

Metrocar Funnel Analysis: written report

By Fedor Dorokhov

Summary

In our comprehensive funnel analysis, we examined the user journey within Metrocar app and identified critical areas for optimization. We found out that the ride completion stage experiences almost a 50% drop-off rate, making that funnel step a primary target to research and improve. iOS seems to be the most popular platform in absolute numbers among our customer base while web platform audience needs to be increased by marketing efforts. Users aged 35-44 outnumbered the others, however, 18-24 group displayed the highest conversion rate in three funnel stages (acceptance, completion and payment), both age categories could be made a prime target demographic. The data supports the feasibility of adopting surge pricing strategies during peak hours, with 8-9 AM and 4-7 PM time periods showing the highest ride requests rate. The stage with the lowest conversion rate (50.77%) in our funnel is ride completion stage, possibly due to low average drivers' rating, long waiting time and especially highly competitive environment during the peak hours.

Context

This project aims to analyze the customer funnel of Metrocar, a ride-sharing app, to identify areas for improvement and optimization. SQL was used for querying the dataset and Tableau BI – for funnel visualizations. Several business questions were asked to uncover valuable insights for improving particular areas of Metrocar funnel. Funnel analysis was conducted to address those five questions:

- 1) What steps of the funnel should we research and improve? Are there any specific drop-off points preventing users from completing their first ride?

- 2) 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?
- 3) What age groups perform best at each stage of our funnel? Which age group(s) likely contain our target customers?
- 4) 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-surfing strategy, what does the distribution of ride requests look like throughout the day?
- 5) What part of our funnel has the lowest conversion rate? What can we do to improve this part of the funnel?

Parameter to toggle between funnel by user and funnel by ride count was added. It is also important to notice that our funnel contains two visualized approaches (*Percent of Previous* and *Percent of Top*) to measure conversion rates on each stages. There are 5 tables included in our database, which generally related to Metrocar funnel steps. Here is how funnel looks like.

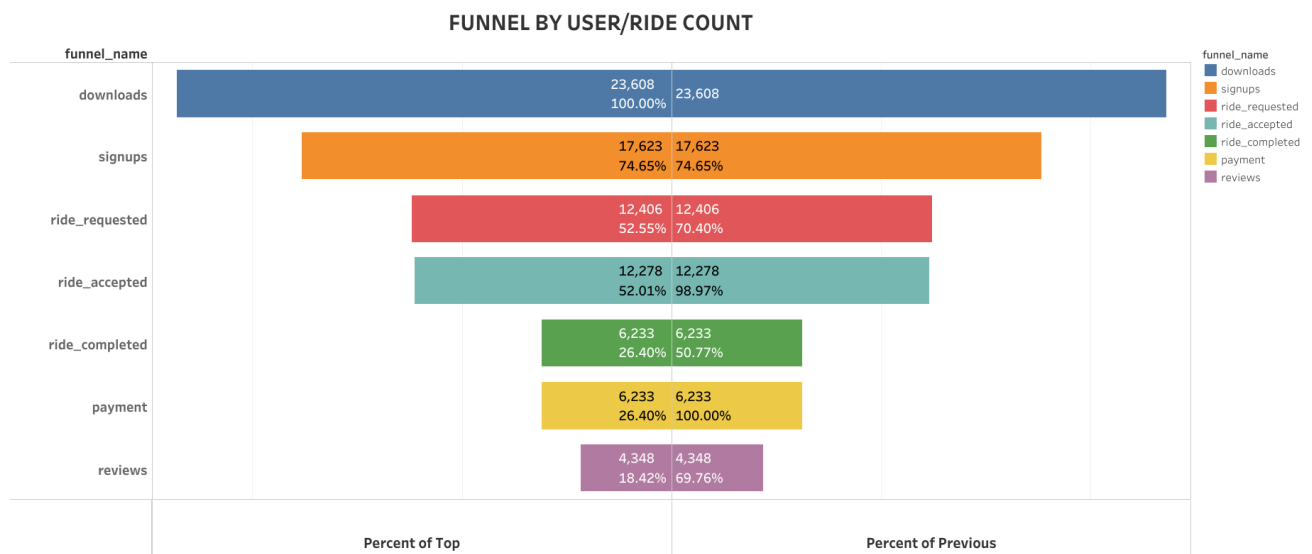
- 1) App Download: A user downloads the Metrocar app from the App Store or Google Play Store.
- 2) Signup: The user creates an account in the Metrocar app, including their name, email, phone number, and payment information.
- 3) Request Ride: The user opens the app and requests a ride by entering their 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.

Results

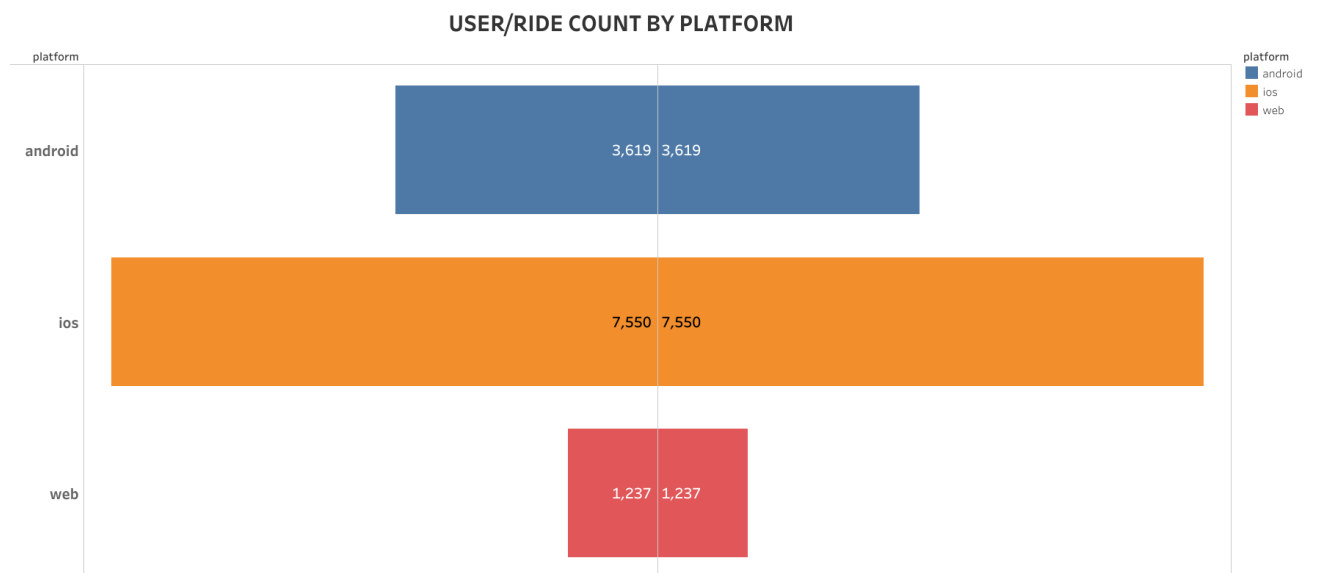
Where Are Users Dropping Off?

Conducting the funnel analysis, the first and most obvious question is which stage of our funnel should be researched and improved in a first place. Observing our dataset based on the drop-off rate between different steps described above we came to the conclusion, that the highest drop-off (almost 50%) appears between the ride acceptance and ride completion stages. In the visualization below it can be seen in terms of conversion rate (the drop-off rate is basically conversion rate subtracted from one).



Platform Insights: Maximizing Marketing Impact

Metrocar currently supports 3 different platforms: ios, android, and web. To understand, where to focus our marketing budget for the upcoming year, we made a dynamic user/ride counter by platform. iOS seems to be the most popular platform, and web appears to be the least. For instance, see the visualization below for ride requests distribution for all platforms (on user level):



The pattern remains the same for all funnel stages, which drives us to pretty straightforward conclusion – **the budget should be made considering both the popularity of iOS platform and the necessity of increasing Metrocar’s audience uses the web platform.**

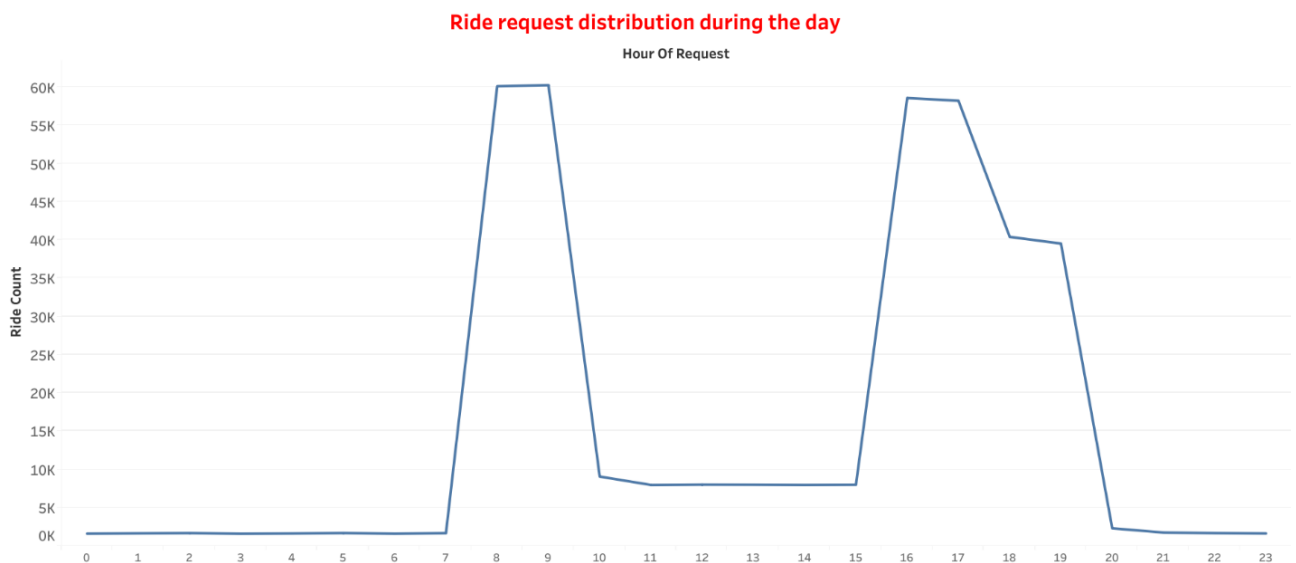
Age Groups and Funnel Performance: Finding Your Audience

When we analyzed which age groups perform best at each stage of our funnel, we arrived at somewhat ambiguous conclusions. On one hand, the 35-44 years old category emerged as an indisputable champion in terms of absolute numbers, boasting 5,181 users. Clearly, this category outnumbered its rivals at every stage of the funnel. However, a different perspective emerges when we consider conversion rates. In fact, 18-24 category showed the best performance in terms of the highest conversion rate in three funnel stages: ride acceptance, ride completion and payment (calculated on ride level).

So, there could be two possible approaches in identifying our target audience: pure numerical, which points on 35-44 category as a clear winner and more complex one, which takes into account the smaller percent of drop-offs in 18-24 group.

Is Surge Pricing Feasible?

Surge pricing is the practice of increasing the price of goods or services when there is the greatest demand for them. To adopt a price-surfing strategy we visualized the distribution of ride requests throughout the day. According to the analyzed data, **time periods when services' prices should be raised are in between 8 and 9 AM and 4 and 7 PM**. The chart below illustrates the rides' frequency distribution by hour.

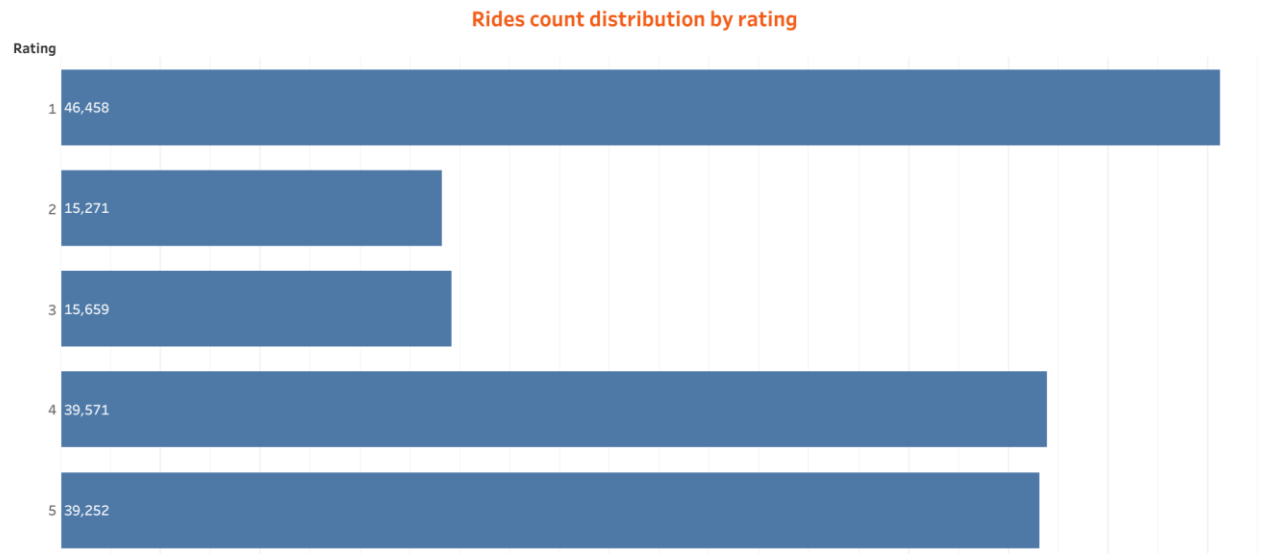


Fixing the Weakest Link in the Funnel

As you probably noticed at “Where Are Users Dropping Off?” section (see the funnel visualization), the stage with the lowest conversion rate (50.77%) in our funnel (according to percent of previous approach) is ride completion stage, in other words, the funnel step, when the ride is requested and accepted by driver, but pickup doesn't happen. There could be several reasons for that:

- 1) The majority of cancellations happens at the hours of greatest demand (see the previous graph), which leads us to the conclusion, that our potential customers **find better offer with our competitors**.
- 2) If there is an option for a user to see driver's average rating in the app after ride's acceptance it also can move our customers to cancel the ride, since the average rating

of Metrocar's drivers is 3.06 from 5, which is definitely not a stellar number. Moreover, the fact that there are more rides with a rating of "1" than any other rating points on the bad level of service as a key issue, see the graph below:



The recommendation here is to pay attention on customers' reviews to identify where such a low average rating is coming from and which part of Metrocar's service users find most unpleasant.

3) The average overall customer's waiting time (the difference between the cancellation and ride request timestamps) is almost 21 minutes. **The possible way to reduce that metric could be hiring more drivers or re-spreading the current ones more equally across the neighborhoods.**

Appendix

SQL queries:

Main funnel dataset, aggregated by platform, age and download date

```
WITH downloads
  AS (SELECT app.platform,
             Date(app.download_ts) AS download_dt,
             s.age_range,
             Count(*)              AS download_count
  FROM   app_downloads app
        LEFT JOIN signups s
             ON s.session_id = app.app_download_key
  GROUP BY app.platform,
           s.age_range,
           Date(app.download_ts)),
signups_users
  AS (SELECT app.platform,
             Date(app.download_ts) AS download_dt,
             s.age_range,
             Count(s.user_id)      AS signup_users
  FROM   signups s
        JOIN app_downloads app
             ON s.session_id = app.app_download_key
  GROUP BY app.platform,
           s.age_range,
           Date(app.download_ts)),
ride_users
  AS (SELECT app.platform,
             Date(app.download_ts) AS download_dt,
             s.age_range,
             Count(DISTINCT r.user_id) AS ride_users,
             Count(r.ride_id)         AS ride_count
  FROM   app_downloads app
```

```

        JOIN signups s
            ON s.session_id = app.app_download_key
        JOIN ride_requests r
            ON r.user_id = s.user_id
    GROUP BY app.platform,
             s.age_range,
             Date(app.download_ts)),
accepted_rides
AS (SELECT app.platform,
           Date(app.download_ts) AS download_dt,
           s.age_range,
           Count(DISTINCT r.user_id) AS accepted_rides,
           Count(r.ride_id) AS ride_count
    FROM app_downloads app
        JOIN signups s
            ON s.session_id = app.app_download_key
        JOIN ride_requests r
            ON r.user_id = s.user_id
    WHERE r.accept_ts IS NOT NULL
    GROUP BY app.platform,
             s.age_range,
             Date(app.download_ts)),
completed_rides
AS (SELECT app.platform,
           Date(app.download_ts) AS download_dt,
           s.age_range,
           Count(DISTINCT r.user_id) AS completed_rides,
           Count(r.ride_id) AS ride_count
    FROM app_downloads app
        JOIN signups s
            ON s.session_id = app.app_download_key
        JOIN ride_requests r
            ON r.user_id = s.user_id
    WHERE r.cancel_ts IS NULL
    GROUP BY app.platform,
             s.age_range,

```



```

        Date(app.download_ts)),
paid_rides
AS (SELECT app.platform,
        Date(app.download_ts)      AS download_dt,
        s.age_range,
        Count(DISTINCT r.user_id) AS paid_rides,
        Count(DISTINCT r.ride_id) AS ride_count
FROM   app_downloads app
JOIN   signups s
        ON s.session_id = app.app_download_key
JOIN   ride_requests r
        ON r.user_id = s.user_id
LEFT JOIN transactions t
        ON r.ride_id = t.ride_id
WHERE  r.accept_ts IS NOT NULL
AND    r.cancel_ts IS NULL
AND    t.charge_status = 'Approved'
GROUP BY app.platform,
        s.age_range,
        Date(app.download_ts)),
users_reviews
AS (SELECT app.platform,
        Date(app.download_ts)      AS download_dt,
        s.age_range,
        Count(DISTINCT rev.user_id) AS reviews,
        Count(rev.ride_id)          AS ride_count
FROM   app_downloads app
JOIN   signups s
        ON s.session_id = app.app_download_key
JOIN   ride_requests r
        ON r.user_id = s.user_id
JOIN   reviews rev
        ON rev.ride_id = r.ride_id
GROUP BY app.platform,
        s.age_range,
        Date(app.download_ts)),

```

```

funnel_stages
AS (SELECT platform,
           0                                     AS funnel_step,
           age_range,
           download_dt,
           'downloads'                          AS funnel_name,
           downloads.download_count AS user_count,
           0                                     AS ride_count
FROM   downloads
UNION
SELECT platform,
           1                                     AS funnel_step,
           age_range,
           download_dt,
           'signups'                          AS funnel_name,
           signups_users.signup_users AS user_count,
           0                                     AS ride_count
FROM   signups_users
UNION
SELECT platform,
           2                                     AS funnel_step,
           age_range,
           download_dt,
           'ride_requested' AS funnel_name,
           ride_users.ride_users AS user_count,
           ride_users.ride_count AS ride_count
FROM   ride_users
UNION
SELECT platform,
           3                                     AS funnel_step,
           age_range,
           download_dt,
           'ride_accepted' AS funnel_name,
           accepted_rides.accepted_rides AS user_count,
           accepted_rides.ride_count AS ride_count
FROM   accepted_rides

```

```

UNION
SELECT platform,
        4                                AS funnel_step,
        age_range,
        download_dt,
        'ride_completed'                AS funnel_name,
        completed_rides.completed_rides AS user_count,
        completed_rides.ride_count      AS ride_count
FROM    completed_rides
UNION
SELECT platform,
        5                                AS funnel_step,
        age_range,
        download_dt,
        'payment'                       AS funnel_name,
        paid_rides.paid_rides           AS user_count,
        paid_rides.ride_count           AS ride_count
FROM    paid_rides
UNION
SELECT platform,
        6                                AS funnel_step,
        age_range,
        download_dt,
        'reviews'                       AS funnel_name,
        users_reviews.reviews           AS user_count,
        users_reviews.ride_count        AS ride_count
FROM    users_reviews
ORDER BY funnel_step,
        platform,
        age_range,
        4)

```

```

SELECT *
FROM    funnel_stages

```

General funnel with drop-off and conversion rates (percent of previous)

```
WITH downloads
  AS (SELECT Count(*) AS download_count
      FROM app_downloads),
signups_users
  AS (SELECT Count(user_id) AS signup_users
      FROM signups),
ride_users
  AS (SELECT Count(DISTINCT user_id) AS ride_users
      FROM ride_requests),
accepted_rides
  AS (SELECT Count(DISTINCT user_id) AS accepted_rides
      FROM ride_requests
      WHERE accept_ts IS NOT NULL),
completed_rides
  AS (SELECT Count(DISTINCT user_id) AS completed_rides
      FROM ride_requests
      WHERE cancel_ts IS NULL),
paid_rides
  AS (SELECT Count(DISTINCT r.user_id) AS paid_rides
      FROM ride_requests r
      left join transactions t
            ON r.ride_id = t.ride_id
      WHERE r.accept_ts IS NOT NULL
            AND r.cancel_ts IS NULL),
users_reviews
  AS (SELECT Count(DISTINCT user_id) AS reviews
      FROM reviews),
funnel_stages
  AS (SELECT 0 AS funnel_step,
            'downloads' AS funnel_name,
            downloads.download_count AS value
      FROM downloads
```

```

UNION
SELECT 1                                AS funnel_step,
      'signups'                          AS funnel_name,
      signups_users.signup_users AS value
FROM   signups_users
UNION
SELECT 2                                AS funnel_step,
      'ride_requested'                   AS funnel_name,
      ride_users.ride_users AS value
FROM   ride_users
UNION
SELECT 3                                AS funnel_step,
      'ride_accepted'                   AS funnel_name,
      accepted_rides.accepted_rides AS value
FROM   accepted_rides
UNION
SELECT 4                                AS funnel_step,
      'ride_completed'                   AS funnel_name,
      completed_rides.completed_rides AS value
FROM   completed_rides
UNION
SELECT 5                                AS funnel_step,
      'ride_paid'                        AS funnel_name,
      paid_rides.paid_rides AS value
FROM   paid_rides
UNION
SELECT 6                                AS funnel_step,
      'reviews'                          AS funnel_name,
      users_reviews.reviews AS value
FROM   users_reviews
ORDER BY funnel_step)
SELECT *,
       value :: FLOAT / Lag(value)
              over (
                  ORDER BY funnel_step) AS convers
ion_rate,

```

```

1 - value :: FLOAT / Lag(value)
                        over (
                            ORDER BY funnel_step) AS drop_of
f
FROM    funnel_stages

```

General funnel with drop-off and conversion rates (percent of top)

```

WITH downloads
  AS (SELECT Count(*) AS download_count
      FROM    app_downloads),
signups_users
  AS (SELECT Count(user_id) AS signup_users
      FROM    signups),
ride_users
  AS (SELECT Count(DISTINCT user_id) AS ride_users
      FROM    ride_requests),
accepted_rides
  AS (SELECT Count(DISTINCT user_id) AS accepted_rides
      FROM    ride_requests
      WHERE   accept_ts IS NOT NULL),
completed_rides
  AS (SELECT Count(DISTINCT user_id) AS completed_rides
      FROM    ride_requests
      WHERE   cancel_ts IS NULL),
paid_rides
  AS (SELECT Count(DISTINCT r.user_id) AS paid_rides
      FROM    ride_requests r
              left join transactions t
                  ON r.ride_id = t.ride_id
      WHERE   r.accept_ts IS NOT NULL
              AND r.cancel_ts IS NULL),
users_reviews
  AS (SELECT Count(DISTINCT user_id) AS reviews
      FROM    reviews),

```

```

funnel_stages
AS (SELECT 0                                AS funnel_step,
        'downloads'                        AS funnel_name,
        downloads.download_count AS value
FROM   downloads
UNION
SELECT 1                                AS funnel_step,
        'signups'                        AS funnel_name,
        signups_users.signup_users AS value
FROM   signups_users
UNION
SELECT 2                                AS funnel_step,
        'ride_requested'                AS funnel_name,
        ride_users.ride_users AS value
FROM   ride_users
UNION
SELECT 3                                AS funnel_step,
        'ride_accepted'                AS funnel_name,
        accepted_rides.accepted_rides AS value
FROM   accepted_rides
UNION
SELECT 4                                AS funnel_step,
        'ride_completed'                AS funnel_name,
        completed_rides.completed_rides AS value
FROM   completed_rides
UNION
SELECT 5                                AS funnel_step,
        'ride_paid'                    AS funnel_name,
        paid_rides.paid_rides AS value
FROM   paid_rides
UNION
SELECT 6                                AS funnel_step,
        'reviews'                      AS funnel_name,
        users_reviews.reviews AS value
FROM   users_reviews
ORDER BY funnel_step)

```

```

SELECT *,
       value :: FLOAT / First_value(value)
                                over (
                                    ORDER BY funnel_step) AS convers
ion_rate,
       1 - value :: FLOAT / First_value(value)
                                over (
                                    ORDER BY funnel_step) AS drop_of
f
FROM   funnel_stages

```

Query for surge-pricing strategy, showing the highest demand hours during the day

```

WITH ride_users
     AS (SELECT Date_part('hour', r.request_ts) AS hour_of_reque
st,
           Count(r.ride_id) AS ride_count
        FROM   ride_requests r
        JOIN   signups s
              ON r.user_id = s.user_id
        GROUP BY Date_part('hour', r.request_ts))
SELECT hour_of_request,
       ride_users.ride_count AS ride_count
FROM   ride_users
ORDER BY hour_of_request

```

Cancelled ride distribution during the day

```

SELECT Date_part('hour', r.request_ts) AS hour_of_request,
       Count(r.ride_id) AS ride_count
FROM   ride_requests r
WHERE  r.cancel_ts IS NOT NULL
       AND r.accept_ts IS NOT NULL

```



```
GROUP BY Date_part('hour', r.request_ts)
ORDER BY 1
```

Average Metrocar's drivers rating

```
SELECT Avg(avg_rate)
FROM (SELECT Avg(rating) AS avg_rate,
              driver_id
       FROM reviews
       GROUP BY 2) sub
```

Ride count distributed by rating

```
SELECT rating,
       Count(ride_id)
FROM reviews
GROUP BY 1
```

Average customer waiting time

```
SELECT Avg(r.cancel_ts - r.request_ts) AS waiting_time
FROM ride_requests r
WHERE r.cancel_ts IS NOT NULL
      AND r.accept_ts IS NOT NULL
```

Tableau visualizations

Funnel analysis visualization + user/ride counter by platform:

https://public.tableau.com/views/FunnelchartUPD/FUNNELBYUSERRIDECOUNT?:language=en-US&publish=yes&:display_count=n&:origin=viz_share_link

Ride requests distribution throughout the day:

https://public.tableau.com/views/Funnelsurgepricing/Riderequestdistributionduringtheday?:language=en-US&:display_count=n&:origin=viz_share_link

Cancellations distribution throughout the day:

https://public.tableau.com/views/Cancelledridedistributionbyhour/Cancelledridedistributionbyhour?:language=en-US&:display_count=n&:origin=viz_share_link

Rides count distribution by rating:

https://public.tableau.com/views/Ridescountdistributionbyrating/Ridescountdistributionbyrating?:language=en-US&:display_count=n&:origin=viz_share_link