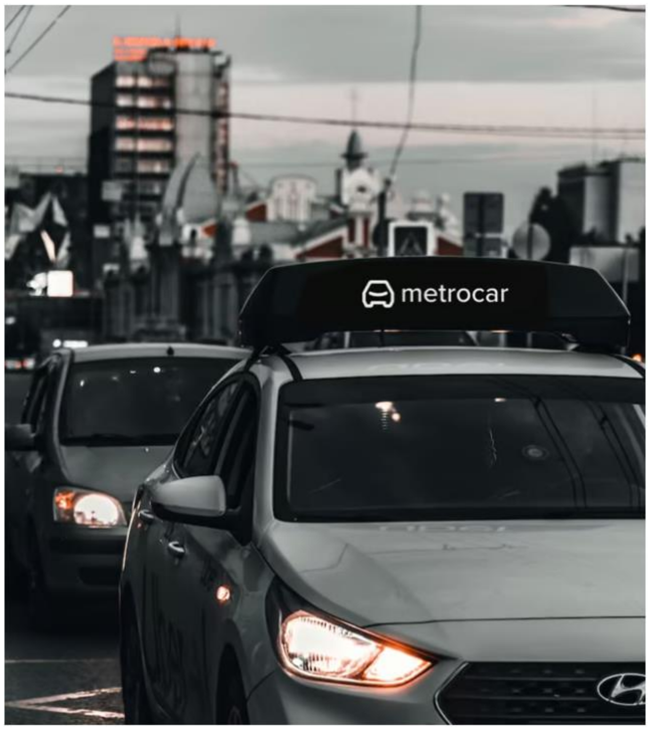


FUNNEL ANALYSIS

Metrocar

Martin Berenstein
MasterSchool | October 2023

Context

This report delves into the systematic analysis of Metrocar, a prominent ride-sharing application, with a primary focus on identifying key areas for enhancement and optimization within its customer funnel. The Metrocar customer funnel consists of several distinct stages:

1. App Download: Users initiate their journey by downloading the Metrocar app from popular platforms like the App Store or Google Play Store.

2. Signup: Following the app download, users create their Metrocar accounts, providing essential information such as their name, email, phone number, and payment details.

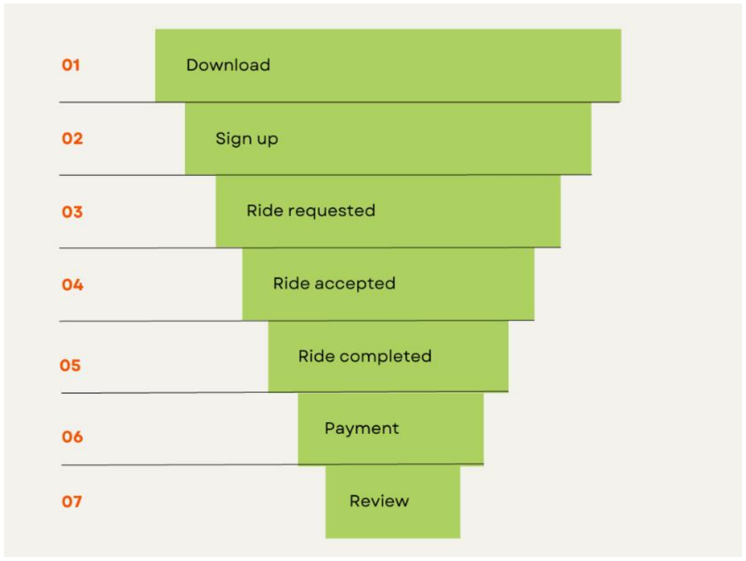
3. Request Ride: Users, now equipped with their Metrocar accounts, open the app to request rides. This stage involves specifying pickup locations, destinations, and ride capacity preferences (ranging from 2 to 6 riders).

4. Driver Acceptance: Nearby drivers come into the picture, receiving, and responding to ride requests by either accepting or declining them.

5. Ride: With successful driver acceptance, users embark on their rides, commencing from the pickup location and concluding at their designated destinations.

6. Payment: After the ride's completion, the Metrocar app automatically processes the payment, with detailed receipts promptly dispatched to users' email inboxes.

7. Review: In the final stage, users are encouraged to rate their driver's performance and provide feedback on their overall ride experience.



Much like in any customer funnel, each of these stages presents opportunities for users to drop off or encounter hurdles. Through funnel analysis, we aim to pinpoint these specific areas for optimization and improvement. This systematic evaluation will not only contribute to enhancing the user experience but also bolster Metrocar's overall standing and success as a ride-sharing platform.

Summary

In our analysis of the ride request funnel, we have identified several critical drop-off points, resulting in just 26.4% of app downloaders completing a ride.

The challenges start with a 25% loss during the transition from app download to sign-up, likely due to the perceived complexity of the sign-up process.

A further 30% drop occurs when transitioning from sign-up to ride request, possibly due to the intricate nature of the ride request process.

Most significantly, a staggering 85% of ride cancellations occur before driver acceptance, with a substantial 50% drop between ride acceptance and completion. These losses are likely influenced by extended waiting times, averaging 15.5 minutes from ride request to pick-up. To solve this issue, it’s recommendable to shorten the waiting times from ride request until the acceptance of that ride by the drivers.

Additionally, a 30% drop from payment to leaving a review indicates the need for streamlined review processes and reminders.

Positively, conversion rates remain unaffected by platform variations, indicating that funnel enhancements can be universally applied. Most users have embraced the app through iOS devices, with Android and web users following suit. Similarly, age groups show consistent conversion rates, emphasizing the efficacy of universal solutions. The age group with the highest download volume is 35-44, closely followed by 25-34. Focusing on optimizing surge pricing during peak hours, specifically between 8 AM – 9 AM and 4 PM – 7 PM, holds the potential to elevate user experience and overall efficiency.

Results

1. Funnel Conversion Insights

Throughout the user journey, several critical drop-off points have been identified, leading to only 26.4% of users who downloaded the app completing a ride.

Firstly, during the transition from downloading the app to signing up, 25% of users were lost. This loss is likely due to the perceived complexity or inconvenience of the signup process.

Moving from the signup stage to requesting a ride saw a 30% drop in users. This drop can be attributed to the intricate nature of the ride request process.

Looking at the user granularity, the most significant drop off occurred from ride acceptance to ride completion, with a 50% loss, giving the idea that customers are canceling rides before even starting the trip. A concerning 85% of these ride cancellations occurred before the driver accepted the ride. These substantial losses are probably due to the long waiting times between the ride request and ride acceptance (an average of 7 minutes) and the waiting time from ride acceptance to pick up (an average of 8.5 minutes), leading to an average of 15.5 minutes of waiting since the user requests a ride until pick up.

Lastly, the shift from payment to leaving a review experienced a 30% loss. This could be because users are unwilling to invest the time or simply forget. Addressing this issue may involve implementing reminders upon reentry to the app and streamlining the review process.

Figure 1: Funnel Analysis at user granularity

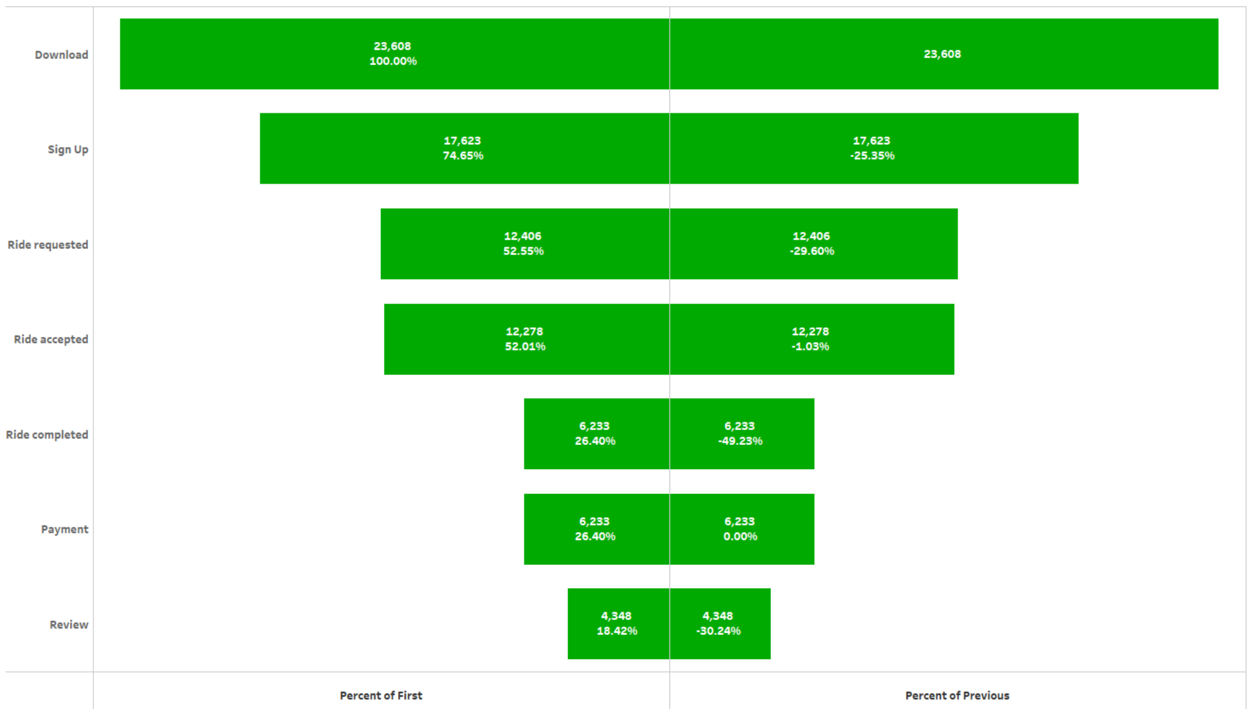
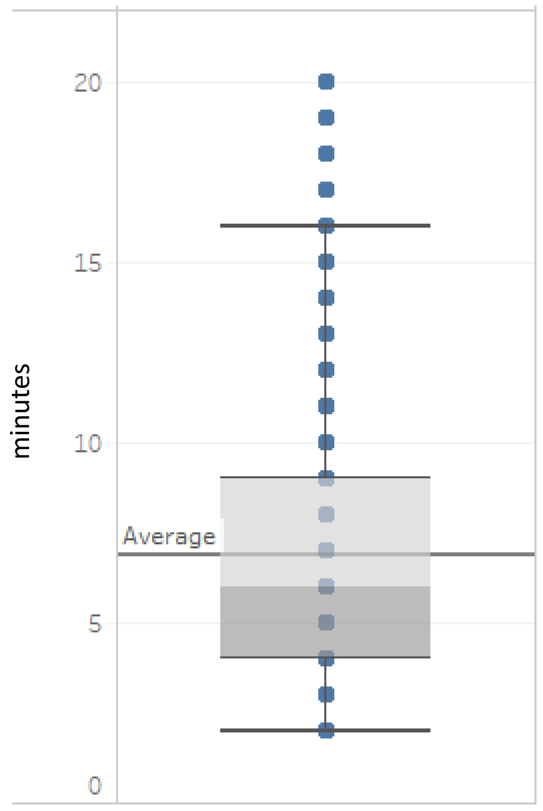
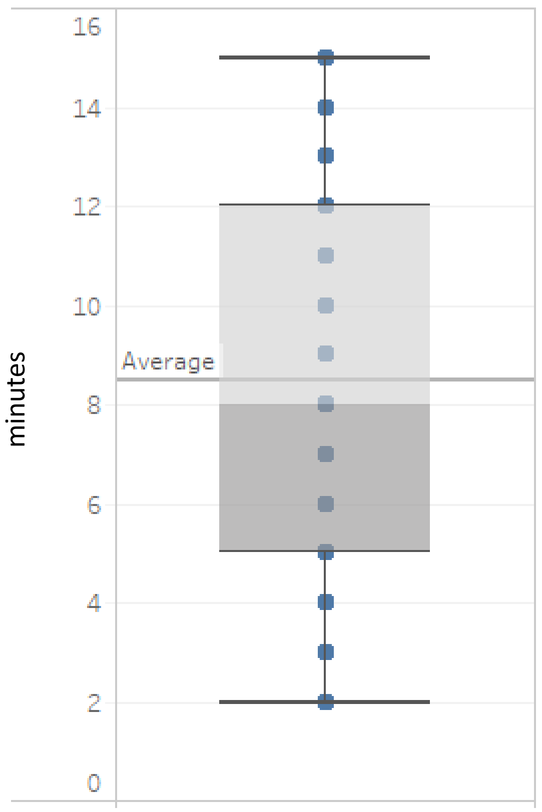


Figure 2: Distribution of waiting times (in minutes)

2.a Waiting until ride acceptance from driver.



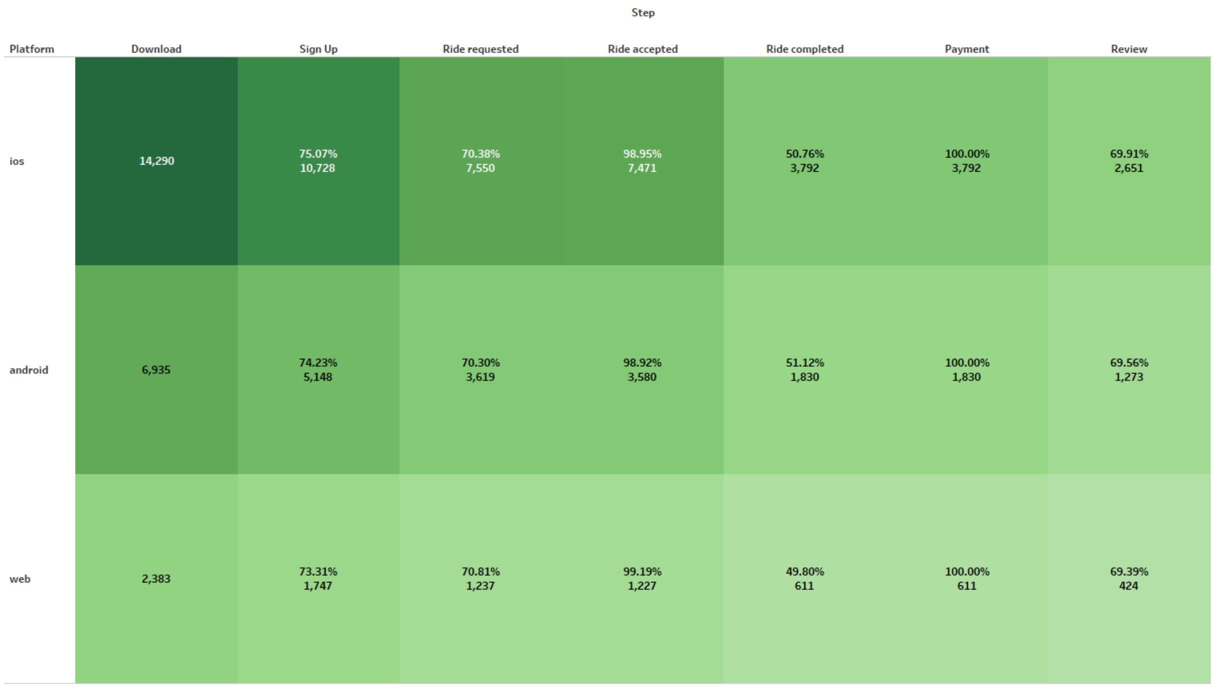
2.b Waiting from ride acceptance until pick up.



2. Platform Insights

Platform analysis revealed that iOS was the most popular platform for downloads, while web-based downloads were the lowest. Despite platform variations, the conversion rates across different steps in the funnel remained consistent. Therefore, improving the user experience and addressing the funnel issues is not limited to any specific platform.

Figure 3: Funnel Analysis for each platform at user granularity

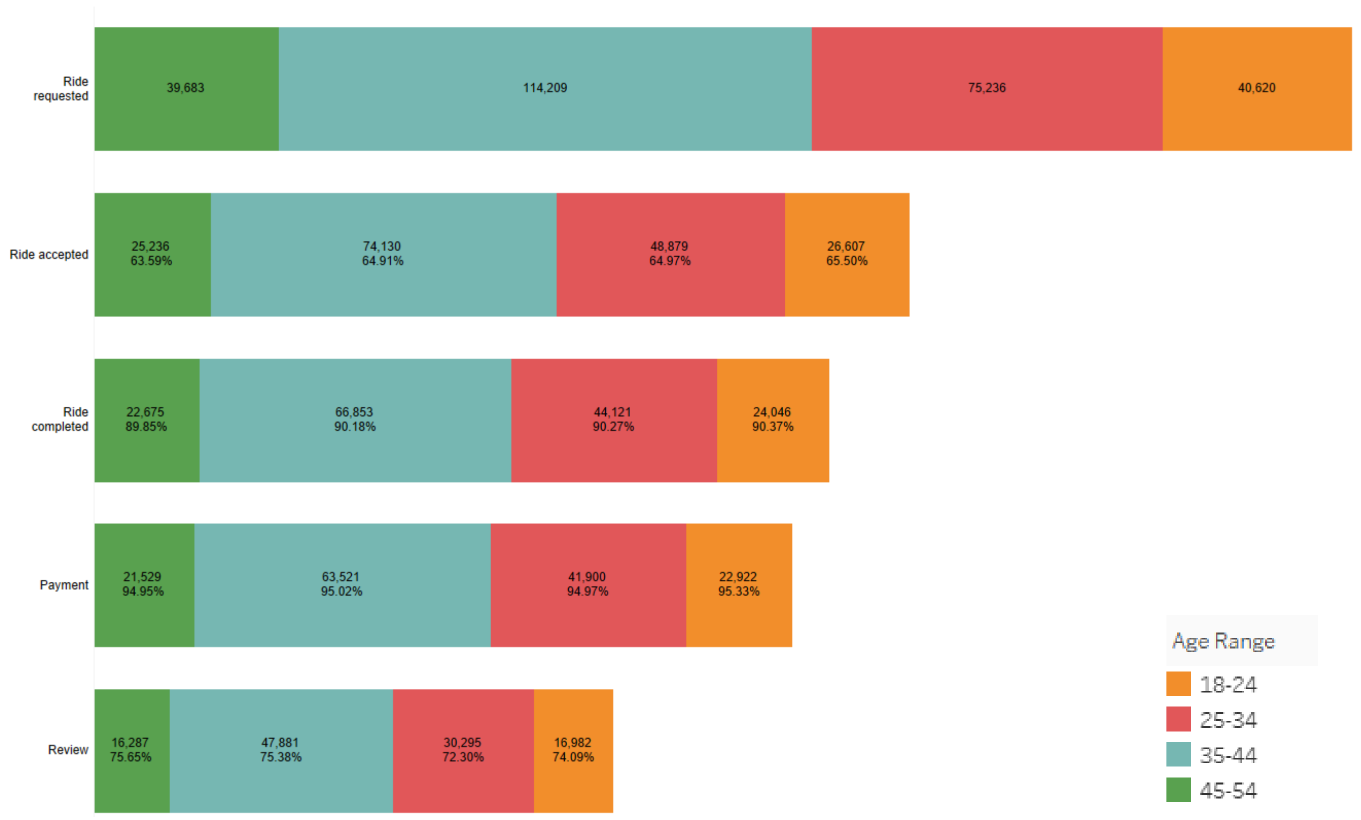


	Step						
Platform	Download	Sign Up	Ride requested	Ride accepted	Ride completed	Payment	Review
ios	14,290	75.07% 10,728	70.38% 7,550	98.95% 7,471	50.76% 3,792	100.00% 3,792	69.91% 2,651
android	6,935	74.23% 5,148	70.30% 3,619	98.92% 3,580	51.12% 1,830	100.00% 1,830	69.56% 1,273
web	2,383	73.31% 1,747	70.81% 1,237	99.19% 1,227	49.80% 611	100.00% 611	69.39% 424

3. Age Range Insights

Users between the ages of 35-44 constituted the largest group of downloads, followed closely by the 25-34 age range. The other two categories (18-24 and 45-54) had similar number of downloads. However, conversion rates across different age groups did not exhibit significant variations. This suggests that the challenges within the funnel are not age-specific and should be addressed universally.

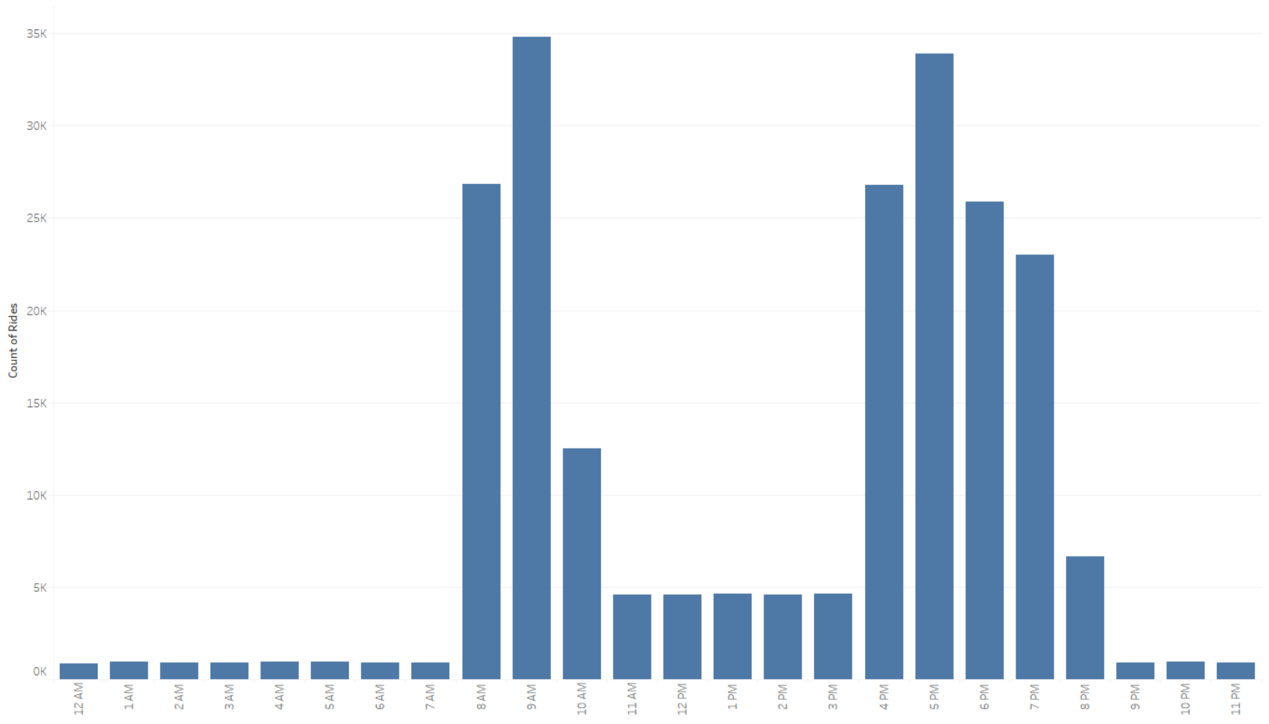
Figure 4: Funnel Analysis for age ranges at user granularity



4. Ride Request Distribution for Surge Pricing

The analysis of ride request distribution throughout the day revealed that the busiest hours were between 8 AM – 9 AM and 4 PM – 7 PM. This pattern remained consistent throughout the week. These insights can be used to optimize surge pricing strategies and improve the user experience during peak hours.

Figure 5: Distribution of rides throughout the day



Appendix

- Presentation
- Tableau Workbook
- SQL query:

```
-- How many times was the app downloaded?

SELECT COUNT(*) as total_downloads
FROM app_downloads;

-- How many users signed up on the app?

SELECT COUNT(*) as total_signups
FROM signups;

-- How many rides were requested through the app?

SELECT COUNT(*) as total_rides_requested
FROM ride_requests;

--How many rides were requested and completed through the app?

WITH user_ride_status AS (
    SELECT
        ride_id,
        MAX(
            CASE
                WHEN dropoff_ts IS NOT NULL
            THEN 1
            ELSE 0
          END
        ) AS rides_completed
    FROM ride_requests
    GROUP BY ride_id
)
SELECT
    COUNT(*) AS total_rides_requested,
    SUM(rides_completed) AS total_rides_completed
FROM user_ride_status;

-- How many rides were requested and how many unique users requested a ride?

SELECT
    COUNT(DISTINCT user_id) AS total_users_ride_requested,
    COUNT(user_id) total_rides_requested
FROM ride_requests;

-- What is the average time of a ride from pick up to drop off?

SELECT
AVG(dropoff_ts-pickup_ts) AS avg_ride_time
FROM ride_requests;

-- How many rides were accepted by a driver?

WITH driver_status AS (
    SELECT
        ride_id,
        MAX(
            CASE
                WHEN accept_ts IS NOT NULL
                THEN 1
                ELSE 0
            END
        ) AS driver_accepted
    FROM ride_requests
    GROUP BY ride_id
)
SELECT
    SUM(driver_accepted) AS total_rides_accepted
FROM driver_status;



-- How many rides did we successfully collect payments and how much was collected?

SELECT
COUNT(DISTINCT ride_id) AS rides_collected_payments,
SUM(purchase_amount_usd)
FROM transactions
WHERE charge_status = 'Approved';

-- How many ride requests happened on each platform?

SELECT
platform,
COUNT(DISTINCT ride_id) AS total_ride_requests
FROM ride_requests r
LEFT JOIN signups s
```

```
ON r.user_id = s.user_id
LEFT JOIN app_downloads d
ON s.session_id = d.app_download_key
GROUP BY platform
;

-- What is the drop-off from users signing up to users requesting a ride?

WITH total_signups AS (
SELECT COUNT(*) AS total_s FROM signups),
total_requests AS (
SELECT COUNT(DISTINCT user_id) AS total_r FROM ride_requests)
SELECT
(total_s - total_r)::float/total_s as drop_off
FROM total_signups, total_requests



-- Waiting time distribution (from ride accepted until pickup)

SELECT
AVG(pickup_ts - accept_ts) AS avg_waiting_time,
MIN(pickup_ts - accept_ts) AS min_waiting_time,
MAX(pickup_ts - accept_ts) AS max_waiting_time
FROM ride_requests



-- Funnel Analysis

WITH Visits AS (
   SELECT
```

```
                                DISTINCT app_download_key AS user_id,
                                platform,
                                age_range,
                                DATE(download_ts) AS Date
    FROM app_downloads v
                    FULL JOIN signups
                    ON session_id = app_download_key

),
Signups AS (
    SELECT DISTINCT s.user_id AS user_id,
            platform,
                    s.age_range,
                    Date
    FROM signups s
            LEFT JOIN Visits v
                    ON session_id = v.user_id
),
ride_requested AS (
    SELECT DISTINCT r.user_id, ride_id, platform,
                    age_range,
                    Date
    FROM ride_requests r
                    LEFT JOIN Signups s
            ON r.user_id = s.user_id
),
ride_accepted AS (
    SELECT DISTINCT
        CASE
            WHEN accept_ts IS NOT NULL
            THEN r.user_id
        END AS user_id,
        CASE
            WHEN accept_ts IS NOT NULL
            THEN r.ride_id
        END AS ride_id,
                              platform,
                    age_range,
                    Date
    FROM ride_requests r
            LEFT JOIN Signups s
            ON r.user_id = s.user_id
),
ride_completed AS (
    SELECT DISTINCT
        CASE
            WHEN dropoff_ts IS NOT NULL
            THEN r.user_id
        END AS user_id,
        CASE
            WHEN dropoff_ts IS NOT NULL
            THEN r.ride_id
        END AS ride_id, platform,
                    age_range,
                    Date
    FROM ride_requests r
            LEFT JOIN Signups s
            ON r.user_id = s.user_id
),
Payment AS (
    SELECT DISTINCT
        CASE
            WHEN charge_status = 'Approved'
            THEN r.user_id
        END AS user_id,
        CASE
            WHEN charge_status = 'Approved'
            THEN t.ride_id
        END AS ride_id, platform,
                    age_range,
                    Date
    FROM transactions t
    LEFT JOIN ride_requests r ON t.ride_id = r.ride_id
            LEFT JOIN Signups s
            ON r.user_id = s.user_id
),
Review AS (
    SELECT DISTINCT r.user_id, r.ride_id, platform,
                    age_range,
                    Date
    FROM reviews r
            LEFT JOIN Signups s
            ON r.user_id = s.user_id
),
steps AS (
    SELECT 'Download' AS step, COUNT(Visits) AS count, null::bigint as ride_count, platform,
age_range, Date FROM Visits GROUP BY platform, age_range, Date
    UNION
    SELECT 'Sign Up' AS step, COUNT(Signups) AS count, null::bigint as ride_count , platform,
age_range, Date  FROM Signups GROUP BY platform, age_range, Date
    UNION
    SELECT 'Ride requested' AS step, COUNT(DISTINCT user_id) AS user_count,
COUNT(DISTINCT ride_id) AS ride_count , platform, age_range, Date  FROM ride_requested
GROUP BY platform, age_range, Date

    UNION

    SELECT 'Ride accepted' AS step, COUNT(DISTINCT user_id) AS user_count,
COUNT(DISTINCT ride_id) AS ride_count , platform, age_range, Date  FROM ride_accepted
GROUP BY platform, age_range, Date

    UNION

    SELECT 'Ride completed' AS step, COUNT(DISTINCT user_id) AS user_count,
COUNT(DISTINCT ride_id) AS ride_count , platform, age_range, Date  FROM ride_completed
GROUP BY platform, age_range, Date

    UNION

    SELECT 'Payment' AS step, COUNT(DISTINCT user_id) AS user_count, COUNT(DISTINCT
ride_id) AS ride_count , platform, age_range, Date  FROM Payment GROUP BY platform,
age_range, Date

    UNION

    SELECT 'Review' AS step, COUNT(DISTINCT user_id) AS user_count, COUNT(DISTINCT
ride_id) AS ride_count , platform, age_range, Date  FROM Review GROUP BY platform,
age_range, Date

)

SELECT

    step,

    platform,

    age_range,

    Date,

                    count AS user_count,

    ride_count


    --LAG(count, 1) OVER () AS previous_count,

    --ROUND((1.0 - count::numeric / LAG(count, 1) OVER ()), 2) AS drop_off

FROM steps

ORDER BY step, platform, age_range,date ASC;
```

```
-- How many bad reviews were realted to the drivers?

SELECT review

FROM reviews

WHERE rating = 1

AND review LIKE '%driver%';


-- How many cancellations were before the driver accepted the ride and what was that average
waiting time?

SELECT

COUNT(ride_id),

AVG(cancel_ts-request_ts) AS avg_time_until_cancellation

FROM ride_requests

WHERE cancel_ts IS NOT NULL and accept_ts IS NULL
```