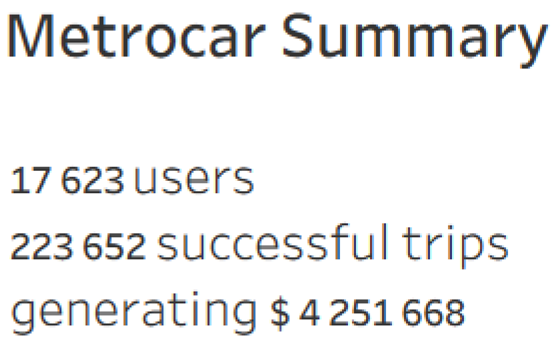
Metrocar_report

📊 Metrocar Dashboard — Key Insights Report

1. KPI Summary (Overall Performance)



Metrocar Summary

17 623 users

223 652 successful trips

generating $ 4 251 668

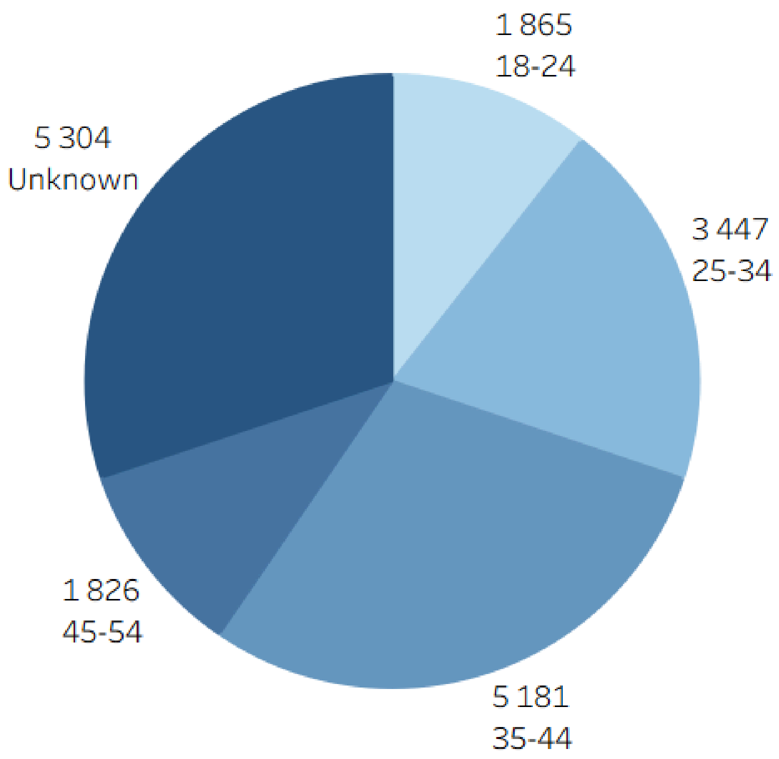
Key metrics:

- **6,233 users** completed at least one ride
- **223,652 successful rides** were completed
- **$4.25M total revenue** generated from approved transactions

Insights:

- The ratio of completed rides to completed users indicates **strong repeat usage**, suggesting that active users tend to take multiple rides.
- Revenue closely follows ride completion volume, confirming that **monetization primarily depends on ride completion rather than user acquisition alone**.

2. User Age Analysis



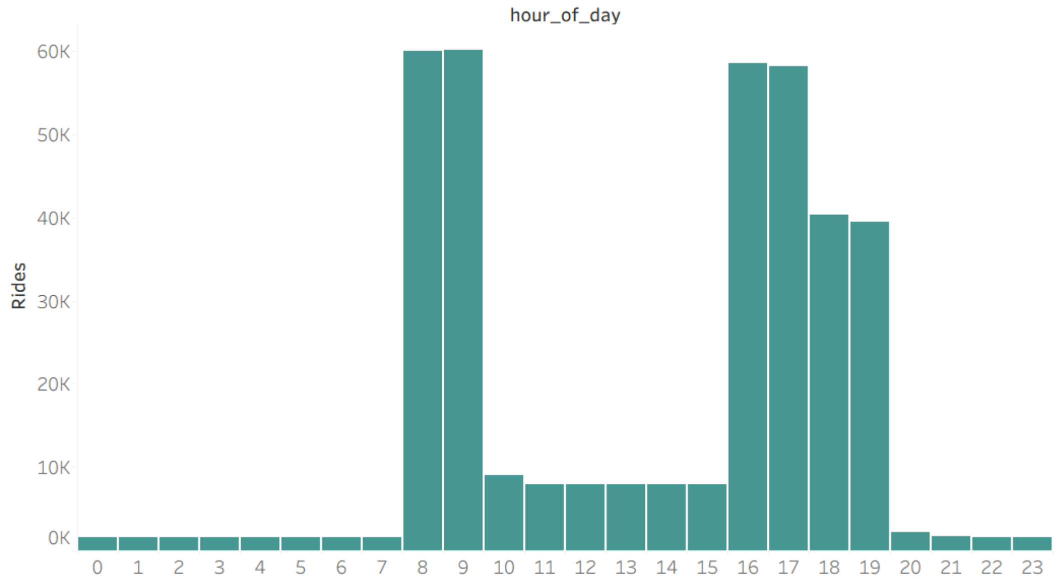
Distribution highlights:

- The **35–44 age group** is the largest segment of active users.
- A significant portion of users is labeled as **"Unknown"**, which may indicate missing demographic data during signup.
- Younger users (18–24) form the smallest identifiable segment.

Insights:

- Core demand comes from **working-age adults**, which aligns with typical urban commuting patterns.
- Improving age data collection could unlock **more accurate targeting and segmentation** for marketing and pricing strategies.

3. Hourly Rides Distribution



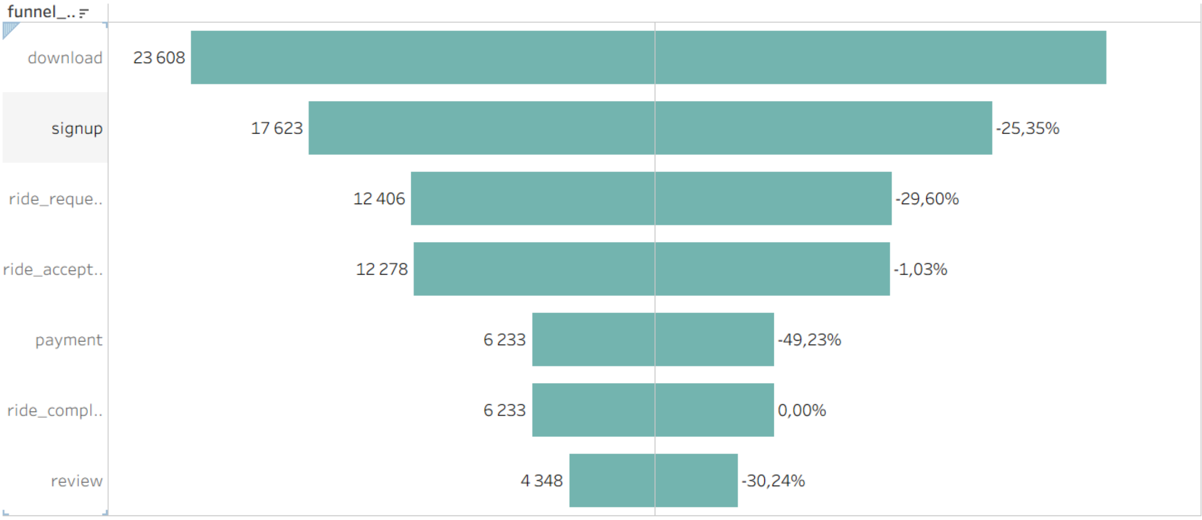
Observed patterns:

- Two clear peaks:
  - **Morning peak:** 8–9 AM
  - **Evening peak:** 4–7 PM
- Very low activity during night hours (0–6 AM).

Insights:

- Ride demand strongly correlates with **commuting hours**, confirming Metrocar's role as a commuter-focused service.
- Off-peak hours represent potential opportunities for **pricing incentives or promotions** to balance demand.

4. Main Users Funnel



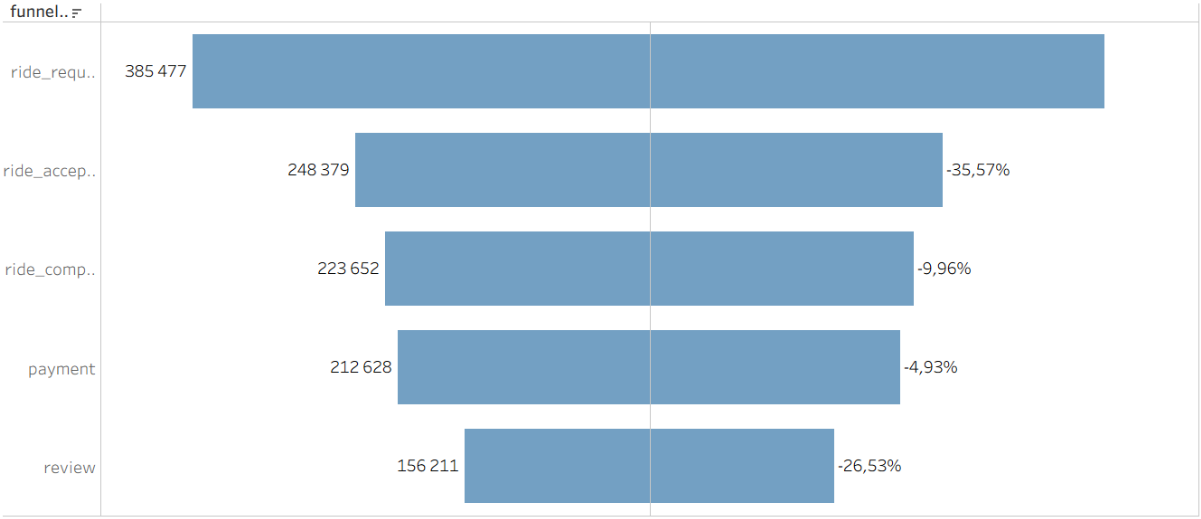
Funnel behavior:

- Significant drop from **download** → **signup**.
- Another major drop from **ride accepted** → **payment**.
- Conversion stabilizes after payment (payment → ride completed).

Insights:

- The **largest friction points** occur early (signup) and at monetization (payment).
- Optimizing onboarding and payment UX could significantly improve **overall conversion rates**.

5. Rides Funnel



Ride lifecycle insights:

- From **ride requested to ride accepted**, ~36% of requests drop.
- Smaller drop between **ride accepted → ride completed**.
- Final drop at **review stage**, which is expected and less critical.

Insights:

- Driver availability and matching efficiency likely drive early funnel losses.
- Improving acceptance rates could directly increase completed rides and revenue.

📌 Strategic Takeaways

- Metrocar is **highly commute-driven**, with predictable demand spikes.
- Retention among active users is strong; **acquisition and conversion** are the main optimization levers.
- Funnel losses indicate clear **UX and operational improvement opportunities** at signup, acceptance, and payment stages.

🧠 SQL Queries Used for the Dashboard

> Below are the final, clean versions of all queries aligned with the visualizations in the dashboard.

---

1. KPI Summary

Completed users & completed rides

```
SELECT
    SUM(number_of_users)
        FILTER (WHERE funnel_name = 'signup')
        AS registered_users,
    SUM(number_of_users)
        FILTER (WHERE funnel_name = 'ride_completed')
        AS completed_users,
    SUM(number_of_rides)
        FILTER (WHERE funnel_name = 'ride_completed')
        AS completed_rides
 FROM funnel_analysis;
```

Total revenue

```
SELECT
    SUM(purchase_amount_usd) AS total_revenue
FROM transactions
WHERE charge_status = 'Approved';
```

2. User Age Analysis (Signup users by age)

```
SELECT
    age_range,
    funnel_name,
    SUM(number_of_users) AS users
FROM funnel_analysis
GROUP BY age_range, funnel_name
ORDER BY age_range, funnel_name;
```

3. Hourly Rides Distribution

```
SELECT
    EXTRACT(HOUR FROM request_ts) AS hour_of_day,
    COUNT(*) AS total_requests
FROM ride_requests
GROUP BY 1
ORDER BY 1;
```

4. Main Users Funnel

```
SELECT
    funnel_step,
    funnel_name,
    SUM(number_of_users) AS users
FROM funnel_analysis
GROUP BY funnel_step, funnel_name
ORDER BY funnel_step;
```

5. Rides Funnel

```
SELECT
    funnel_step,
    funnel_name,
    SUM(number_of_rides) AS rides
FROM funnel_analysis
GROUP BY funnel_step, funnel_name
ORDER BY funnel_step;
```