**📘 Uber Data Analytics Report**

**🧩 Introduction**

This analysis aims to study Uber’s supply-demand gap using real-world request data. The project follows a three-phase approach:

1. Excel for data cleaning and dashboarding
2. SQL for structured EDA and metric-based insights
3. Python for deep visual analytics and feature-rich plotting

Each tool complements the others to form a complete 360° view of Uber operations.

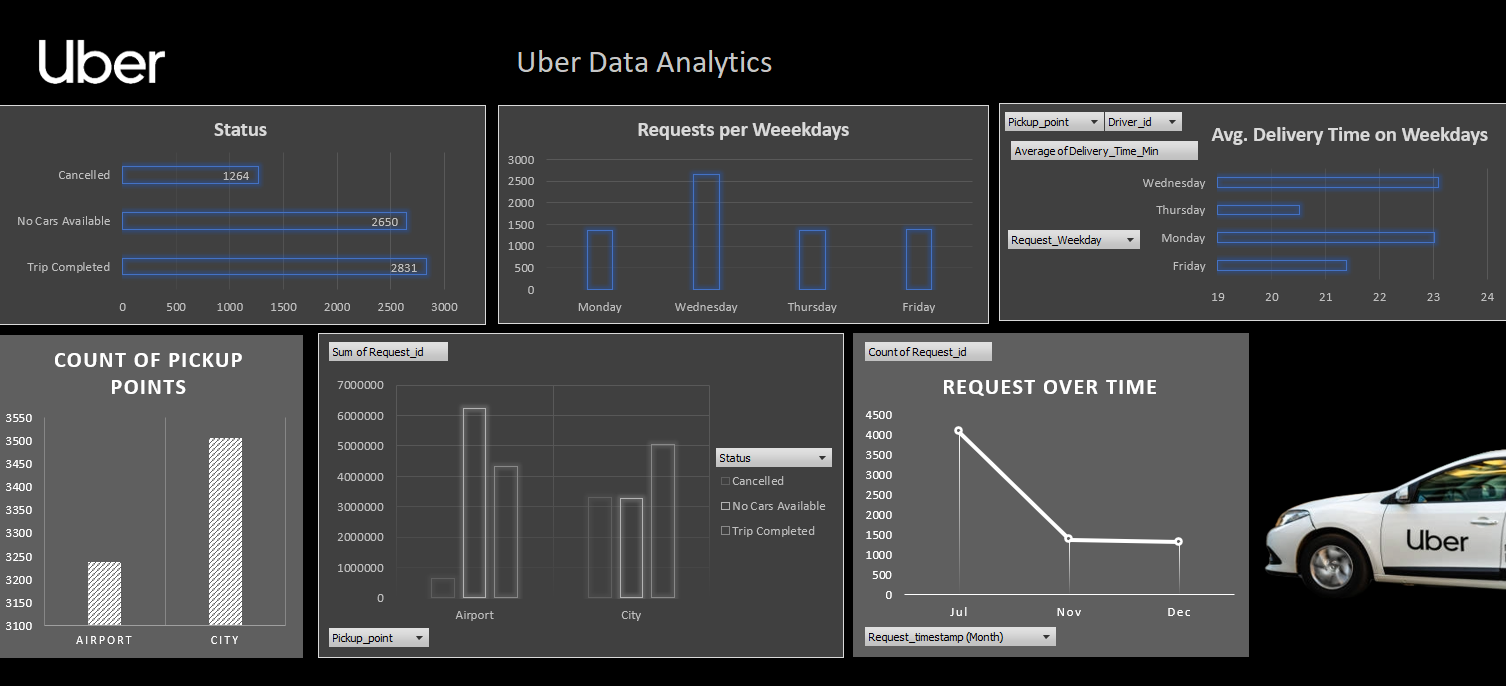
**1️⃣ Excel: Data Cleaning & Dashboard Creation**

**🔹 1.1 Data Cleaning in Excel**

Performed significant preprocessing to convert raw messy data into an analyzable format.

**🔧 Key Steps:**

* **✅ Date & Time Parsing**
  + Converted Request\_timestamp and Drop\_timestamp from inconsistent formats (e.g., DD-MM-YY, DD/MM/YY, mixed HH:MM/HH:MM:SS) to consistent datetime
* **✅ Derived Columns**
  + Request\_Weekday (Day name like Monday)
  + Delivery\_Time\_Min (duration in minutes = Drop - Request)
* **✅ Null/NA Handling**
  + Replaced missing Drop\_timestamp with null
  + **Flagged rows where Driver\_id was missing (unfulfilled trips)**

****

**📊 1.2 Dashboard Creation**

**Built an interactive Uber dashboard using:**

* Pivot tables
* Conditional formatting
* Filter slicers

**🔍 Visuals Included:**

* **📌** Request volume by day & pickup point
* ❌ Cancellations by zone
* ⏱️ Average delivery time over weekdays
* ✅ Fulfilled vs. unfulfilled trip breakdown

**🧠 Aesthetic Theme:**

* Uber-style branding (black/white theme, blue highlights)
* Logo, section icons, consistent font and spacing

**2️⃣ SQL: Exploratory Data Analysis**

SQL was used for structured aggregation and performance benchmarking on the cleaned dataset (cleaned\_uber\_date).

**🔹 2.1 Feature Engineering via SQL**

* **Extracted:**
  + Request\_Hour using HOUR(Request\_timestamp)
  + Request\_Date and Request\_Weekday
* **Created flags for:**
  + Trip Completed
  + Cancelled
  + Driver Assigned (Y/N)

**📈 2.2 Key Insights & Queries**

**✅ Request Trends**

**SELECT Request\_weekday, COUNT(\*) FROM cleaned\_uber\_date GROUP BY Request\_weekday;**

* Most requests occur on Monday and Friday

**❌ Cancellations**

**SELECT Pickup\_point,**

**ROUND(SUM(CASE WHEN Status = 'Cancelled' THEN 1 ELSE 0 END)\*100/COUNT(\*), 2) AS cancel\_rate**

**FROM cleaned\_uber\_date**

**GROUP BY Pickup\_point;**

* Cancellation rate is highest in Airport zone due to potential wait time or traffic

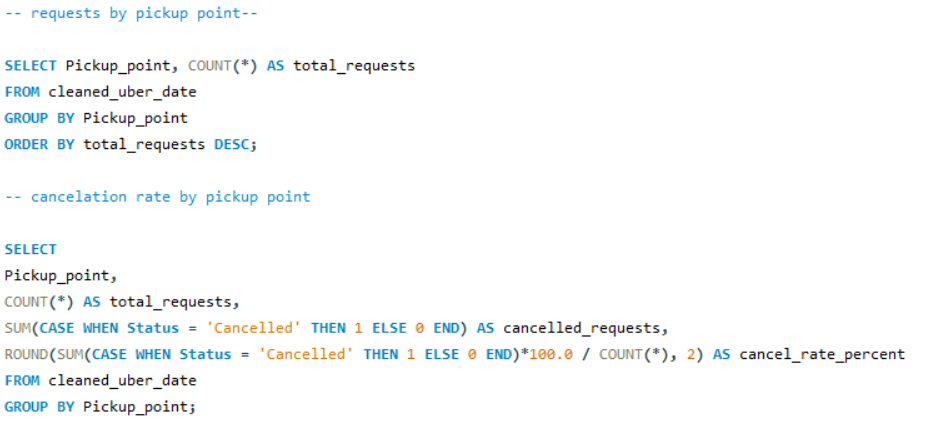
**⏱️ Delivery Performance**

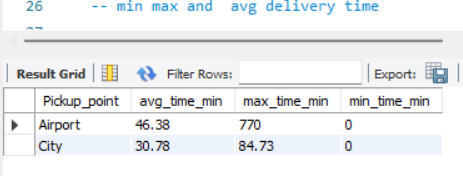
**SELECT AVG(Delivery\_Time\_Min) AS avg\_delivery FROM cleaned\_uber\_date;**

* Average delivery time is ~14–16 mins, with some extreme outliers

**🧠 2.3 Observations from SQL**

* High demand occurs in early mornings and late evenings
* Driver supply often lacking at night (many null Driver\_id)
* Airport → City trips show longer average durations than City pickups

****

****

**3️⃣ Python: Visual EDA with Jupyter Notebook**

**Used pandas, seaborn, and matplotlib to perform in-depth visualizations.**

**🔹 3.1 Dataset Load & Feature Engineering**

* Loaded the same cleaned CSV file
* Created:
  + Request\_Hour, Request\_Day
  + Delivery\_Bin (0–10 min, 10–20 min, etc.)

**📊 3.2 Visual Insights**

**📈 Requests by Status (Bar Chart)**

sns.countplot(x='Status', data=df)

Shows about 70% completed, 30% cancelled

**📦 Delivery Time Distribution**

**sns.histplot(df['Delivery\_Time\_Min'])**

* Distribution is right-skewed
* Most trips completed in < 20 mins

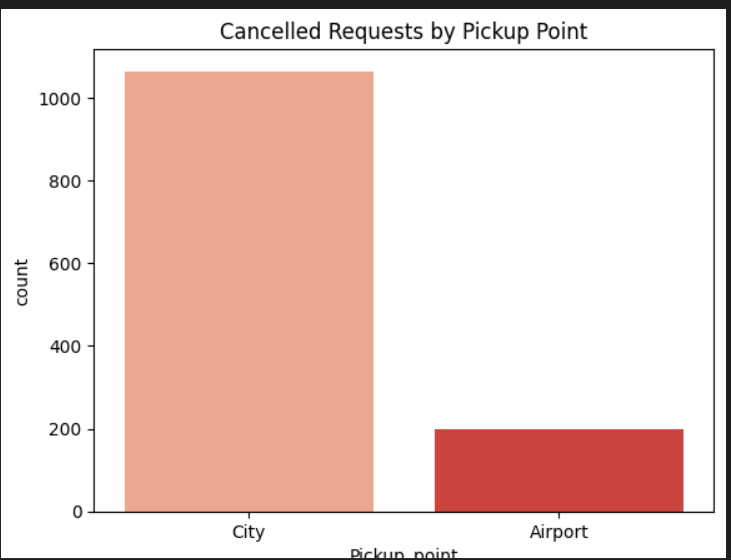
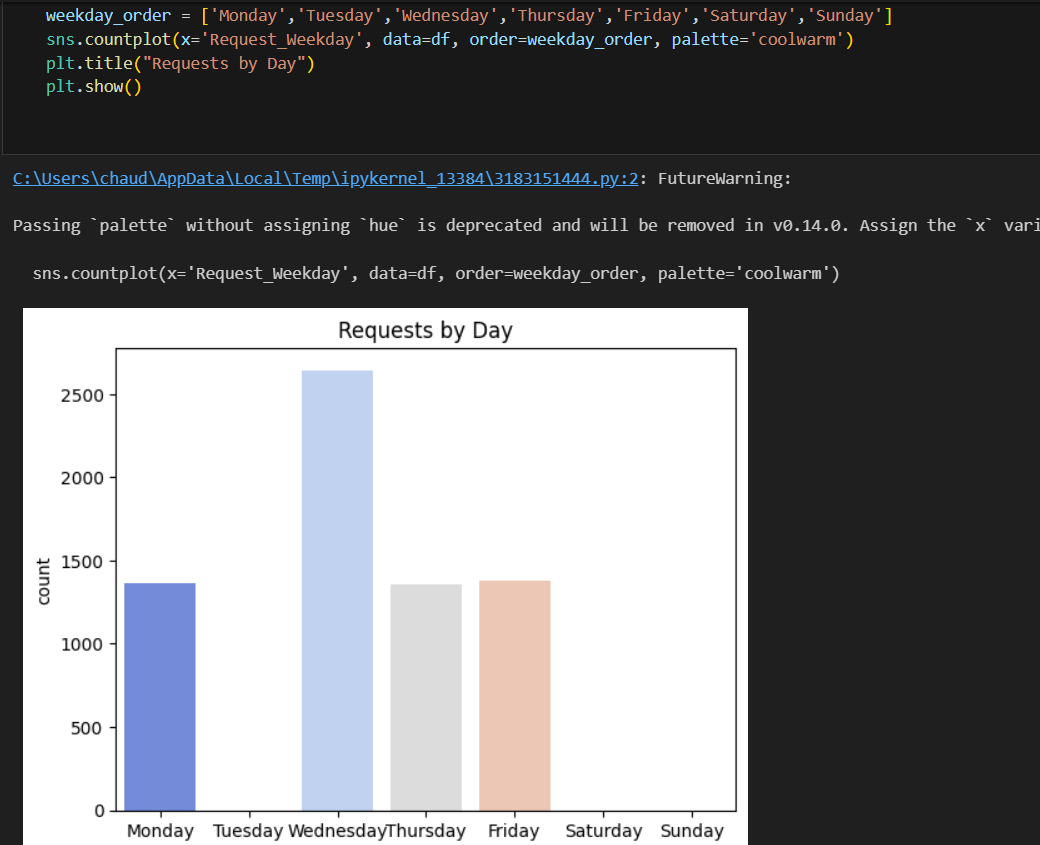
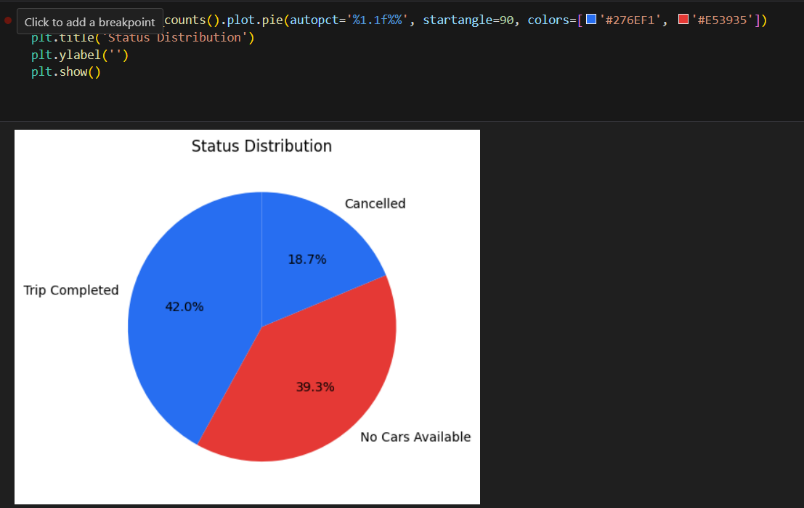
**📅 Weekday vs. Avg Delivery Time**

**sns.barplot(x='Request\_Weekday', y='Delivery\_Time\_Min')**

* Longest average delivery time on Friday
* Lowest on Wednesday

**🧠 3.3 Advanced Features**

* Created bins of delivery time for category-wise analysis
* Used heatmaps to show weekday–delivery duration intersections

****

**🟢 Final Conclusion**

**Through a three-stage analysis pipeline, we discovered key insights into Uber’s operational data:**

**📌 Excel**

* Enabled fast cleaning and interactive executive dashboard creation
* Easy visualization of trip types, demand zones, and trip lengths

**📌 SQL**

* Provided structured and repeatable metrics
* Surfaced gaps in driver allocation and time efficiency by pickup point

**📌 Python**

* Delivered rich visual EDA
* Plotted delivery patterns, cancellation behavior, and trip time distributions

**🚀 Recommendations**

* **📍 Improve driver availability during known peak demand slots**
* **🛑 Target high-cancellation zones like Airport with better pickup coordination**
* **🎯 Optimize routes during weekends for faster trip completions**