Uber Supply-Demand Gap Analysis

Name: Prajaktha Poojary

Tools Used: Excel, SQL, Python (Pandas, Seaborn)

Date: 22-06-2025

# 1. Project Objective

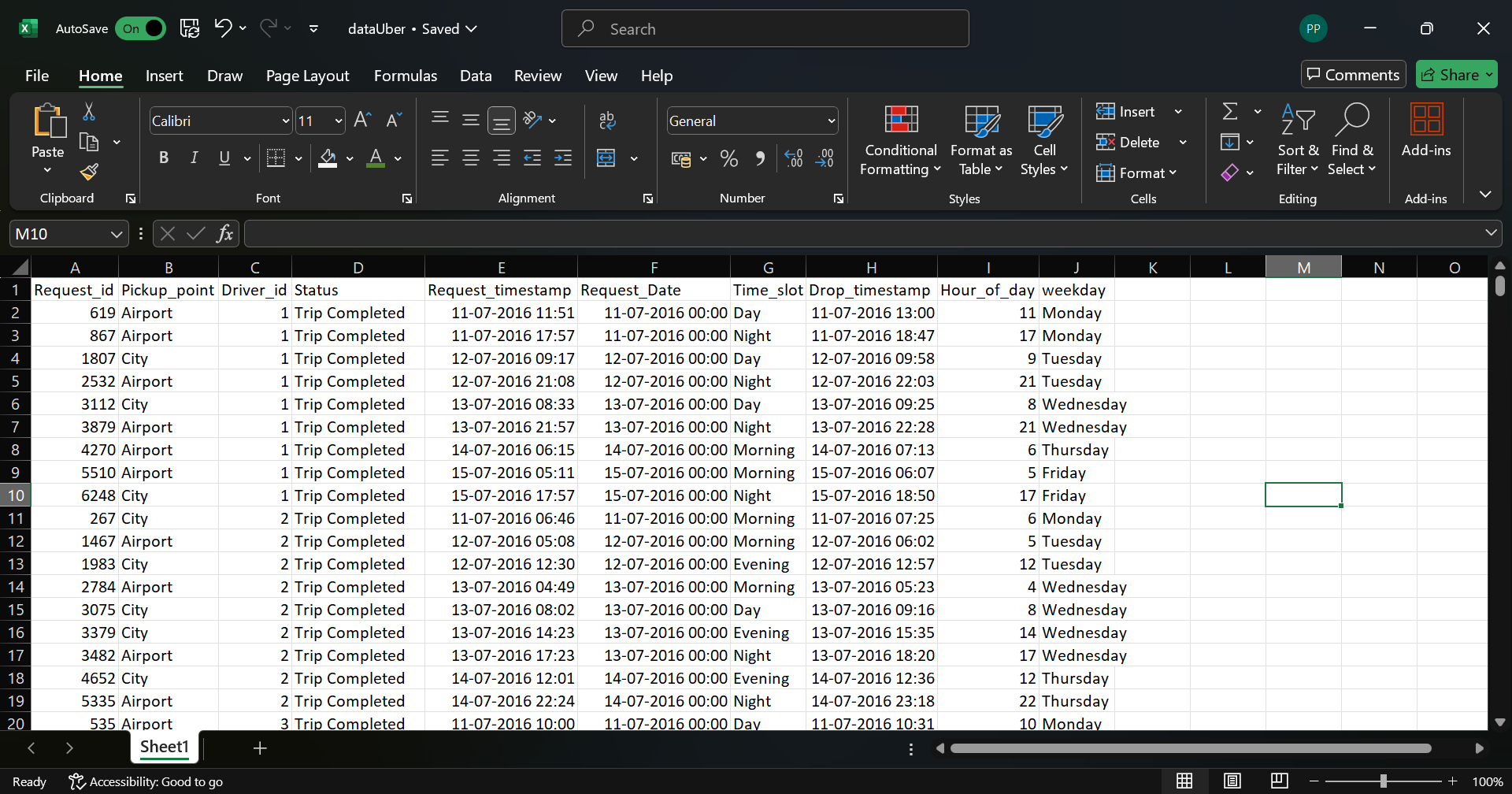
The main goal of this project is to analyze Uber ride request data to identify gaps between supply and demand using Excel, SQL, and Python. The goal is to derive actionable insights to help Uber improve driver allocation and reduce ride failures.

# 2. Excel Summary

File name: uber\_excel\_cleaned.xlsx

**🧼 Data Cleaning Done in Excel:**

* Removed or fixed missing values (like Drop Timestamps for cancelled rides)
* Converted Request\_timestamp and Drop\_timestamp to proper datetime format
* Extracted new columns using formulas:
  + Hour\_of\_day using =HOUR()
  + Weekday using =TEXT(date,"dddd")
  + Time\_slot using IF() and nested conditions



**📊 Pivot Tables Created:**

* **Status by Pickup Point** – to see how many rides were completed, cancelled, or had no cars from City and Airport
* **Hourly Requests by Status** – to identify peak hours for each status
* **Weekday-wise Request Volume** – to check daily demand trends
* **Time Slot-wise Requests** – to analyze demand distribution across morning, evening, night, etc.

**📈 Dashboard Features:**

* Added all charts in a single sheet called Dashboard
* Used slicers for **Pickup Point** and **Weekday** to make the dashboard interactive
* Aligned charts like:
  + Bar Chart (Ride Status Distribution)
  + Pie Chart (Pickup Point Split)
  + Column Chart (Hourly Demand)
  + Line Chart (Weekday Trends)

**💡 Insights from Excel Dashboard:**

* Most ride failures happened at the **Airport during night**
* **Morning hours (7 AM – 10 AM)** saw the highest number of cancellations
* **Weekdays have more ride requests** than weekends

# 3. SQL Summary

- File name: uber\_queries.sql

Table Name: uber\_requests

Key Queries:

**📥 What I Did:**

* Imported the cleaned CSV (cleanedReport.csv) into MySQL
* Created a table named uber\_requests with appropriate data types
* Used SQL queries to explore patterns in ride request data

**📌 Important SQL Queries Used:**

* **Total number of requests by Status**

SELECT Status, COUNT(\*) FROM uber\_requests GROUP BY Status;

* **Failed requests (Cancelled/No Cars) by Hour of Day**

SELECT Hour\_of\_day, COUNT(\*)

FROM uber\_requests

WHERE Status IN ('Cancelled', 'No Cars Available')

GROUP BY Hour\_of\_day;

* **Sql query for Pickup Point-wise No Cars Available count**

SELECT Pickup\_point, COUNT(\*)

FROM uber\_requests

WHERE Status = 'No Cars Available'

GROUP BY Pickup\_point;

* **Ride status distribution on each weekday**

SELECT weekday, Status, COUNT(\*)

FROM uber\_requests

GROUP BY weekday, Status;

**💡 Insights from SQL Analysis:**

* Most cancellations happen between **7 AM and 10 AM**
* **Airport** has the highest number of “No Cars Available” responses
* **Weekdays** show a higher number of failed rides compared to weekends

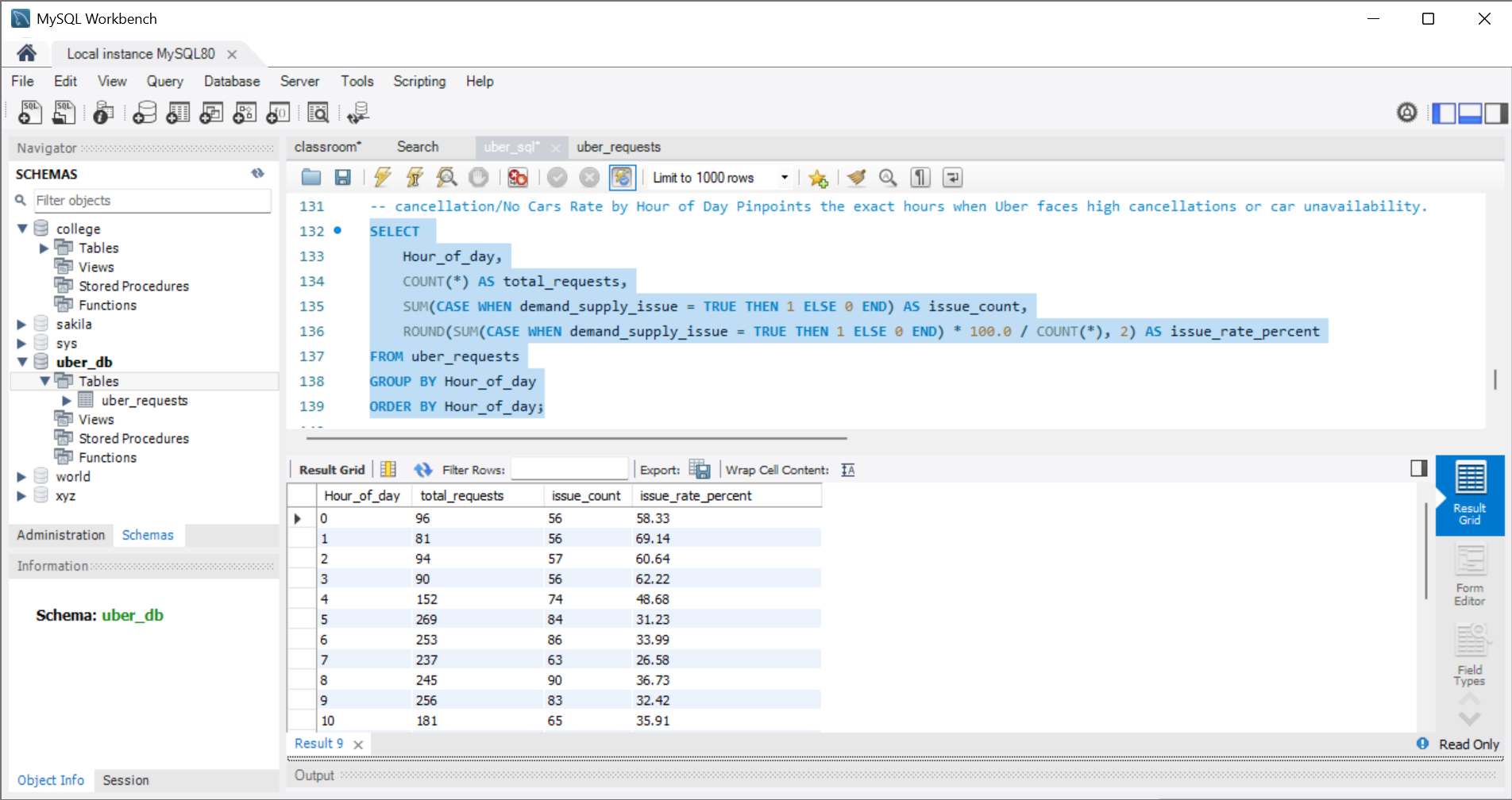


Fig: Cancellation/No cars rate by hour of day

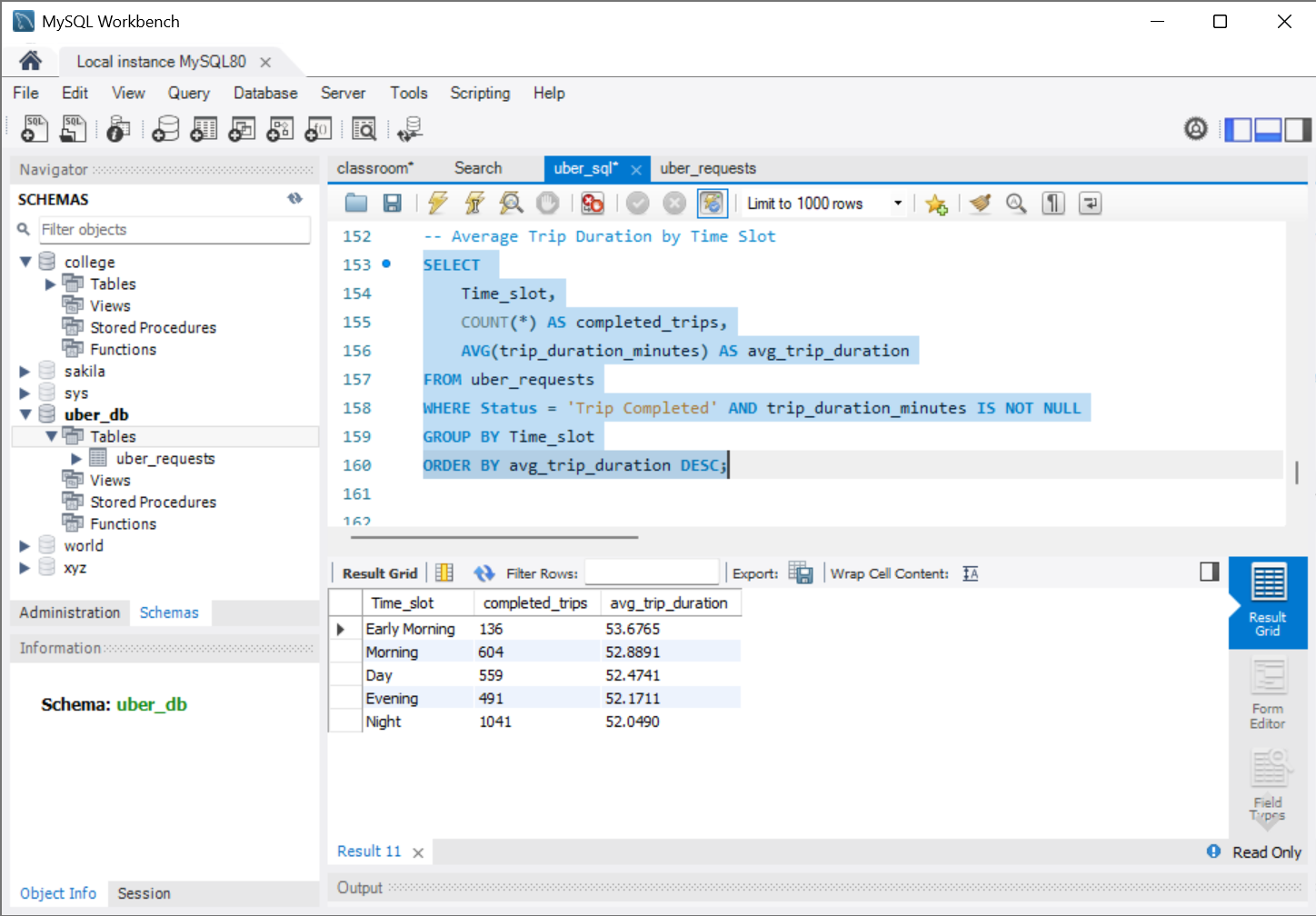


Fig:average Trip Duration by Time Slot

# 4. Python EDA Summary

**File name:** Uber\_EDA\_UBM\_Visualizations.ipynb  
**Tools Used:** Pandas, Seaborn, Matplotlib, NumPy

**🧹 Data Cleaning & Feature Engineering:**

* Loaded the cleaned dataset using pandas.read\_csv()
* Converted Request\_timestamp and Drop\_timestamp to datetime objects
* Created new columns for better analysis:
  + Hour\_of\_day – extracted using .dt.hour
  + Weekday – using .dt.day\_name()
  + Trip\_duration – calculated from request and drop time (for completed trips)
  + is\_weekend – flagged Saturday/Sunday
  + Issue – flagged cancelled or “No Cars Available”
  + Time\_slot\_filled – filled missing/unavailable time slots based on hour

**📈 Charts Created:**

* Total of **20+ charts** categorized under:
  + **Univariate Analysis** (e.g., Status, Hour, Pickup Point)
  + **Bivariate Analysis** (e.g., Status vs Hour, Pickup vs Issue)
  + **Multivariate Analysis** (e.g., Status vs Hour vs Pickup Point, Heatmaps)

**💡 Insights from Python EDA:**

* Peak demand is during **Morning (7–10 AM)** and **Evening (5–8 PM)**
* **Airport** faces the highest “No Cars Available” issues, especially at Night
* **Trip durations** are longer from the Airport and during Day/Evening slots
* **Most ride failures** happen on **weekdays during morning rush hours**

**📊 Libraries Used:**

* pandas for data wrangling
* seaborn and matplotlib.pyplot for visualizations
* warnings and datetime for data handling

# 5. Recommendations

1. Increase night-shift drivers, especially at the Airport  
2. Offer bonuses for early morning and weekend availability  
3. Introduce predictive driver positioning based on historical peaks  
4. Improve app messaging during high-failure time slots

# 6. Conclusion

This project successfully combines Excel, SQL, and Python to deliver end-to-end data analysis for Uber. The insights clearly show where Uber faces demand-supply mismatches and provide practical recommendations to improve service and user satisfaction.