093

c) List of unique Client ID who completed at least one order, also show each client's total money spent, and the total order(s) completed. Order the list by total money spent (descending), then by total order(s) completed (descending)

```sql
SELECT requestor_client_id AS client_id, SUM(total_price) AS Total_money_spent,
    COUNT(order_status = 2) AS Total_order_complete
FROM vanorder
WHERE order_status = 2
GROUP BY client_id
ORDER BY Total_money_spent DESC, Total_order_complete DESC;
```

(10 rows)

client_id	Total_money_spent	Total_order_complete
48	1904	8
197	1710	1
16	1332	1
17	1190	1
10	1115	1
14	906	5
213	843	2
82	765	3
12	719	3
38	640	2

d) List of all drivers who took order(s) (regardless of whether they eventually complete the order), also show each driver's total income and total order(s) completed. Order the list by total income (descending), then by total order(s) completed

```sql
SELECT vi.servicer_auth as Driver_id,
IFNULL(SUM(vo.total_price), 0) as Total_income,
(IF(IFNULL(SUM(vo.total_price), 0)=0, 0, COUNT(vo.servicer_auth))) as Total_orders_completed
FROM vaninterest vi
LEFT JOIN vanorder vo
ON vi.idvanOrder = vo.idvanOrder
    AND vi.servicer_auth = vo.servicer_auth
    AND vi.order_subset_assigned = vo.order_subset
    AND vo.order_status = 2
GROUP BY Driver_id
ORDER BY Total_income DESC, Total_orders_completed DESC, Driver_id ASC
;
```

Result

(10 rows)

Driv...	Total_income	Total_orders_completed
96	1710	1
150	1332	1
266	1190	1
252	1115	1
97	755	2
202	632	2
145	615	1
146	602	2
55	594	1
11	586	1

e) List of driver ID who took orders, but never complete an order?

```sql
SELECT vi.servicer_auth as Driver_id
FROM vaninterest vi
LEFT JOIN vanorder vo
ON vi.idvanOrder = vo.idvanOrder
    AND vi.servicer_auth = vo.servicer_auth
    AND vi.order_subset_assigned = vo.order_subset
    AND vo.order_status = 2
GROUP BY Driver_id
HAVING (IF(IFNULL(SUM(vo.total_price), 0)=0, 0, COUNT(vo.servicer_auth)))  = 0
ORDER BY Driver_id ASC;
```

Result

(10 rows)

Driv...
▶ 21
284
285
286
287
288
289
290
291
292

Part 2

Visualization and statistical knowledge

The date align left, which means that this is not a date format

Driver Response Timestamp		Order Create Timestamp		Order Id
24/3/2017	22:58:31	24/3/2017	22:57:38	1
25/3/2017	1:04:53	25/3/2017	1:04:53	2
25/3/2017	1:06:16	25/3/2017	1:06:16	3
25/3/2017	1:08:40	25/3/2017	1:07:03	4
25/3/2017	2:34:18	25/3/2017	2:34:08	5
25/3/2017	3:41:19	25/3/2017	3:41:05	6
25/3/2017	4:13:57	25/3/2017	4:13:44	7
25/3/2017	4:43:32	25/3/2017	4:43:11	8
25/3/2017	4:47:57	25/3/2017	4:47:05	9
25/3/2017	5:04:46	25/3/2017	5:04:37	10

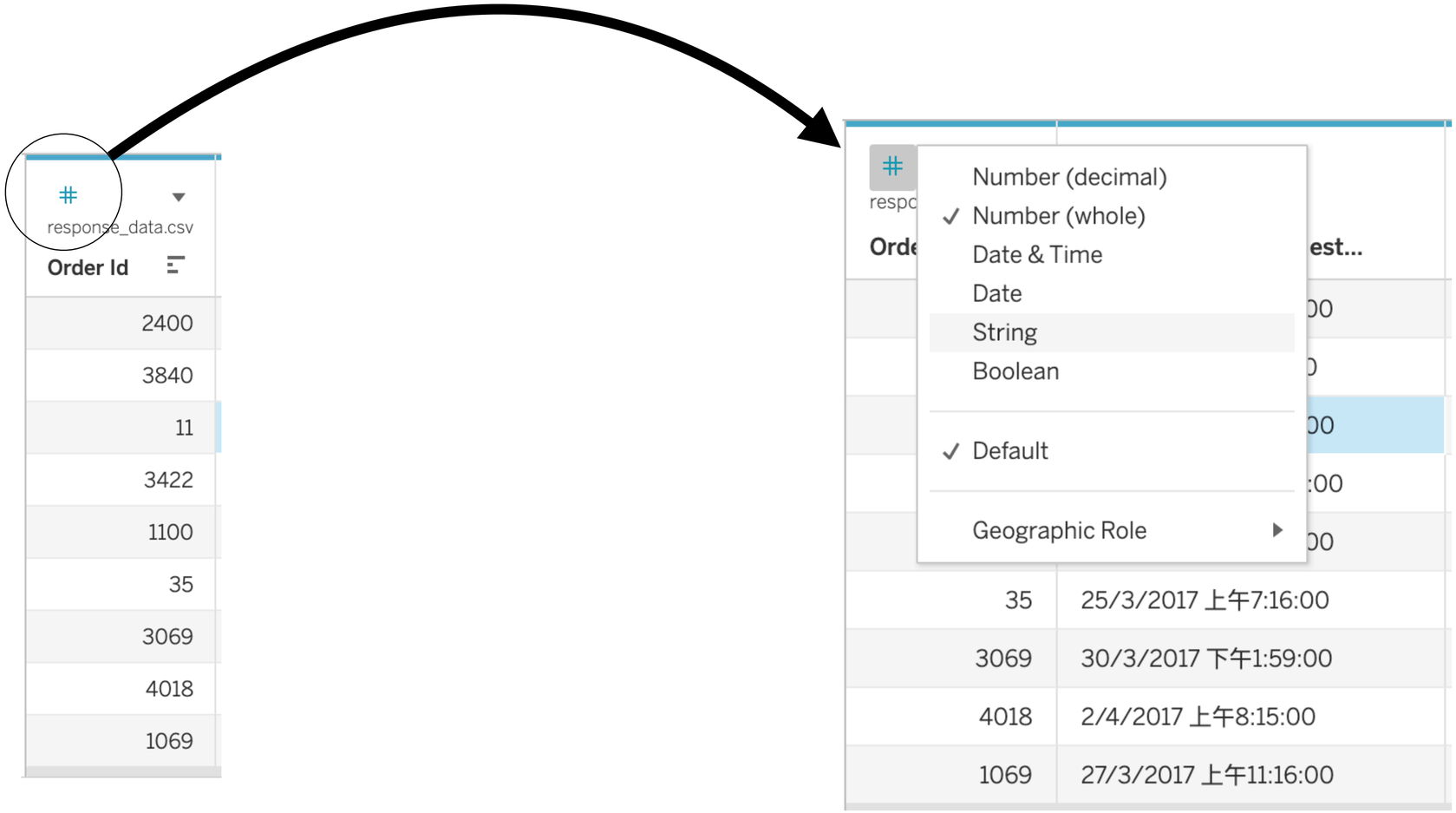
The date align right

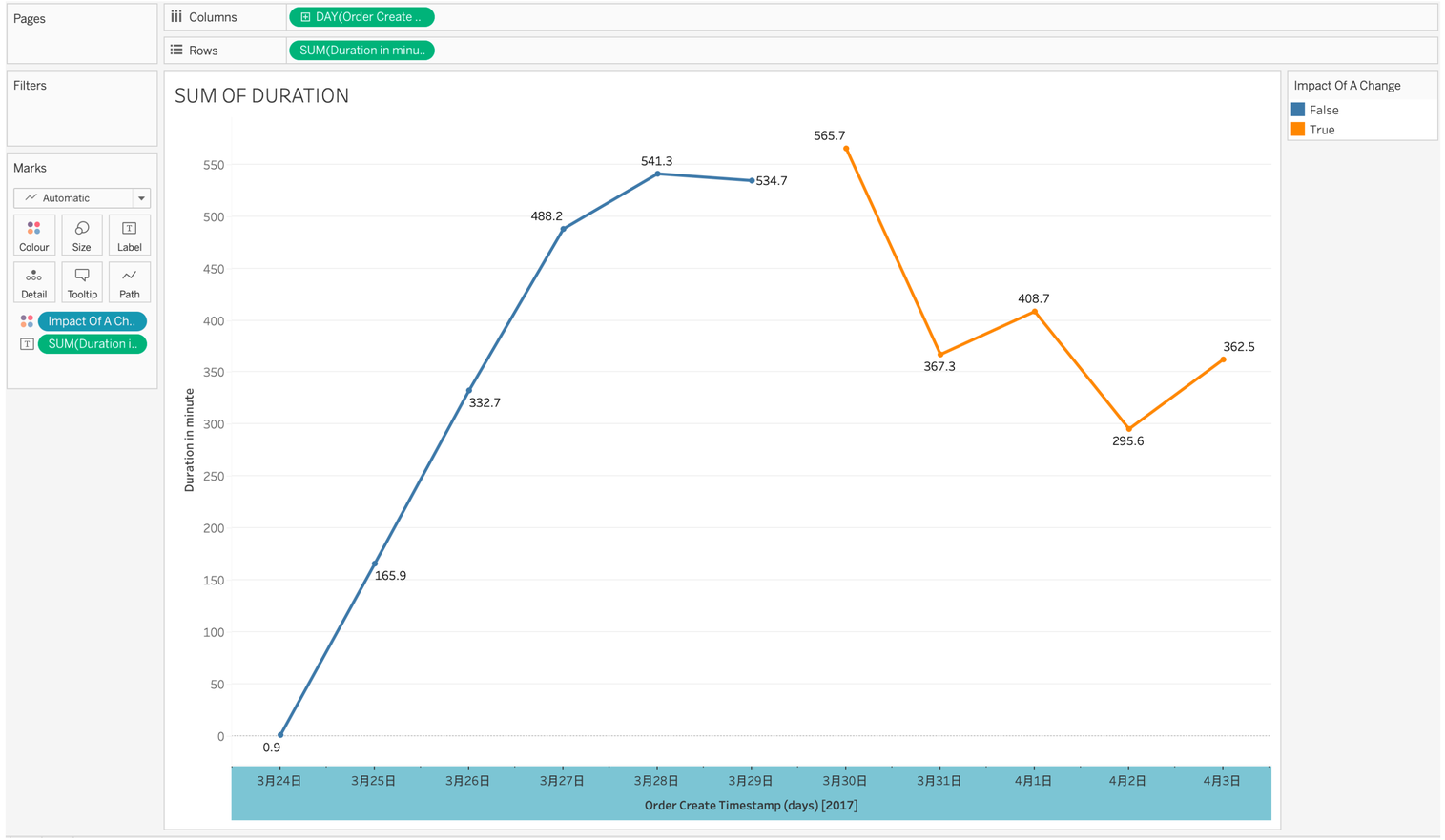
Driver Response Timestamp	Order Create Timestamp	Order Id
3/24/17 22:58:31	3/24/17 22:57:38	1
3/25/17 1:04:53	3/25/17 1:04:53	2
3/25/17 1:06:16	3/25/17 1:06:16	3
3/25/17 1:08:40	3/25/17 1:07:03	4
3/25/17 2:34:18	3/25/17 2:34:08	5
3/25/17 3:41:19	3/25/17 3:41:05	6
3/25/17 4:13:57	3/25/17 4:13:44	7
3/25/17 4:43:32	3/25/17 4:43:11	8
3/25/17 4:47:57	3/25/17 4:47:05	9
3/25/17 5:04:46	3/25/17 5:04:37	10
3/25/17 5:54:43	3/25/17 5:05:28	11
3/25/17 5:16:51	3/25/17 5:16:43	12
3/25/17 5:18:04	3/25/17 5:17:51	13
3/25/17 5:21:58	3/25/17 5:21:44	14
3/25/17 5:23:42	3/25/17 5:22:44	15
3/25/17 5:23:01	3/25/17 5:22:53	16
3/25/17 5:32:45	3/25/17 5:32:14	17
3/25/17 5:34:24	3/25/17 5:32:47	18
3/25/17 5:33:28	3/25/17 5:33:20	19
3/25/17 5:41:00	3/25/17 5:40:20	20
3/25/17 5:40:40	3/25/17 5:40:32	21
3/25/17 5:44:12	3/25/17 5:43:20	22
3/25/17 5:45:49	3/25/17 5:45:40	23
3/25/17 5:45:52	3/25/17 5:45:43	24
3/25/17 5:47:28	3/25/17 5:47:22	25
3/25/17 5:50:40	3/25/17 5:50:27	26
3/25/17 5:51:56	3/25/17 5:51:38	27
3/25/17 5:58:16	3/25/17 5:57:27	28
3/25/17 6:11:50	3/25/17 6:11:28	29
3/25/17 6:13:11	3/25/17 6:12:25	30

COMPLETED TABLE

	A	B	C	D	E	F	G
1	Order Id	Driver Response Timestamp	Order Create Timestamp	impact of a change	Duration	Duration in second	Duration in minute
2	2400	29/3/2017 13:04	29/3/2017 11:56	FALSE	1:08:22	4102	68.4
3	3840	1/4/2017 15:08	1/4/2017 14:02	TRUE	1:05:45	3945	65.7
4	11	25/3/2017 5:54	25/3/2017 5:05	FALSE	0:49:15	2955	49.2
5	3422	31/3/2017 10:48	31/3/2017 9:59	TRUE	0:48:44	2924	48.7
6	1100	27/3/2017 11:41	27/3/2017 10:54	FALSE	0:46:51	2811	46.8
7	35	25/3/2017 7:16	25/3/2017 6:30	FALSE	0:46:23	2783	46.4
8	3069	30/3/2017 13:59	30/3/2017 13:13	TRUE	0:45:53	2753	45.9
9	4018	2/4/2017 8:15	2/4/2017 7:30	TRUE	0:45:24	2724	45.4
10	1069	27/3/2017 11:16	27/3/2017 10:31	FALSE	0:44:39	2679	44.6

- Impact of a change: =IF([Order Create Timestamp] >=DATEVALUE(“2017-03-30 12:00:00”),TRUE,FALSE)
- Duration: ＝Driver Response Timestamp - Order Create Timestamp
- Duration in second：Duration * 86400
- Duration in minute：Duration * 1440

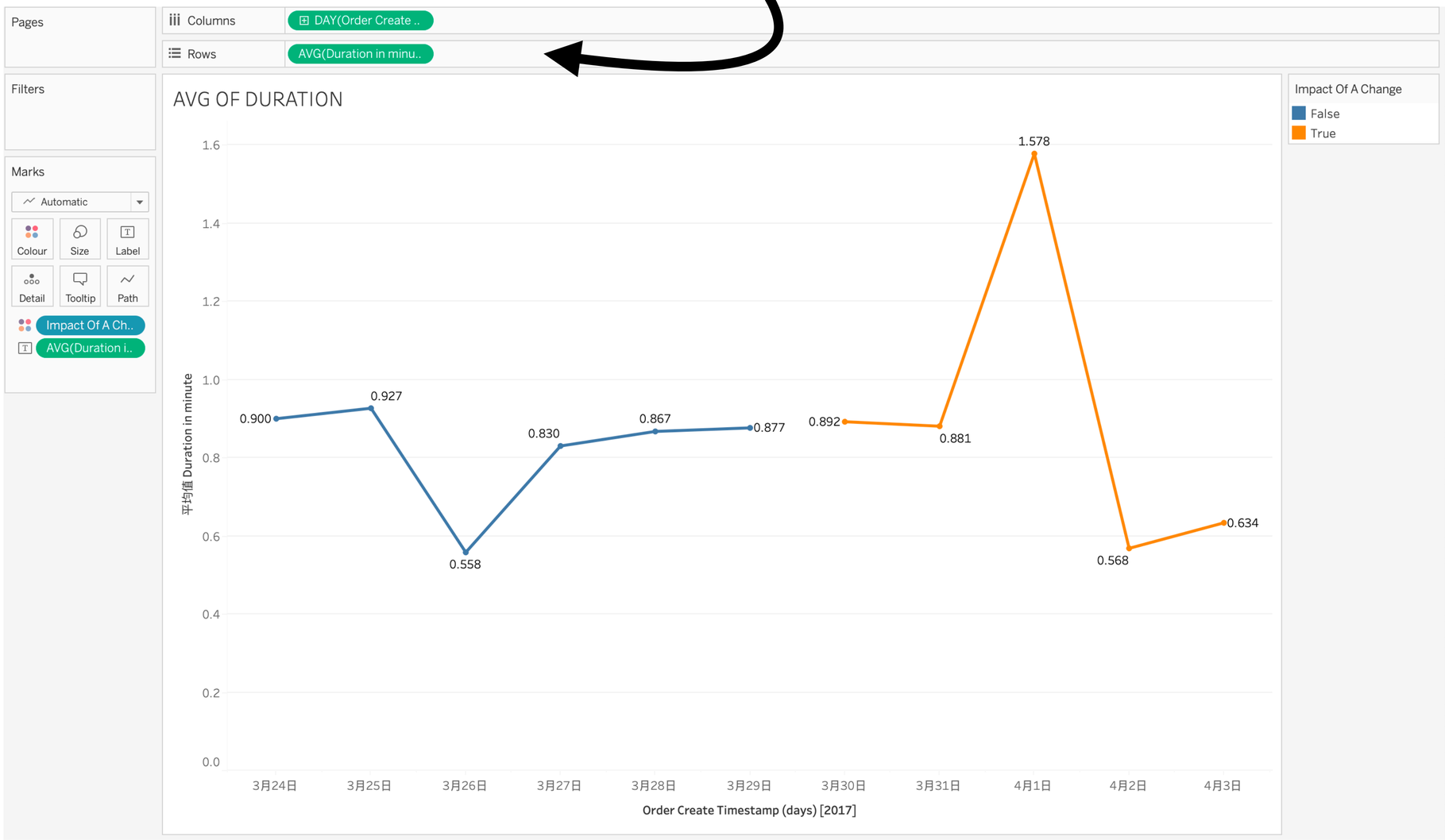

response_data.csv
Order Id
2400
3840
11
3422
1100
35
3069
4018
1069
Number (decimal)
Number (whole)
Date & Time
Date
String
Boolean
Default
Geographic Role
35
25/3/2017 上午7:16:00
3069
30/3/2017 下午1:59:00
4018
2/4/2017 上午8:15:00
1069
27/3/2017 上午11:16:00



- The orange line represents the sum of duration after the Impact of change
- The sum of duration decreases after the change

However, this is not enough to prove that the result is positive, so…

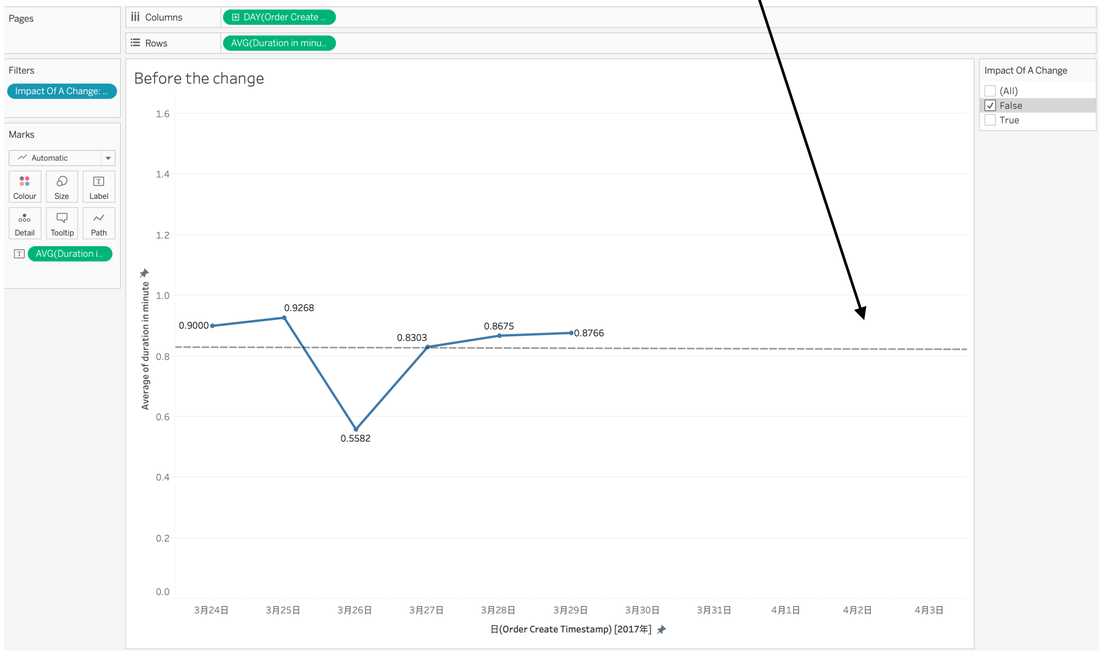
The sum of duration changed to be average of duration



Remove the colour

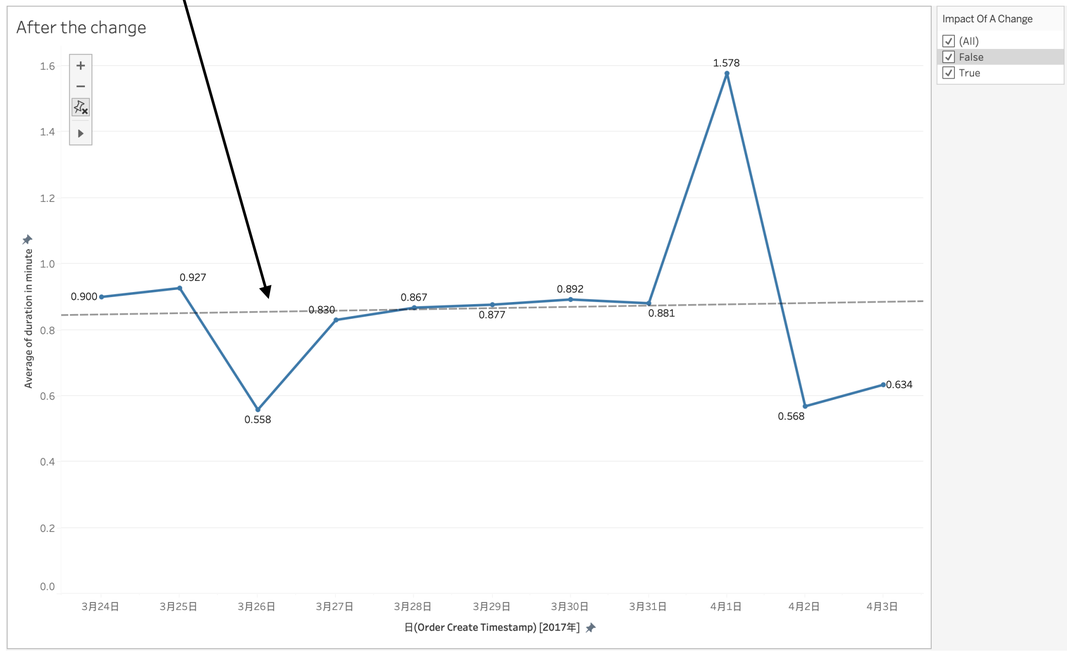
- The sum of duration can be affected by the no. of daily orders → Therefore, the sum function is not a good estimator
- The average of duration can represent a better estimation on improvement in stead

Before the change implied, the trend line is flat



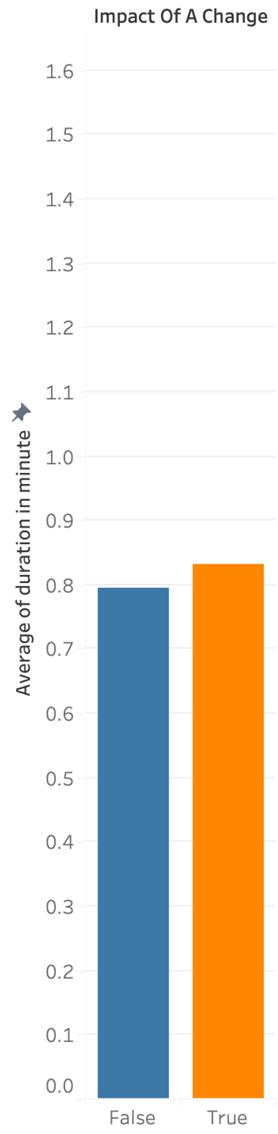
BEFORE

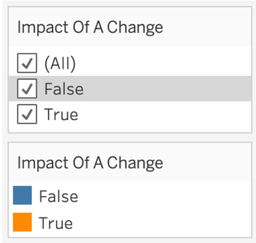
After the change implied, the trend line goes a little bit upward



AFTER

Before and After





The duration increased after the change implied.

a) What is the resulting impact from this change?

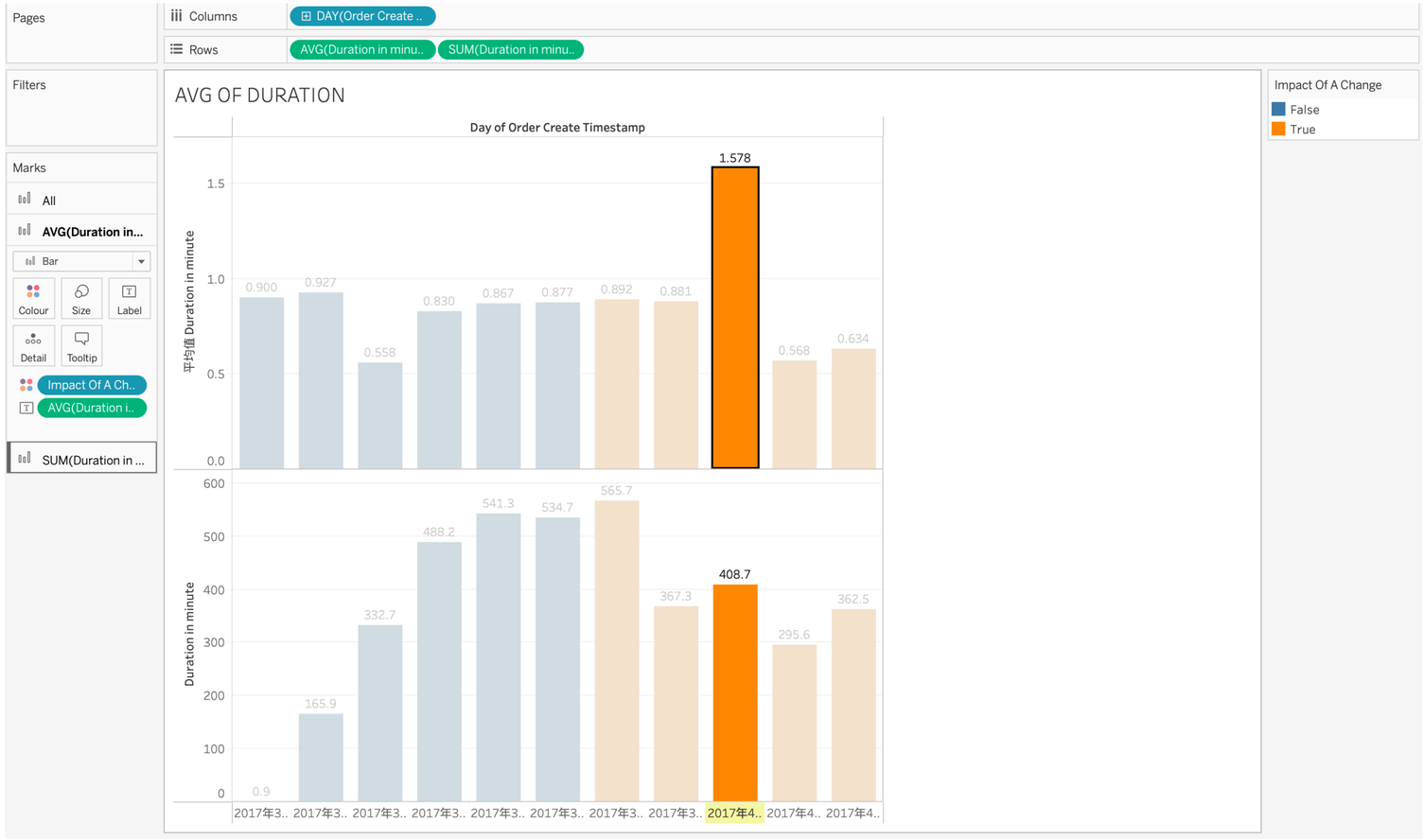
- The average of duration from create timestamp to driver response timestamp had been increase
- The change is negative to an improvement of duration

(Base on the provided data)

b) Any additional insights or observations you think are worth noting?

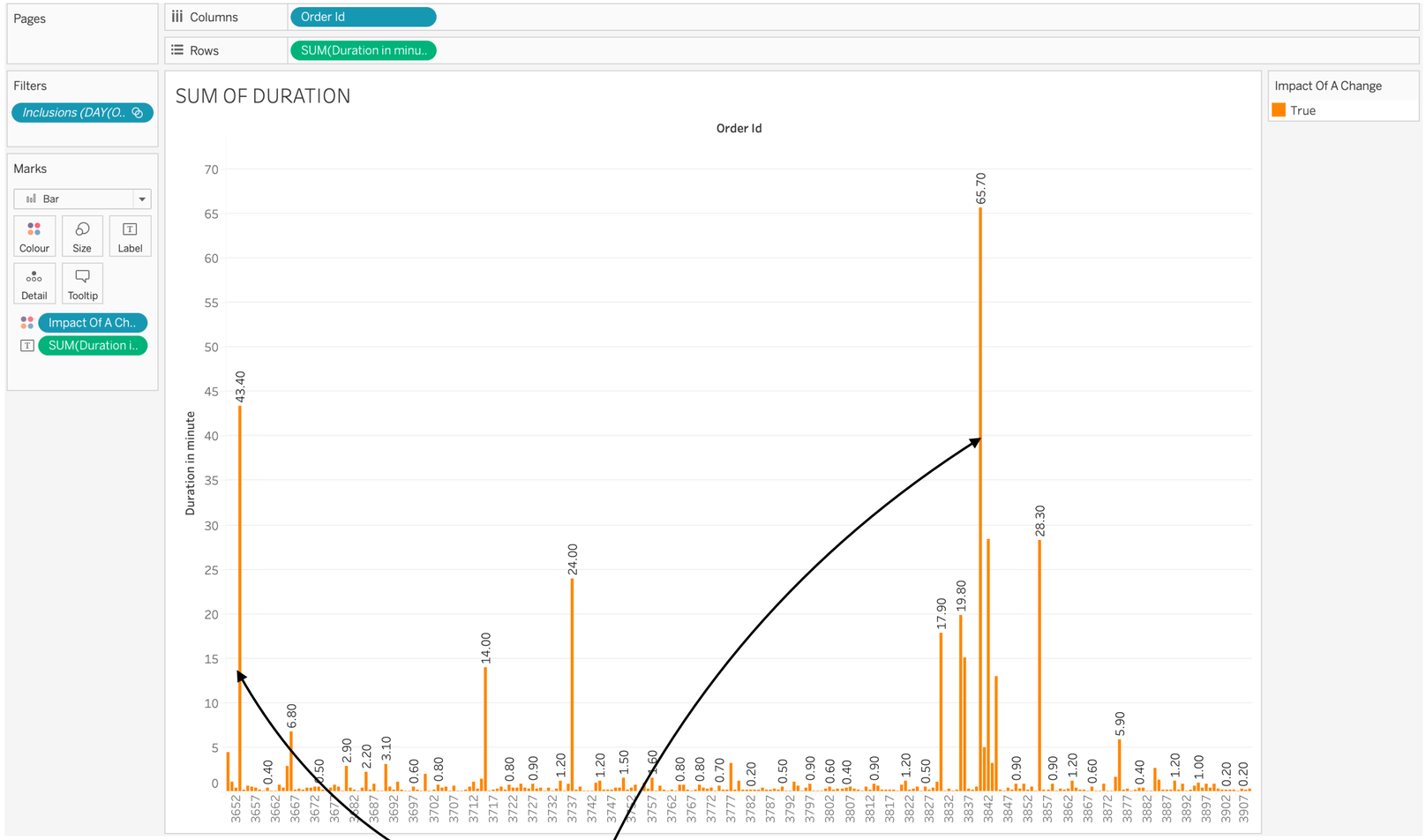
Outstanding data

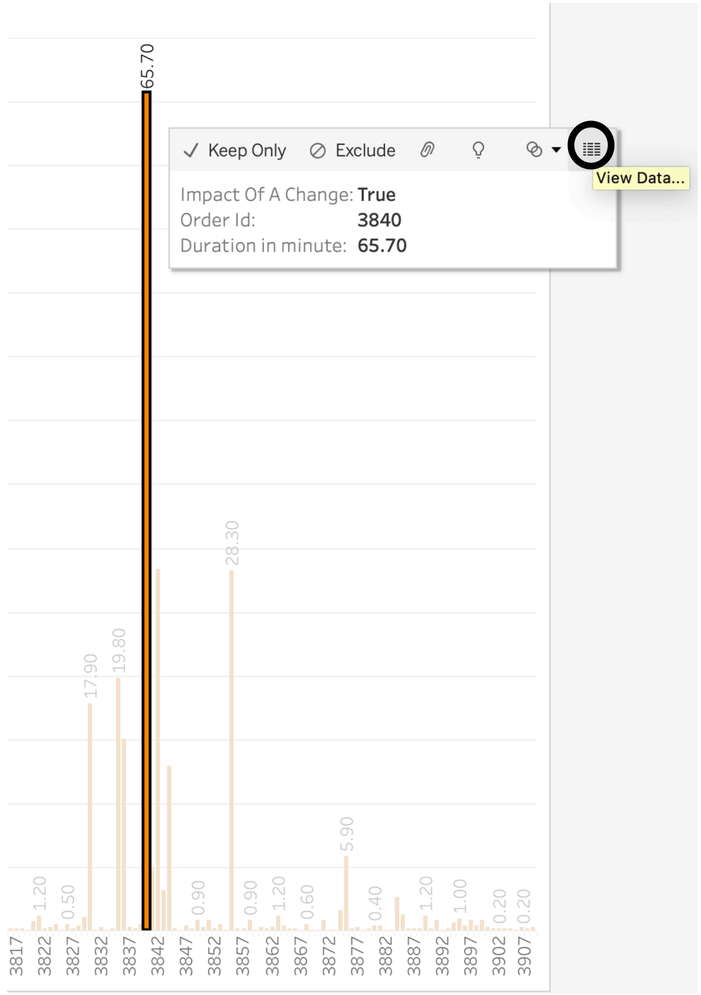
- The outstanding data are worthy to be investigated



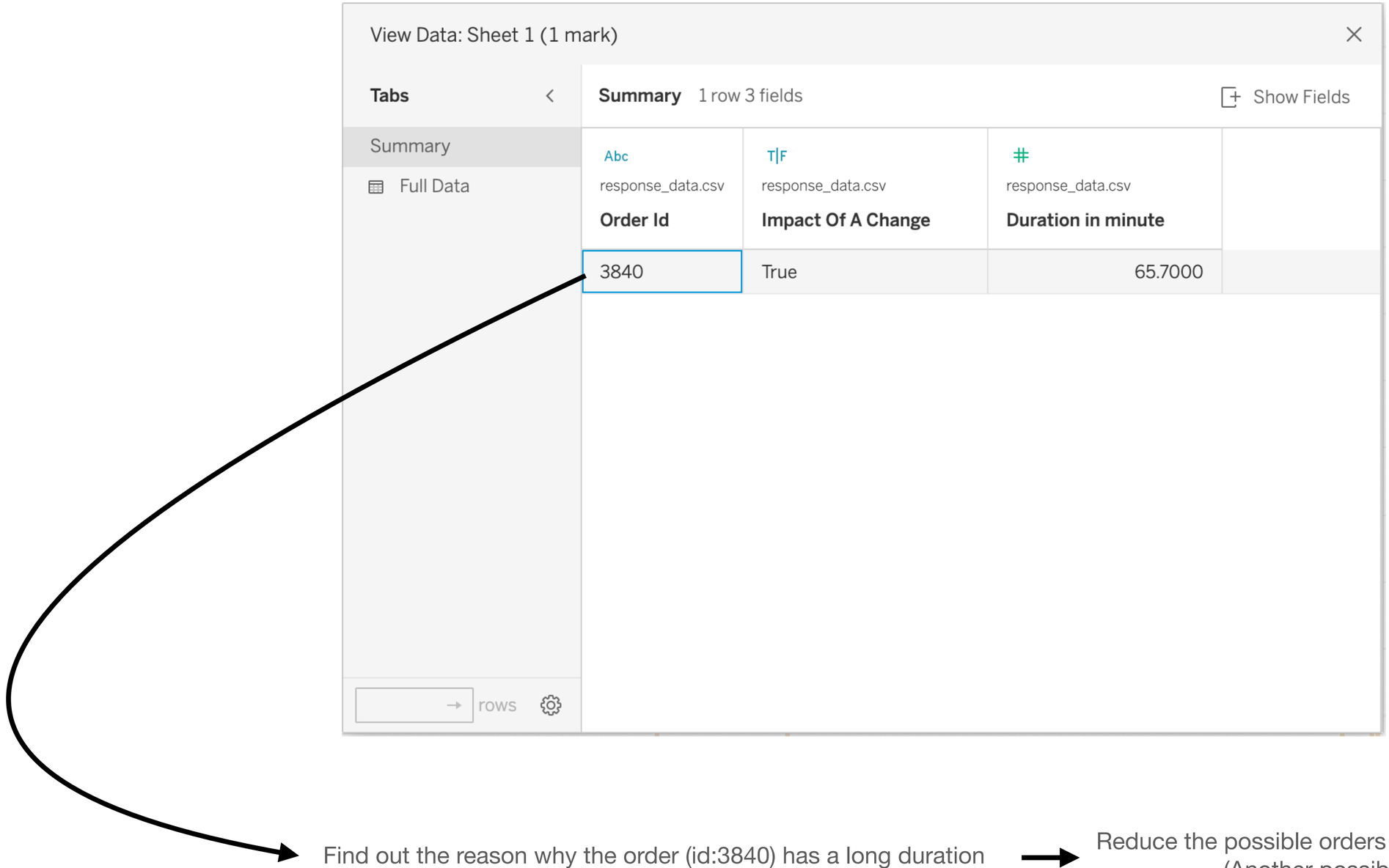
To investigate, keep only

Put order id into the columns





Identify the outstanding orders



Find out the reason why the order (id:3840) has a long duration → Reduce the possible orders of long duration in the future (Another possible enhancement)

c) Any observations regarding the testing process or environment? How would you improve or redesign this experiment?

- Observations have been mentioned by pages 9-16
- To improve this experiment：
  - More data type should be included, such as van type, user ranking ,driver ranking, driving years of driver, special requests of each order… etc
  - Longer time range before the experiment can help to enhance the confident interval and estimate the seasonal index for Time-series analysis
  - Comments from drivers can be important (non-numerical data)

Thank you