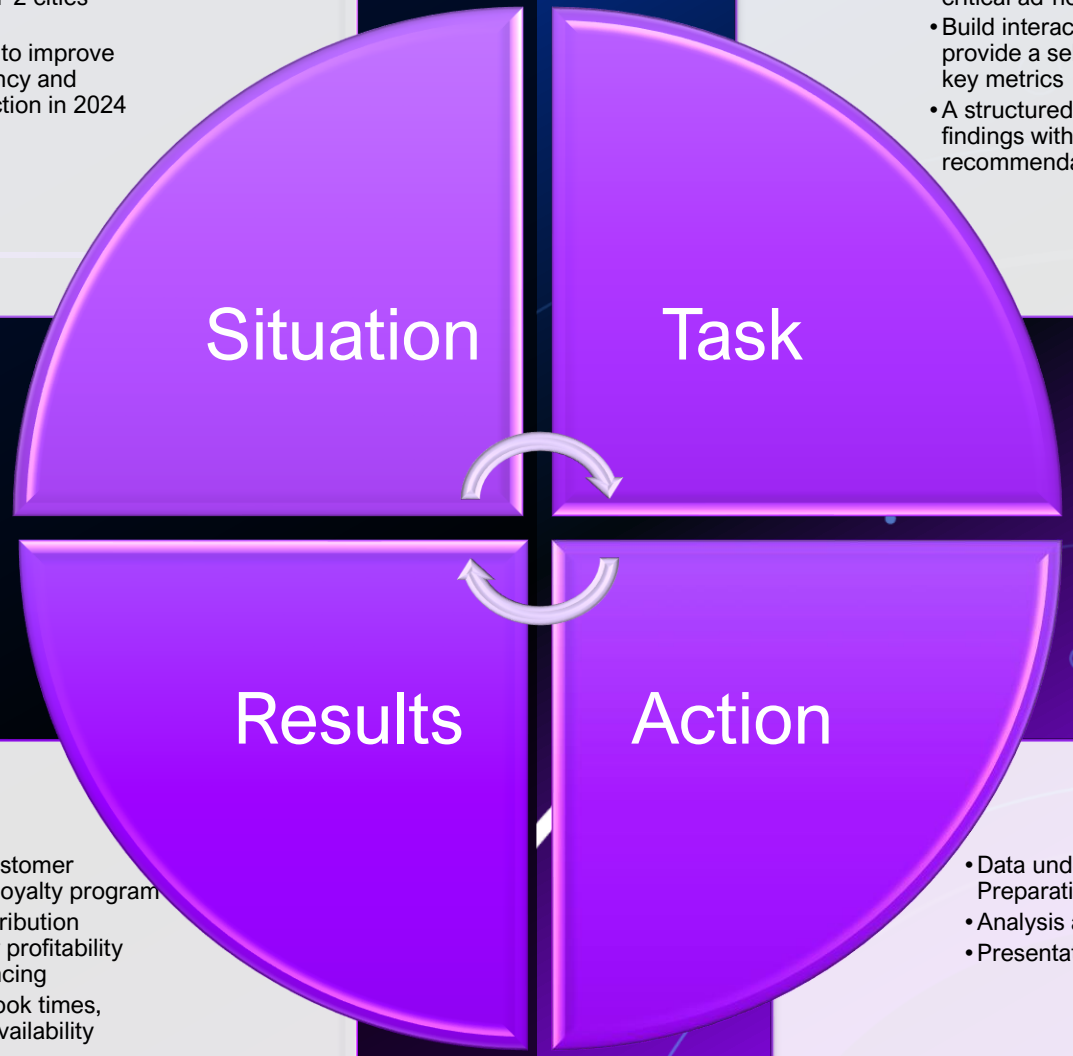
OPTIMIZING TRANSPORTATION EFFICIENCY : BUSINESS INSIGHT FROM UBER DATA ANALYTICS

S.T.A.R FRAMEWORK



- Uber is a cab service company operating in 10 tier-2 cities across india
- The company aim to improve operational efficiency and passenger satisfaction in 2024

- SQL based reports for business critical ad-hoc request
- Build interactive dashboard that provide a self explanatory view of key metrics
- A structured presentation of findings with actionable recommendations

- Boost repeated customer retention through loyalty program
- Optimized trip distribution strategy for higher profitability and demand balancing
- Identifying peak book times, improving driver availability

- Data understanding and Preparation
- Analysis and Insight Generations
- Presentation and Story-telling

UBER MODEL

Source: www.proschoolonline.com

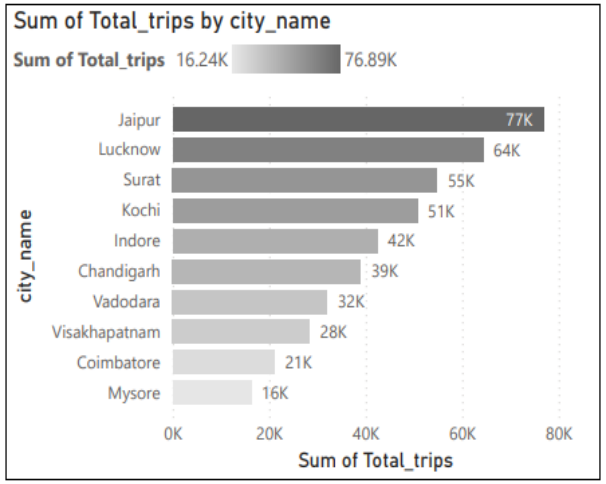
Business Model of Uber

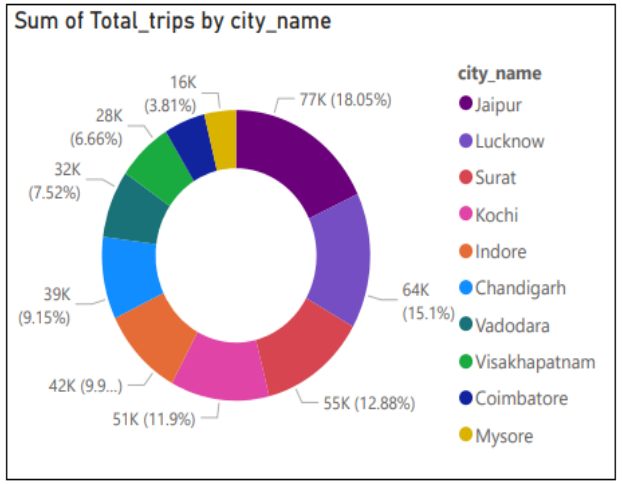
- Uber has its footprints in 600 cities spread across 65 countries.
- On average, 15 million trips are completed every day.
- Has completed over 5 billion trips globally.
- The valuation of uber stood at $000 billion.
- An average uber driver earns $000 per month.
- There are around 3 million uber drivers globally.

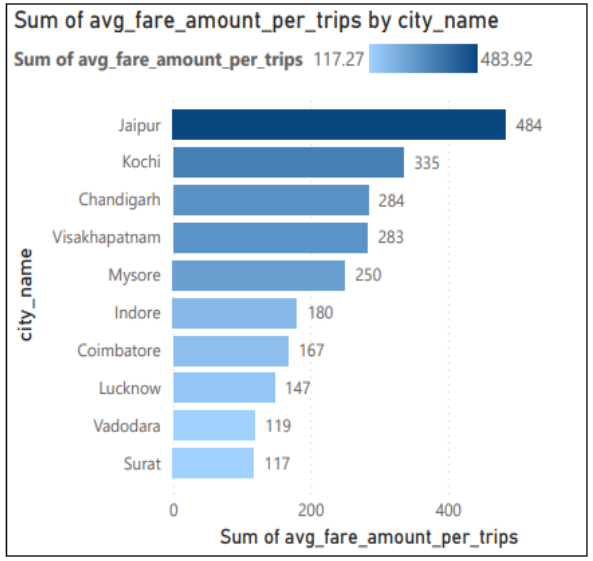
BUSINESS REQUETS 1

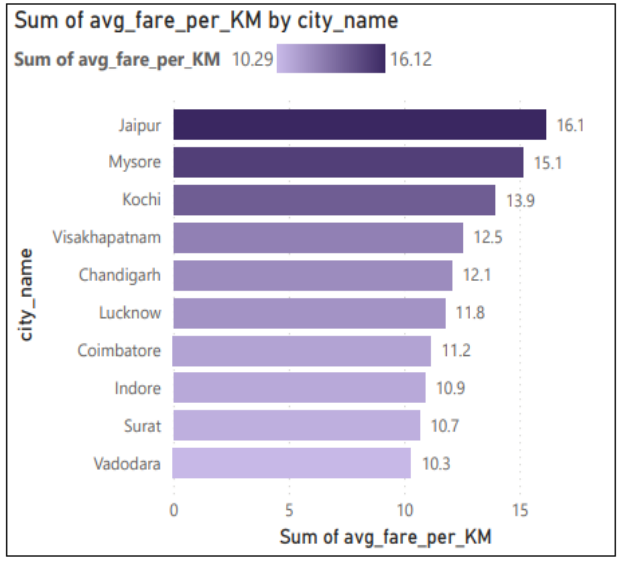
City-Level Fare and Trip Summary: -

Generate a report that display the total trips, average fare per KM, average fare per trips, and the percentage contribution of each city's trips to the overall trips. This report will help in assessing trip volume, pricing efficiency, and each city's contribution to the overall trip count.










```sql
WITH cities AS (
SELECT
    fact_trips.city_id,
    dim_city.city_name,
        ROUND(AVG(
            fact_trips.distance_travelled_km
            ),0) AS avg_distance_travelled,
        SUM(
            fact_trips.distance_travelled_km
            ) AS Total_distance_travelled,
        ROUND(AVG(
            fact_trips.fare_amount
            ),2) AS avg_fare_amount_per_trips,
        SUM(
            fact_trips.fare_amount
            ) AS Total_fare_amount,
    COUNT(*) AS Total_trips
FROM
    fact_trips
INNER JOIN
    dim_city
ON
    fact_trips.city_id = dim_city.city_id
GROUP BY
    city_id,
    city_name
)
SELECT
    cities.city_name,
    cities.Total_trips,
    cities.avg_fare_amount_per_trips,
        ROUND((
            cities.Total_fare_amount / cities.Total_distance_travelled
            ),2) AS avg_fare_per_KM,
        ROUND((
            cities.Total_trips / (SELECT
            SUM(Total_trips)
            FROM cities) * 100
            ),1) AS Percentage
FROM
    cities
GROUP BY
    cities.city_id,
    cities.city_name ;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

city_name	Total_trips	avg_fare_amount_per_trips	avg_fare_per_KM	Percentage
Visakhapatnam	28366	282.67	12.53	6.7
Chandigarh	38981	283.69	12.06	9.2
Surat	54843	117.27	10.66	12.9
Vadodara	32026	118.57	10.29	7.5
Mysore	16238	249.71	15.14	3.8

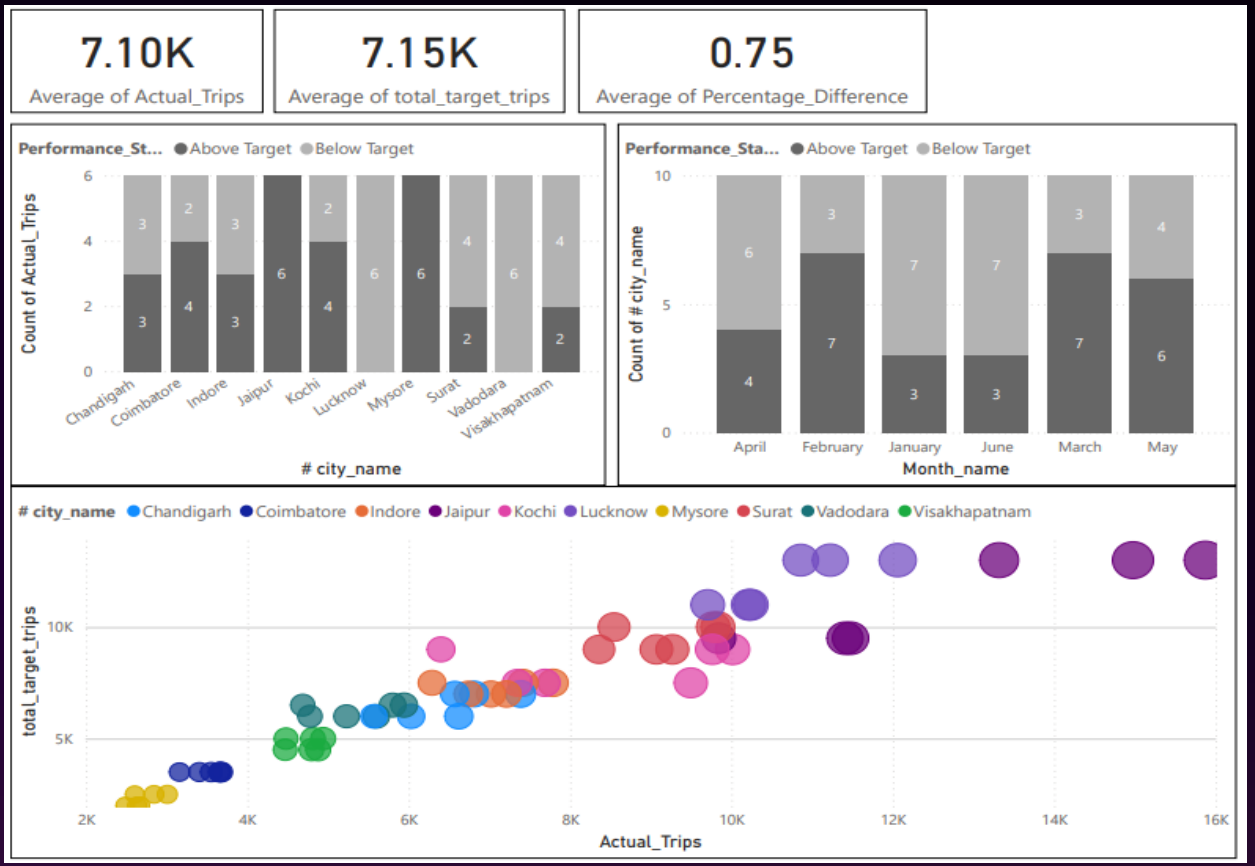
BUSINESS REQUETS 2

Monthly City-Level Trips Target Performance Report :-

Generate a report that evaluates the target performance of trips at the monthly and city level. For each city and month, compare the actual trips with the target trips and categorized the performance.

- If actual trips are greater than target trips “Above Target”
- If actual trips are less than equal to target trips “Below Target”

Additionally, calculated the % difference between actual and target trips to quantify the performance gap.




```sql
WITH actual AS (
    SELECT
        MONTHNAME(fact_trips.date) AS Month_name,
        fact_trips.city_id,
        dim_city.city_name,
        COUNT(fact_trips.trip_id) AS Actual_Trips
    FROM fact_trips
    INNER JOIN
        dim_city
    ON
        fact_trips.city_id = dim_city.city_id
    GROUP BY
        Month_name,
        city_id,
        city_name
)
    SELECT
        actual.city_name,
        actual.Month_name,
        actual.Actual_Trips,
        monthly_target_trips.total_target_trips,
            (CASE
                WHEN actual.Actual_Trips > monthly_target_trips.total_target_trips
                THEN 'Above Target'
                WHEN actual.Actual_Trips <= monthly_target_trips.total_target_trips
                THEN 'Below Target'
            END
            ) AS Performance_Status,
        (
                (actual.Actual_Trips - monthly_target_trips.total_target_trips) / monthly_target_trips.total_target_trips * 100
        ) AS Percentage_Difference
    FROM
        actual
    INNER JOIN
        monthly_target_trips
    ON
        actual.Month_name = MONTHNAME(monthly_target_trips.month)
    AND
        actual.city_id = monthly_target_trips.city_id ;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

city_name	Month_name	Actual_Trips	total_target_trips	Performance_Status	Percentage_Difference
Indore	March	7019	7000	Above Target	0.2714
Mysore	May	3007	2500	Above Target	20.2800
Lucknow	April	10212	11000	Below Target	-7.1636
Vadodara	February	5228	6000	Below Target	-12.8667
Kochi	May	10014	9000	Above Target	11.2667

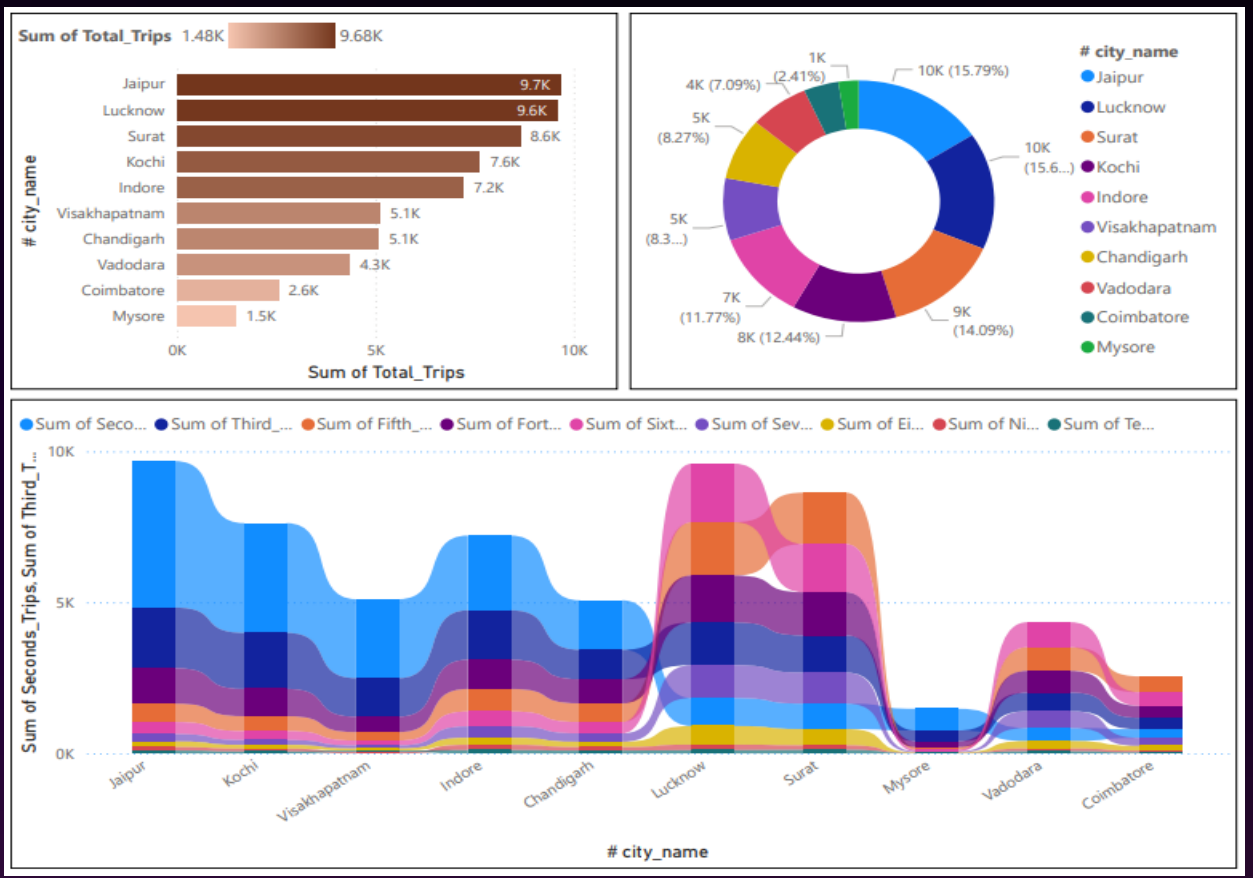
BUSINESS REQUETS 3

City-Level Request Passenger Trips Frequency Report:-

Generate a report that show that percentage distribution of repeat passenger by the number of trips they have taken each city. Calculating the percentage of repeat passenger who took 2 Trips, 3 Trips and so on up to 10 Trips.

Each columns should represent a trips count category, displaying the percentage of repeat passenger who fall into that category put of total repeat passenger for that city.

This report will help identify cities with repeat trips frequency, which can indicates strong customer loyalty or frequently usage patterns.




```sql
WITH cities AS (
    SELECT
        dim_city.city_name,
            SUM( CASE WHEN dim_repeat_trip_distribution.trip_count = '2-Trips' THEN dim_repeat_trip_distribution.repeat_passenger_count
        END ) AS 'Seconds_Trips',
            SUM( CASE WHEN dim_repeat_trip_distribution.trip_count = '3-Trips' THEN dim_repeat_trip_distribution.repeat_passenger_count
        END ) AS 'Third_Trips',
            SUM( CASE WHEN dim_repeat_trip_distribution.trip_count = '4-Trips' THEN dim_repeat_trip_distribution.repeat_passenger_count
        END ) AS 'Forth_Trips',
            SUM( CASE WHEN dim_repeat_trip_distribution.trip_count = '5-Trips' THEN dim_repeat_trip_distribution.repeat_passenger_count
        END ) AS 'Fifth_Trips',
            SUM( CASE WHEN dim_repeat_trip_distribution.trip_count = '6-Trips' THEN dim_repeat_trip_distribution.repeat_passenger_count
        END ) AS 'Sixth_Trips',
            SUM( CASE WHEN dim_repeat_trip_distribution.trip_count = '7-Trips' THEN dim_repeat_trip_distribution.repeat_passenger_count
        END ) AS 'Seventh_Trips',
            SUM( CASE WHEN dim_repeat_trip_distribution.trip_count = '8-Trips' THEN dim_repeat_trip_distribution.repeat_passenger_count
        END ) AS 'Eighth_Trips',
            SUM( CASE WHEN dim_repeat_trip_distribution.trip_count = '9-Trips' THEN dim_repeat_trip_distribution.repeat_passenger_count
        END ) AS 'Ninth_Trips',
            SUM( CASE WHEN dim_repeat_trip_distribution.trip_count = '10-Trips' THEN dim_repeat_trip_distribution.repeat_passenger_count
        END ) AS 'Tenth_Trips'
    FROM dim_repeat_trip_distribution
    INNER JOIN dim_city
    ON dim_repeat_trip_distribution.city_id = dim_city.city_id
    GROUP BY
        city_name
)
SELECT *,
    (
        cities.Seconds_Trips + cities.Third_Trips + cities.Forth_Trips +
        cities.Fifth_Trips + cities.Sixth_Trips + cities.Seventh_Trips +
        cities.Eighth_Trips + cities.Ninth_Trips + cities.Tenth_Trips
    ) AS 'Total_Trips',
        ROUND((
            (
        cities.Seconds_Trips + cities.Third_Trips + cities.Forth_Trips +
        cities.Fifth_Trips + cities.Sixth_Trips + cities.Seventh_Trips +
        cities.Eighth_Trips + cities.Ninth_Trips + cities.Tenth_Trips
    ) * 100 / ( SELECT
                    SUM(
                    cities.Seconds_Trips + cities.Third_Trips + cities.Forth_Trips +
                    cities.Fifth_Trips + cities.Sixth_Trips + cities.Seventh_Trips +
                    cities.Eighth_Trips + cities.Ninth_Trips + cities.Tenth_Trips
                    )
                FROM cities
    )
    ),2) AS Percentage
FROM cities ;
```

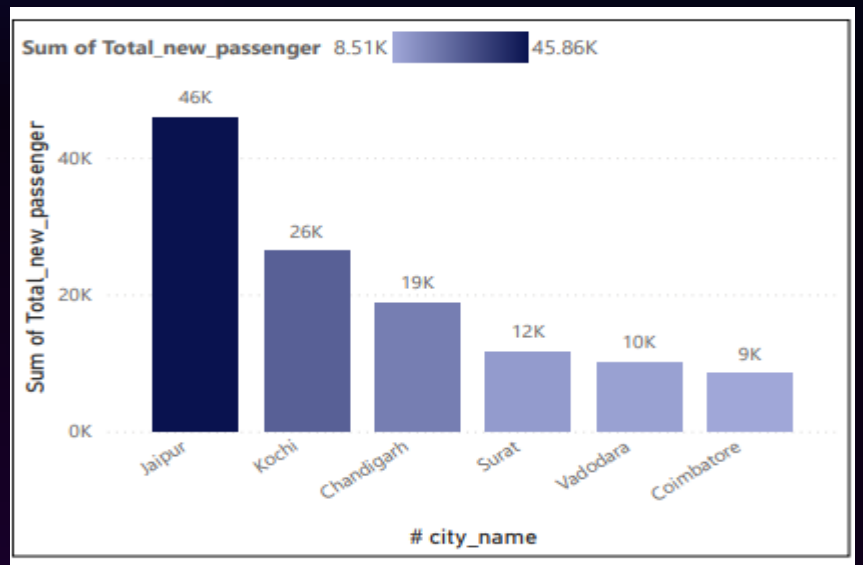
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

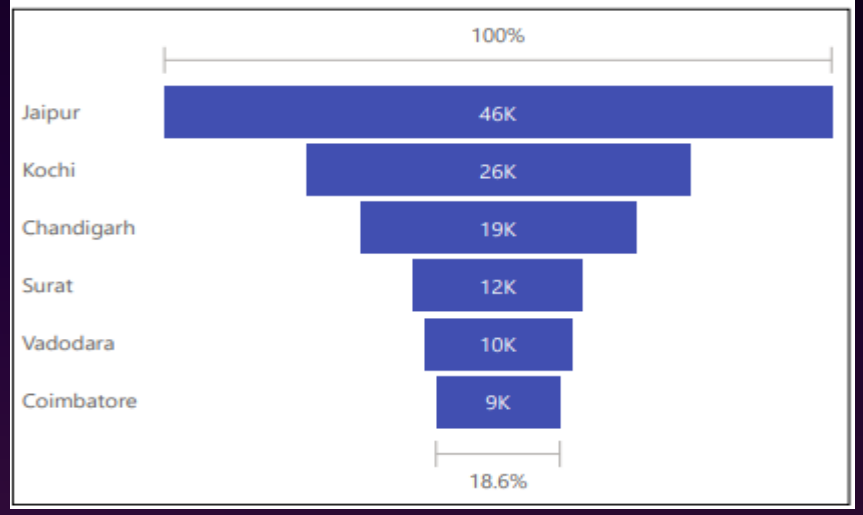
city_name	Seconds_Trips	Third_Trips	Forth_Trips	Fifth_Trips	Sixth_Trips	Seventh_Trips	Eighth_Trips	Ninth_Trips	Tenth_Trips	Total_Trips	Percentage
Visakhapatnam	2618	1275	510	278	163	101	71	45	47	5108	8.33
Chandigarh	1638	976	798	619	376	278	176	118	91	5070	8.27
Surat	843	1232	1430	1706	1594	1027	539	150	117	8638	14.09
Vadodara	429	616	718	785	829	559	251	89	70	4346	7.09
Mysore	720	361	188	86	60	26	21	8	7	1477	2.41

BUSINESS REQUETS 4

Identify Cities With Highest and Lowest Total new Passenger :-

Generate a report that calculates the total new passenger for each city and ranks them based on this value. Identify the Top 3 cities with the highest number of new passenger as well as the bottom 3 cities with the lowest number of new passenger, categorized them as the “Top 3” or “Bottom 3” accordingly.






```sql
WITH cities AS (
    SELECT
        dim_city.city_name,
        COUNT(
            CASE
                WHEN fact_trips.passenger_type = 'new'
                THEN 1
            END
        ) AS Total_new_passenger,
        RANK()
            OVER(
                ORDER BY
                COUNT(
            CASE
                WHEN fact_trips.passenger_type = 'new'
                THEN 1
            END
        ) DESC
        ) AS Top_Rank_City,
        RANK()
            OVER(
                ORDER BY
                COUNT(
            CASE
                WHEN fact_trips.passenger_type = 'new'
                THEN 1
            END
        ) ASC
        ) AS Bottom_Rank_City
    FROM fact_trips
    INNER JOIN dim_city
    ON fact_trips.city_id = dim_city.city_id
    GROUP BY
        city_name
)
SELECT
    cities.city_name,
    cities.Total_new_passenger,
    CASE
        WHEN cities.Top_Rank_City < 4
        THEN 'Top_3_City'
        WHEN cities.Bottom_Rank_city < 4
        THEN 'Bottom_3_City'
    END AS Total
FROM cities
WHERE
    Top_Rank_City < 4
    OR
        Bottom_Rank_City < 4 ;
```

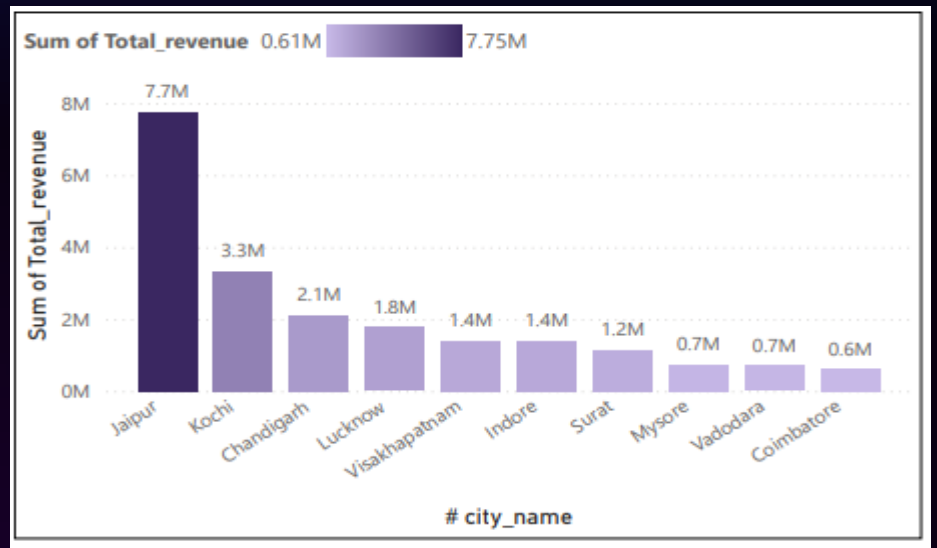
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

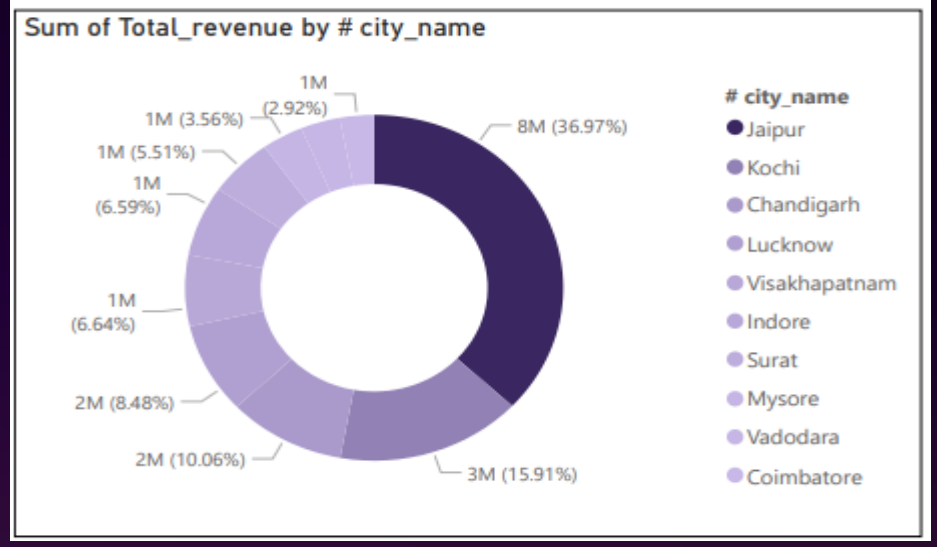
city_name	Total_new_passenger	Total
Coimbatore	8514	Bottom_3_City
Vadodara	10127	Bottom_3_City
Surat	11626	Bottom_3_City
Chandigarh	18908	Top_3_City
Kochi	26416	Top_3_City

BUSINESS REQUETS 5

Identify Month with Highest Revenue For Each City :-

Generate a report that identifies the month with the highest revenue for each city. For each city display the month name, the revenue amount for that month, and the percentage contribution of that month's revenue to the city's total revenue.






```sql
WITH cities AS (
    SELECT
        dim_city.city_name,
        MONTHNAME(
            fact_trips.date
        ) AS Month_name,
        SUM(
            fact_trips.fare_amount
        ) AS Total_revenue,
            RANK()
                OVER(
                    PARTITION BY city_name
                ORDER BY SUM(
                    fact_trips.fare_amount
        ) DESC
        ) AS Ranking
    FROM
        fact_trips
    INNER JOIN
        dim_city
    ON
        fact_trips.city_id = dim_city.city_id
    GROUP BY
        city_name,
        Month_name
)
    SELECT
        cities.city_name,
        cities.Month_name,
        cities.Total_revenue,
    ROUND((
        cities.Total_revenue / (
            SELECT
                SUM(Total_revenue)
            FROM cities
            ) * 100
    ),2) AS Percentage_contribution
FROM cities ;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

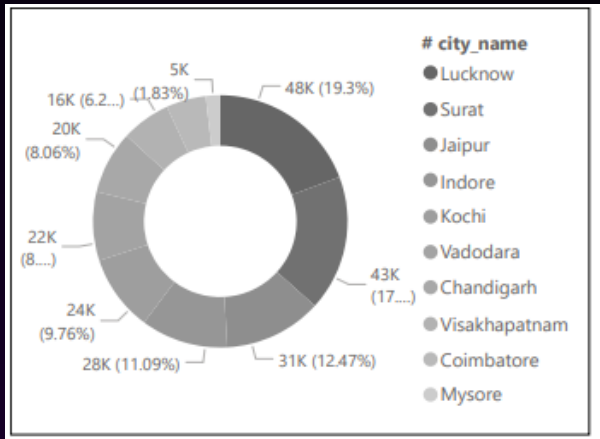
city_name	Month_name	Total_revenue	Percentage_contribution
Chandigarh	February	2108290	1.95
Chandigarh	January	1934293	1.79
Chandigarh	March	1863793	1.72
Chandigarh	May	1850777	1.71
Chandigarh	June	1717223	1.59

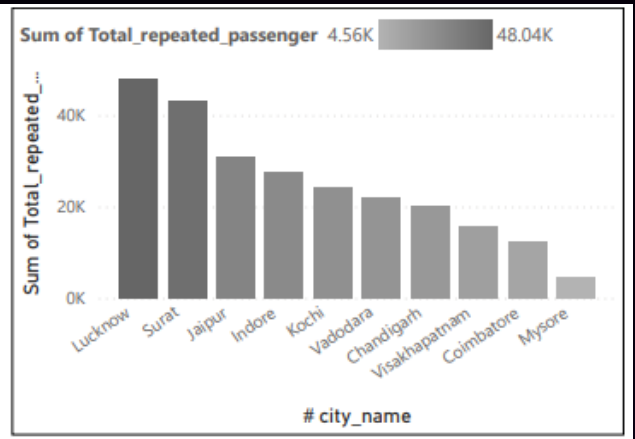
BUSINESS REQUETS 6

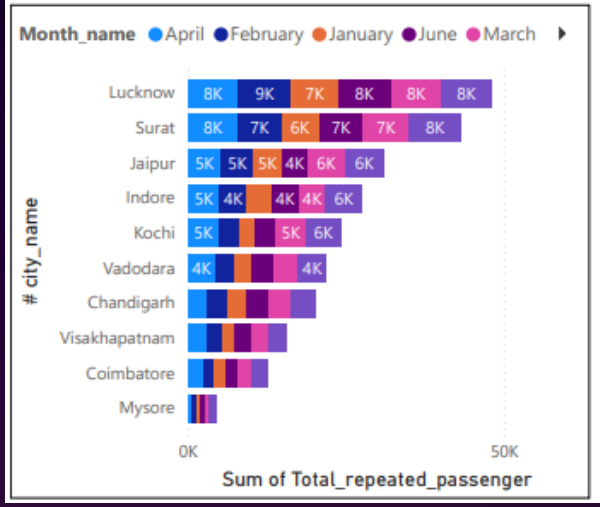
Repeat Passenger Rate Analysis :-

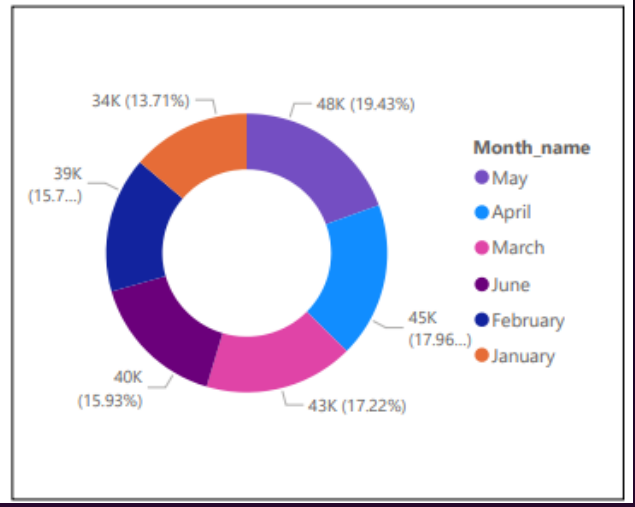
Generate a report that calculates two metrics

- Monthly repeat passenger rate :- Calculate the repeat passenger rate for each city and month by comparing the number of repeat passenger to the total passenger.
- City-wide repeat passenger rate :- Calculate the overall repeat passenger rate for each city, considering all passenger across month.









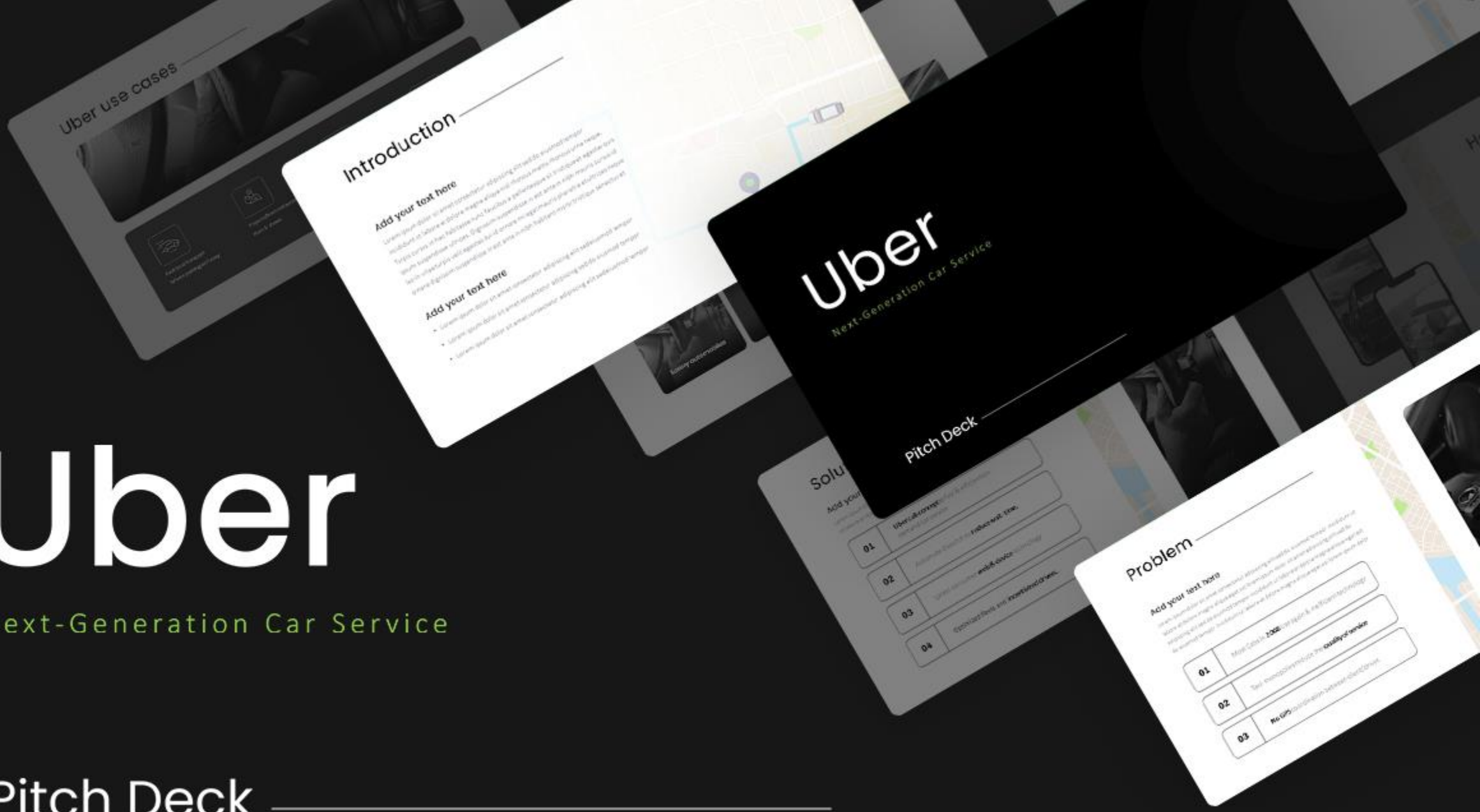

```sql
WITH cities AS (
SELECT
    dim_city.city_name,
    MONTHNAME(
        fact_trips.date
    ) AS Month_name,
    COUNT(*) AS Total_passenger,
    COUNT(
        CASE
            WHEN fact_trips.passenger_type = 'repeated'
            THEN 1
        END
    ) AS Total_repeated_passenger
FROM fact_trips
INNER JOIN dim_city
ON fact_trips.city_id = dim_city.city_id
GROUP BY
    city_name,
    Month_name
)
SELECT *,
ROUND((
    cities.Total_repeated_passenger / cities.Total_passenger * 100
),2) AS Monthly_repeated_passenger_percentage
FROM cities ;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

city_name	Month_name	Total_passenger	Total_repeated_passenger	Monthly_repeated_passenger_percentage
Chandigarh	April	5566	3070	55.16
Chandigarh	February	7387	3283	44.44
Chandigarh	January	6810	2890	42.44
Chandigarh	June	6029	3599	59.69
Chandigarh	March	6569	3341	50.86

KEY TAKEAWAYS

- Jaipur is the Top perform in tips, revenue and new passenger acquisition
- Lucknow and Surat have high repeat passenger rates, indicating good customer retention
- April and May are peak months, while January and June need improvement
- There is a close match between actual and target trips, which is a positive operational efficiency indicators



Uber

Next-Generation Car Service

Pitch Deck