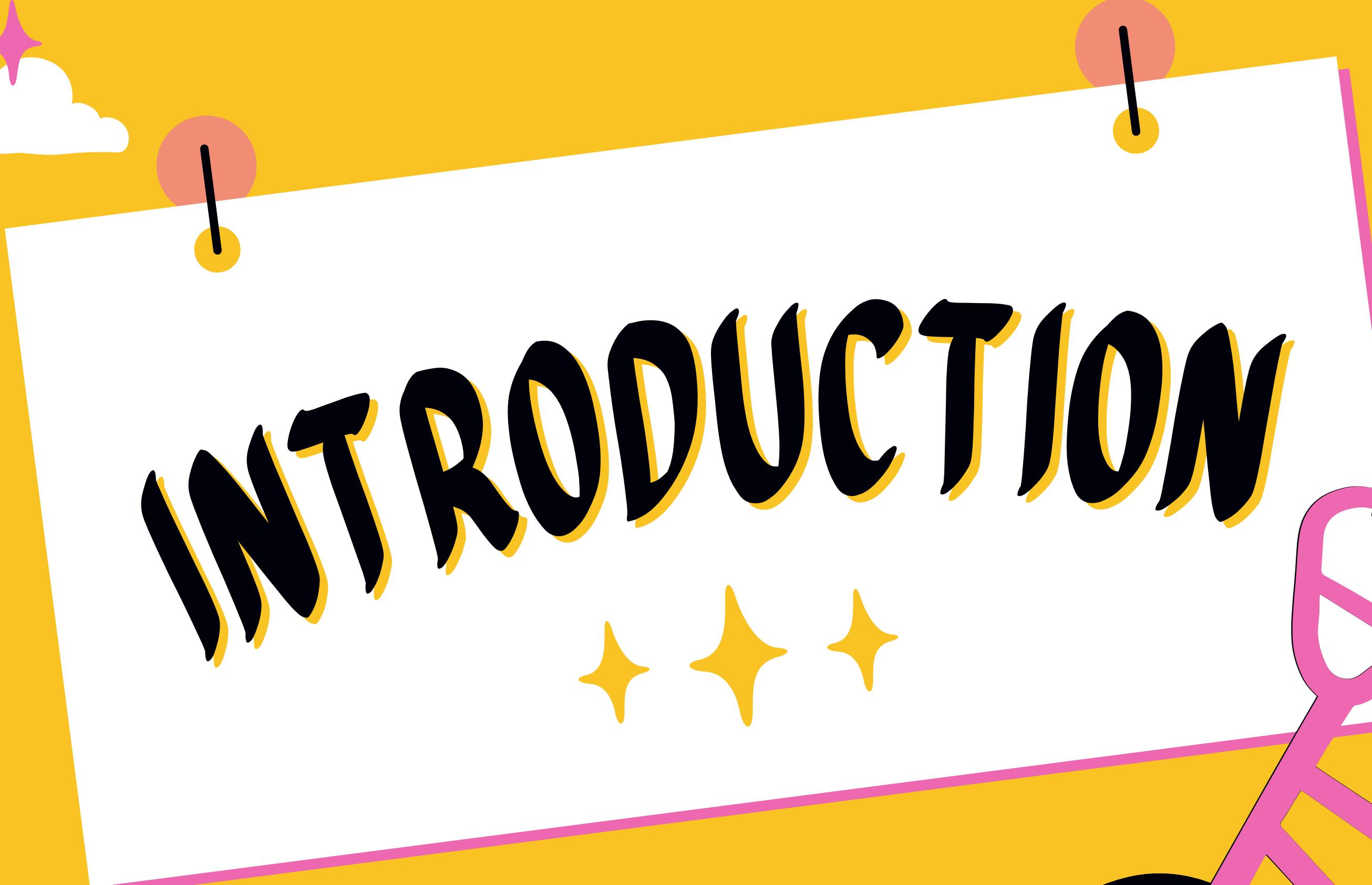


COODCABS TRANSPORTATION ANALYSIS

DOMAIN: TRANSPORTATION & MOBILITY



# INTRODUCTION



**Goodcabs**, a cab service company founded two years ago, has established a strong presence in the Indian market by prioritizing tier-2 cities. Unlike other providers, Goodcabs focuses on empowering local drivers to earn a sustainable income in their hometowns while delivering top-notch service to passengers. Currently operating in ten tier-2 cities across India, the company has set ambitious goals to drive growth and enhance passenger satisfaction.

To achieve these objectives, the **Goodcabs** management team is evaluating performance across key metrics such as

1. Trip volume,
2. Passenger satisfaction levels,
3. Repeat passenger rates,
4. Trip distribution, and
5. The ratio of new to repeat passengers.





# TASK

- 1. 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.
- 2. 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.
- 3. Create a Dashboard:** We design a dashboard showcasing our metrics and analysis.
- 4. 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.

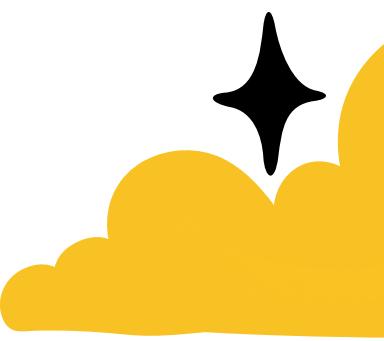


# BUSINESS QUERIES

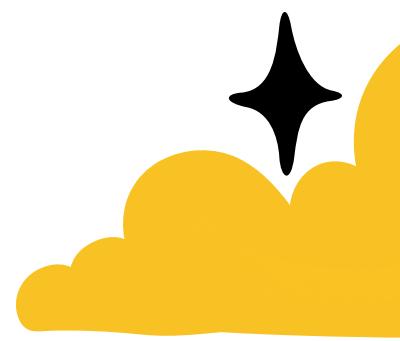
Generate a report that displays the total trips, average fare per km, average fare per trip, and the percentage contribution of each city's trips to the overall trips. This report will help in assessing trip volume, pricing efficiency, and each city's contribution to the overall trip count.

Fields:

- city\_name
- total\_trips
- avg\_fare\_per\_km
- avg\_fare\_per\_trip
- %\_contribution\_to\_total\_trips



```
-- Business Request-1 City-Level fare and trip summary report --
with cte as(
select
dc.city_name,
count(ft.trip_id) as 'total_trips',
sum(ft.fare_amount) as 'total_revenue',
sum(ft.distance_travelled_km) as 'total_distance_travelled'
from dim_city dc join fact_trips ft
on dc.city_id = ft.city_id
group by dc.city_name)
select *,
(total_revenue/total_distance_travelled) as 'avg_fare_per_km',
(total_revenue/total_trips) as 'avg_fare_per_trip',
concat(round((total_trips * 100 / (SELECT SUM(total_trips) FROM cte)),2),'%') as '%_contribution_to_total_trips'
from cte;
```



	city_name	total_trips	total_revenue	total_distance_travelled	avg_fare_per_km	avg_fare_per_trip	%_contribution_to_total_trips
1	Chandigarh	38981	11058401	916783	12.062179	283.686950	9%
2	Coimbatore	21104	3523992	316121	11.147604	166.982183	4%
3	Indore	42456	7635228	700629	10.897676	179.838609	9%
4	Jaipur	76888	37207497	2308418	16.118180	483.918127	18%
5	Kochi	50702	16997596	1220167	13.930548	335.245079	11%
6	Lucknow	64299	9463551	804571	11.762232	147.180376	15%
7	Mysore	16238	4054745	267877	15.136592	249.707168	3%
8	Surat	54843	6431599	603122	10.663844	117.272924	12%
9	Vadodara	32026	3797200	368867	10.294225	118.566164	7%
10	Visakhapatnam	28366	8018282	639765	12.533167	282.672283	6%



Generate a report that evaluates the target performance for trips at the monthly and city level. For each city and month, compare the actual total trips with the target trips and categorise the performance as follows:

- If actual trips are greater than target trips, mark it as 'Above Target'.
- If actual trips are less than or equal to target trips, mark it as 'Below Target'.

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

Fields:

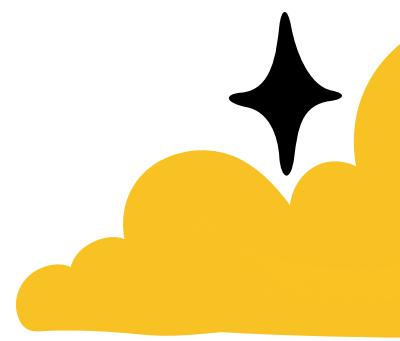
- city\_name
- month\_name
- actual\_trips
- target\_trips
- performance\_status
- %\_difference

```
-- Business Request-2 Monthly City-Level Trips Target Performance Report --
with actual_trips as (
select city_id, DATENAME(month, date) as 'month_name', DATENAME(year, date) as 'year', COUNT(*) as 'actual_trips'
from fact_trips
group by city_id, DATENAME(month, date), DATENAME(year, date)),

target_trips as (
select city_id, DATENAME(month, month) as 'month_name', DATENAME(year, month) as 'year', SUM(total_target_trips) as 'target_trips'
from monthly_target_trips
group by city_id, DATENAME(month, month), DATENAME(year, month)
),

result_query as (
select c1.city_id, c1.month_name, c1.actual_trips, c2.target_trips,
case
when c1.actual_trips > c2.target_trips then 'Above Target' else 'Below Target' end as 'performance_status',
(c1.actual_trips - c2.target_trips)*100/ c2.target_trips as '%_difference'
from actual_trips c1 join target_trips c2
on c1.city_id = c2.city_id and c1.month_name = c2.month_name and c1.year = c2.year)

select c3.city_id, dc.city_name , c3.month_name, c3.actual_trips, c3.target_trips,
case when c3.actual_trips > c3.target_trips then 'Above Target' else 'Below Target' end as 'performance_status',
(c3.actual_trips - c3.target_trips)*100/ c3.target_trips as '%_difference'
from result_query c3 join dim_city dc
on c3.city_id = dc.city_id;
```



	city_id	city_name	month_name	actual_trips	target_trips	performance_status	%_difference
1	AP01	Visakhapatnam	January	4468	4500	Below Target	-0.711111
2	AP01	Visakhapatnam	June	4478	5000	Below Target	-10.440000
3	AP01	Visakhapatnam	March	4877	4500	Above Target	8.377777
4	AP01	Visakhapatnam	April	4938	5000	Below Target	-1.240000
5	AP01	Visakhapatnam	May	4812	5000	Below Target	-3.760000
6	AP01	Visakhapatnam	February	4793	4500	Above Target	6.511111
7	CH01	Chandigarh	March	6569	7000	Below Target	-6.157142
8	CH01	Chandigarh	February	7387	7000	Above Target	5.528571
9	CH01	Chandigarh	January	6810	7000	Below Target	-2.714285
10	CH01	Chandigarh	April	5566	6000	Below Target	-7.233333
11	CH01	Chandigarh	June	6029	6000	Above Target	0.483333
12	CH01	Chandigarh	May	6620	6000	Above Target	10.333333
13	GJ01	Surat	May	9774	10000	Below Target	-2.260000
14	GJ01	Surat	April	9831	10000	Below Target	-1.690000
15	GJ01	Surat	June	8544	10000	Below Target	-14.560000
16	GJ01	Surat	March	9267	9000	Above Target	2.966666
17	GJ01	Surat	January	8358	9000	Below Target	-7.133333
18	GJ01	Surat	February	9069	9000	Above Target	0.766666
19	GJ02	Vadodara	May	5799	6500	Below Target	-10.784615
20	GJ02	Vadodara	April	5941	6500	Below Target	-8.600000
21	GJ02	Vadodara	January	4775	6000	Below Target	-20.416666
22	GJ02	Vadodara	March	5598	6000	Below Target	-6.700000
23	GJ02	Vadodara	February	5228	6000	Below Target	-12.866666
24	GJ02	Vadodara	June	4685	6500	Below Target	-27.923076
25	KA01	Mysore	April	2603	2500	Above Target	4.120000
26	KA01	Mysore	January	2485	2000	Above Target	24.250000
27	KA01	Mysore	May	3007	2500	Above Target	20.280000
28	KA01	Mysore	February	2668	2000	Above Target	33.400000
29	KA01	Mysore	June	2842	2500	Above Target	13.680000
30	KA01	Mysore	March	2633	2000	Above Target	31.650000

31	KL01	Kochi	May	10014	9000	Above Target	11.266666
32	KL01	Kochi	April	9762	9000	Above Target	8.466666
33	KL01	Kochi	June	6399	9000	Below Target	-28.900000
34	KL01	Kochi	February	7688	7500	Above Target	2.506666
35	KL01	Kochi	March	9495	7500	Above Target	26.600000
36	KL01	Kochi	January	7344	7500	Below Target	-2.080000
37	MP01	Indore	March	7019	7000	Above Target	0.271428
38	MP01	Indore	February	7210	7000	Above Target	3.000000
39	MP01	Indore	January	6737	7000	Below Target	-3.757142
40	MP01	Indore	April	7415	7500	Below Target	-1.133333
41	MP01	Indore	June	6288	7500	Below Target	-16.160000
42	MP01	Indore	May	7787	7500	Above Target	3.826666
43	RJ01	Jaipur	June	9842	9500	Above Target	3.600000
44	RJ01	Jaipur	February	15872	13000	Above Target	22.092307
45	RJ01	Jaipur	January	14976	13000	Above Target	15.200000
46	RJ01	Jaipur	March	13317	13000	Above Target	2.438461
47	RJ01	Jaipur	May	11475	9500	Above Target	20.789473
48	RJ01	Jaipur	April	11406	9500	Above Target	20.063157
49	TN01	Coimbatore	May	3550	3500	Above Target	1.428571
50	TN01	Coimbatore	January	3651	3500	Above Target	4.314285
51	TN01	Coimbatore	June	3158	3500	Below Target	-9.771428
52	TN01	Coimbatore	February	3404	3500	Below Target	-2.742857
53	TN01	Coimbatore	March	3680	3500	Above Target	5.142857
54	TN01	Coimbatore	April	3661	3500	Above Target	4.600000
55	UP01	Lucknow	February	12060	13000	Below Target	-7.230769
56	UP01	Lucknow	January	10858	13000	Below Target	-16.476923
57	UP01	Lucknow	March	11224	13000	Below Target	-13.661538
58	UP01	Lucknow	June	10240	11000	Below Target	-6.909090
59	UP01	Lucknow	April	10212	11000	Below Target	-7.163636
60	UP01	Lucknow	May	9705	11000	Below Target	-11.772727



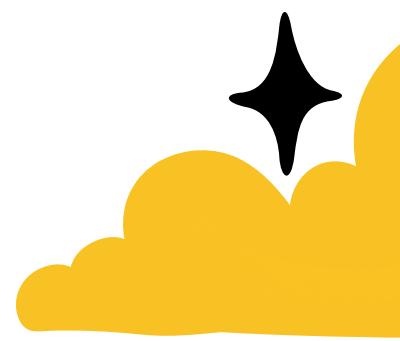
Generate a report that shows the percentage distribution of repeat passengers by the number of trips they have taken in each city. Calculate the percentage of repeat passengers who took 2 trips, 3 trips, and so on, up to 10 trips.

Each column should represent a trip count category, displaying the percentage of repeat passengers who fall into that category out of the total repeat passengers for that city.

This report will help identify cities with high repeat trip frequency, which can indicate strong customer loyalty or frequent usage patterns.

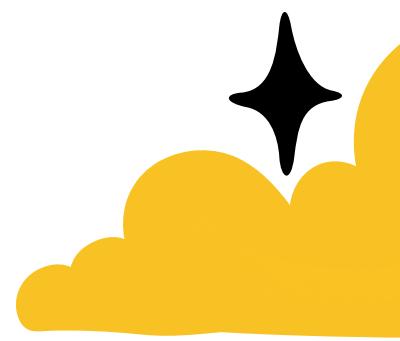
Fields:

- city\_name
- 2-Trips
- 3-Trips
- 4-Trips
- 5-Trips
- 6-Trips
- 7-Trips
- 8-Trips
- 9-Trips
- 10-Trips

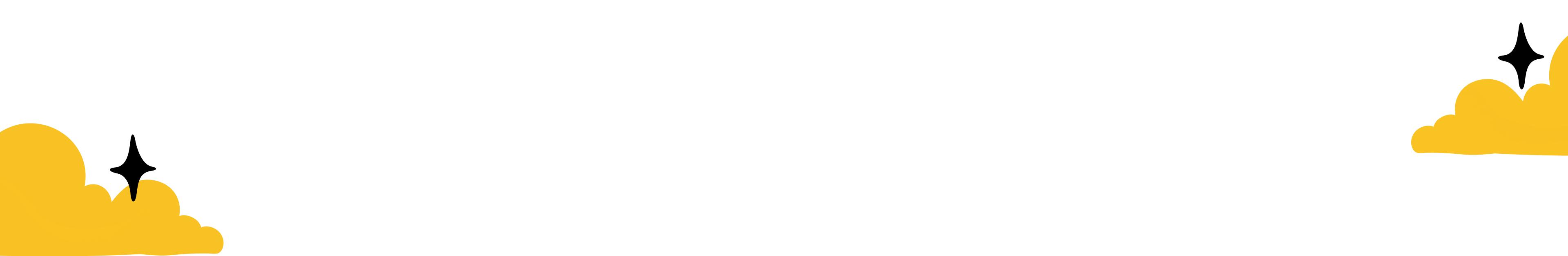


```
-- Business Request-3 City-Level Repeat passenger trip frequency report --
with cte as (
    select dc.city_id, td.trip_count, SUM(td.repeat_passenger_count) as 'repeat_passenger_per_trips'
    from dim_city dc join dim_repeat_trip_distribution td
    on dc.city_id = td.city_id
    group by dc.city_id, td.trip_count),

cte1 as (
    select c1.city_id,
        sum(case when c1.trip_count = '9-Trips' then c1.repeat_passenger_per_trips end)*100/sum(c1.repeat_passenger_per_trips) as '9-Trips',
        sum(case when c1.trip_count = '8-Trips' then c1.repeat_passenger_per_trips end)*100/sum(c1.repeat_passenger_per_trips) as '8-Trips',
        sum(case when c1.trip_count = '7-Trips' then c1.repeat_passenger_per_trips end)*100/sum(c1.repeat_passenger_per_trips) as '7-Trips',
        sum(case when c1.trip_count = '6-Trips' then c1.repeat_passenger_per_trips end)*100/sum(c1.repeat_passenger_per_trips) as '6-Trips',
        sum(case when c1.trip_count = '5-Trips' then c1.repeat_passenger_per_trips end)*100/sum(c1.repeat_passenger_per_trips) as '5-Trips',
        sum(case when c1.trip_count = '4-Trips' then c1.repeat_passenger_per_trips end)*100/sum(c1.repeat_passenger_per_trips) as '4-Trips',
        sum(case when c1.trip_count = '3-Trips' then c1.repeat_passenger_per_trips end)*100/sum(c1.repeat_passenger_per_trips) as '3-Trips',
        sum(case when c1.trip_count = '2-Trips' then c1.repeat_passenger_per_trips end)*100/sum(c1.repeat_passenger_per_trips) as '2-Trips',
        sum(case when c1.trip_count = '10-Trips' then c1.repeat_passenger_per_trips end)*100/sum(c1.repeat_passenger_per_trips) as '10-Trips'
    from cte c1
    group by c1.city_id
)
select * from cte1;
```



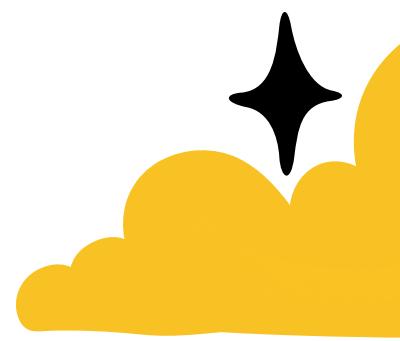
	city_id	9-Trips	8-Trips	7-Trips	6-Trips	5-Trips	4-Trips	3-Trips	2-Trips	10-Trips
1	AP01	0.880971	1.389976	1.977290	3.191072	5.442443	9.984338	24.960845	51.252936	0.920125
2	CH01	2.327416	3.471400	5.483234	7.416173	12.209072	15.739644	19.250493	32.307692	1.794871
3	GJ01	1.736513	6.239870	11.889326	18.453345	19.749942	16.554758	14.262560	9.759203	1.354480
4	GJ02	2.047860	5.775425	12.862402	19.075011	18.062586	16.520938	14.173953	9.871145	1.610676
5	KA01	0.541638	1.421800	1.760324	4.062288	5.822613	12.728503	24.441435	48.747461	0.473933
6	KL01	1.206399	1.652242	2.111198	3.907684	6.477838	11.814843	24.350904	47.665879	0.813008
7	MP01	2.383592	3.256651	5.238359	6.845898	10.338137	13.400776	22.685698	34.340354	1.510532
8	RJ01	1.198099	1.900433	2.520140	4.131377	6.290022	12.115265	20.729188	50.144598	0.970873
9	TN01	2.312818	6.154449	10.466483	17.640141	20.619364	15.562524	14.817718	11.211289	1.215209
10	UP01	1.906845	6.429092	11.326456	20.183390	18.422423	16.202980	14.765030	9.659268	1.104511



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

Fields:

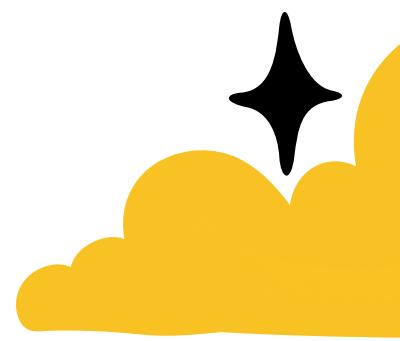
- city\_name
- total\_new\_passengers
- city\_category



```
-- Business Request-4 Identify cities with highest and lowest total new passengers --
with cte as (
    select dc.city_name, SUM(new_passengers) as 'total_new_passengers'
    from fact_passenger_summary fps join dim_city dc on
    fps.city_id = dc.city_id
    group by dc.city_name),

cte1 as (
    select *,
    RANK() over(order by total_new_passengers) as 'lowest_rank',
    RANK() over(order by total_new_passengers desc) as 'highest_rank'
    from cte c1)

select c2.city_name, c2.total_new_passengers, case when highest_rank <=3 then 'Top 3' else 'Bottom 3' end as 'city_category'
from cte1 c2
where highest_rank <=3 or lowest_rank <=3;
```



	city_name	total_new_passengers	city_category
1	Jaipur	45856	Top 3
2	Kochi	26416	Top 3
3	Chandigarh	18908	Top 3
4	Surat	11626	Bottom 3
5	Vadodara	10127	Bottom 3
6	Coimbatore	8514	Bottom 3



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.

Fields:

- city\_name
- highest\_revenue\_month
- revenue
- percentage\_contribution (%)



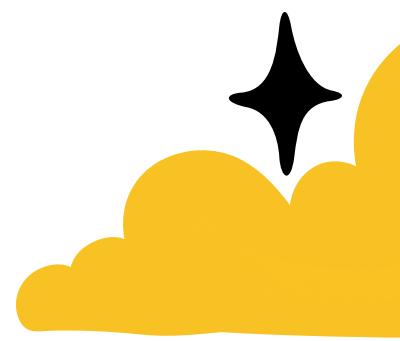
```
-- Business Request-5 Identify month with highest revenue for each city --
with cte as (
    select city_id, SUM(fare_amount) as 'total_revenue'
    from fact_trips
    group by city_id),

cte1 as (
    select city_id, DATENAME(month, date) as 'monthName', SUM(fare_amount) as 'revenue'
    from fact_trips
    group by city_id, DATENAME(month, date)),

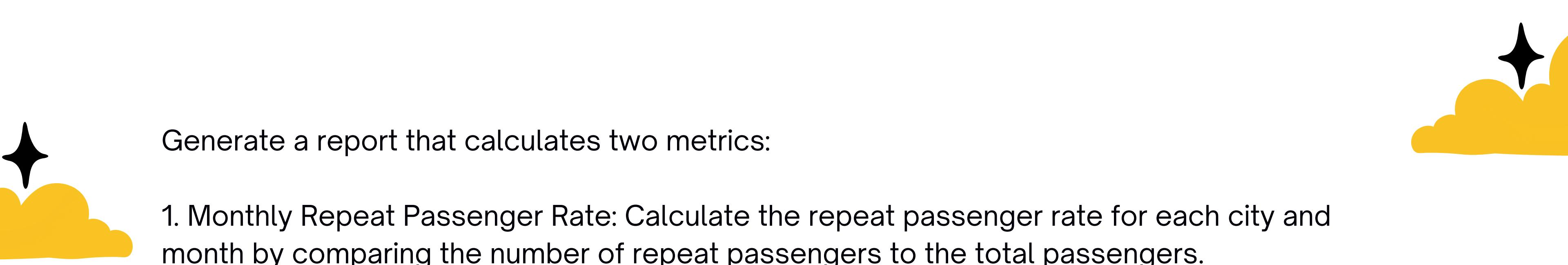
cte2 as (
    select c1.city_id, c2.monthName, c2.revenue, c1.total_revenue, cast(round((c2.revenue*100/c1.total_revenue),2) as decimal(10,2)) as '%_contribution'
    from cte c1 join cte1 c2
    on c1.city_id = c2.city_id),

cte3 as (
    select dc.city_name, (c3.monthName) as 'highest_revenue_month', c3.revenue, c3.[%_contribution]
    from cte2 c3 join dim_city dc
    on c3.city_id = dc.city_id
)

select * from (
    select *,
        RANK() over(partition by city_name order by revenue desc) as 'rnk'
    from cte3) as A
where A.rnk = 1;
```



	city_name	highest_revenue_month	revenue	%_contribution	rnk
1	Chandigarh	February	2108290	19.07	1
2	Coimbatore	April	612431	17.38	1
3	Indore	May	1380996	18.09	1
4	Jaipur	February	7747202	20.82	1
5	Kochi	May	3333746	19.61	1
6	Lucknow	February	1777269	18.78	1
7	Mysore	May	745170	18.38	1
8	Surat	April	1154909	17.96	1
9	Vadodara	April	706250	18.60	1
10	Visakhapatnam	April	1390682	17.34	1



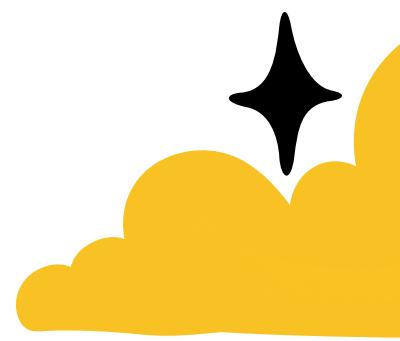
Generate a report that calculates two metrics:

1. Monthly Repeat Passenger Rate: Calculate the repeat passenger rate for each city and month by comparing the number of repeat passengers to the total passengers.
2. City-wise Repeat Passenger Rate: Calculate the overall repeat passenger rate for each city, considering all passengers across months.

These metrics will provide insights into monthly repeat trends as well as the overall repeat behaviour for each city.

Fields:

- city\_name
- month
- total\_passengers
- repeat\_passengers
- monthly\_repeat\_passenger\_rate (%): Repeat passenger rate at the city and month level.
- city\_repeat\_passenger\_rate (%): Overall repeat passenger rate for each city, aggregated across months.

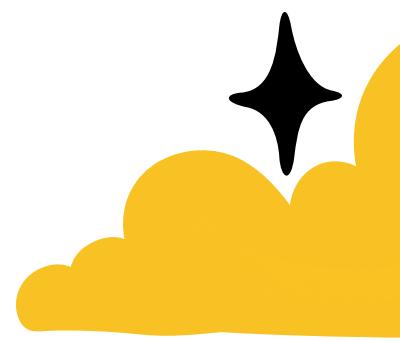


```
-- Business Request-6 Repeat Passenger Rate Analysis --
with cte as (
    select city_id, DATENAME(month, month) as 'month', repeat_passengers, total_passengers,
    cast(round((repeat_passengers*100/total_passengers),2) as decimal(10,2)) as 'repeat_passenger_rate(%)'
    from fact_passenger_summary),

    cte1 as (
        select cte.city_id,
        cast(round((SUM(cte.repeat_passengers)*100/SUM(cte.total_passengers)),2) as decimal(10,2)) as 'overall_repeat_passengers(%)'
        from cte
        group by city_id),

    cte2 as (
        select cte.city_id, cte.month, cte.repeat_passengers, cte.total_passengers, cte.[repeat_passenger_rate(%)], cte1.[overall_repeat_passengers(%)]
        from cte join cte1
        on cte.city_id = cte1.city_id)

select dc.city_name, cte2.month, cte2.repeat_passengers, cte2.total_passengers, cte2.[repeat_passenger_rate(%)],
cte2.[overall_repeat_passengers(%)]
from cte2 join dim_city dc on
cte2.city_id = dc.city_id;
```



	city_name	month	repeat_passengers	total_passenger	repeat_passenger_rate(%)	overall_repeat_passenger(%)
1	Visakhapatnam	January	650	3163	20.55	28.61
2	Visakhapatnam	February	790	3170	24.92	28.61
3	Visakhapatnam	March	923	3093	29.84	28.61
4	Visakhapatnam	April	992	2837	34.97	28.61
5	Visakhapatnam	May	951	2890	32.91	28.61
6	Visakhapatnam	June	802	2702	29.68	28.61
7	Chandigarh	June	867	3297	26.30	21.14
8	Chandigarh	May	969	3699	26.20	21.14
9	Chandigarh	April	789	3285	24.02	21.14
10	Chandigarh	March	872	4100	21.27	21.14
11	Chandigarh	February	853	4957	17.21	21.14
12	Chandigarh	January	720	4640	15.52	21.14
13	Surat	January	1184	3616	32.74	42.63
14	Surat	February	1313	3567	36.81	42.63
15	Surat	March	1494	3440	43.43	42.63
16	Surat	April	1551	3394	45.70	42.63
17	Surat	May	1606	3217	49.92	42.63
18	Surat	June	1490	3030	49.17	42.63
19	Vadodara	June	703	1807	38.90	30.03
20	Vadodara	May	868	2256	38.48	30.03
21	Vadodara	April	862	2499	34.49	30.03
22	Vadodara	March	759	2522	30.10	30.03
23	Vadodara	February	610	2756	22.13	30.03
24	Vadodara	January	544	2633	20.66	30.03
25	Mysore	January	172	2129	8.08	11.23
26	Mysore	February	183	2290	7.99	11.23
27	Mysore	March	208	2194	9.48	11.23
28	Mysore	April	236	2072	11.39	11.23
29	Mysore	May	349	2270	15.37	11.23
30	Mysore	June	329	2203	14.93	11.23
31	Kochi	June	1049	4060	25.84	22.40
32	Kochi	May	1853	6222	29.78	22.40
33	Kochi	April	1576	6515	24.19	22.40

34	Kochi	March	1348	6213	21.70	22.40
35	Kochi	February	1005	5372	18.71	22.40
36	Kochi	January	795	5660	14.05	22.40
37	Indore	January	1033	3876	26.65	32.68
38	Indore	February	1103	3981	27.71	32.68
39	Indore	March	1091	3833	28.46	32.68
40	Indore	April	1295	3646	35.52	32.68
41	Indore	May	1563	3591	43.53	32.68
42	Indore	June	1131	3152	35.88	32.68
43	Jaipur	June	1181	6956	16.98	17.43
44	Jaipur	May	1842	7174	25.68	17.43
45	Jaipur	April	1736	7856	22.10	17.43
46	Jaipur	March	1840	9257	19.88	17.43
47	Jaipur	February	1661	12450	13.34	17.43
48	Jaipur	January	1422	11845	12.01	17.43
49	Coimbatore	January	392	2214	17.71	23.05
50	Coimbatore	February	346	1993	17.36	23.05
51	Coimbatore	March	427	1965	21.73	23.05
52	Coimbatore	April	480	1722	27.87	23.05
53	Coimbatore	May	504	1543	32.66	23.05
54	Coimbatore	June	402	1628	24.69	23.05
55	Lucknow	June	1727	3698	46.70	37.12
56	Lucknow	May	1662	3487	47.66	37.12
57	Lucknow	April	1496	3807	39.30	37.12
58	Lucknow	March	1622	4781	33.93	37.12
59	Lucknow	February	1659	5188	31.98	37.12
60	Lucknow	January	1431	4896	29.23	37.12

# DASHBOARD

I used Power BI to create an interactive dashboard that showcases key metrics such as total passengers, new passengers, repeat passengers, and total revenue. The analysis is segmented by passenger type and day type (weekend vs. weekdays). This project turned raw data into actionable insights, enabling more informed decision-making.



January

February

March

April

May

June

New

Repeated

Weekday

Weekend

All



Total Passengers

238309<sup>✓</sup>

LM: (Blank) (+Infinity%)



Total New Passengers

176998<sup>✓</sup>

LM: (Blank) (+Infinity%)

Total Repeat Passengers

61311<sup>✓</sup>

LM: (Blank) (+Infinity%)

Total Revenue

108188091<sup>✓</sup>

LM: (Blank) (+Infinity%)

Passenger Rating

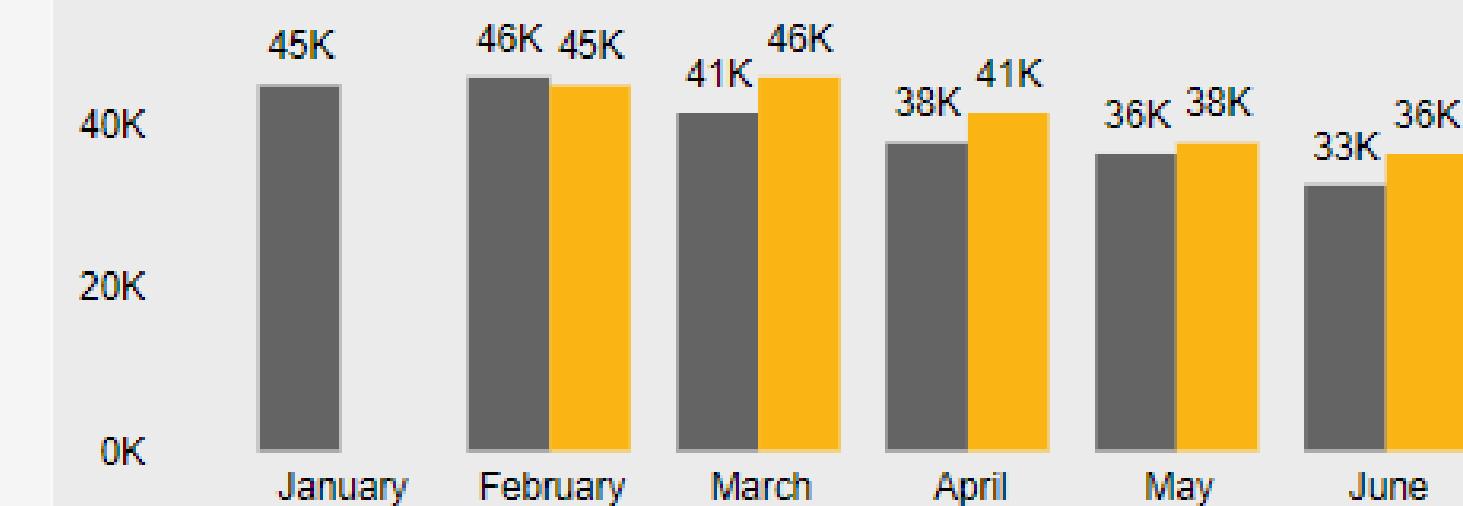


City	Passenger Rating
Mysore	
Jaipur	
Kochi	
Visakhapatnam	
Chandigarh	
Coimbatore	
Indore	
Vadodara	

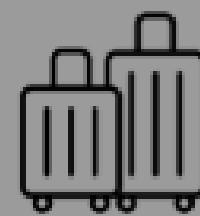
Driver Rating

City	Driver Rating
Kochi	8.99
Visakhapatnam	8.99
Jaipur	8.99
Mysore	8.98
Chandigarh	7.72
Coimbatore	7.69
Indore	7.65
Vadodara	6.65
Lucknow	6.62

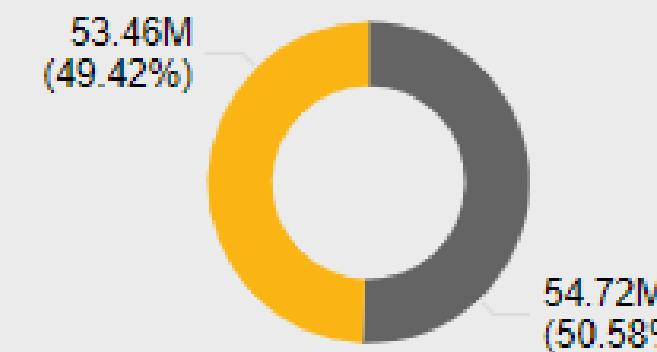
Total Passengers By Month



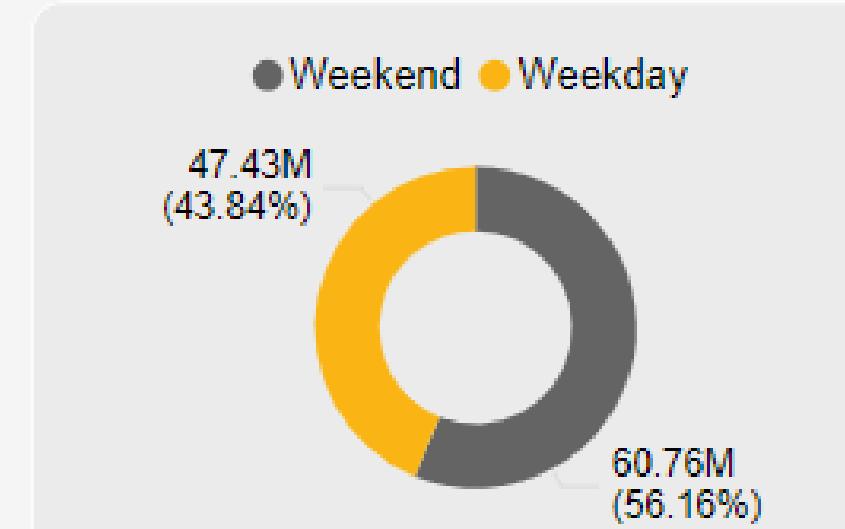
Revenue By Passenger Type



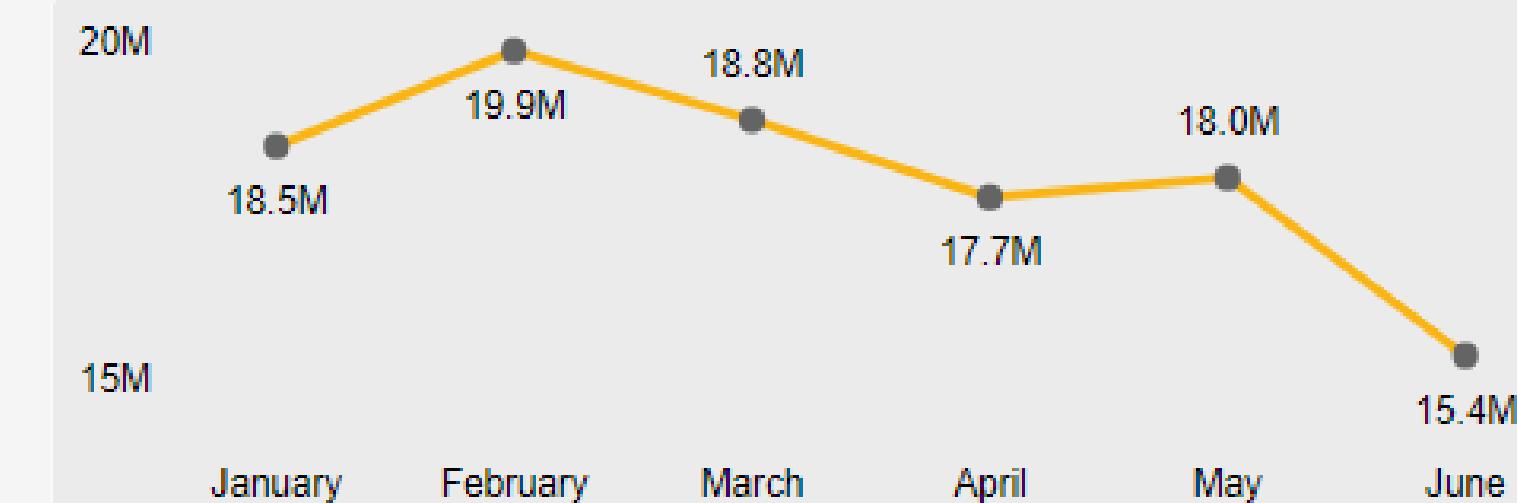
● Repeated ● New



Revenue By Day Type



Revenue By Month





January

February

March

April

May

June

New

Repeated

Weekday

Weekend

All



## Total Trips

425903

LM: (Blank) (+Infinity%)



## Total Distance Travelled

8146320

LM: (Blank) (+Infinity%)

## Average Fare Per Km

13.28

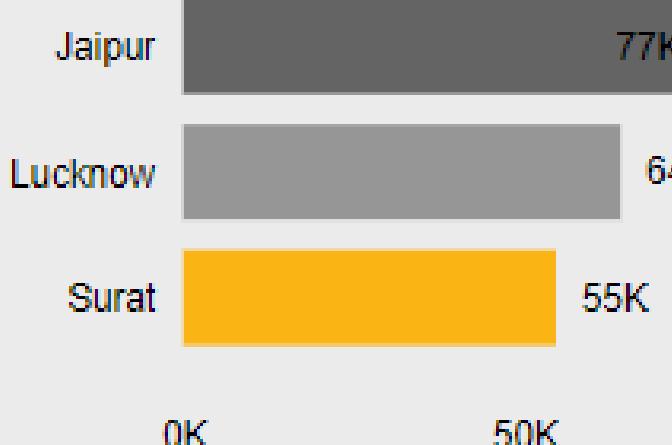
LM: (Blank) (+Infinity%)

## Total Revenue

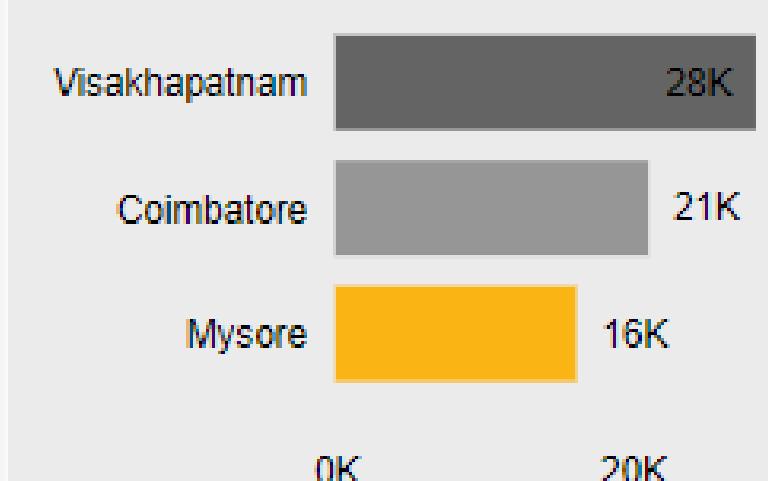
108188091

LM: (Blank) (+Infinity%)

## Top 3 City By Total Trips



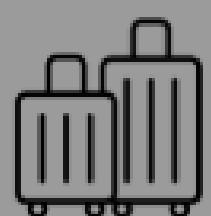
## Bottom 3 City By Total Trips



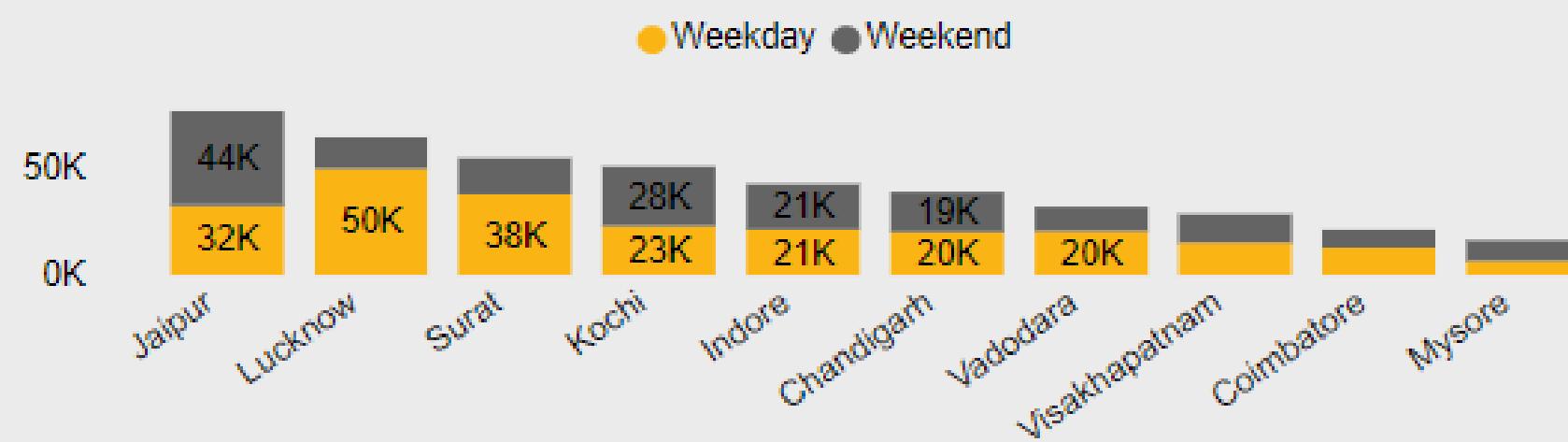
## Repeat Passengers Frequency By City

City	10-Trips	2-Trips	3-Trips	4-Trips	5-Trips	6-Trips
Chandigarh	1.79%	32.31%	19.25%	15.74%	12.21%	7.42%
Coimbatore	1.22%	11.21%	14.82%	15.56%	20.62%	17.64%
Indore	1.51%	34.34%	22.69%	13.40%	10.34%	6.85%
Jaipur	0.97%	50.14%	20.73%	12.12%	6.29%	4.13%
Kochi	0.81%	47.67%	24.35%	11.81%	6.48%	3.91%
Lucknow	1.10%	9.66%	14.77%	16.20%	18.42%	20.18%
Mysore	0.47%	48.75%	24.44%	12.73%	5.82%	4.06%
Surat	1.35%	9.76%	14.26%	16.55%	19.75%	18.45%
Vadodara	1.61%	9.87%	14.17%	16.52%	18.06%	19.08%

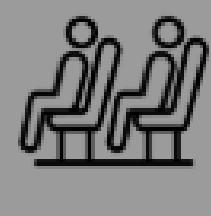
## Total Trips By City and Day Type



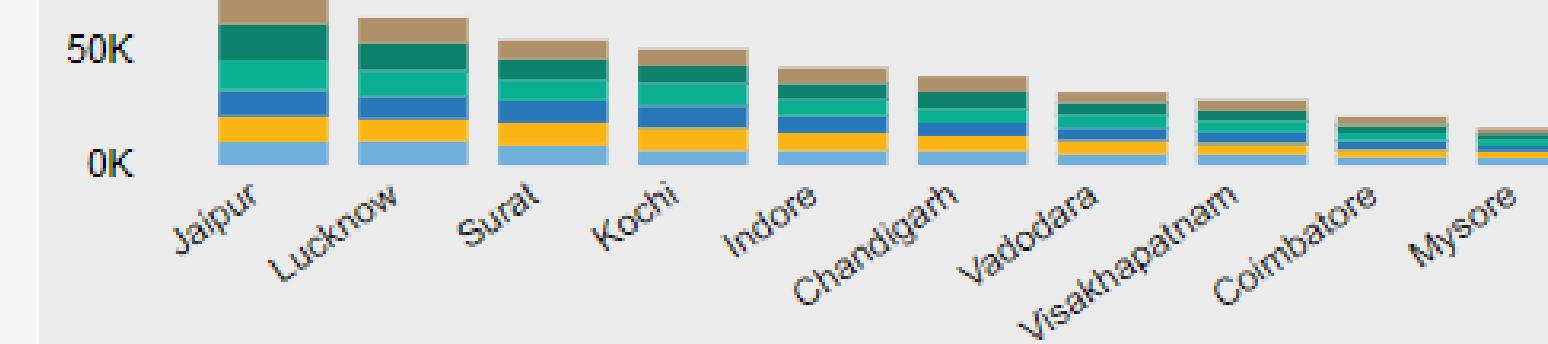
● Weekday ● Weekend



## Total Trips By City and Month



● June ● May ● April ● March ● February ● January





January

February

March

April

May

June

New

Repeated

Weekday

Weekend

All

▼

## Repeat Passengers

61311

LM: (Blank) (+Infinity%)

## Repeat Passengers Rate %

25.73%

LM: (Blank) (+Infinity%)

## New vs Repeated Passengers

2.89

LM: (Blank) (+Infinity%)

## Total Revenue

108188091

LM: (Blank) (+Infinity%)



## Target Achievement KPIs

City	Month	Trips % Diff	Passenger % Diff
Chandigarh	January	-2.71%	
Chandigarh	February	5.53%	
Chandigarh	March	-6.16%	
Chandigarh	April	-7.23%	
Chandigarh	May	10.33%	
Chandigarh	June	0.48%	
Coimbatore	January	4.31%	
Coimbatore	February	-2.74%	
Coimbatore	March	-5.44%	

## RPR % By City and Month

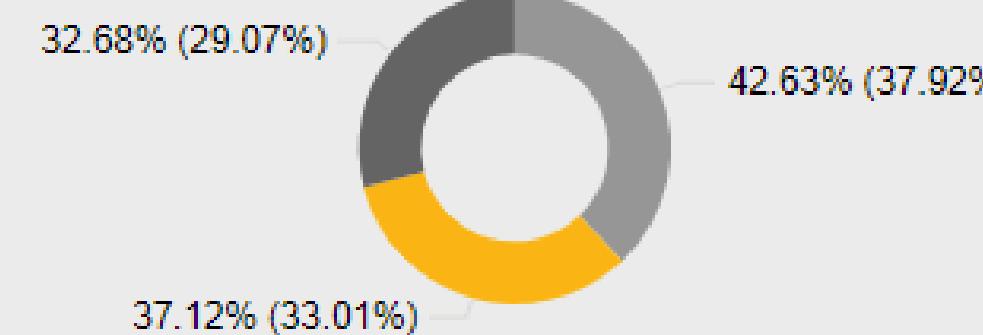
City	January	February	March
Surat	32.74%	36.81%	43.43%
Lucknow	29.23%	31.98%	33.93%
Indore	26.65%	27.71%	28.46%
Vadodara	20.66%	22.13%	30.10%
Visakhapatnam	20.55%	24.92%	29.84%
Coimbatore	17.71%	17.36%	21.73%
Kochi	14.05%	18.71%	21.70%
Chandigarh	15.52%	17.21%	21.27%
Jaipur	12.01%	13.34%	19.88%

## Average Fare Per Trip By City

City	Avg. Fare Per Trips	Avg. Trips
Jaipur	483.92	1.00
Kochi	335.25	1.00
Chandigarh	283.69	1.00
Visakhapatnam	282.67	1.00
Mysore	249.71	1.00
Indore	179.84	1.00
Coimbatore	166.98	1.00
Lucknow	147.18	1.00
Vadodara	118.57	1.00

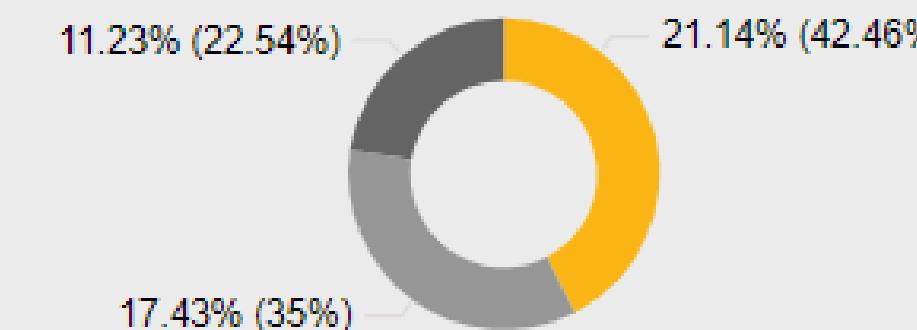
## Top 3 City By RPR %

● Surat ● Lucknow ● Indore



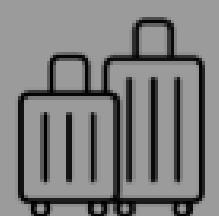
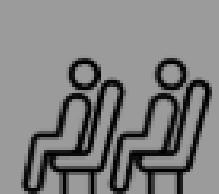
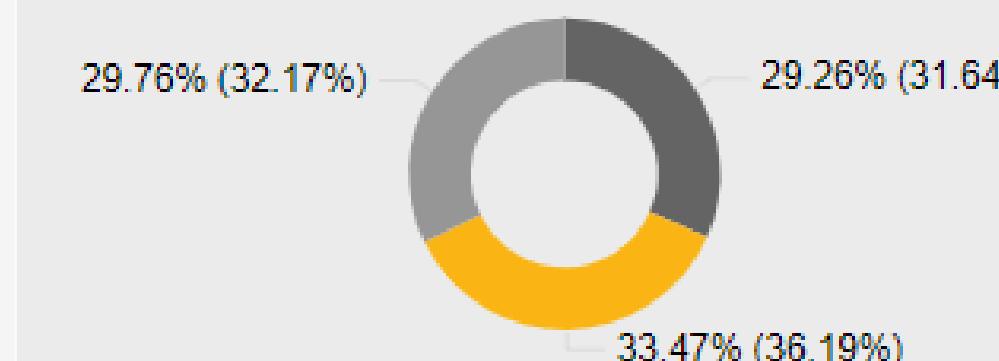
## Bottom 3 City By RPR %

● Chandigarh ● Jaipur ● Mysore



## Top 3 Months By RPR %

● April ● May ● June



# INSIGHTS

- Jaipur, Lucknow, and Surat emerged as the top three cities with the highest total trip volumes, while Vishakhapatnam, Coimbatore, and Mysore recorded the lowest trip volumes.
- 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.
- Jaipur has the highest average fare per trip, indicating potential pricing efficiency or higher willingness to pay. In contrast, Surat has the lowest average fare per trip, suggesting possible pricing adjustments or market factors.
- 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.

# INSIGHTS

- Mysore achieved the highest passenger ratings, with new passengers scoring an average of 8.98 and repeat passengers 7.98. Meanwhile, Kochi recorded the highest driver ratings at 8.99, reflecting superior driver performance.
- Demand peaks were observed in February and March, while June experienced the lowest demand for trips, likely due to seasonal variations.
- 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.

# INSIGHTS

- Jaipur showed the strongest preference for trips during weekends compared to weekdays, followed by Lucknow, Surat, and Kochi. This trend was consistent across all cities, where weekends recorded higher trip volumes.
- We can enhance weekend trip volumes further by running city-specific weekend discounts or special offers.
- Passengers in Vishakhapatnam were most likely to take exactly two trips, more than in any other city.

# INSIGHTS

- Surat and Lucknow exhibited the highest repeat passenger rates, reflecting strong customer retention. Conversely, Jaipur and Mysore showed the lowest repeat passenger rates, indicating a need for improved strategies.
- We can continue to prioritize driver performance to sustain high ratings, as observed in Kochi. We can use feedback from Mysore's new and repeat passengers to standardize quality benchmarks across cities.



# **FURTHER ANALYSIS & RECOMMENDATIONS**

What factors (such as quality of service, competitive pricing, or city demographics) might contribute to higher or lower repeat passenger rates in different cities? Are there correlations with socioeconomic or lifestyle patterns in these cities?

- Repeat rates depend on service quality, pricing, and the people of the city. Cities with good services and fair prices, like Surat, and Lucknow have more repeat passengers, while poor service or high prices can lower repeat passenger rates.

How do tourism seasons or local events (festivals, conferences) impact Goodcabs's demand patterns? Would tailoring marketing efforts to these events increase trip volume in tourism-oriented cities?

- We can see tourism seasons or local events impact Goodcabs's demand and services. It can lead to demand spikes in cities like Jaipur and Surat. Yes, we can use marketing efforts to these events which can attract more passengers in these cities. We can target the promotions and also we can have campaigns during festivals in these cities to attract more people.

What emerging mobility trends (such as electric vehicle adoption, green energy use) are impacting the cab service market in tier-2 cities? Should Goodcabs consider integrating electric vehicles or eco-friendly initiatives to stay competitive?

- Electric vehicle adoption and green energy initiatives are growing trends that could impact the cab service market. It can attract environmentally conscious passengers and also improve the business.

Are there opportunities for Goodcabs to partner with local businesses (such as hotels, malls, or event venues) to boost demand and improve customer loyalty? Could these partnerships drive more traffic, especially in tourism-heavy or high-footfall areas?

- Goodcabs could collaborate with local hotels, malls, or event venues to drive more traffic. For example, offering ride packages for tourists who are staying at hotels, or we can provide shuttle services for events or conferences, to increase the demand.

To make goodcabs more data-driven and improve its performance across key metrics (such as repeat passenger rate, customer satisfaction, new passengers and trip volume), what additional data should Goodcabs collect? Consider data that could provide deeper insights into customer behaviour, operational efficiency, and market trends.

- Goodcabs should collect data on what customers like, how well the service is running, and trends in the market. This data will provide deeper insights into customer behavior and help optimize operations for better performance.



THANK YOU!

