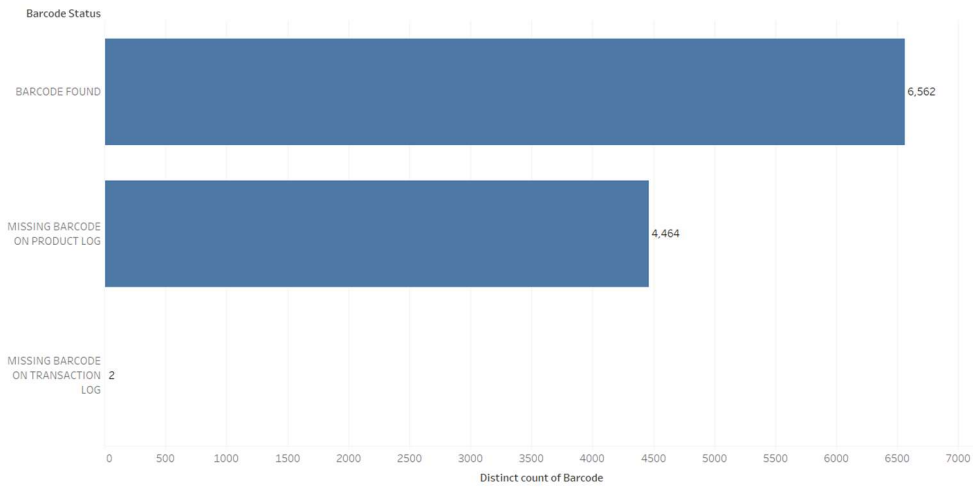
Exercise:

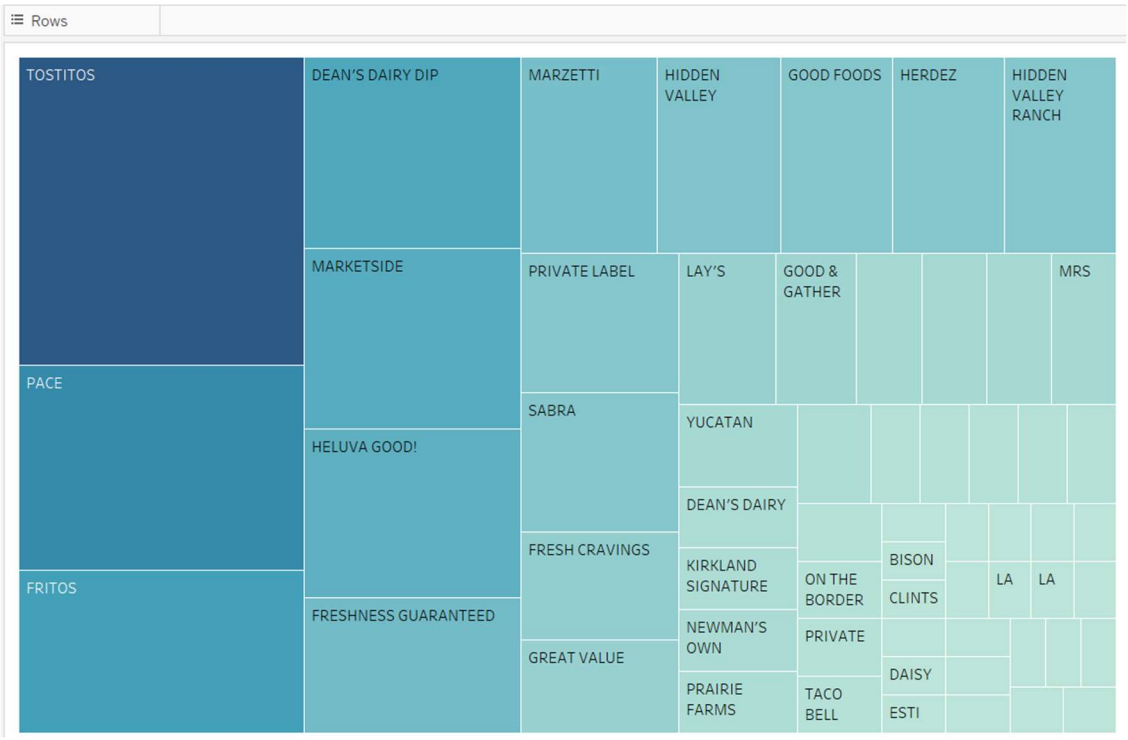
First: explore the data

Review the unstructured csv files and answer the following questions with code that supports your conclusions:

- Are there any data quality issues present?
- ---------- USERS TABLE ----------
    - The users table is missing a lot of records that are present in the transactions table.
        - ```
          SELECT COUNT(DISTINCT USER_ID) FROM TRANSACTIONS_EV A WHERE NOT EXISTS(SELECT 1 FROM USERS_D WHERE ID = A.USER_ID)
          ```

User Status	users
KNOWN USER	91
UNKNOWN USER	17,603
        -
    - People that use Fetch are apparently long-lived. The users table contains around 60 folks that are over 100 years old. Several of them were around at the turn of the last century which is impressive. I'm guessing some of these records should have a '20' for the year rather than a '19', but not entirely sure where the cutoff would be other than arbitrarily picking a maximum age.
        - ```
          SELECT *
          FROM users_d
          WHERE birth_date < DATEADD(YEAR, -100, GETDATE());
          ```
    - The gender data has 12 variations, many of which are essentially the same thing. Need to standardize how it's being stored.
        - ```
          SELECT DISTINCT GENDER FROM USERS_D
          ```
- ---------- PRODUCTS TABLE ----------
    - Similar to the users table this table is also missing a lot of records that are present in the transactions table.
          ```




```
SELECT * FROM TRANSACTIONS_EV A WHERE NOT EXISTS(SELECT 1 FROM PRODUCTS_D WHERE BARCODE = A.BARCODE)
```

- There's duplication within the table and a number of nulls, which is bad as barcode is our main linkage between the transaction table and the product table.

```
SELECT BARCODE, COUNT(1) FROM PRODUCTS_D GROUP BY BARCODE HAVING COUNT(1) > 1  --184 NUMERIC BARCODES ARE IN THERE/DUPLICATED, ALSO 4025 EMPTY BARCODES
```

- ---------- TRANSACTIONS TABLE ----------

- Some issues centered around the final_quantity and final_sale columns. Specifically, there were many instances where either one or the other would be null. I was able to merge some of the records using combination of barcode/receipt, and user/store/date, but there are still null records left. One assumption I did make based on looking at the differences is that these records are all legitimate, but they need to be merged. There were the exact same number of records with missing quantities and those with missing sales, which leads me to believe a better merge is necessary. Adding additional fields to the users table and maybe some additional product details could help with this endeavor.

```
--JOIN THE SPLIT RECORDS BASED ON RECEIPT ID AND BARCODE------------------------
```

```
select a.receipt_id, a.purchase_date, a.scan_date, a.store_name, a.user_id, a.barcode, a.final_quantity, b.final_sale
from transactions_ev a inner join transactions_ev b ON a.receipt_id = b.receipt_id and a.barcode = b.barcode
where a.final_sale is null and b.final_quantity is null

--HERES ANOTHER ONE I FOUND THAT WORKED THAT I'M STILL RELATIVELY CONFIDENT IN (SAME STORE, SAME USER, SAME DATE)
select a.receipt_id, a.purchase_date, a.scan_date, a.store_name, a.user_id, a.barcode, a.final_quantity, b.final_sale
```

```
 from transactions_ev a inner join transactions_ev b ON a.user_id =
b.user_id and a.store_name = b.store_name and a.purchase_date =
b.purchase_date
where a.final_sale is null and b.final_quantity is null and
a.receipt_id != b.receipt_id order by a.user_id, a.store_name
```

- `Select count(1) from transactions_ev where final_quantity is null  --12500`
- `select count(1) from transactions_ev where final_sale is null -12500`

- There was also an issue with the transaction log specifically where it had a number of missing barcodes

Barcode Status	Barcode (group)	
MISSING BARCODE ON TRANSACTION LOG	-1	8
	<NA>	5,696
MISSING BARCODE ON PRODUCT LOG	Numeric Barcodes > 0	19,297

-
-
```
SELECT BARCODE, COUNT(1) FROM TRANSACTIONS_EV WHERE BARCODE < 0
GROUP BY BARCODE
```

- Are there any fields that are challenging to understand?
  - For me the fields weren't as challenging to understand as the logging method especially as it relates to the transaction table. For a while I thought that a large number of those records were either bad or duplicated. After spending some time looking through it, I'm more inclined to think that they are in need of a merge. I think the datasets could use some standardization in terms of field types and think that the tables would be more useful with additional elements.

We recommend using SQL or python and data visualization to examine the data.

Second: provide SQL queries

Answer three of the following questions with at least one question coming from the closed-ended and one from the open-ended question set. Each question should be answered using one query.

Closed-ended questions:

- What are the top 5 brands by receipts scanned among users 21 and over?
- What are the top 5 brands by sales among users that have had their account for at least six months?
- What is the percentage of sales in the Health & Wellness category by generation?
  - My first step here was to define what years the generations are classified into. For this I googled it to get a range of years for each generation.

- ```
  SELECT CASE WHEN YEAR(b.birth_date) BETWEEN 1901 AND 1924 THEN 'The
  Greatest Generation'
          WHEN YEAR(b.birth_date) BETWEEN 1925 AND 1945 THEN 'The Silent
  Generation'
          WHEN YEAR(b.birth_date) BETWEEN 1946 AND 1964 THEN 'Baby
  Boomers'
          WHEN YEAR(b.birth_date) BETWEEN 1965 AND 1980 THEN 'Generation
  X'
          WHEN YEAR(b.birth_date) BETWEEN 1981 AND 1996 THEN 'Millennials'
          WHEN YEAR(b.birth_date) BETWEEN 1997 AND 2012 THEN 'Generation
  Z'
          WHEN YEAR(b.birth_date) BETWEEN 2013 AND 2022 THEN 'Generation
  Alpha'
          WHEN YEAR(b.birth_date) >= 2025 THEN 'Generation Beta'
          ELSE 'Unknown Generation'
      END AS Generation,
      (SUM(a.final_sale) / SUM(SUM(a.final_sale)) OVER ()) * 100 AS
  Perc_of_Sales
  FROM transactions_ev a JOIN users_d b ON a.user_id = b.id JOIN
  products_d c ON a.barcode = c.barcode
  WHERE c.category_1 = 'Health & Wellness' AND a.final_sale IS NOT NULL
  GROUP BY CASE WHEN YEAR(b.birth_date) BETWEEN 1901 AND 1924 THEN 'The
  Greatest Generation'
          WHEN YEAR(b.birth_date) BETWEEN 1925 AND 1945 THEN 'The Silent
  Generation'
          WHEN YEAR(b.birth_date) BETWEEN 1946 AND 1964 THEN 'Baby
  Boomers'
          WHEN YEAR(b.birth_date) BETWEEN 1965 AND 1980 THEN 'Generation
  X'
          WHEN YEAR(b.birth_date) BETWEEN 1981 AND 1996 THEN 'Millennials'
          WHEN YEAR(b.birth_date) BETWEEN 1997 AND 2012 THEN 'Generation
  Z'
          WHEN YEAR(b.birth_date) BETWEEN 2013 AND 2022 THEN 'Generation
  Alpha'
          WHEN YEAR(b.birth_date) >= 2025 THEN 'Generation Beta'
          ELSE 'Unknown Generation' END ORDER BY Perc_of_Sales DESC;
  ```

Open-ended questions: for these, make assumptions and clearly state them when answering the question.

- Who are Fetch's power users?
- Which is the leading brand in the Dips & Salsa category?
    - Tostitos is the leader in the dataset. It is worth noting that since the product log is incomplete (see image 2 below), and brand name comes from Product Log, this result could be skewed
  ```




- ```
  SELECT c.category_2, Brand, COUNT(a.receipt_id) AS transaction_count
  FROM transactions_ev a
  INNER JOIN products_d c ON a.barcode = c.barcode
  WHERE c.category_2 = 'Dips & Salsa' AND a.final_sale IS NOT NULL and
  brand is not null
  GROUP BY c.category_2, brand ORDER BY transaction_count DESC;
  ```

- At what percent has Fetch grown year over year?
  - For this specific question I have to assume you are referring to user growth as opposed to transaction growth, as the transaction table only contains roughly 3 months of data from 2024. I also assume you are looking for growth in a full 12 month period, not just since the beginning of 2024 which would only be a 9 month growth rate. Based on those assumptions, user growth has increased by 18.15% in the last 12 months.
  - ```
    SELECT total_users, last_year_total_users,
    CAST((total_users - last_year_total_users) as decimal(18,2))/
    last_year_total_users * 100 as growth_last_12_months
    FROM(
    SELECT
        COUNT(DISTINCT CASE
            WHEN CAST(t.created_date AS DATETIMEOFFSET) < d.dt_id
            THEN t.id
            ELSE NULL
        END) AS last_year_total_users,
        COUNT(DISTINCT t.id) AS total_users
    FROM t_users t
    ```

```
CROSS JOIN (
    SELECT DATEADD(DAY, -365, MAX(CAST(created_date AS DATETIMEOFFSET))) AS
dt_id
    FROM t_users
) d) e;
```

Third: communicate with stakeholders

Construct an email or slack message that is understandable to a product or business leader who is not familiar with your day-to-day work. Summarize the results of your investigation. Include:

- Key data quality issues and outstanding questions about the data
- One interesting trend in the data
    - Use a finding from part 2 or come up with a new insight
- Request for action: explain what additional help, info, etc. you need to make sense of the data and resolve any outstanding issues

SEE SEPARATE ATTACHMENT.