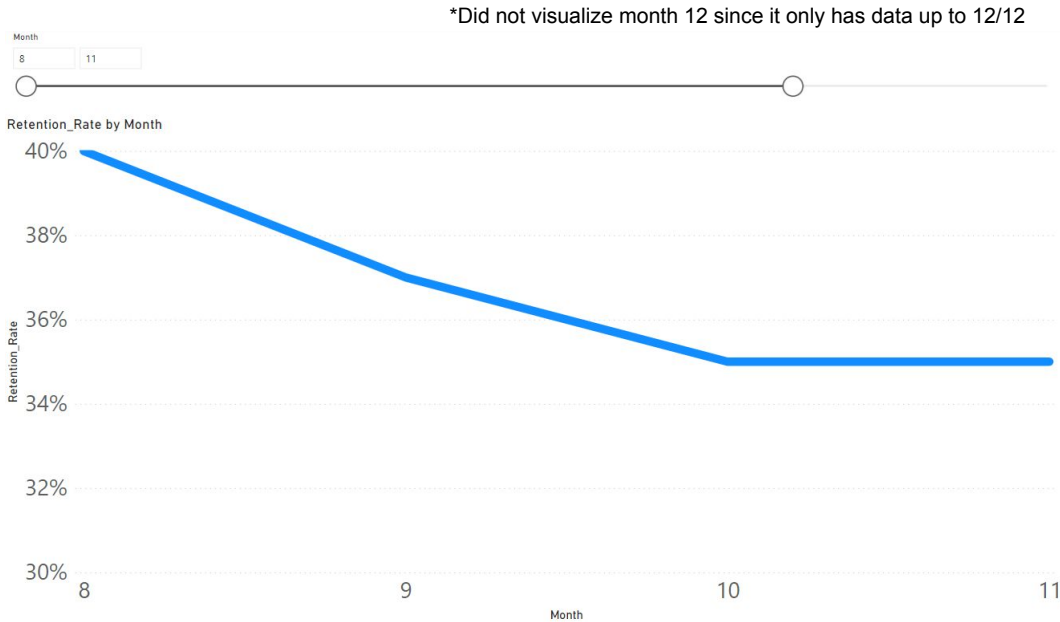


Part 1: Question 1

```
SELECT Year, Month, COUNT(DISTINCT userId) AS Total_User_Count, COUNT(DISTINCT Retained) AS Retained_User_Count,
ROUND(COUNT(DISTINCT Retained)/COUNT(DISTINCT userId),2) AS Retention_Rate
FROM
(
    SELECT EXTRACT(YEAR FROM timestamp) AS Year, EXTRACT(MONTH FROM Timestamp) AS Month, timestamp, userid, IF(event =
'Order Completed', userid, NULL) AS Retained, event
    FROM `tradesy-interviews.takehome.events`
)
GROUP BY Year, Month
ORDER BY Year, Month
```

SQL Output:

Year	Month	Total_User_Count	Retained_User_Count	Retention_Rate
2018	8	28354	11469	0.4
2018	9	30831	11526	0.37
2018	10	32943	11610	0.35
2018	11	32815	11331	0.35
2018	12	20926	4608	0.22



Part 1: Question 2 - SQL

Assumptions:

- Don't include month 12 since it is not complete (only goes to 12/12)
- Month 2 retention = 2 successive months of a user having an "Order Completed" event

SQL Output:

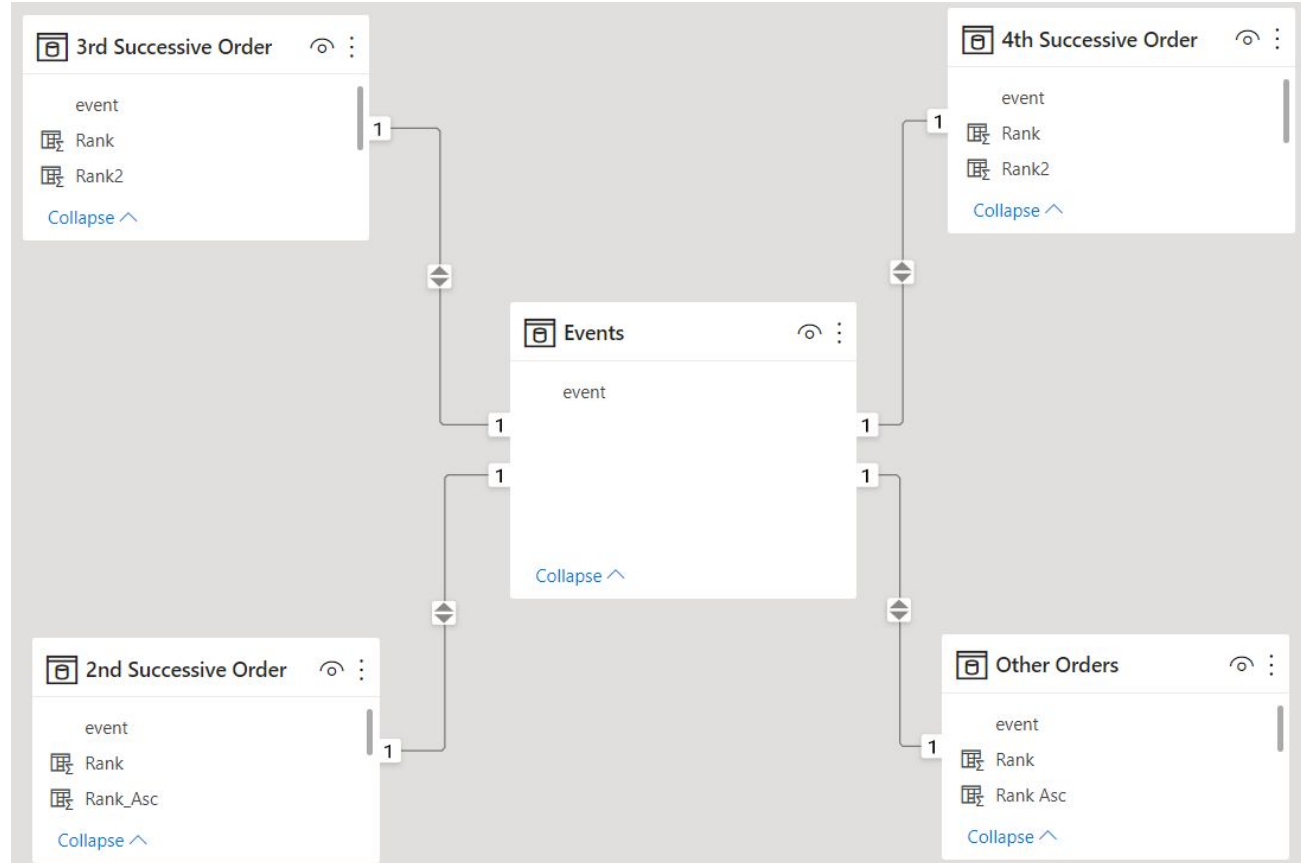
event	Total_Event_Count	Total_User_Count
Collection Viewed	119799	2081
Closet Viewed	125782	3290
Product Favorited	122625	3353
Filter Selected	83356	1964
Products Searched	150825	3846
Order Completed	8270	4877
Followed Search Viewed	201	69
Product Recommendation Clicked	22	11

```
WITH T1 AS (SELECT DISTINCT Month AS Month2, Retained FROM
(
SELECT
    Month, Retained,
    IF(Lag_Month1 = Month - 1, 1, 0) Two_successive,
    IF(Lag_Month2 = Lag_Month1 - 1, 1, 0) Three_successive,
    IF(Lag_Month3 = Lag_Month2 - 1, 1, 0) Four_successive
FROM
(
SELECT
    Month, Retained,
    LAG(Month, 1) OVER (PARTITION BY Retained ORDER BY Month) AS Lag_Month1,
    LAG(Month, 2) OVER (PARTITION BY Retained ORDER BY Month) AS Lag_Month2,
    LAG(Month, 3) OVER (PARTITION BY Retained ORDER BY Month) AS Lag_Month3,
FROM
(
SELECT DISTINCT EXTRACT(MONTH FROM Timestamp) AS Month, userid AS Retained
FROM `tradesy-interviews.takehome.events`
WHERE event = 'Order Completed' AND EXTRACT(MONTH FROM Timestamp) <> 12
)
)
)
WHERE Two_successive = 1 AND Three_successive = 0 AND Four_successive = 0)

SELECT
    event,
    COUNT(event) AS Total_Event_Count,
    COUNT(DISTINCT userId) AS Total_User_Count
FROM `tradesy-interviews.takehome.events`
INNER JOIN T1
    ON Retained = userid AND Month2 = EXTRACT(MONTH FROM Timestamp)
WHERE EXTRACT(MONTH FROM Timestamp) <> 12
GROUP BY event
```

Part 1: Question 2 - Power BI Model

- Ran the same query 3 more times but filtered to different order types.
- “Other Orders” is comprised of users with “New” and/or lapsed (non-successive) orders. Used as primary baseline.
- Also included perspectives on month 3 and month 4 retention for comparative purposes.



Part 1: Question 2 - Analysis

1 = New and Lapsed (Non-successive) Orders
2 = 2nd Successive Orders (Month 2 Retention)
3 = 3rd Successive Orders (Month 3 Retention)
4 = 4th Successive Orders (Month 4 Retention)

User Count to Event Type Correlations

Distinct User Count

event	1	2	3	4
Order Completed	37188	4877	1241	407
Products Searched	27752	3846	973	305
Product Favorited	19922	3353	890	301
Closet Viewed	22020	3290	857	291
Collection Viewed	14323	2081	521	186
Filter Selected	12780	1964	536	176
Followed Search Viewed	228	69	20	12
Product Recommendation Clicked	303	11		
Total	134516	19491	5038	1678

Rank Desc: User Count

event	1	2	3	4
Order Completed	1	1	1	1
Products Searched	2	2	2	2
Product Favorited	4	3	3	3
Closet Viewed	3	4	4	4
Collection Viewed	5	5	6	5
Filter Selected	6	6	5	6
Followed Search Viewed	8	7	7	7
Product Recommendation Clicked	7	6		
Total	36	36	28	28

Rank Asc: Scatterplot



Event Count to Event Type Correlations

Total Event Count

event	1	2	3	4
Products Searched	568766	150825	56318	20513
Closet Viewed	418325	125782	55963	25825
Product Favorited	466296	122625	43940	16673
Collection Viewed	513837	119799	41297	12031
Filter Selected	339268	83356	32203	11433
Order Completed	49589	8270	2605	969
Followed Search Viewed	667	201	55	34
Product Recommendation Clicked	509	22		
Total	2457257	610880	232381	87478

Rank Desc: Event Count

event	1	2	3	4
Products Searched	1	1	1	2
Closet Viewed	4	2	2	1
Product Favorited	3	3	3	3
Collection Viewed	2	4	4	4
Filter Selected	5	5	5	5
Order Completed	6	6	6	6
Followed Search Viewed	7	7	7	7
Product Recommendation Clicked	8	6		
Total	36	36	28	28

Rank Asc: Scatterplot



- Month 2 retention is relatively less correlated with “Product Recommendation Clicked” users vs New and Lapsed baseline. Months 3-4 retention saw no activity.
- Month 2 retention is relatively more correlated with “Closet Viewed” event count vs New and Lapsed baseline.
- Month 2 retention is relatively less correlated with “Collection view” vs New and Lapsed baseline.
- With a bit more time, I'd take a look at % to total by event and order type to get a bit more precision here.

Part 2: Aggregate

Assumptions:

- cancelled orders are considered
- order ids are sequential

Press Alt+F1 for Accessibility Options.

```
1 SELECT AVG((Order2_Amount - amount)) AS Avg_Diff
2 FROM
3 (
4     SELECT
5         RANK() OVER (PARTITION BY user_id ORDER BY id) AS id_Rank,
6         id,
7         user_id,
8         amount,
9         LEAD(amount) OVER (PARTITION BY user_id ORDER BY id) AS Order2_Amount,
10        status
11     FROM `tradesy-interviews.takehome.orders`
12 )
13 WHERE id_Rank = 1
```

Processing location: US

Query results

 SAVE RESULTS

 EXPLORE DATA ▾

Query complete (0.5 sec elapsed, 117.2 KB processed)

Job information

Results

JSON

Execution details

Row	Avg_Diff
-----	----------

1	0.6566501070440529
---	--------------------

```
SELECT AVG((Order2_Amount - amount)) AS Avg_Diff
FROM
(
    SELECT
        RANK() OVER (PARTITION BY user_id ORDER BY
id) AS id_Rank,
        id,
        user_id,
        amount,
        LEAD(amount) OVER (PARTITION BY user_id
ORDER BY id) AS Order2_Amount,
        status
    FROM `tradesy-interviews.takehome.orders`
)
WHERE id_Rank = 1
```

Avg “amount” difference between second and first order is approx. \$0.66

Part 2: by user_id

Press Alt+F1 for Accessibility Options.

```
1 SELECT user_id, AVG((Order2_Amount - amount)) AS Avg_Diff
2 FROM
3 (
4     SELECT
5         RANK() OVER (PARTITION BY user_id ORDER BY id) AS id_Rank,
6         id,
7         user_id,
8         amount,
9         LEAD(amount) OVER (PARTITION BY user_id ORDER BY id) AS Order2_Amount,
10        status
11     FROM `tradesy-interviews.takehome.orders`
12 )
13 WHERE id_Rank = 1
14 GROUP BY user_id
15 ORDER BY user_id
```

Processing location: US

Query results

 SAVE RESULTS

 EXPLORE DATA ▾

Query complete (0.7 sec elapsed, 117.2 KB processed)

Job information

Results

JSON

Execution details

Row	user_id	Avg_Diff
1	1	6.947029180000001
2	2	-13.174637591
3	3	-5.998022607
4	4	12.01940669
5	5	5.548770000000001
6	6	7.789846662
7	7	-0.9717244239999996

```
SELECT user_id, AVG((Order2_Amount - amount)) AS Avg_Diff
FROM
(
    SELECT
        RANK() OVER (PARTITION BY user_id ORDER BY id) AS
id_Rank,
        id,
        user_id,
        amount,
        LEAD(amount) OVER (PARTITION BY user_id ORDER BY id)
AS Order2_Amount,
        status
    FROM `tradesy-interviews.takehome.orders`
)
WHERE id_Rank = 1
GROUP BY user_id
ORDER BY user_id
```