12/
01/
25

PRESENTED BY
RANA BASAK

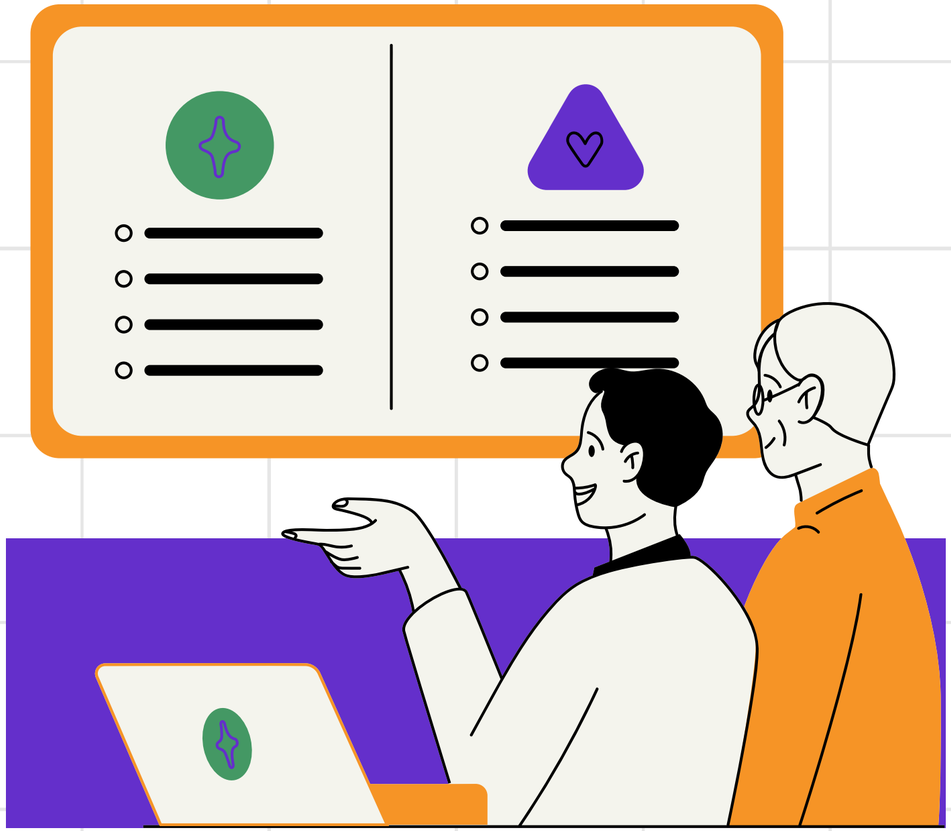
GOOD CAB

PRESENTATION

PROJECT SUMMARY

Goodcabs, a rapidly growing cab service company in India, has made a strong impact by focusing on tier-2 cities. With a commitment to supporting local drivers and ensuring high-quality service for passengers, Goodcabs operates across ten tier-2 cities. For 2024, the company aims to drive growth and improve passenger satisfaction, with key performance targets in mind.

The management team, led by Bruce Haryali (Chief of Operations), seeks insights into Goodcabs' performance across critical metrics, including trip volume, passenger satisfaction, repeat passenger rate, trip distribution, and the balance between new and repeat passengers



RANA BASAK

TASKS

Understand the Data: We begin by thoroughly reviewing the metadata and analyzing the datasets. This is a critical first step to ensure a strong foundation for our analysis.

Start with Key Questions: We then refer to the primary_and_secondary_questions.pdf document as per guide. We can use any preferred tool to analyze the data and answer these questions.

Create a Dashboard: We design a dashboard showcasing our metrics and analysis.

Address Business Queries: We consult the ad-hoc-requests.pdf document for important business questions and these will require generating reports using SQL-based queries.

RANA BASAK

BUSINESS-SPECIFIC SQL REPORTS

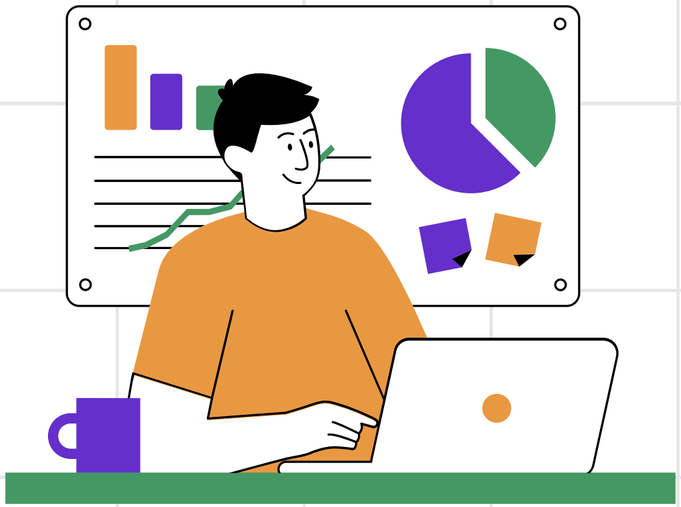
Business-Specific SQL Reports: Review the ad-hoc-requests for specific business questions that require SQL-based reports. Created and present these reports using SQL queries.

RANA BASAK

BUSINESS REQUEST - 1

Displays the total trips, average fare per km, average fare per trip, and the percentage contribution of each city's trips to the overall trips.

```sql
SELECT
    CITY_NAME,
    COUNT(TRIP_ID) AS Total_Trips,
    SUM(FARE_AMOUNT) / SUM(DISTANCE_TRAVELLED_KM) AS Avg_fare_per_km,
    SUM(FARE_AMOUNT) / COUNT(TRIP_ID) AS Avg_fare_per_trip,
    round(100*(COUNT(TRIP_ID)*1.0) / (
        SELECT COUNT(TRIP_ID) FROM FACT_TRIPS
    ),2) AS percentage_contribution_to_total_trips
FROM
    FACT_TRIPS
    JOIN DIM_CITY ON FACT_TRIPS.CITY_ID = DIM_CITY.CITY_ID
GROUP BY
    CITY_NAME
```



RANA BASAK

BUSINESS REQUEST - 1

Displays the total trips, average fare per km, average fare per trip, and the percentage contribution of each city's trips to the overall trips.

CITY_NAME	Total_Trips	Avg_fare_per_km	Avg_fare_per_trip	Total_trip_perc
Visakhapatnam	28366	12.5332	282.6723	6.66
Chandigarh	38981	12.0622	283.6870	9.15
Surat	54843	10.6638	117.2729	12.88
Vadodara	32026	10.2942	118.5662	7.52
Mysore	16238	15.1366	249.7072	3.81
Kochi	50702	13.9305	335.2451	11.90
Indore	42456	10.8977	179.8386	9.97
Jaipur	76888	16.1182	483.9181	18.05
Coimbatore	21104	11.1476	166.9822	4.96
Lucknow	64299	11.7622	147.1804	15.10

10 rows in set (6.19 sec)

RANA BASAK

BUSINESS REQUEST - 2

Evaluates the target performance for trips at the monthly and city level. It compares actual total trips with target trips and categorizes the performance.

```sql
WITH cte1 AS(
  SELECT city_name, month_name, COUNT(trip_id) AS actual_trips
  FROM dim_city c
  JOIN fact_trips t ON c.city_id=t.city_id
  JOIN dim_date d ON d.date=t.date
  GROUP BY 1,2)
  SELECT c.city_name, c.month_name, actual_trips, target_trips,
         CASE WHEN actual_trips > target_trips THEN "Above Target"
              WHEN actual_trips <= target_trips THEN "Below Target"
        END AS performance_status,
          ROUND((actual_trips-target_trips)/target_trips*100,2) AS "difference_%"
FROM cte1 c
JOIN (SELECT  city_name, MONTHNAME(month) AS month_name, total_target_trips AS target_trips
     FROM targets_db.monthly_target_trips t
     JOIN dim_city c ON t.city_id=c.city_id) x ON c.city_name=x.city_name AND c.month_name=x.month_name
ORDER BY 6 DESC;

```

RANA BASAK

BUSINESS REQUEST - 2

Evaluates the target performance for trips at the monthly and city level. It compares actual total trips with target trips and categorizes the performance.

city_name	month_name	actual_trips	target_trips	performance_status	difference_%
Mysore	February	2668	2000	Above Target	33.40
Mysore	March	2633	2000	Above Target	31.65
Kochi	March	9495	7500	Above Target	26.60
Mysore	January	2485	2000	Above Target	24.25
Jaipur	February	15872	13000	Above Target	22.09
Jaipur	May	11475	9500	Above Target	20.79
Mysore	May	3007	2500	Above Target	20.28
Jaipur	April	11406	9500	Above Target	20.06
Jaipur	January	14976	13000	Above Target	15.20
Mysore	June	2842	2500	Above Target	13.68
Kochi	May	10014	9000	Above Target	11.27
Chandigarh	May	6620	6000	Above Target	10.33
Kochi	April	9762	9000	Above Target	8.47
Visakhapatnam	March	4877	4500	Above Target	8.38
Visakhapatnam	February	4793	4500	Above Target	6.51
Chandigarh	February	7387	7000	Above Target	5.53
Coimbatore	March	3680	3500	Above Target	5.14
Coimbatore	April	3661	3500	Above Target	4.60
Coimbatore	January	3651	3500	Above Target	4.31
Mysore	April	2603	2500	Above Target	4.12
Indore	May	7787	7500	Above Target	3.83
Jaipur	June	9842	9500	Above Target	3.60
Indore	February	7210	7000	Above Target	3.00
Surat	March	9267	9000	Above Target	2.97
Kochi	February	7688	7500	Above Target	2.51
Jaipur	March	13317	13000	Above Target	2.44
Coimbatore	May	3550	3500	Above Target	1.43
Surat	February	9069	9000	Above Target	0.77
Chandigarh	June	6029	6000	Above Target	0.48
Indore	March	7019	7000	Above Target	0.27
Indore	March	7019	7000	Above Target	0.27
Visakhapatnam	January	4468	4500	Below Target	-0.71
Indore	April	7415	7500	Below Target	-1.13
Visakhapatnam	April	4938	5000	Below Target	-1.24
Surat	April	9831	10000	Below Target	-1.69
Kochi	January	7344	7500	Below Target	-2.08
Surat	May	9774	10000	Below Target	-2.26
Chandigarh	January	6810	7000	Below Target	-2.71
Coimbatore	February	3404	3500	Below Target	-2.74
Visakhapatnam	May	4812	5000	Below Target	-3.76
Indore	January	6737	7000	Below Target	-3.76
Chandigarh	March	6569	7000	Below Target	-6.16
Vadodara	March	5598	6000	Below Target	-6.70
Lucknow	June	10240	11000	Below Target	-6.91
Surat	January	8358	9000	Below Target	-7.13
Lucknow	April	10212	11000	Below Target	-7.16
Lucknow	February	12060	13000	Below Target	-7.23
Chandigarh	April	5566	6000	Below Target	-7.23
Vadodara	April	5941	6500	Below Target	-8.60
Coimbatore	June	3158	3500	Below Target	-9.77
Visakhapatnam	June	4478	5000	Below Target	-10.44
Vadodara	May	5799	6500	Below Target	-10.78
Lucknow	May	9705	11000	Below Target	-11.77
Vadodara	February	5228	6000	Below Target	-12.87
Lucknow	March	11224	13000	Below Target	-13.66
Surat	June	8544	10000	Below Target	-14.56
Indore	June	6288	7500	Below Target	-16.16
Lucknow	January	10858	13000	Below Target	-16.48
Vadodara	January	4775	6000	Below Target	-20.42
Vadodara	June	4685	6500	Below Target	-27.92
Kochi	June	6399	9000	Below Target	-28.90

60 rows in set (6.38 sec)

RANA BASAK

BUSINESS REQUEST - 3

Shows the percentage distribution of repeat passengers by the number of trips they have taken in each city.

```sql
SELECT
    CITY_name,
    ROUND(SUM(  CASE
            WHEN TRIP_COUNT = '2-Trips' THEN REPEAT_PASSENGER_COUNT
            ELSE 0
                END *1.00)*100/SUM(REPEAT_PASSENGER_COUNT),2) AS TRIP_2,
    ROUND(SUM(  CASE
            WHEN TRIP_COUNT = '3-Trips' THEN REPEAT_PASSENGER_COUNT
            ELSE 0
                END *1.00)*100/SUM(REPEAT_PASSENGER_COUNT),2) AS TRIP_3,
    ROUND(SUM(  CASE
            WHEN TRIP_COUNT = '4-Trips' THEN REPEAT_PASSENGER_COUNT
            ELSE 0
                END *1.00)*100/SUM(REPEAT_PASSENGER_COUNT),2) AS TRIP_4,
    ROUND(SUM( CASE
            WHEN TRIP_COUNT = '5-Trips' THEN REPEAT_PASSENGER_COUNT
            ELSE 0
                END *1.00 )*100/SUM(REPEAT_PASSENGER_COUNT),2) AS TRIP_5,
    ROUND(SUM(  CASE
            WHEN TRIP_COUNT = '6-Trips' THEN REPEAT_PASSENGER_COUNT
            ELSE 0
                END *1.00)*100/SUM(REPEAT_PASSENGER_COUNT),2) AS TRIP_6,
    ROUND(SUM(  CASE
            WHEN TRIP_COUNT = '7-Trips' THEN REPEAT_PASSENGER_COUNT
            ELSE 0
                END *1.00)*100/SUM(REPEAT_PASSENGER_COUNT),2) AS TRIP_7,
    ROUND(SUM( CASE
            WHEN TRIP_COUNT = '8-Trips' THEN REPEAT_PASSENGER_COUNT
            ELSE 0
                END *1.00)*100/SUM(REPEAT_PASSENGER_COUNT),2) AS TRIP_8,
    ROUND(SUM( CASE
            WHEN TRIP_COUNT = '9-Trips' THEN REPEAT_PASSENGER_COUNT
            ELSE 0
                END *1.00 )*100/SUM(REPEAT_PASSENGER_COUNT),2) AS TRIP_9,
    ROUND(SUM( CASE
            WHEN TRIP_COUNT = '10-Trips' THEN REPEAT_PASSENGER_COUNT
            ELSE 0
                END *1.00 )*100/SUM(REPEAT_PASSENGER_COUNT),2) AS TRIP_10
FROM DIM_REPEAT_TRIP_DISTRIBUTION
JOIN dim_city ON dim_city.city_id = DIM_REPEAT_TRIP_DISTRIBUTION.city_id
GROUP BY CITY_name;
```

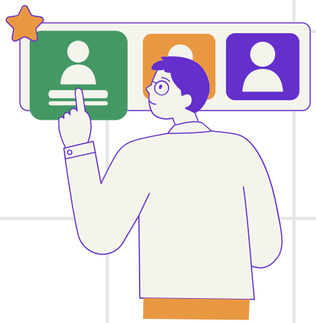
RANA BASAK

BUSINESS REQUEST - 3

Shows the percentage distribution of repeat passengers by the number of trips they have taken in each city.

CITY_name	TRIP_2	TRIP_3	TRIP_4	TRIP_5	TRIP_6	TRIP_7	TRIP_8	TRIP_9	TRIP_10
Visakhapatnam	51.25	24.96	9.98	5.44	3.19	1.98	1.39	0.88	0.92
Chandigarh	32.31	19.25	15.74	12.21	7.42	5.48	3.47	2.33	1.79
Surat	9.76	14.26	16.55	19.75	18.45	11.89	6.24	1.74	1.35
Vadodara	9.87	14.17	16.52	18.06	19.08	12.86	5.78	2.05	1.61
Mysore	48.75	24.44	12.73	5.82	4.06	1.76	1.42	0.54	0.47
Kochi	47.67	24.35	11.81	6.48	3.91	2.11	1.65	1.21	0.81
Indore	34.34	22.69	13.40	10.34	6.85	5.24	3.26	2.38	1.51
Jaipur	50.14	20.73	12.12	6.29	4.13	2.52	1.90	1.20	0.97
Coimbatore	11.21	14.82	15.56	20.62	17.64	10.47	6.15	2.31	1.22
Lucknow	9.66	14.77	16.20	18.42	20.18	11.33	6.43	1.91	1.10

10 rows in set (0.01 sec)



RANA BASAK

BUSINESS REQUEST - 4

This report calculates the total new passengers for each city and ranks them. It identifies the top 3 cities with the highest number of new passengers as well as the bottom 3 cities with the lowest.

```sql
WITH city_sum AS (
    SELECT d.city_name, SUM(f.new_passengers) AS total_passengers
    FROM fact_passenger_summary f
    JOIN dim_city d ON f.city_id = d.city_id
    GROUP BY d.city_name
),
ranked_state AS (
    SELECT city_name, total_passengers,
    ROW_NUMBER() OVER(ORDER BY total_passengers DESC) AS descrank,
    ROW_NUMBER() OVER(ORDER BY total_passengers ASC) AS ascrank
    FROM city_sum
)
SELECT city_name, total_passengers,
  CASE WHEN descrank <= 3 THEN 'Top 3'
       WHEN ascrank <= 3 THEN 'Bottom 3'
  END AS category
FROM ranked_state
WHERE descrank <= 3 OR ascrank <= 3
ORDER BY total_passengers DESC;
```

RANA BASAK

BUSINESS REQUEST - 4

This report calculates the total new passengers for each city and ranks them. It identifies the top 3 cities with the highest number of new passengers as well as the bottom 3 cities with the lowest.

city_name	total_passengers	category
Jaipur	45856	Top 3
Kochi	26416	Top 3
Chandigarh	18908	Top 3
Surat	11626	Bottom 3
Vadodara	10127	Bottom 3
Coimbatore	8514	Bottom 3

6 rows in set (0.04 sec)

RANA BASAK

BUSINESS REQUEST - 5

This report identifies the month with the highest revenue for each city, displaying the month name, revenue amount, and the percentage contribution of that month's revenue to the city's total revenue.

```sql
WITH first_month AS (
    SELECT city_name, monthname(date) AS Month_name, SUM(fare_amount) AS total_fare_amount,
      ROW_NUMBER() OVER (PARTITION BY city_name ORDER BY SUM(fare_amount) DESC) AS ranking
  FROM fact_trips
  JOIN
      dim_city ON dim_city.city_id = fact_trips.city_id
  GROUP BY city_name, monthname(date)
  ORDER BY total_fare_amount DESC),
states AS (
    SELECT city_name, SUM(total_fare_amount) AS cur
    FROM first_month
    GROUP BY city_name )
SELECT states.city_name, total_fare_amount, Month_name, ROUND(100 * total_fare_amount / cur, 2) as percentage
FROM first_month
JOIN states ON first_month.city_name = states.city_name
WHERE ranking = 1;
```

RANA BASAK

BUSINESS REQUEST - 5

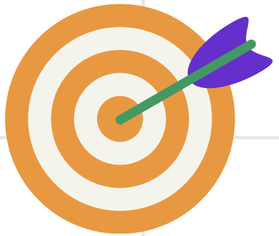
This report identifies the month with the highest revenue for each city, displaying the month name, revenue amount, and the percentage contribution of that month's revenue to the city's total revenue.

city_name	total_fare_amount	Month_name	percentage
Jaipur	7747202	February	20.82
Kochi	3333746	May	19.61
Chandigarh	2108290	February	19.07
Lucknow	1777269	February	18.78
Visakhapatnam	1390682	April	17.34
Indore	1380996	May	18.09
Surat	1154909	April	17.96
Mysore	745170	May	18.38
Vadodara	706250	April	18.60
Coimbatore	612431	April	17.38

10 rows in set (1.61 sec)

RANA BASAK

BUSINESS REQUEST - 6

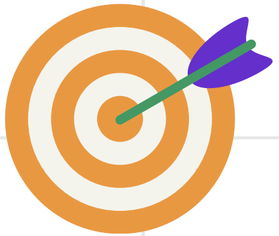


This report calculates the repeat passenger rate both monthly and city-wide. It provides insights into repeat passenger behavior and trends across months.

```sql
WITH cte AS (
  SELECT c.city_name, monthname(month) AS month, SUM(total_passengers) AS total_passengers,
         SUM(repeat_passengers) AS repeat_passengers,
         ROUND((SUM(repeat_passengers * 1.00) / SUM(total_passengers)) * 100, 2) AS monthly_rep_pass_rate
  FROM fact_passenger_summary ps
  JOIN dim_city c ON ps.city_id = c.city_id
  GROUP BY 1, 2)
SELECT c.city_name, month, c.total_passengers, c.repeat_passengers,
       monthly_rep_pass_rate, city_rep_pass_rate
FROM cte c
JOIN (SELECT c.city_name, SUM(total_passengers) AS total_passengers,
             SUM(repeat_passengers) AS repeat_passengers,
             ROUND((SUM(1.00 * repeat_passengers) / SUM(total_passengers)) * 100, 2) AS city_rep_pass_rate
      FROM fact_passenger_summary ps
      JOIN dim_city c ON ps.city_id = c.city_id
      GROUP BY 1) x ON c.city_name = x.city_name
ORDER BY c.city_name;
```

RANA BASAK

BUSINESS REQUEST - 6



This report calculates the repeat passenger rate both monthly and city-wide. It provides insights into repeat passenger behavior and trends across months.

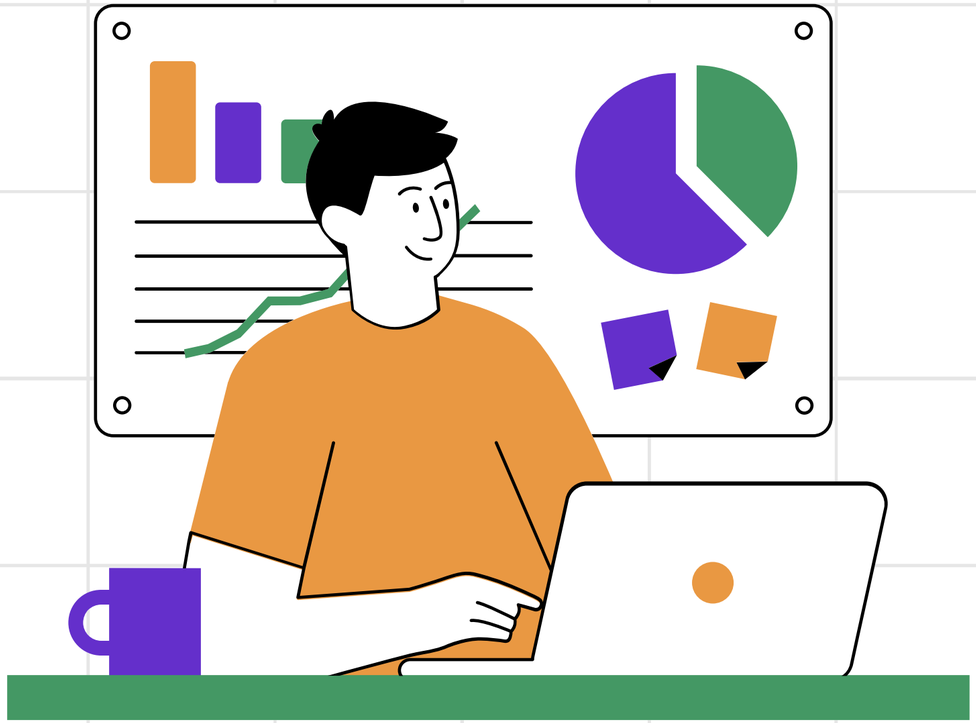
city_name	month	total_passengers	repeat_passengers	monthly_rep_pass_rate	city_rep_pass_rate
Chandigarh	February	4957	853	17.21	21.14
Chandigarh	January	4640	720	15.52	21.14
Chandigarh	March	4100	872	21.27	21.14
Chandigarh	April	3285	789	24.02	21.14
Chandigarh	May	3699	969	26.20	21.14
Chandigarh	June	3297	867	26.30	21.14
Coimbatore	June	1628	402	24.69	23.05
Coimbatore	April	1722	480	27.87	23.05
Coimbatore	March	1965	427	21.73	23.05
Coimbatore	February	1993	346	17.36	23.05
Coimbatore	January	2214	392	17.71	23.05
Coimbatore	May	1543	504	32.66	23.05
Indore	June	3152	1131	35.88	32.68
Indore	May	3591	1563	43.53	32.68
Indore	April	3646	1295	35.52	32.68
Indore	March	3833	1091	28.46	32.68
Indore	January	3876	1033	26.65	32.68
Indore	February	3981	1103	27.71	32.68
Jaipur	January	11845	1422	12.01	17.43
Jaipur	April	7856	1736	22.10	17.43
Jaipur	June	6956	1181	16.98	17.43
Jaipur	May	7174	1842	25.68	17.43
Jaipur	March	9257	1840	19.88	17.43
Jaipur	February	12450	1661	13.34	17.43
Kochi	January	5660	795	14.05	22.40
Kochi	February	5372	1005	18.71	22.40
Kochi	March	6213	1348	21.70	22.40
Kochi	April	6515	1576	24.19	22.40
Kochi	May	6222	1853	29.78	22.40
Kochi	June	4060	1049	25.84	22.40
Lucknow	January	4896	1431	29.23	37.12
Lucknow	February	5188	1659	31.98	37.12
Lucknow	March	4781	1622	33.93	37.12
Lucknow	April	3807	1496	39.30	37.12
Lucknow	May	3487	1662	47.66	37.12
Lucknow	June	3698	1727	46.70	37.12
Mysore	June	2203	329	14.93	11.23
Mysore	May	2270	349	15.37	11.23
Mysore	April	2072	236	11.39	11.23
Mysore	March	2194	208	9.48	11.23
Mysore	February	2290	183	7.99	11.23
Mysore	January	2129	172	8.08	11.23
Surat	January	3616	1184	32.74	42.63
Surat	June	3030	1490	49.17	42.63
Surat	May	3217	1606	49.92	42.63
Surat	April	3394	1551	45.70	42.63
Surat	March	3440	1494	43.43	42.63
Surat	February	3567	1313	36.81	42.63
Vadodara	June	1807	703	38.90	30.03
Vadodara	May	2256	868	38.48	30.03
Vadodara	April	2499	862	34.49	30.03
Vadodara	March	2522	759	30.10	30.03
Vadodara	February	2756	610	22.13	30.03
Vadodara	January	2633	544	20.66	30.03
Visakhapatnam	February	3170	790	24.92	28.61
Visakhapatnam	March	3093	923	29.84	28.61
Visakhapatnam	April	2837	992	34.97	28.61
Visakhapatnam	May	2890	951	32.91	28.61
Visakhapatnam	June	2702	802	29.68	28.61
Visakhapatnam	January	3163	650	20.55	28.61

60 rows in set (0.04 sec)

RANA BASAK

DASHBOARD

Primary & Secondary Questions



RANA BASAK

GOOD CAB

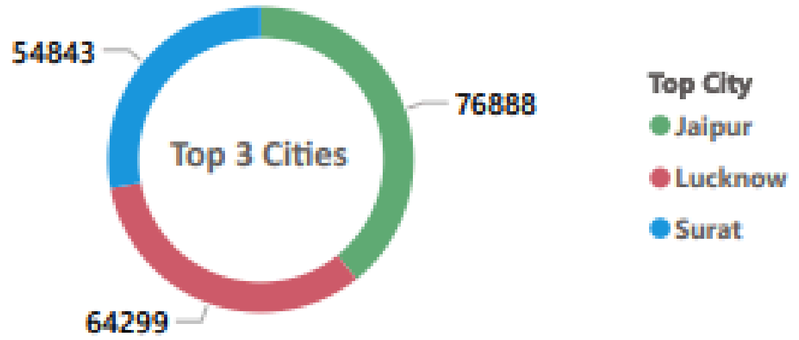
Month: All

Trips | Passenger | Comparison

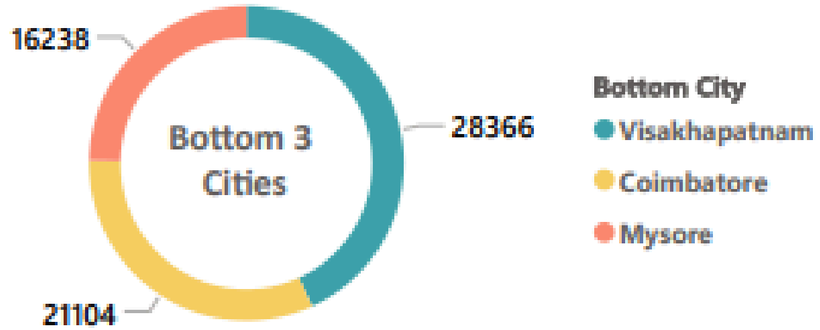
Analyzed By
RANA BASAK

Total Passenger	Total Trips	Total New Passenger	Average Distance	Average fare
238309	425903	176998	19.13	254.02

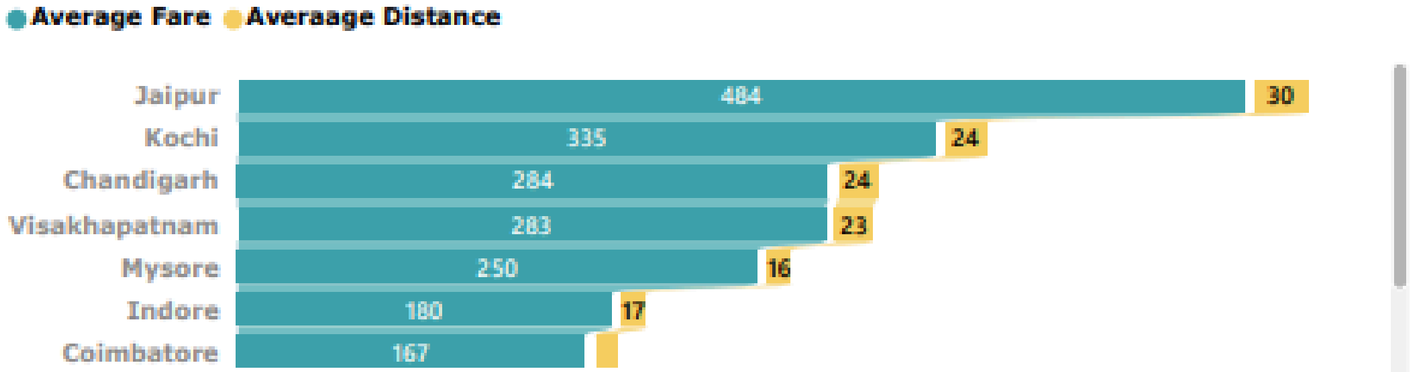
Maximum Trips



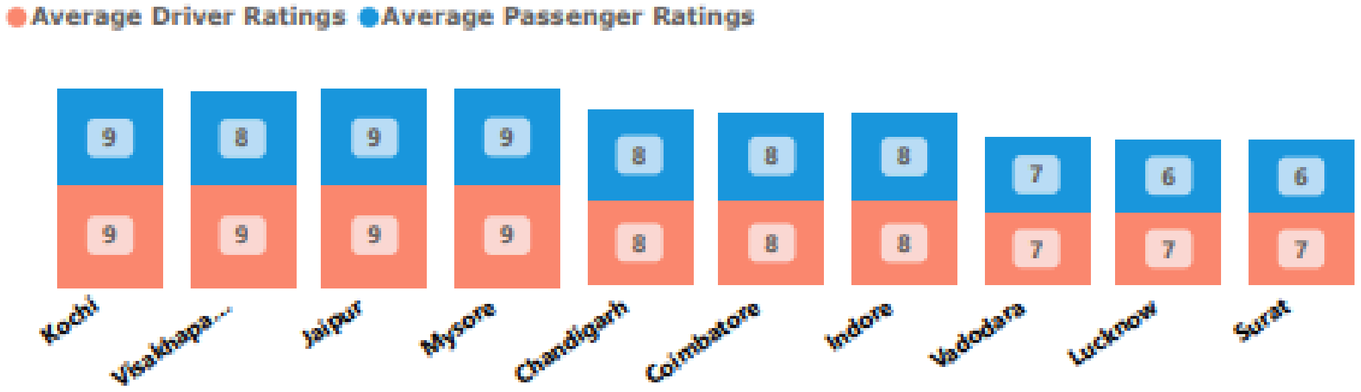
Minimum Trips



Average Distance (Km) & Average Fare For Each City



Driver Rating V.S Passenger Rating



GOOD CAB

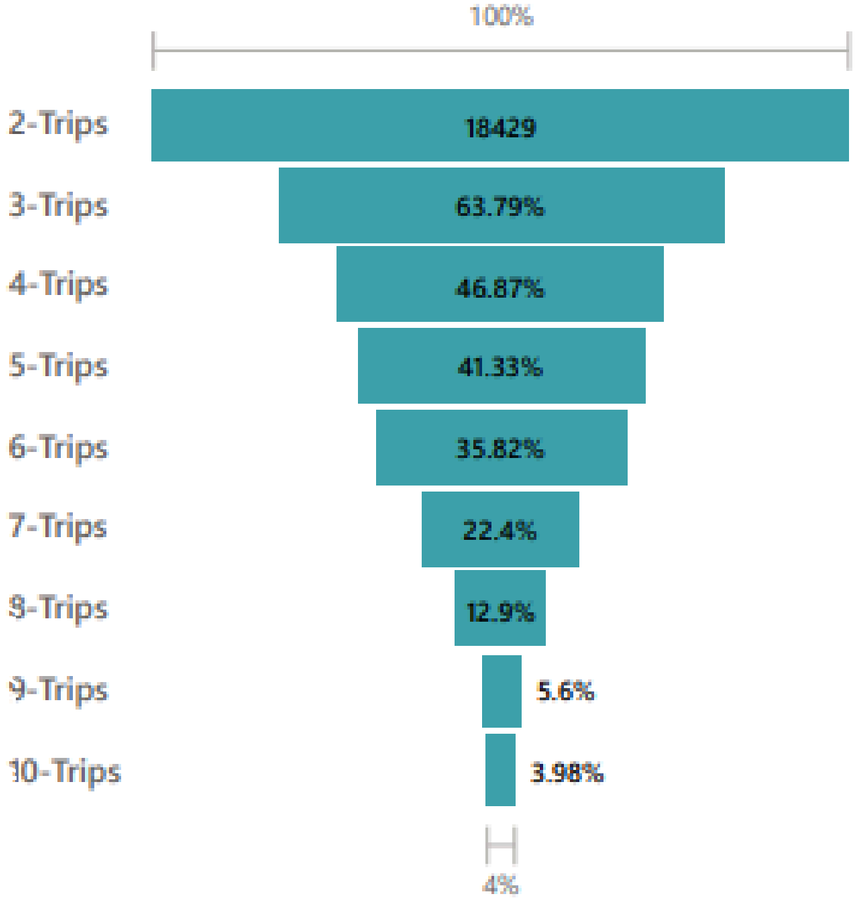
City: All

Trips | Passenger | Comparison

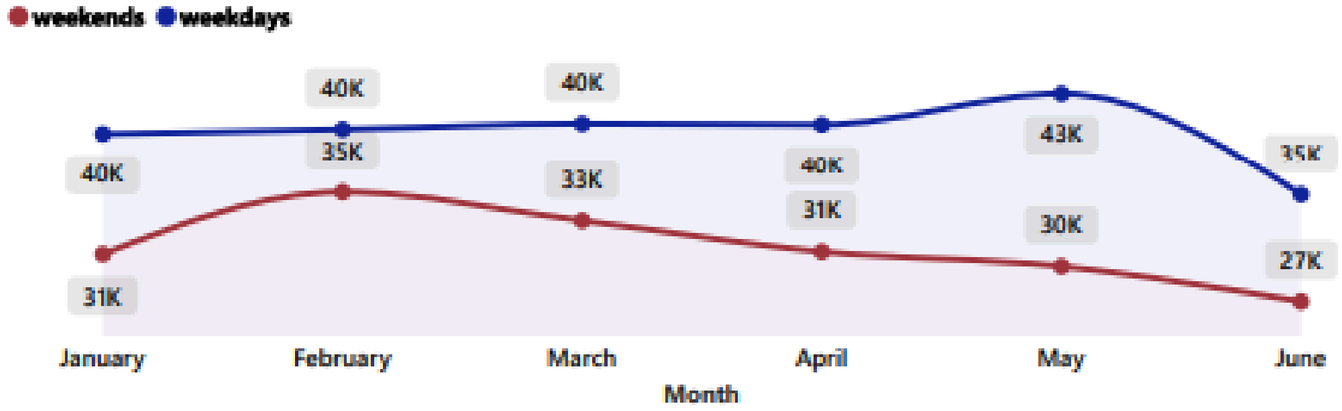
Analyze By
RANA BASAK

Total Passenger	Total Trips	Total New Passenger	Weekdays Trip	Weekends Trip
238309	425903	176998	238338	187565

Repeat Passengers



Weekdays V.S Weekends Trips



Highest & Least Trip Month

City	Top Month	Max Trips	Bottom Month	Minimum Trips
Chandigarh	February	7387	April	5566
Coimbatore	March	3680	June	3158
Indore	May	7787	June	6288
Jaipur	February	15872	June	9842
Kochi	May	10014	June	6399
Lucknow	February	12060	May	9705

GOOD CAB

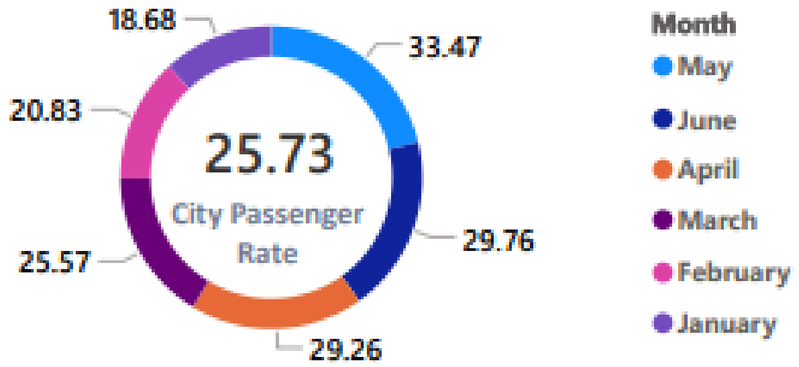
City: All

Trips | Passenger | Comparison

Analyzed By
RANA BASAK

Total Passenger	Total Trips	Total New Passenger	Total Repeat Passenger	Average Rating
238309	425903	176998	61311	7.66

Passenger Rate By Month



City Target Ratings

City	Passenger Rating	Target Ratting	% Diffrenece
Jaipur	8.58	8.25	4.05
Mysore	8.70	8.50	2.37
Kochi	8.52	8.50	0.19
Chandigarh	7.98	8.00	-0.29
Visakhapatnam	8.43	8.50	-0.79

Monthly Target Trips

Month name	Total Trips	Target Trips	% Diffenerece
February	75379	72500	3.97
May	72543	70500	2.90
March	73679	72500	1.63
April	71335	70500	1.18
January	70462	72500	-2.81
June	62505	70500	-11.34

Monthly Target Passenger

Month	New Passenger	Target Passenger	% Difference
April	26620	25000	6.48
January	36329	36700	-1.01
February	36201	36700	-1.36
May	24182	25000	-3.27
June	22852	25000	-8.59
March	30814	36700	-16.04

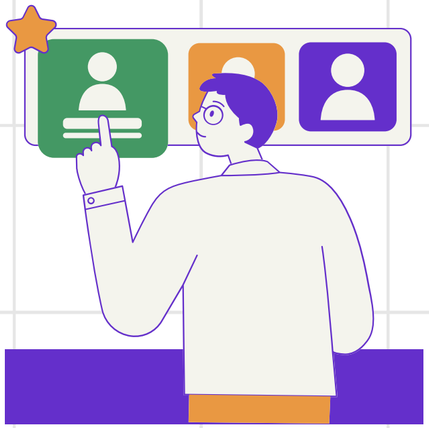
INSIGHTS

- **Trip Volume**: Jaipur, Lucknow, and Surat had the highest total trip volumes, while Vishakhapatnam, Coimbatore, and Mysore had the lowest.
- **Fare Analysis:** Jaipur had the highest average fare per trip, suggesting better pricing efficiency, while Surat had the lowest average fare, indicating potential pricing adjustments.
- **Passenger Ratings:** Mysore achieved the highest passenger ratings, with new passengers scoring 8.98 and repeat passengers 7.98.
- **Driver Performance:** Kochi recorded the highest driver ratings at 8.99, highlighting superior driver performance.
- **Demand Patterns:** Demand peaked in February and March, while June experienced the lowest demand due to seasonal variations. Weekends consistently showed higher trip volumes across all cities.
- **Customer Retention:** Surat and Lucknow had the highest repeat passenger rates, while Jaipur and Mysore had the lowest, suggesting the need for better customer retention strategies.

RANA BASAK

RECOMONDATIONS

- We can implement localized marketing campaigns and promotions to boost awareness and trip volumes. We can promote the service more in cities like Vishakhapatnam, Coimbatore, and Mysore with offers and discounts to attract new riders.
- We can review fares in Surat and other low-fare cities to ensure they are reasonable and profitable and we can use flexible pricing during busy times to maximize revenue.
- We should leverage the demand spikes in February and March by introducing seasonal offers or promotional campaigns and also we can develop strategies to boost demand in off-peak months, such as June, through partnerships with local events or businesses.
- We can enhance weekend trip volumes further by running city-specific weekend discounts or special offers.



RANA BASAK

FURTHER ANALYSIS

- Repeat passenger rates are influenced by service quality, pricing, and local factors. Cities like Surat and Lucknow, with good services and fair prices, have higher repeat rates, while poor service or high prices can reduce repeat passenger rates.

- Tourism seasons and local events drive demand spikes in cities like Jaipur and Surat. Goodcabs can capitalize on this by targeting marketing efforts and promotions around these events and festivals to attract more passengers.

- Electric vehicle adoption and green energy initiatives are growing trends that can attract environmentally conscious passengers and boost the cab service market.

RANA BASAK

FURTHER ANALYSIS

- Goodcabs can collaborate with local hotels, malls, and event venues by offering ride packages for tourists and shuttle services for events or conferences to increase demand.

- Goodcabs should collect customer feedback, service performance data, and market trends to gain insights into customer behavior and optimize operations for improved performance.

RANA BASAK

THANK YOU!

RANA BASAK

12/91/25.