METROCAR FUNNEL ANALYSIS

November 2023
John Ofoegbu

TABLE OF CONTENT

SUMMARY

This project analyzed the customer funnel of Metrocar, a ride-sharing app (like Uber/Lyft), to identify areas of improvement and optimization. The funnel starts with the app download step, runs through sign-up, ride requested, ride accepted, ride completed, payment steps and terminates when a review is provided by the user. The step from ride acceptance to ride completion showed the lowest conversion rate (≈51%) and consequently the highest drop off rate (≈ 49%). At the platform level, ios (60.6%) and android (29.2%) make up a combined ≈90% of Metrocar's users while the rest are web users (10.2%). Considering the composition of users by age, the age groups 25-34 (≈28%) and 35-44 (≈42%) comprise a total of ≈70% of all users. Peak ride requests occur in the mornings (8-10am), trailed by afternoons (4-6pm) and followed closely in the evenings (6-8pm). Therefore, the recommended focus of further research should be on the cause of the low conversion of ride requests to ride completion. Moreover, more marketing budget should be allocated to attracting patronage from users on the ios and android communication platforms within the 25-34 and 35-44 age ranges. Furthermore, the most appropriate implementation of surge pricing would be in the mornings (8-10am), afternoons (4-6pm) and evenings (6-8pm).

CONTEXT

Metrocar's business model is based on a platform that connects riders with drivers through a mobile application. Metrocar acts as an intermediary between riders and drivers, providing a user-friendly platform to connect them and facilitate the ride-hailing process.

Funnel analysis allows businesses and organizations identify where users drop off or convert, helping them to ultimately increase desired outcomes, such as sales, sign-ups, or conversions. It is widely used in e-commerce, marketing, and product development to drive growth and revenue.

In this case, Metrocar's stakeholders have asked several business questions that can uncover valuable insights for improving specific areas of the customer funnel.

The Business Questions

You will need to analyze the data and make recommendations based on the following business questions:

- What steps of the funnel should we research and improve? Are there any specific drop-off points preventing users from completing their first ride?
- Metrocar currently supports 3 different platforms: ios, android, and web. To recommend where to focus our marketing budget for the upcoming year, what insights can we make based on the platform?

- What age groups perform best at each stage of our funnel? Which age group(s) likely contain our target customers?
- Surge pricing is the practice of increasing the price of goods or services when there is the greatest demand for them. If we want to adopt a price-surging strategy, what does the distribution of ride requests look like throughout the day?
- What part of our funnel has the lowest conversion rate? What can we do to improve this part of the funnel?

Metrocar's Funnel

The customer funnel for Metrocar typically includes the following stages:

1. App Download: A user downloads the Metrocar app from the App Store or Google Play Store.
2. Signup: The user creates an account in the Metrocar app, including their name, email, phone number, and payment information.
3. Request Ride: The user opens the app and requests a ride by entering their pickup location, destination, and ride capacity (2 to 6 riders).
4. Driver Acceptance: A nearby driver receives the ride request and accepts the ride.
5. Ride: The driver arrives at the pickup location, and the user gets in the car and rides to their destination.
6. Payment: After the ride, the user is charged automatically through the app, and a receipt is sent to their email.
7. Review: The user is prompted to rate their driver and leave a review of their ride experience.

The Dataset

This dataset is inspired by publicly available datasets for Uber/Lyft. The data for this dataset was generated specifically for this project.

The data can be accessed through Beekeeper using the following URL
postgres://Test:bQNxVzJL4g6u@ep-noisy-flower-846766-pooler.us-east-2.aws.neon.tech/Metrocar

Dataset Structure

Below is a description of each table and its columns.

- **app_downloads**: contains information about app downloads

    - app_download_key: unique id of an app download
    - platform: ios, android or web
    - download_ts: download timestamp
- **signups**: contains information about new user signups
    - user_id: primary id for a user
    - session_id: id of app download
    - signup_ts: signup timestamp
    - age_range: the age range the user belongs to
- **ride_requests**: contains information about rides
    - ride_id: primary id for a ride
    - user_id: foreign key to user (requester)
    - driver_id: foreign key to driver
    - request_ts: ride request timestamp
    - accept_ts: driver accept timestamp
    - pickup_location: pickup coordinates
    - destination_location: destination coordinates
    - pickup_ts: pickup timestamp
    - dropoff_ts: dropoff timestamp
    - cancel_ts: ride cancel timestamp (accept, pickup and dropoff timestamps may be null)
- **transactions**: contains information about financial transactions based on completed rides:
    - ride_id: foreign key to ride
    - purchase_amount_usd: purchase amount in USD
    - charge_status: approved, cancelled
    - transaction_ts: transaction timestamp
- **reviews**: contains information about driver reviews once rides are completed
    - review_id: primary id of review
    - ride_id: foreign key to ride
    - driver_id: foreign key to driver
    - user_id: foreign key to user (requester)

- rating: rating from 0 to 5
- free_response: text response given by user/requester

RESULTS

Number of times App was downloaded: 23,608.

Number of users who signed up on the App: 17,623.

Number of rides that were requested through the App: 385,477.

Number of rides that were requested and completed: 6,233.

Number of unique users who requested rides: 12,406.

Average time of a ride: 00:52:36.738773.

Number of rides accepted by drivers: 248,379.

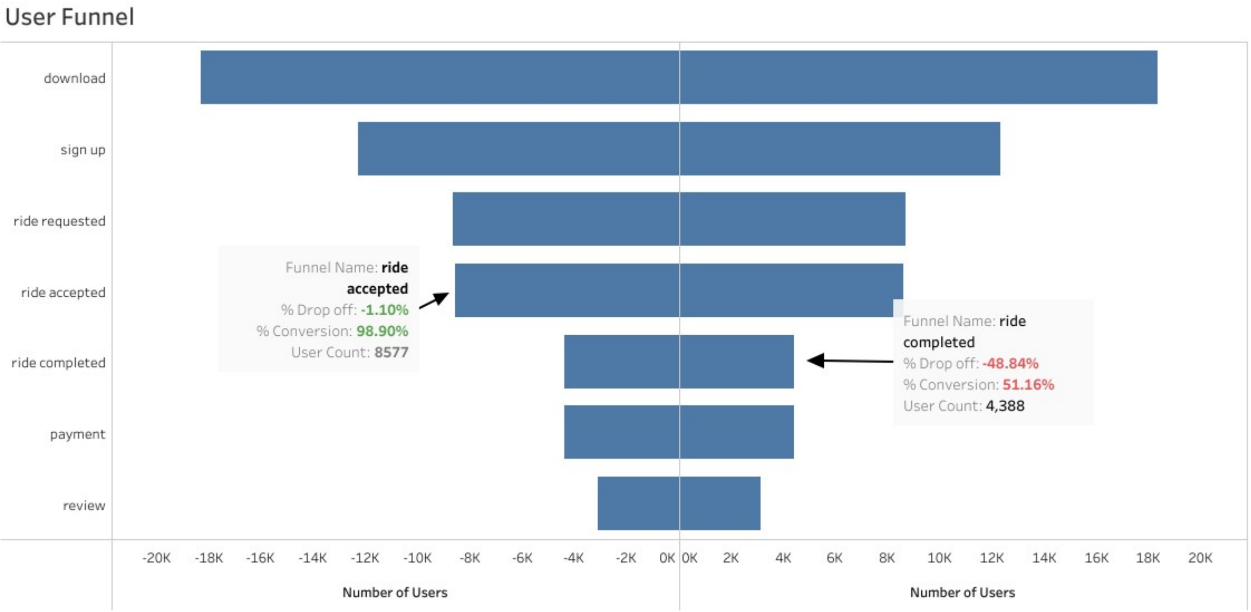
Number of rides with successful payment, and the total amount collected: 212628 & $4251667.61.

Number of ride requests per platform: 38,467 (web); 112,317 (android); 234,693 (ios).

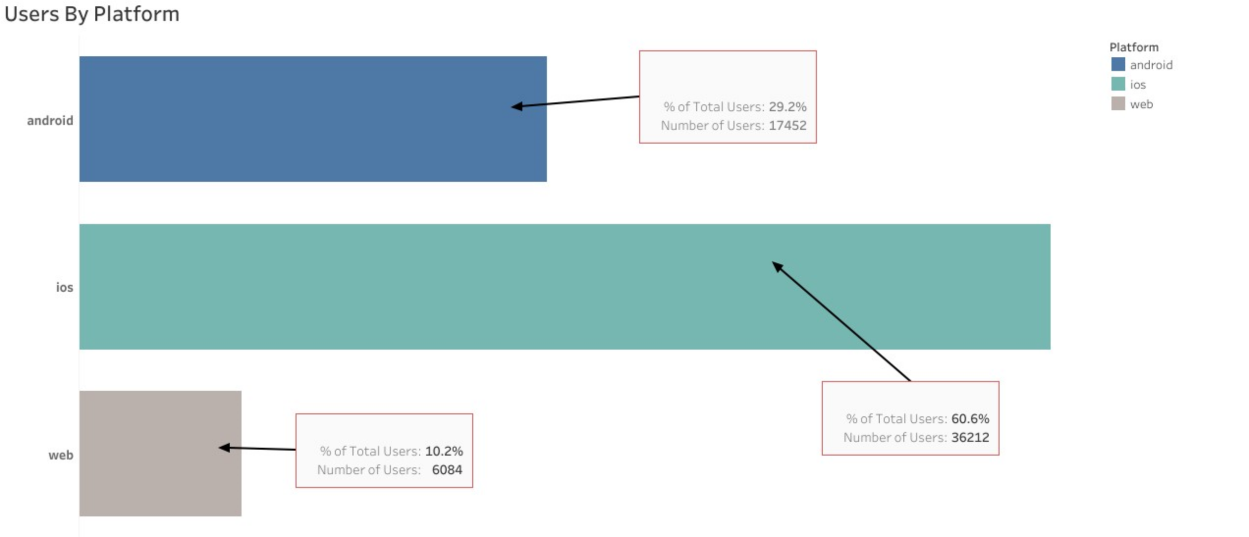
Funnel drop-off data:

Funnel name	**User count**	**Lag**	**Percentage drop-off**	**Percentage conversion rate**
download	23608	0	0	0
sign up	17623	23608	25	75
ride requested	12406	17623	30	70
ride accepted	12278	12406	1	99
payment	6233	12278	49	51
ride completed	6233	6233	0	100
review	4348	6233	30	70

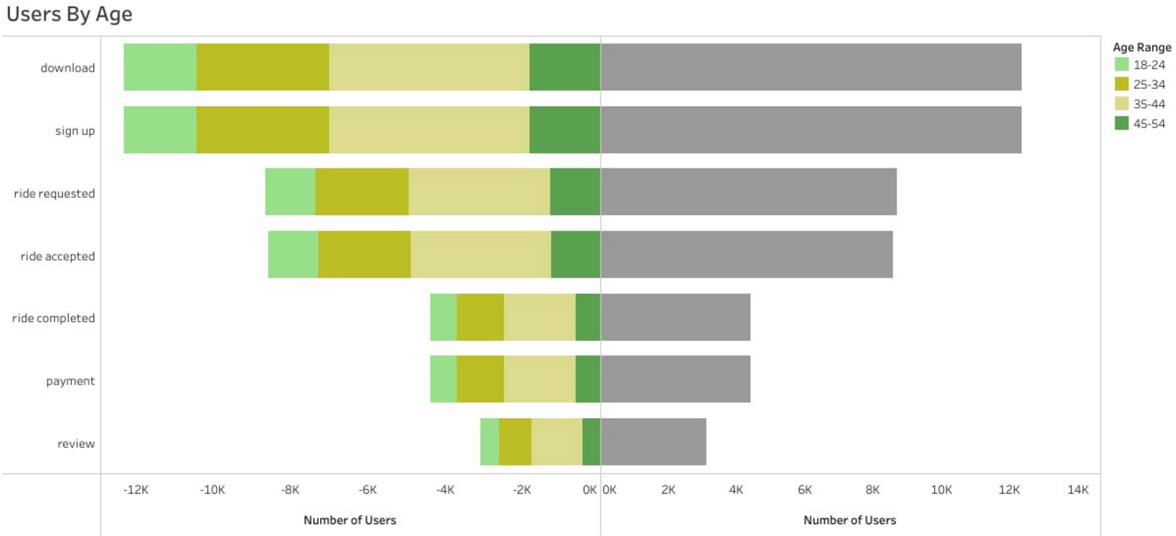
User funnel showing point of greatest drop-off rate/lowest conversion rate.



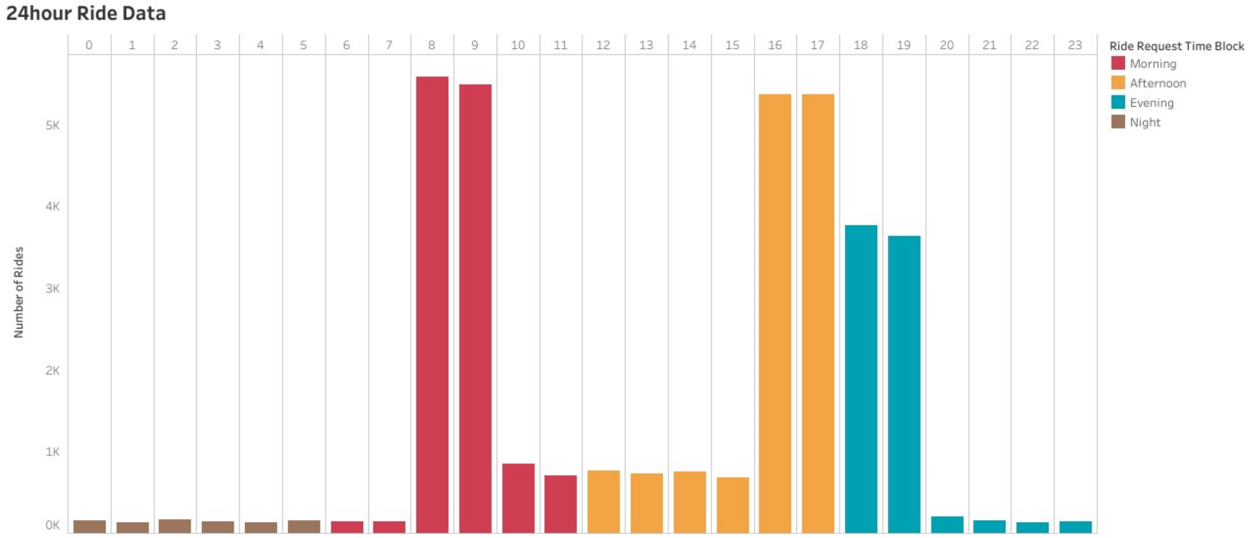
Users by Platform



Users By Age



24-hour ride data



RECOMMENDATIONS

It is recommended that the focus of further research should be on the cause of the low conversion of ride requests to ride completion. Moreover, more marketing budget should be allocated to attracting patronage from users on the ios and android communication platforms within the 25-34 and 35-44 age ranges. Furthermore, the most appropriate implementation of surge pricing would be in the mornings (8-10am), afternoons (4-6pm) and evenings (6-8pm).

APPENDIX

- SQL Codes:

1. Number of times App was downloaded

   ```
   SELECT COUNT (app_download_key)
   FROM app_downloads
   ```

2. Number of users who signed up on the App

   ```
   SELECT COUNT (DISTINCT user_id)
   FROM signups
   ```

3. Number of rides that were requested throught the App

   ```
   SELECT COUNT (ride_id)
   FROM ride_requests
   ```

4. Number of rides that were requested and completed

   ```
   SELECT COUNT (ride_id)
   FROM ride_requests
   WHERE cancel_ts IS NULL
   ```

   OR

   ```
   SELECT COUNT (ride_id)
   FROM ride_requests
   WHERE dropoff_ts IS NOT NULL
   AND pickup_ts IS NOT NULL
   ```

5. No of unique users who requested rides

```
SELECT COUNT (ride_id) no_of_reqstd_rides, COUNT (DISTINCT user_id) no_of_unique_users
FROM ride_requests
```

6. Average time of a ride

```
SELECT AVG(dropoff_ts - pickup_ts) avg_time_of_a_ride
FROM ride_requests
```

7. Number of rides accepted by driver

```
SELECT COUNT (ride_id)
FROM ride_requests
WHERE accept_ts IS NOT NULL
```

8. Number of rides that paid successfully and the total amount collected

```
SELECT COUNT(ride_id) no_of_rides_that_paid_succesfully,
SUM (purchase_amount_usd) total_amt_collected
FROM transactions
WHERE charge_status = 'Approved'
```

9. Number of ride requests per platform

```
SELECT COUNT (r.user_id), a.platform
FROM app_downloads a
RIGHT JOIN signups s
```

```
ON a.app_download_key = s.session_id
JOIN ride_requests r
ON s.user_id = r.user_id
GROUP BY 2
ORDER BY 1
```

10. % Drop-off & % Conversion

```
WITH
visitors AS (

 SELECT  DISTINCT  app_download_key, s.user_id
 FROM app_downloads a
        LEFT JOIN signups s
        ON a.app_download_key = s.session_id
        ),

get_on AS (

 SELECT DISTINCT s.user_id
 FROM signups s),

want_ride AS (

 SELECT DISTINCT r.user_id
 FROM get_on g
 JOIN ride_requests r
 ON g.user_id = r.user_id),
driver_accepted AS (

  SELECT DISTINCT r.user_id
  FROM want_ride w
  JOIN ride_requests r
  ON w.user_id = r.user_id
  WHERE accept_ts IS NOT NULL),


completed_rides AS (

  SELECT DISTINCT r.user_id
  FROM driver_accepted d
  JOIN ride_requests r
  ON d.user_id = r.user_id

  WHERE r.cancel_ts IS NULL),


successful_payments AS (

  SELECT DISTINCT r.user_id
  FROM completed_rides c
  JOIN ride_requests r
  ON c.user_id = r.user_id
  JOIN transactions t
  ON r.ride_id = t.ride_id
  WHERE t.charge_status = 'Approved'),
     reviews AS (

      SELECT DISTINCT r.user_id
      FROM successful_payments sp
      JOIN reviews rev
      ON sp.user_id = rev.user_id
      JOIN ride_requests r
      ON rev.user_id = r.user_id),

     steps AS (

SELECT  'download' AS funnel_name, COUNT (app_download_key)
FROM visitors

UNION

      SELECT  'sign up' AS funnel_name, COUNT (*)
      FROM get_on

      UNION

      SELECT  'ride requested' AS funnel_name, COUNT(*)
      FROM want_ride

      UNION

      SELECT  'ride accepted' AS funnel_name, COUNT (*)
      FROM driver_accepted

      UNION

      SELECT  'ride completed' AS funnel_name, COUNT (*)
      FROM completed_rides

      UNION

      SELECT  'payment' AS funnel_name, COUNT (*)
      FROM successful_payments

      UNION

      SELECT  'review' AS funnel_name, COUNT(*)
      FROM reviews

      ORDER BY COUNT DESC)

      SELECT
                  funnel_name,
                        COUNT user_count,
         COALESCE (LAG(COUNT, 1) OVER (),0) AS lag,
         COALESCE (ROUND((1.0 - COUNT::NUMERIC/LAG(COUNT, 1) OVER ()),2)*100,0) AS percentage_drop_off,
         COALESCE ((1-ROUND((1.0 - COUNT::NUMERIC/LAG(COUNT, 1) OVER ()),2))*100,0) AS percentage_conversion_rate
      FROM steps
```

Query for Resulting data used in data visualization.

```
WITH visitors AS(

  SELECT DISTINCT app_download_key,

(SELECT r.ride_id
FROM app_downloads a
LEFT JOIN signups s
ON a.app_download_key = s.session_id
FULL JOIN ride_requests r
ON s.user_id = r.user_id
GROUP BY a.download_ts, r.ride_id
HAVING a.download_ts < MIN (a.download_ts)), a.platform platform, DATE (a.download_ts) download_dt, s.age_range,  r.request_ts



FROM app_downloads a
LEFT JOIN signups s
ON a.app_download_key = s.session_id
FULL JOIN ride_requests r
ON s.user_id = r.user_id
```

```
          ),

get_in AS (

   SELECT

 (SELECT  r.ride_id
 FROM app_downloads a
 JOIN signups s
 ON s.session_id = a.app_download_key
          LEFT JOIN ride_requests r
          ON s.user_id = r.user_id
 GROUP BY r.ride_id, s.signup_ts
 HAVING s.signup_ts < MIN(s.signup_ts)), s.user_id, a.platform platform, DATE (a.download_ts)
download_dt, s.age_range, r.request_ts



 FROM app_downloads a
 JOIN signups s
 ON s.session_id = a.app_download_key
          LEFT JOIN ride_requests r
          ON s.user_id = r.user_id
          ),


want_ride AS (
  SELECT DISTINCT r.user_id, r.ride_id, a.platform platform, DATE (a.download_ts) download_dt,
s.age_range,  r.request_ts
  FROM get_in g
  JOIN ride_requests r
  ON g.user_id = r.user_id
            JOIN signups s
            ON r.user_id = s.user_id
            JOIN app_downloads a
            ON s.session_id = a.app_download_key),


driver_accepted AS (


  SELECT DISTINCT r.user_id, r.ride_id, a.platform platform, DATE (a.download_ts) download_dt,
s.age_range,r.request_ts
  FROM want_ride w
  JOIN ride_requests r
  ON w.user_id = r.user_id
  JOIN signups s
  ON s.user_id = r.user_id
  JOIN app_downloads a
  ON a.app_download_key = s.session_id
  WHERE accept_ts IS NOT NULL),


completed_rides AS (


  SELECT DISTINCT r.user_id, r.ride_id, a.platform platform, DATE (a.download_ts) download_dt,
s.age_range, r.request_ts
  FROM driver_accepted d
  JOIN ride_requests r
  ON d.user_id = r.user_id
  JOIN signups s
  ON s.user_id = r.user_id
  JOIN app_downloads a
  ON a.app_download_key = s.session_id

  WHERE r.cancel_ts IS NULL),



successful_payments AS (

  SELECT DISTINCT r.user_id, t.ride_id, a.platform platform, DATE (a.download_ts) download_dt,
s.age_range, r.request_ts
  FROM completed_rides c
  JOIN ride_requests r
  ON c.user_id = r.user_id
  JOIN transactions t
  ON r.ride_id = t.ride_id
  JOIN signups s
  ON s.user_id = r.user_id
  JOIN app_downloads a
  ON s.session_id = a.app_download_key
  WHERE t.charge_status = 'Approved'),



reviews AS (
  SELECT DISTINCT r.user_id, rev.ride_id, a.platform platform, DATE (a.download_ts) download_dt,
s.age_range, r.request_ts
  FROM successful_payments sp
  JOIN reviews rev
  ON sp.user_id = rev.user_id
  JOIN ride_requests r
  ON rev.user_id = r.user_id
  JOIN signups s
  ON s.user_id = r.user_id
  JOIN app_downloads a
  ON a.app_download_key = s.session_id)



  SELECT 0 AS funnel_step, 'download' AS funnel_name, COUNT (DISTINCT app_download_key)
user_count,
        COUNT(ride_id) ride_count, platform, download_dt, age_range, request_ts
  FROM visitors
  GROUP BY 5,6,7,8


  UNION

  SELECT 1 AS funnel_step, 'sign up' AS funnel_name, COUNT(DISTINCT user_id) user_count,
  COUNT(ride_id) ride_count, platform, download_dt, age_range, request_ts
  FROM get_in
  GROUP BY 5,6,7,8
```

```
UNION

SELECT 2 AS funnel_step, 'ride requested' AS funnel_name, COUNT(DISTINCT user_id) user_count,
COUNT(ride_id) ride_count, platform, download_dt, age_range, request_ts
FROM want_ride
GROUP BY 5,6,7,8

UNION

SELECT 3 AS funnel_step, 'ride accepted' AS funnel_name, COUNT(DISTINCT user_id) user_count,
COUNT(ride_id) ride_count, platform, download_dt, age_range, request_ts
FROM driver_accepted
GROUP BY 5,6,7,8

UNION

SELECT 4 AS funnel_step, 'ride completed' AS funnel_name, COUNT(DISTINCT user_id) user_count,
COUNT(ride_id) ride_count, platform, download_dt, age_range, request_ts
FROM completed_rides
GROUP BY 5,6,7,8

UNION

SELECT 5 AS funnel_step, 'payment' AS funnel_name, COUNT(DISTINCT user_id) user_count,
COUNT(ride_id) ride_count, platform, download_dt, age_range, request_ts
FROM successful_payments
GROUP BY 5,6,7,8
```

```
UNION


SELECT 6 AS funnel_step, 'review' AS funnel_name, COUNT(DISTINCT user_id) user_count,

COUNT(ride_id) ride_count, platform, download_dt, age_range, request_ts

FROM reviews

GROUP BY 5,6,7,8



ORDER BY funnel_step, platform, age_range, download_dt
```

- Link to Tableau Dashboard: Mazi's Metrocar | Tableau Public
- Link to YouTube video presentation of summary: https://youtu.be/jOAZ80QvoHw
- Lint to csv files used in the visualization: https://docs.google.com/spreadsheets/d/1IG5wsoZQbFqVuAYpfV4YvCwwQFABPMn3fXeGF4QimYk/edit#gid=0