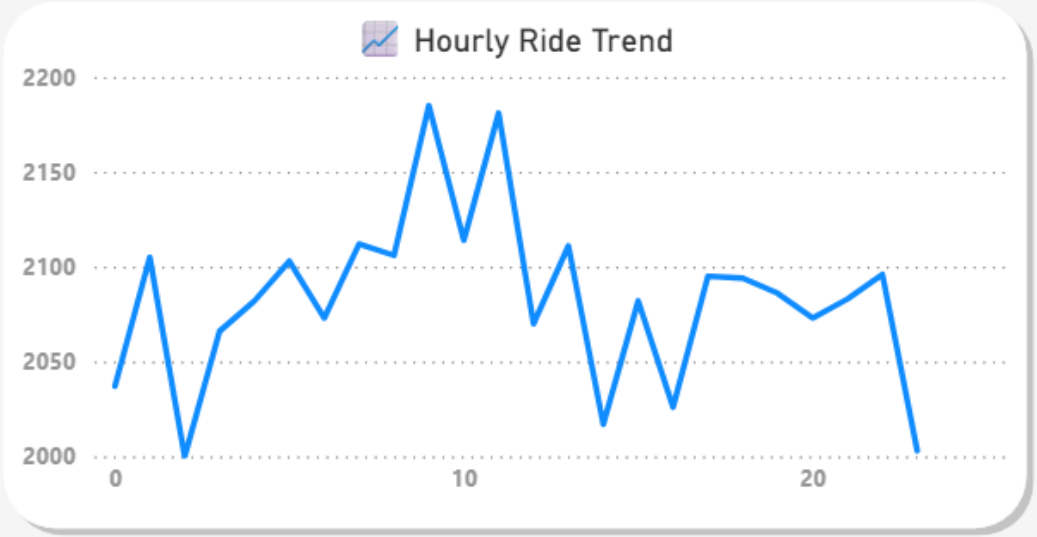
Rides Analysis

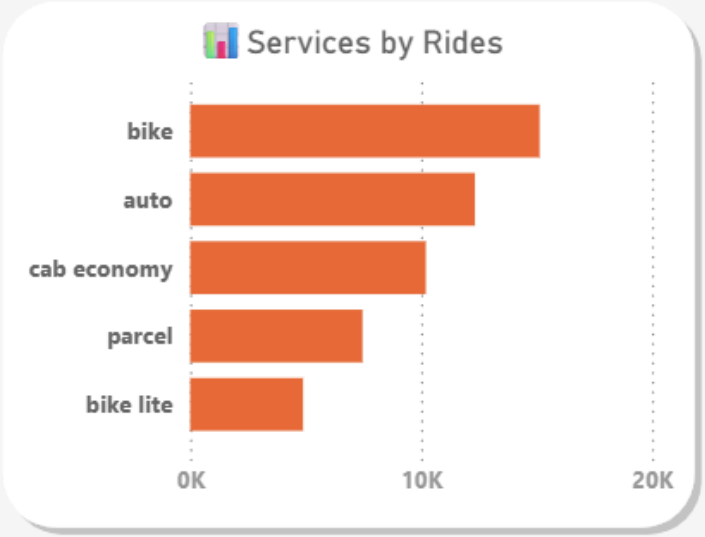
Python(pandas) , SQL , Power bi

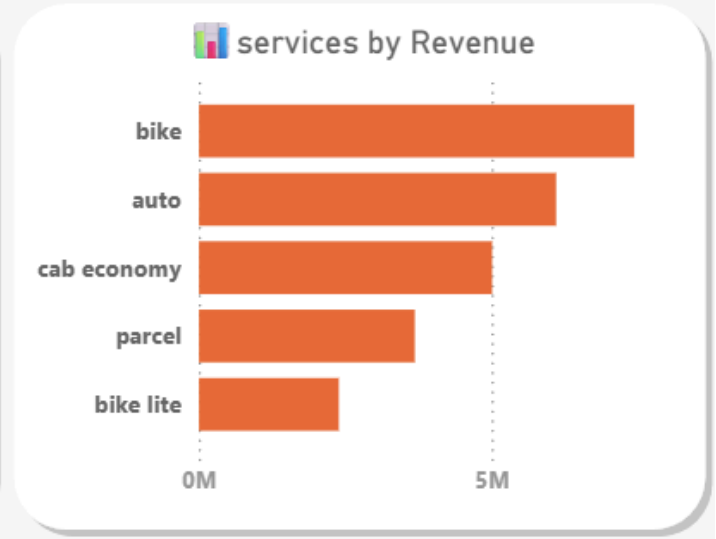
Prince kumar yadav

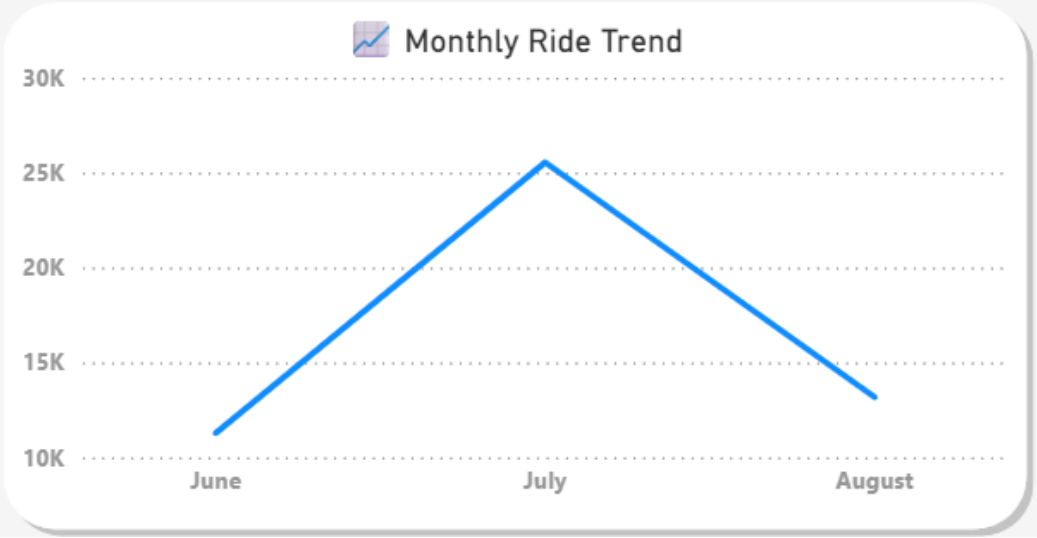
Rides Analysis

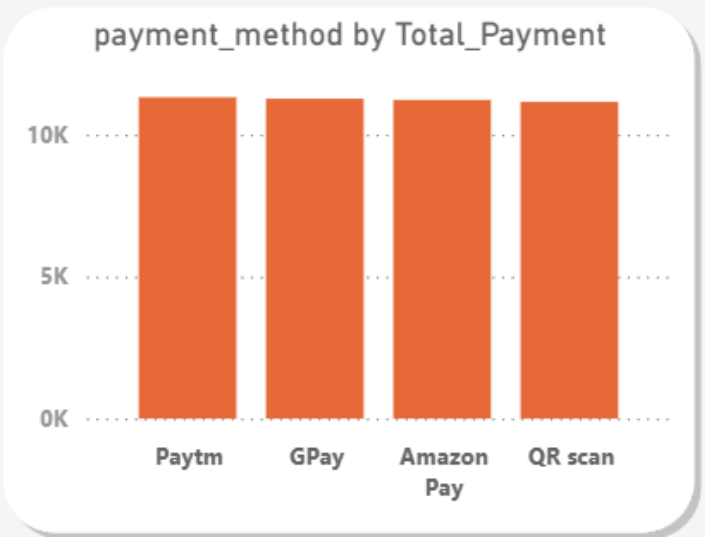
Total_Rides	Total_Revenue	Completed_Rides	%Completed_Rides
50K	24.6M	45K	89.9%

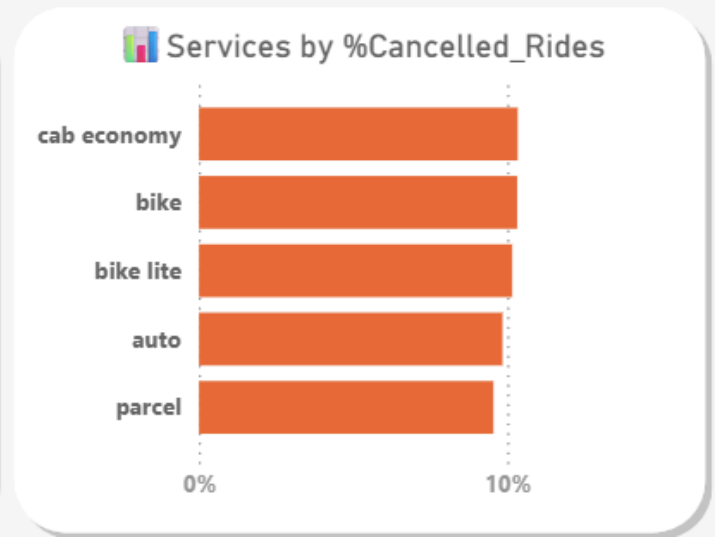


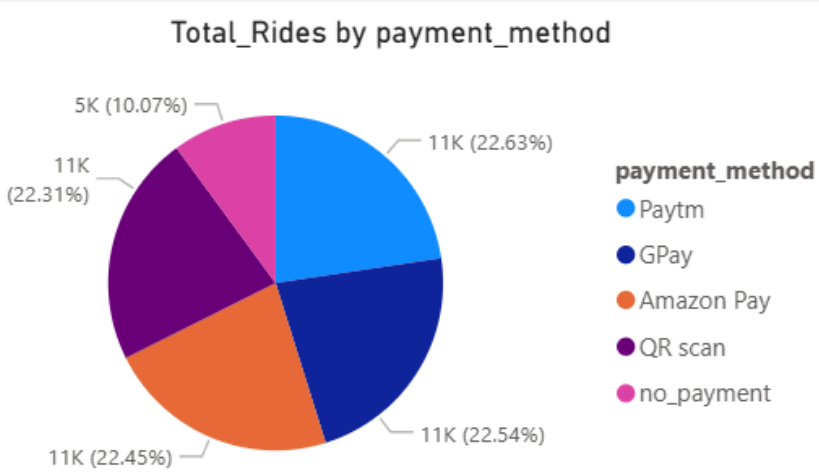












payment_method

- [ ] Amazon Pay
- [ ] GPay
- [ ] no_payment
- [ ] Paytm
- [ ] QR scan

services

- [ ] auto
- [ ] bike
- [ ] bike lite
- [ ] cab economy
- [ ] parcel

Hour

- [ ] 0
- [ ] 1
- [ ] 2
- [ ] 3
- [ ] 4
- [ ] 5
- [ ] 6

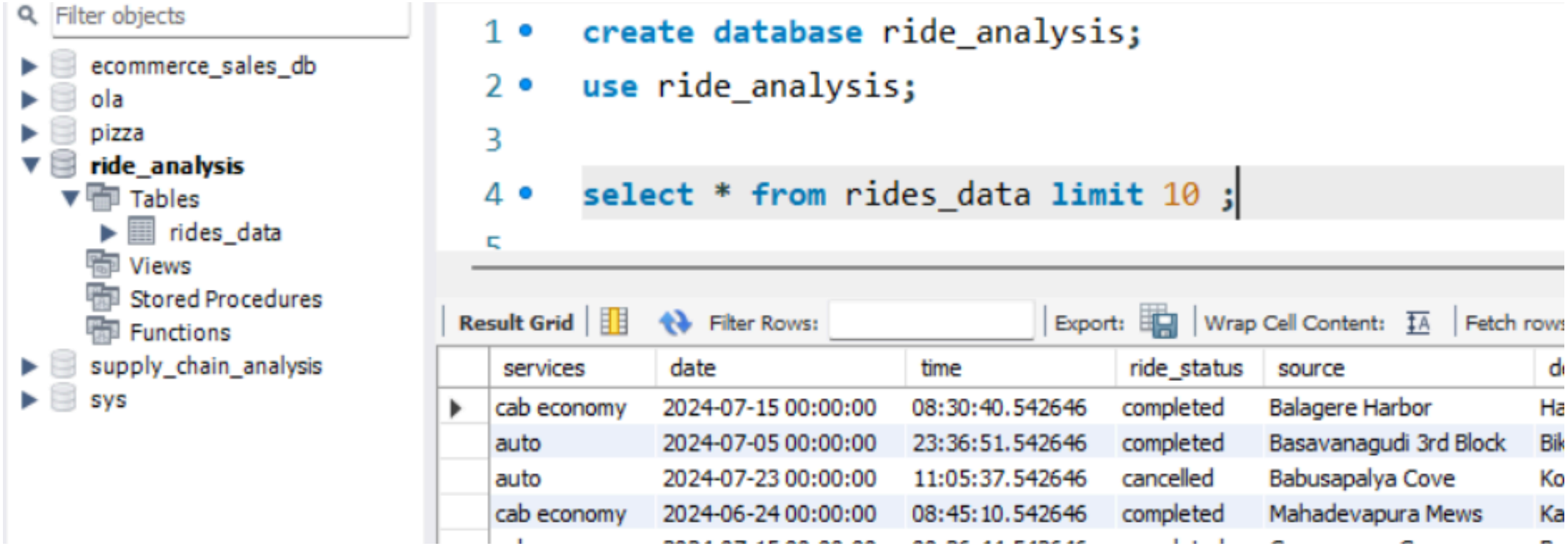
Max Rides by Source

source	destination	Total_Rides	Average_Fare_per_Completed_Ride	Total_Revenue
Anekal Woods	Yelahanka Complex	2	328.8	657.6
Basaveshwaranagar Place	Kalena Agrahara Fork	2	504.0	1008.1
HSR Layout Crescent	Tavarekere View	2	704.2	704.2
Jayanagar Cut	Yelahanka Landing	2	411.3	822.6
Kundalahalli 6th Stage	Gottigere 5th Stage	2	808.2	1616.4
Magadi Road 5th Stage	Rachenahalli Landing	2	287.2	574.4
Naganathapura Close	Koramangala 7th Block Vista	2	692.2	1384.4
Total		**14**	**520.6**	**6767.7**

ride_status

- [ ] cancelled
- [ ] completed

```python
import pandas as pd
import numpy as np

# Load the Dataset

df = pd.read_csv('rides_data.csv')

# Convert date

df['date'] = pd.to_datetime(df['date'], errors='coerce')


# Convert columns to proper format

numeric_cols = ['duration' , 'distance' , 'ride_charge' , 'misc_charge' , 'total_fare' ]
df[numeric_cols] = df[numeric_cols].apply(pd.to_numeric, errors='coerce')

# Replace NaN

cancelled_ride = df['ride_status'].str.lower() == 'cancelled'
df.loc[cancelled_ride, ['ride_charge', 'misc_charge', 'total_fare']] = 0

df['payment_method'] = df['payment_method'].fillna('no_payment')

# Add calculated columns

df['duration_hours'] = df['duration']/60
df['avg_speed_kmph'] = df['distance']/df['duration_hours']

df.to_csv("clean_rides_data.csv", index=False)
print("\n✅ Cleaned data saved as 'cleaned_rides_data.csv'")
```


Filter objects
ecommerce_sales_db
ola
pizza
ride_analysis
Tables
rides_data
Views
Stored Procedures
Functions
supply_chain_analysis
sys
1 • create database ride_analysis;
2 • use ride_analysis;
3
4 • select * from rides_data limit 10 ;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Fetch rows
services
date
time
ride_status
source
cab economy
2024-07-15 00:00:00
08:30:40.542646
completed
Balagere Harbor
auto
2024-07-05 00:00:00
23:36:51.542646
completed
Basavanagudi 3rd Block
auto
2024-07-23 00:00:00
11:05:37.542646
cancelled
Babusapalya Cove
cab economy
2024-06-24 00:00:00
08:45:10.542646
completed
Mahadevapura Mews

-- KPIs --

```
7
8         -- Total Rides --
9
10 •    SELECT
11          COUNT(*) AS total_rides
12      FROM
13          rides_data;
14
```

Result Grid | Filter Rows: | Export:

total_rides
50000

```sql
-- Completed vs Cancelled Rides --

SELECT
    ride_status , COUNT(*) AS total_rides
FROM
    rides_data
GROUP BY 1;
```

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

ride_status	total_rides
completed	44964
cancelled	5036

```sql
-- Total Revenue (only completed rides) --

SELECT
    round(SUM(total_fare),2) AS total_revenue
FROM
    rides_data
WHERE
    ride_status = 'completed';
```

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

total_revenue
24612983.05

```sql
-- Average Fare per Ride --

SELECT
    round(AVG(total_fare),2) AS avg_fare
FROM
    rides_data
WHERE
    ride_status = 'completed';
```

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

avg_fare
547.39

```sql
-- 2. Service Type Analysis --
SELECT
    services,
    COUNT(*) AS total_rides,
    ROUND(SUM(total_fare), 2) AS total_revenue,
    ROUND(AVG(total_fare), 2) AS avg_fare,
    ROUND(AVG(distance), 2) AS avg_distance,
    SUM(CASE
        WHEN ride_status = 'cancelled' THEN 1
        ELSE 0
    END) AS cancelled_rides
FROM
    rides_data
GROUP BY 1
ORDER BY total_revenue DESC;
```

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

services	total_rides	total_revenue	avg_fare	avg_distance	cancelled_rides
bike	15128	7432783.95	491.33	25.39	1561
auto	12327	6099731.32	494.83	25.6	1213
cab economy	10202	5006233.04	490.71	25.5	1054
parcel	7459	3686514.15	494.24	25.64	712
bike lite	4884	2387720.59	488.89	25.7	496

```sql
-- 3. Payment Method Insights --

SELECT
    payment_method,
    COUNT(*) AS total_orders,
    ROUND(SUM(total_fare), 2) AS total_revenue
FROM
    rides_data
GROUP BY 1
ORDER BY total_revenue DESC;
```

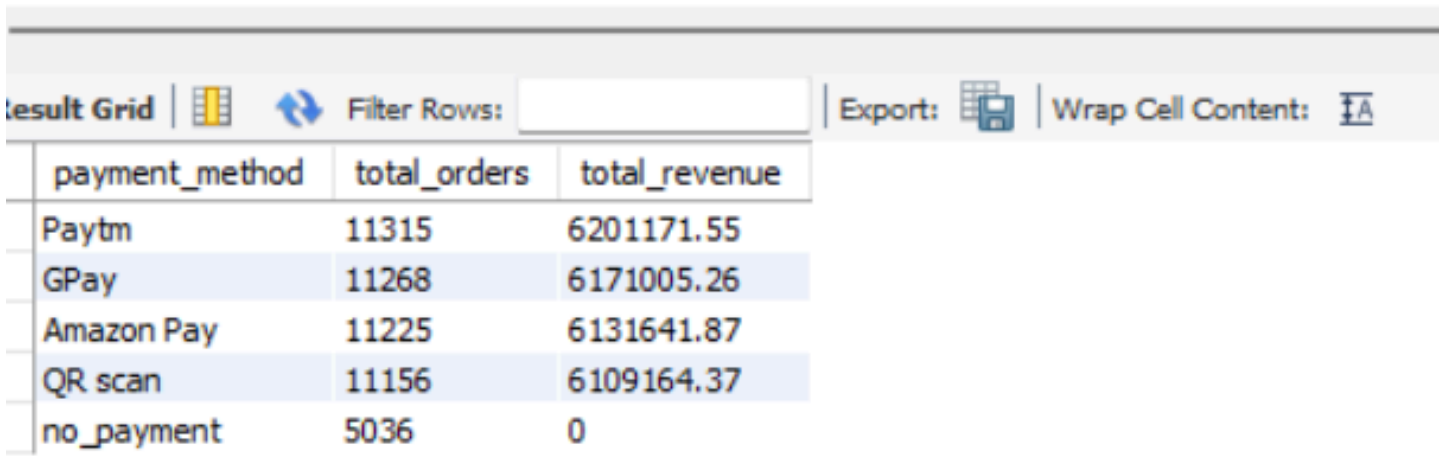
payment_method	total_orders	total_revenue
Paytm	11315	6201171.55
GPay	11268	6171005.26
Amazon Pay	11225	6131641.87
QR scan	11156	6109164.37
no_payment	5036	0

```sql
-- 4. Top Routes --
SELECT
    source,
    destination,
    COUNT(*) AS rides,
    ROUND(AVG(total_fare), 2) AS avg_fare,
    ROUND(AVG(distance), 2) AS avg_distance,
    ROUND(AVG(duration), 2) AS avg_duration
FROM
    rides_data
WHERE
    ride_status = 'completed'
GROUP BY 1 , 2
ORDER BY rides DESC
LIMIT 10;
```

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

source	destination	rides	avg_fare	avg_distance	avg_duration
Anekal Woods	Yelahanka Complex	2	328.78	29.22	18.50
Magadi Road 5th Stage	Rachenahalli Landing	2	287.17	26.55	83.50
Kundalahalli 6th Stage	Gottigere 5th Stage	2	808.2	33.14	53.50
Naganathapura Close	Koramangala 7th Block Vista	2	692.21	14.61	31.00
Jayanagar Cut	Yelahanka Landing	2	411.31	23.62	65.00
Basaveshwaranagar Place	Kalena Agrahara Fork	2	504.03	41.68	95.50

```sql
-- 3. Payment Method Insights --

SELECT
    payment_method,
    COUNT(*) AS total_orders,
    ROUND(SUM(total_fare), 2) AS total_revenue
FROM
    rides_data
GROUP BY 1
ORDER BY total_revenue DESC;
```



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

payment_method	total_orders	total_revenue
Paytm	11315	6201171.55
GPay	11268	6171005.26
Amazon Pay	11225	6131641.87
QR scan	11156	6109164.37
no_payment	5036	0