



Labmentix

Uber Request Data Insights

Mentor: Mr. Vipul Sonawane

Submitted By: Surabhi Kharkate

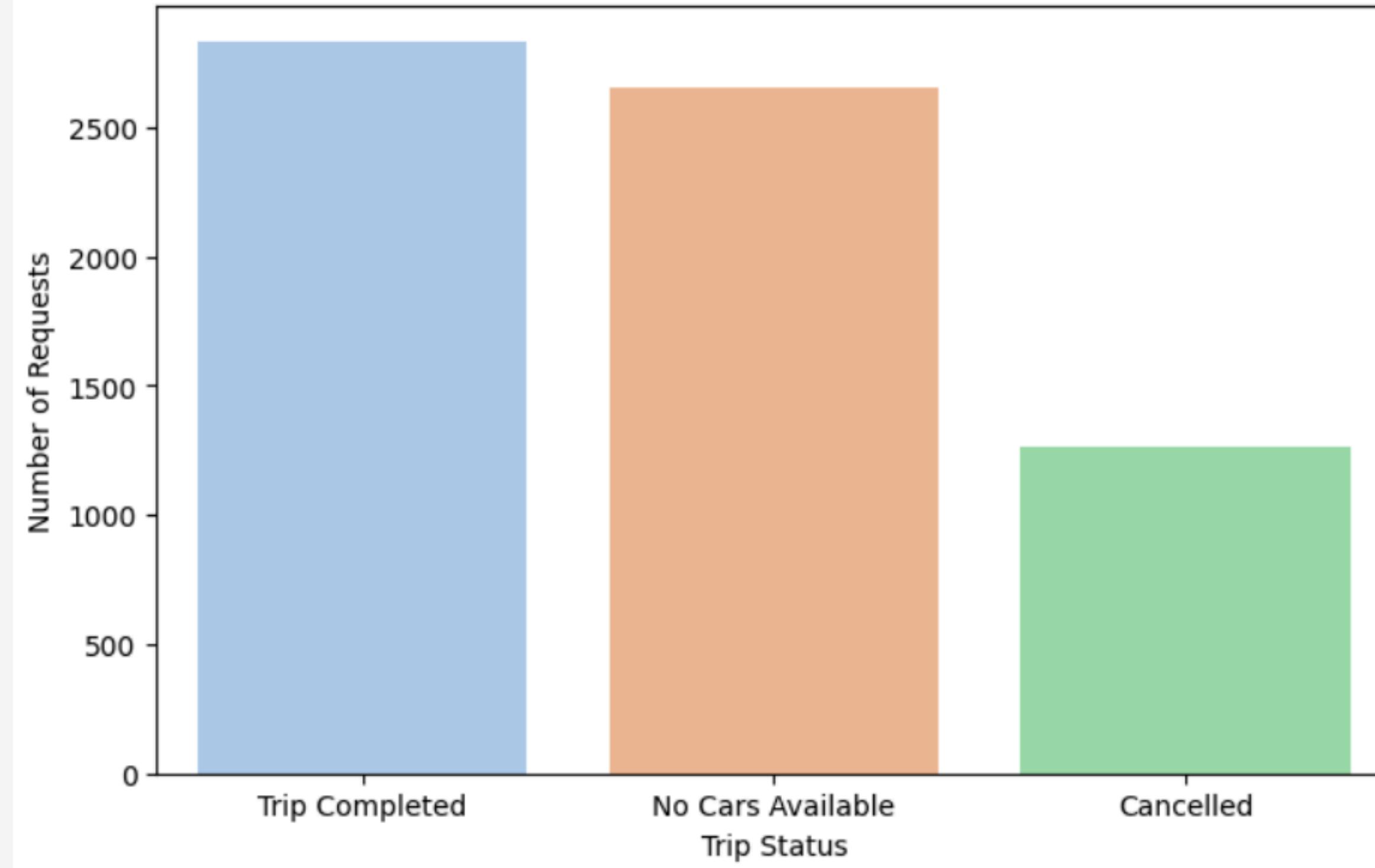
Problem Statement

Uber is facing issues like ride cancellations and driver unavailability, especially during peak hours. These problems reduce customer satisfaction and affect business efficiency. This project aims to analyze Uber's ride request data to uncover patterns in demand, trip status, and pickup locations, helping improve service reliability and resource allocation.

Using EDA

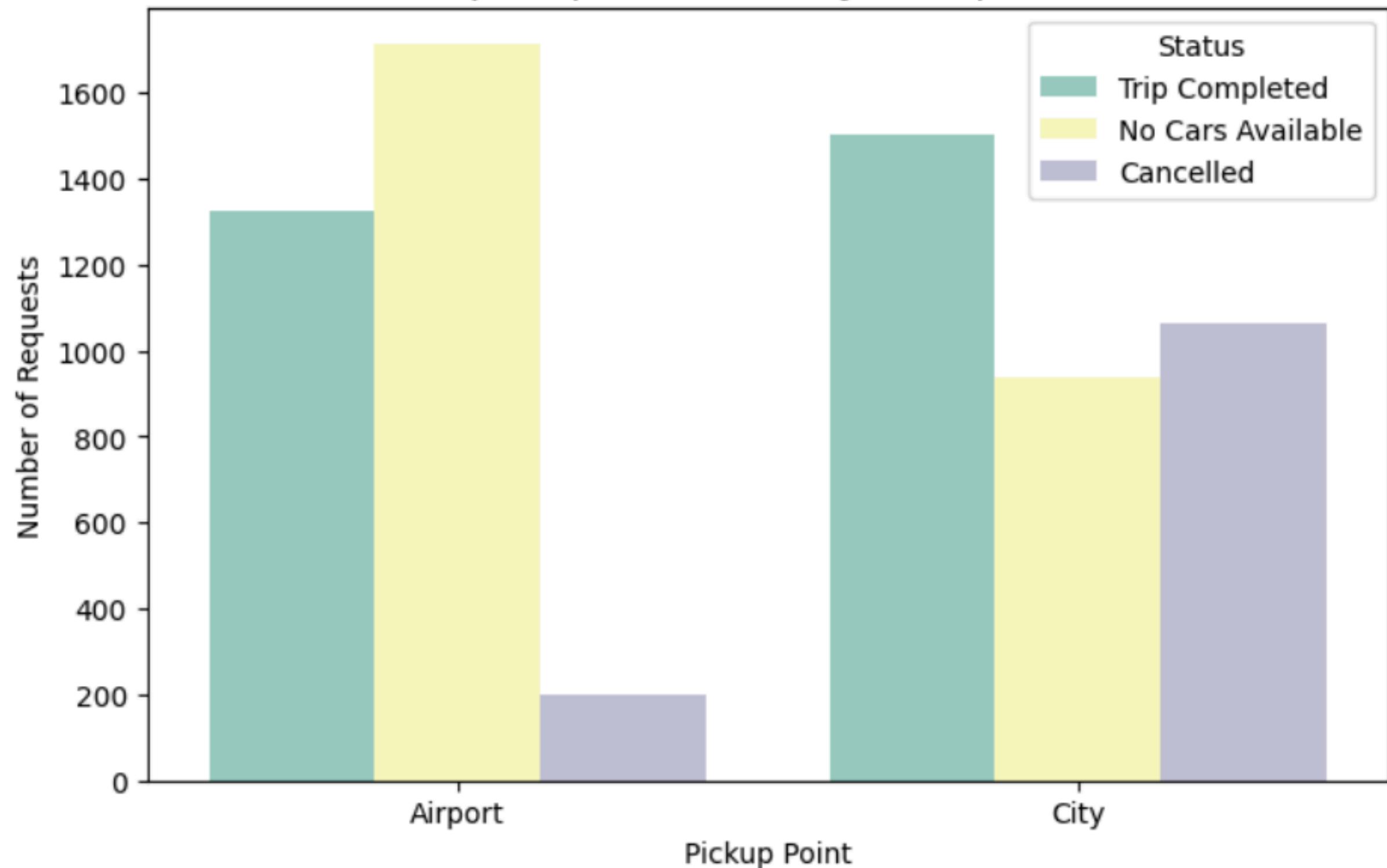
- Goal: Analyze Uber ride data to find trends and service gaps.
- Peak demand: Morning (5–9 AM) & Evening (5–9 PM).
- Failures: Many rides were cancelled or had no drivers in peak hours.
- Pickup trends:
 - City → More cancellations
 - Airport → More No Cars Available
- Trip Duration: Mostly between 20–80 mins.
- Insight: Data helps optimize drivers & reduce lost trips.

Overall Trip Request Status Distribution



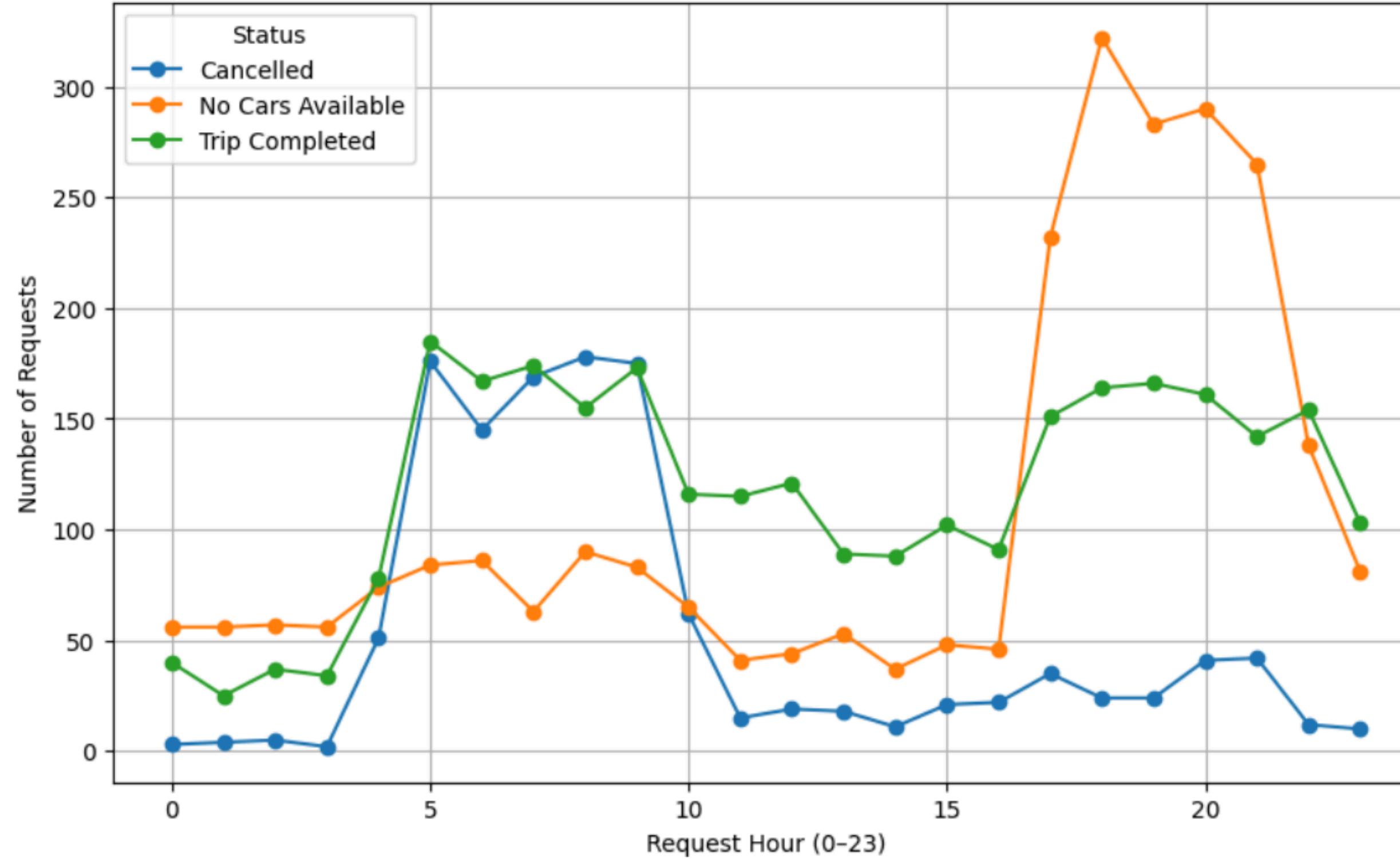
This chart provides a high-level overview of how Uber ride requests end – whether they are completed, cancelled, or unavailable. It's a foundational step to understand overall service performance and operational issues.

Trip Request Status by Pickup Point



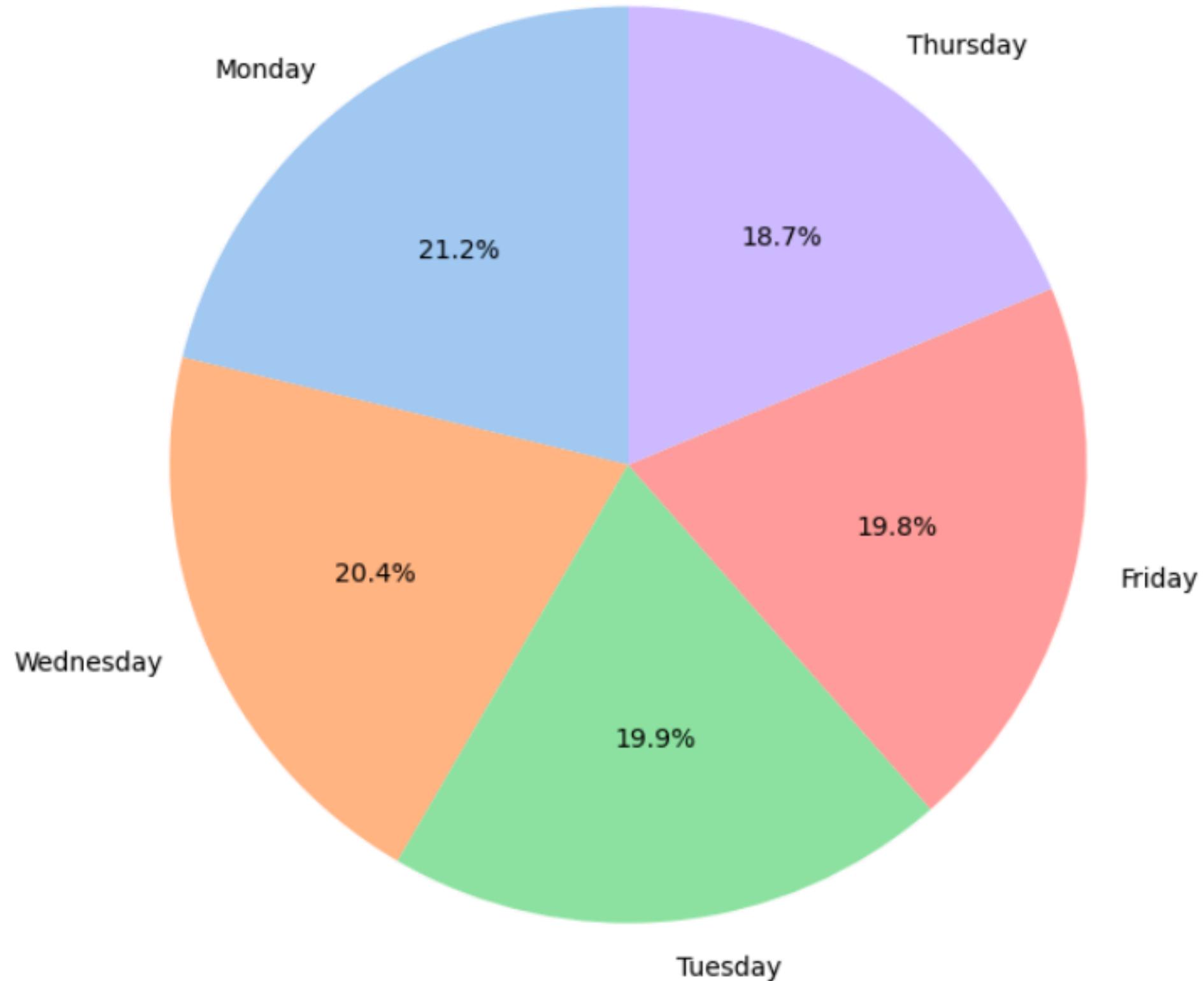
This chart breaks down trip outcomes based on the pickup location (City vs Airport), helping identify where issues are concentrated geographically.

Trip Status Trend Across Hours of the Day



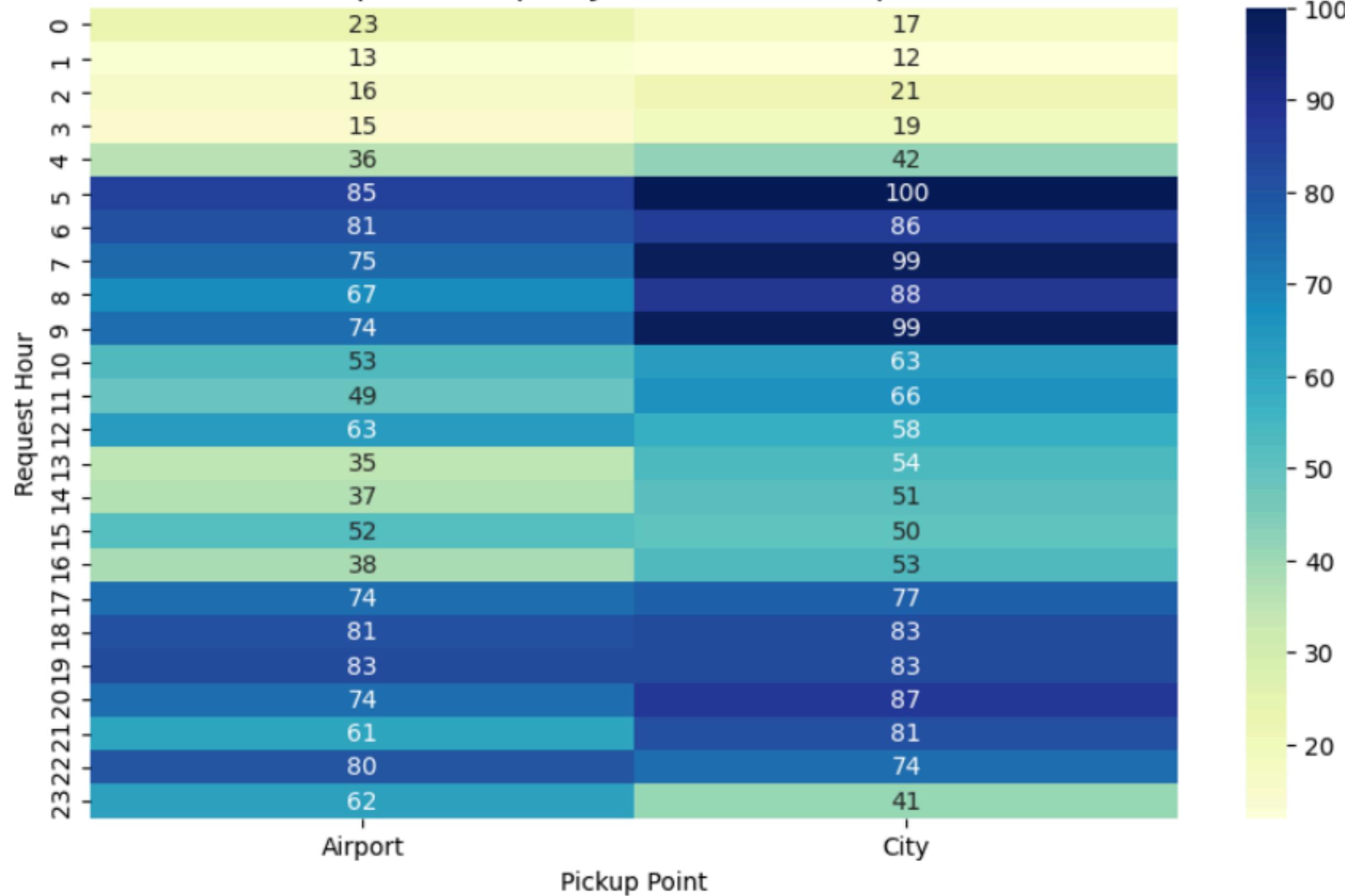
A line chart helps visualize the trend of demand and service status over the course of a day. It's perfect for showing time-based patterns, unlike bar charts that are better for static comparisons.

Trip Completions by Day of the Week

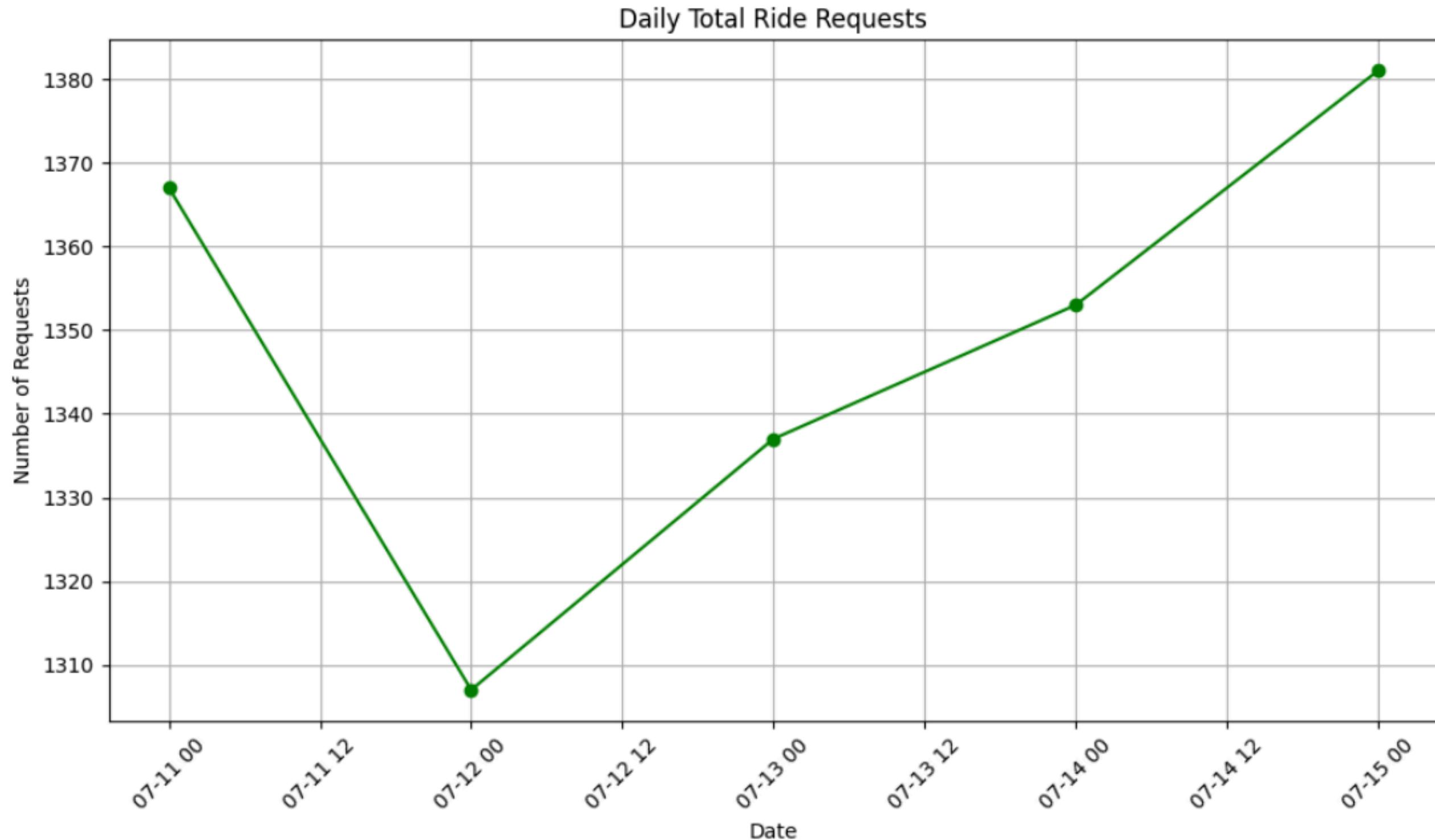


A pie chart is ideal for showing proportions. This visualization reveals which day of the week contributes the most to total trip completions.

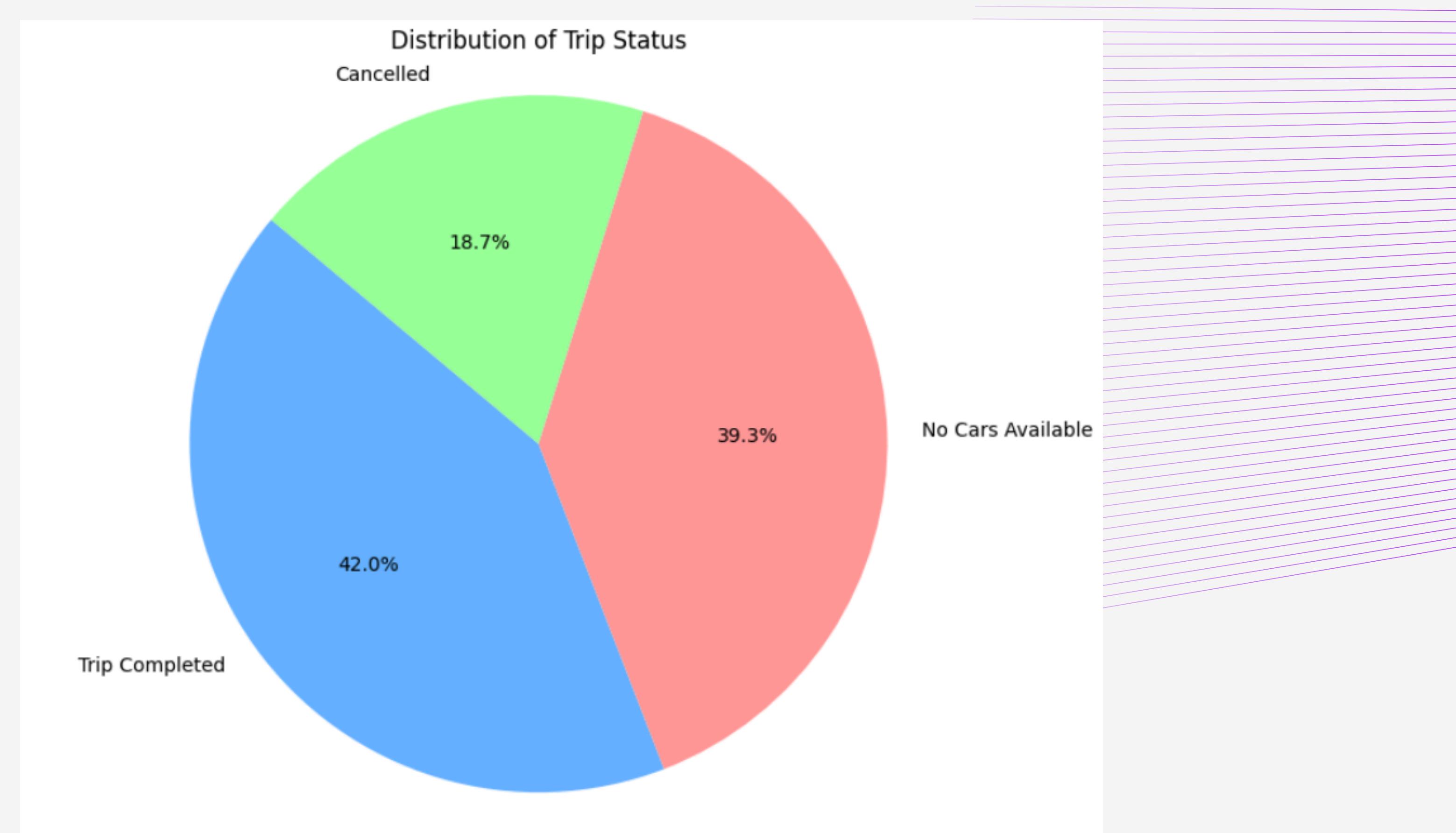
Completed Trips by Hour and Pickup Point



Unlike bar or pie charts, a heatmap allows you to spot when and where successful trips are happening (or not). It combines two variables (hour + pickup location).

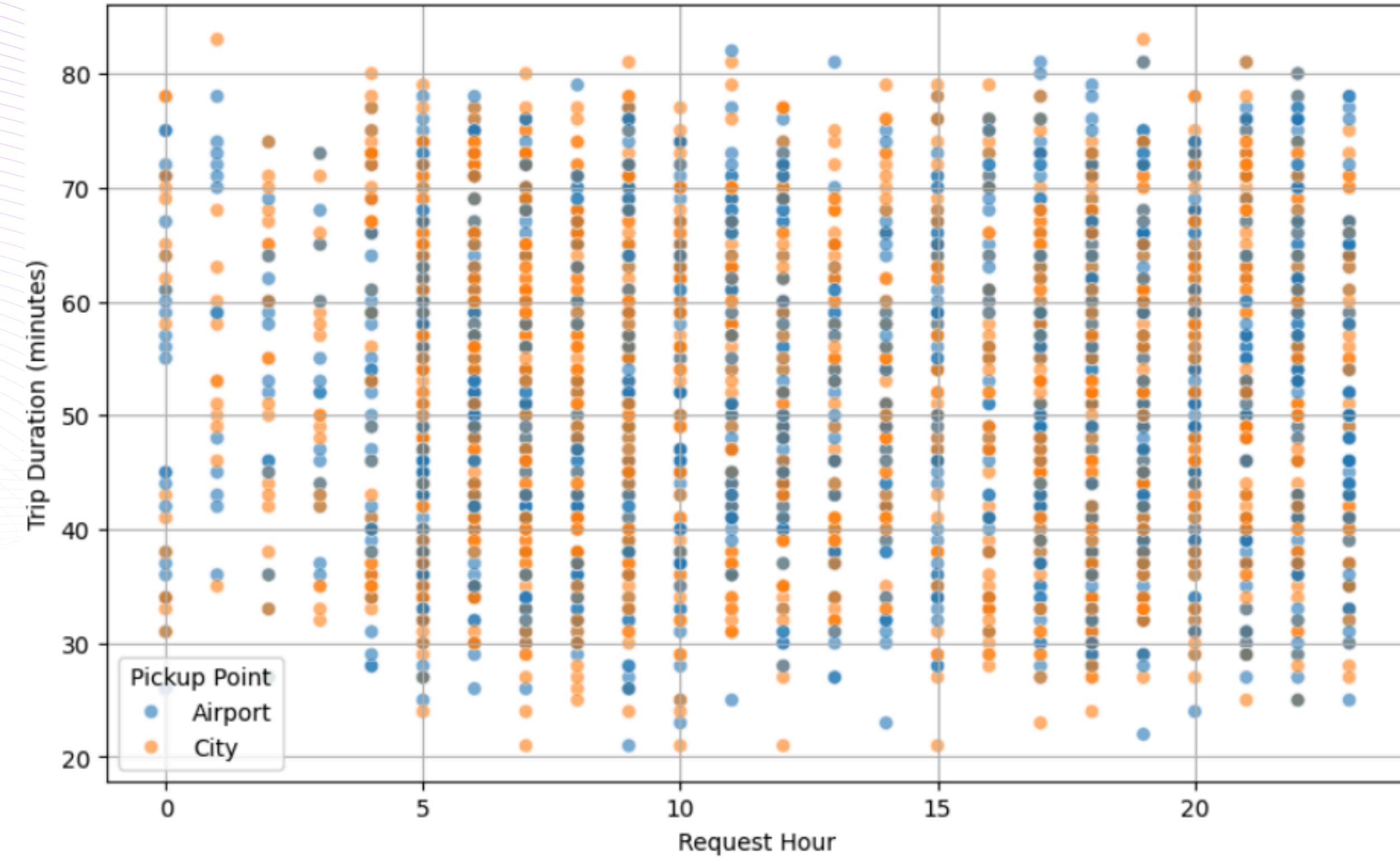


- A line chart clearly shows request patterns over time.
- Helps visualize demand trends and peak days.



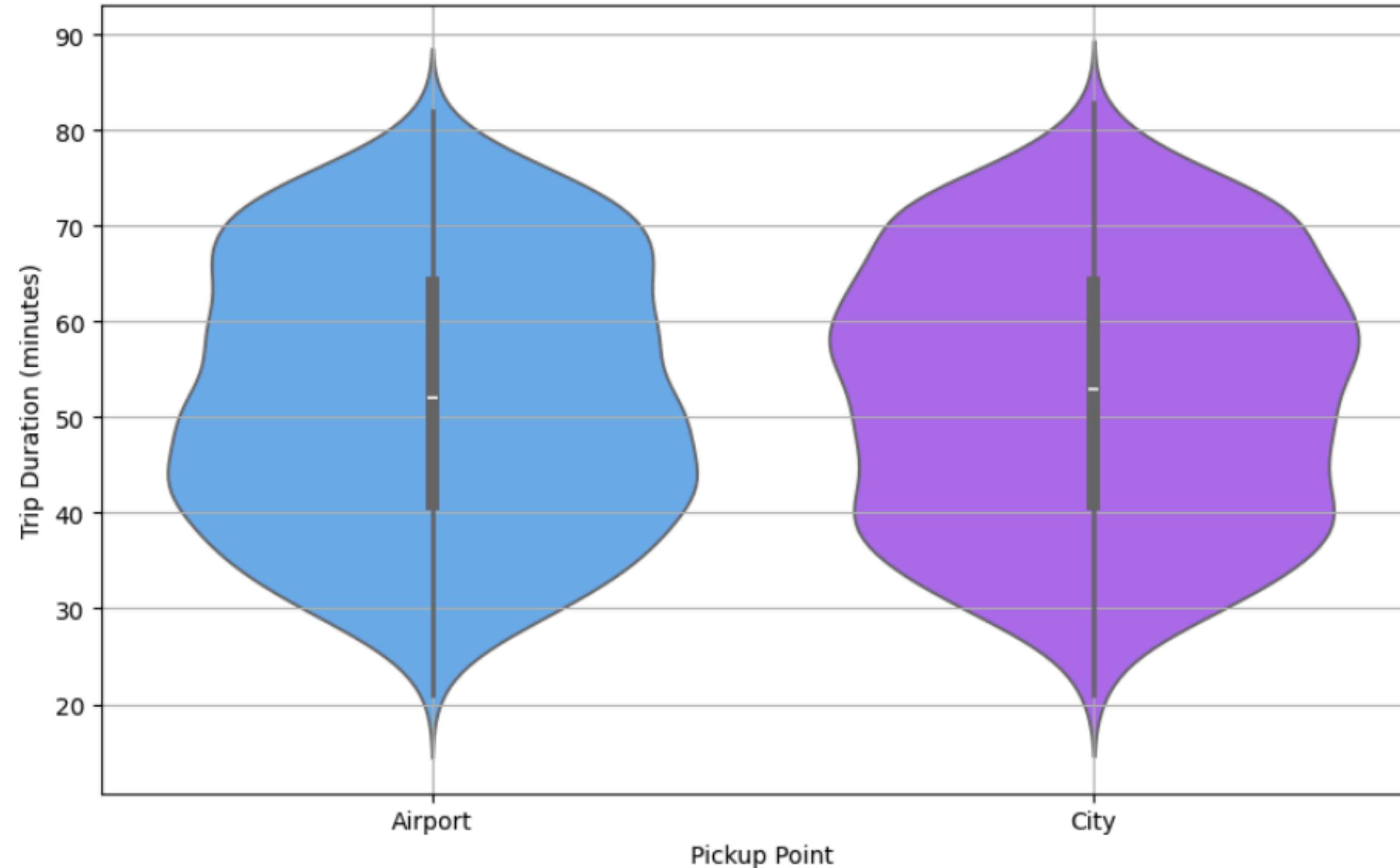
A pie chart quickly shows the percentage share of each trip status. Visual and easy to interpret.

Trip Duration vs Request Hour (Completed Trips Only)



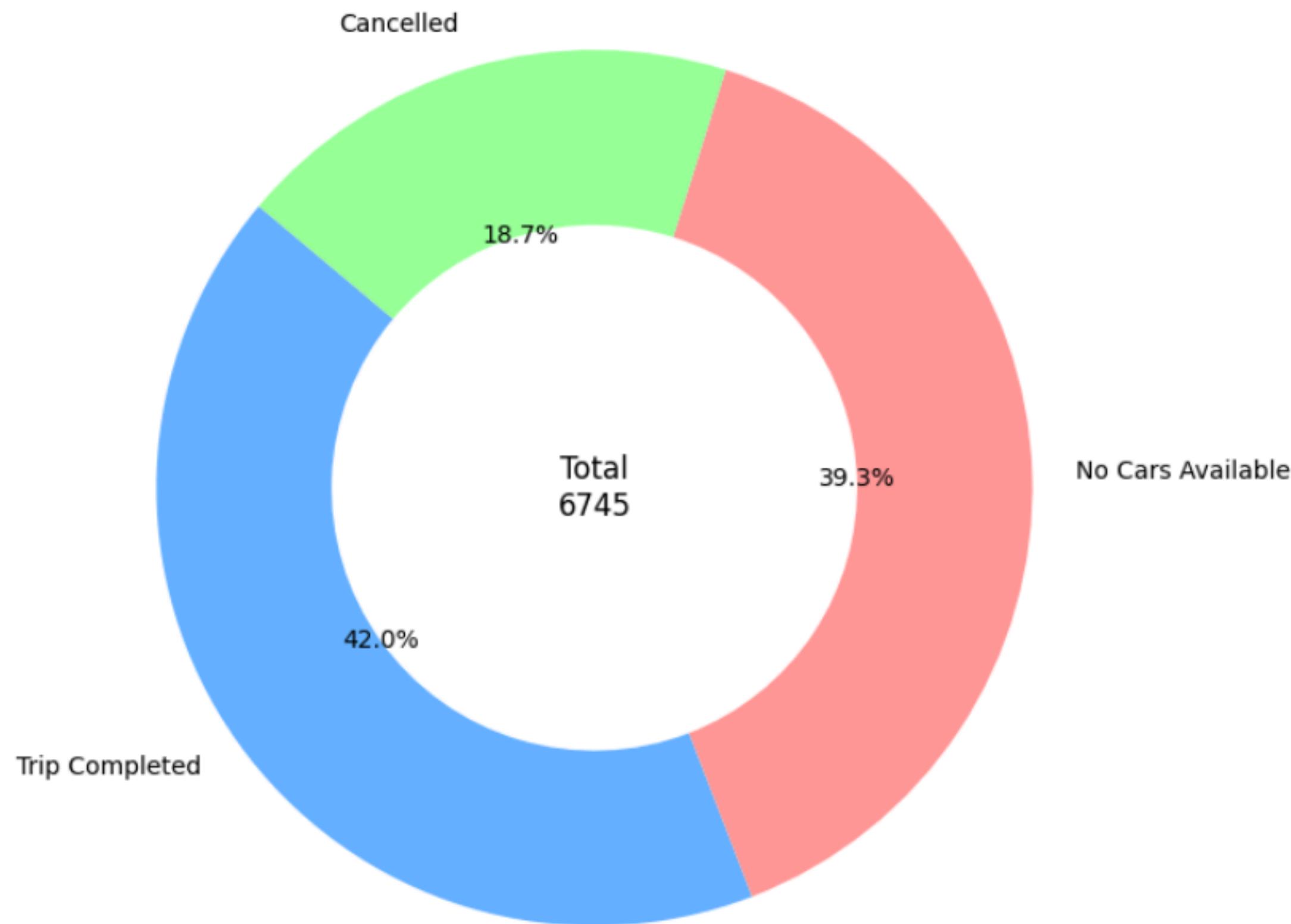
A scatter plot shows the distribution and variation of trip durations over the day.

Violin Plot of Trip Duration by Pickup Point (Completed Trips)



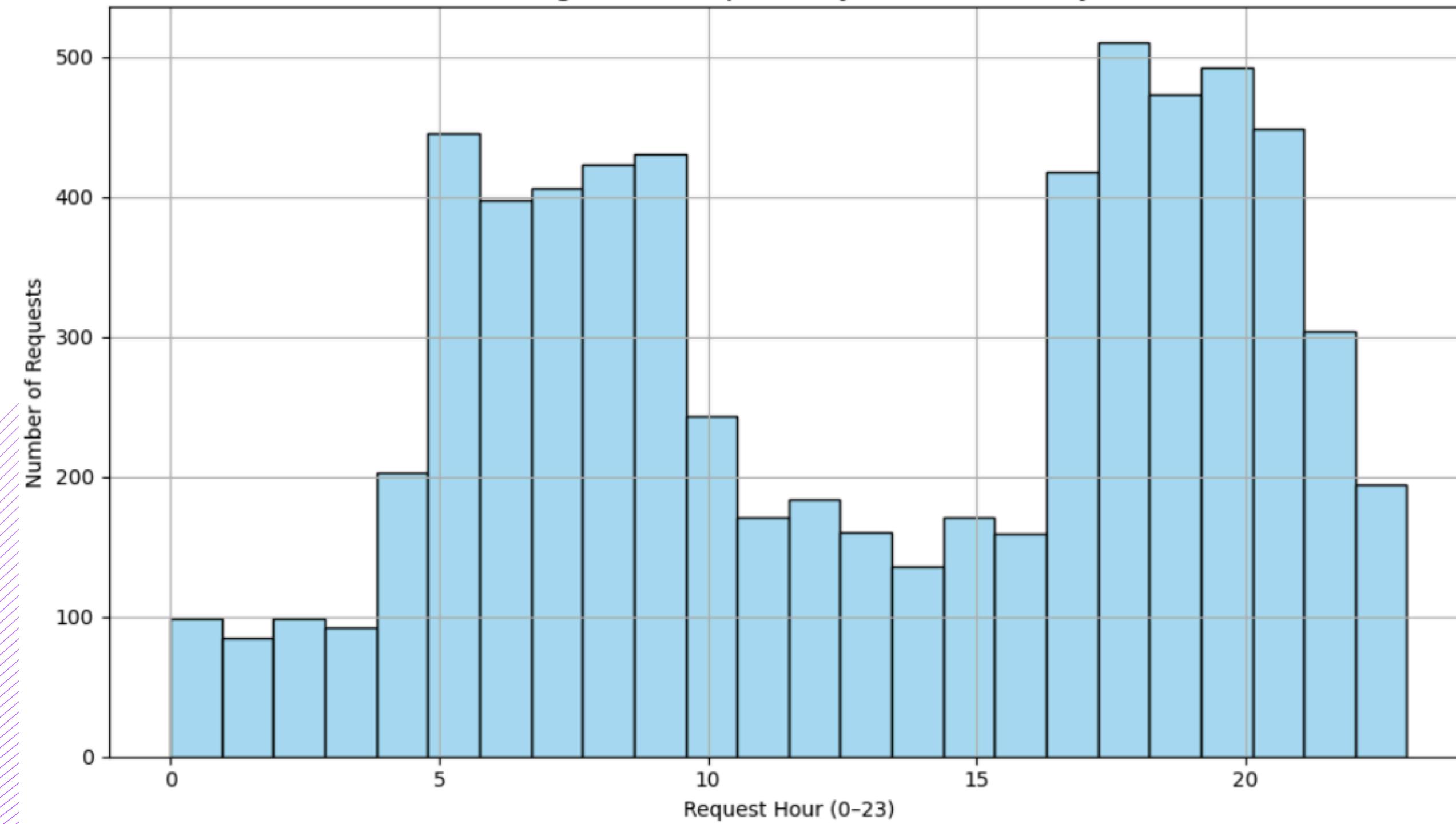
It offers more insight than a box plot by showing the density and distribution shape of trip durations.

Trip Status Distribution – Donut Chart



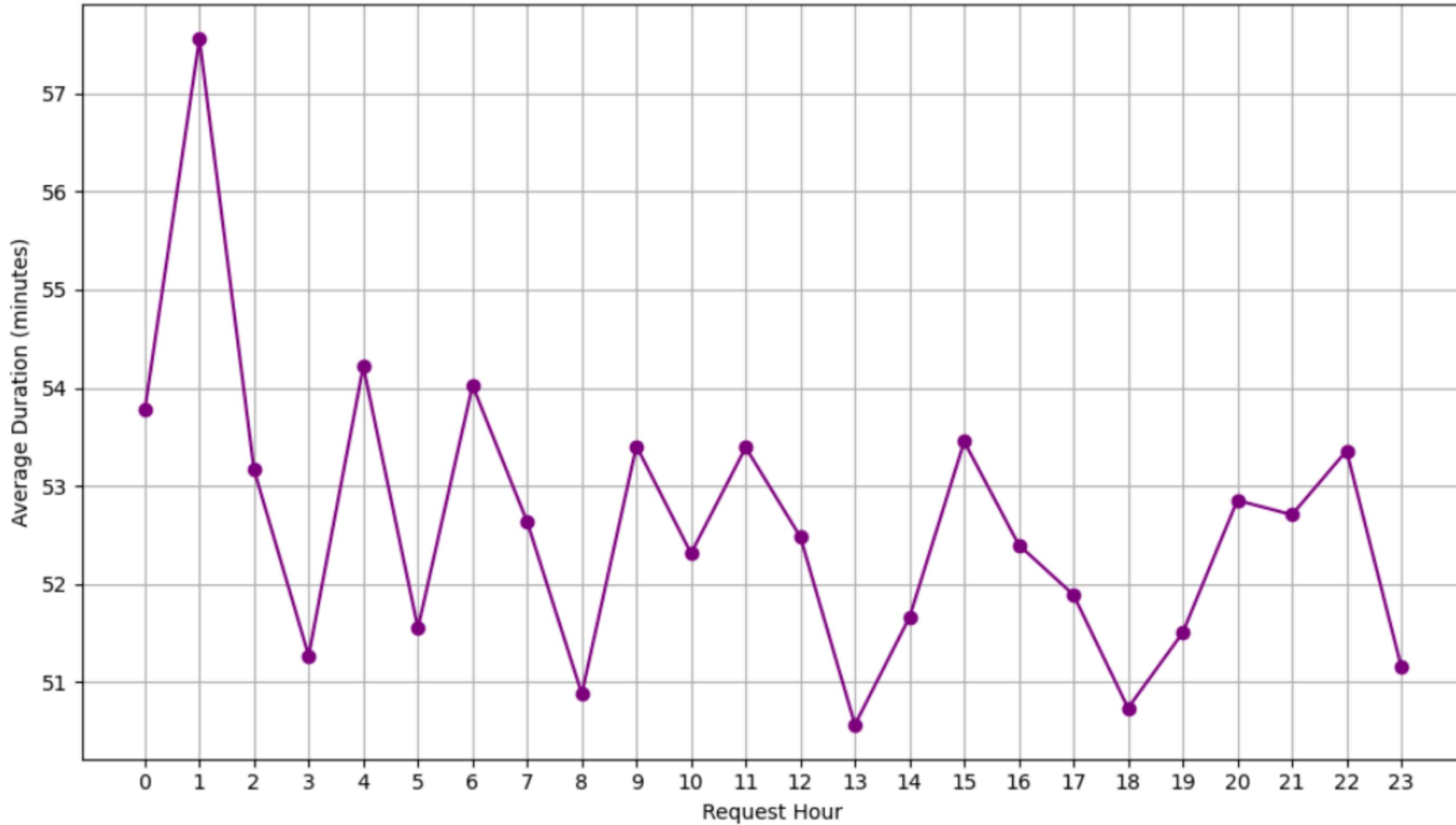
It presents the same insights as a pie chart but adds clarity with the total value in the center.

Histogram of Requests by Hour of the Day

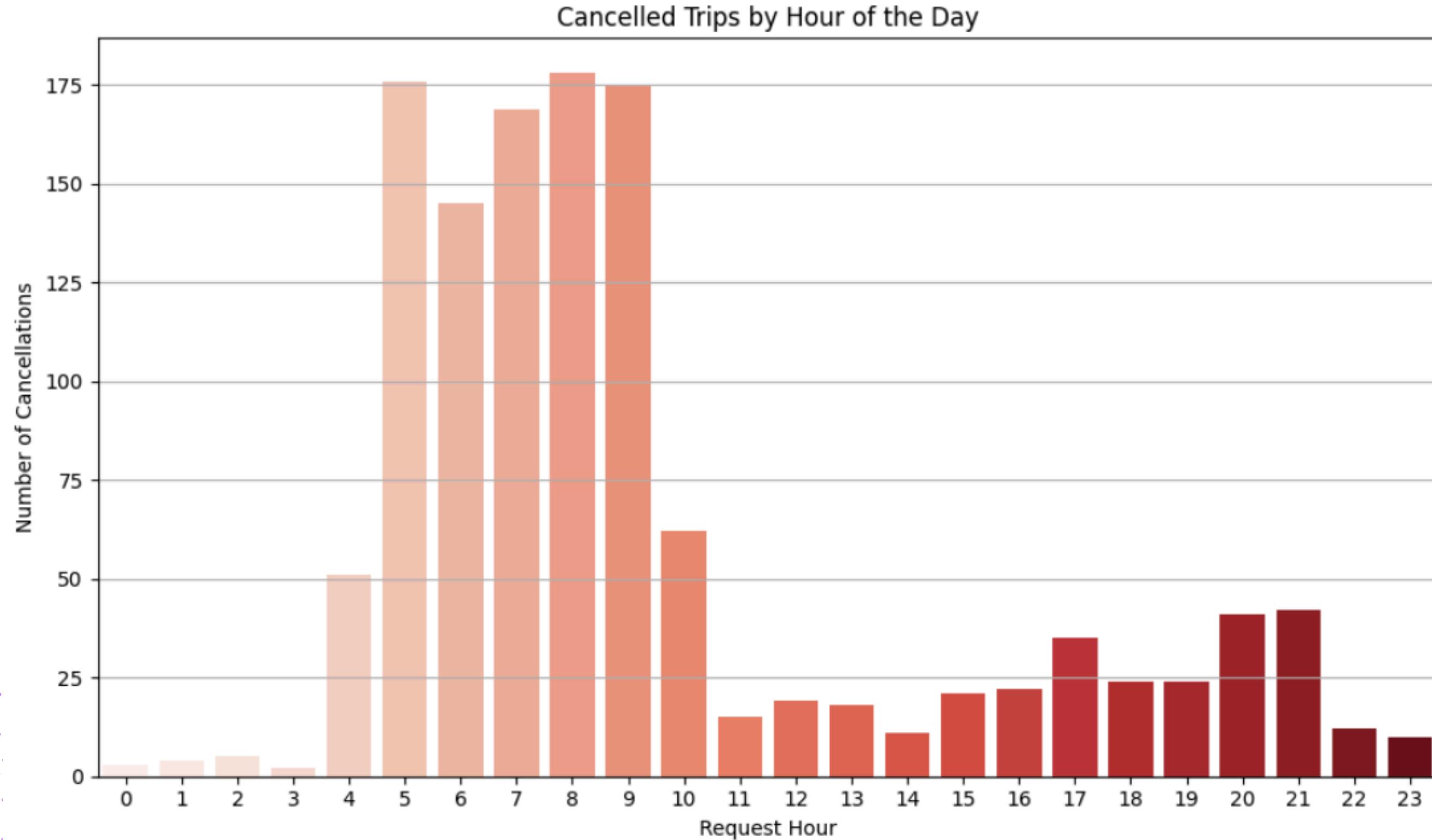


Shows the distribution of ride requests throughout the day.

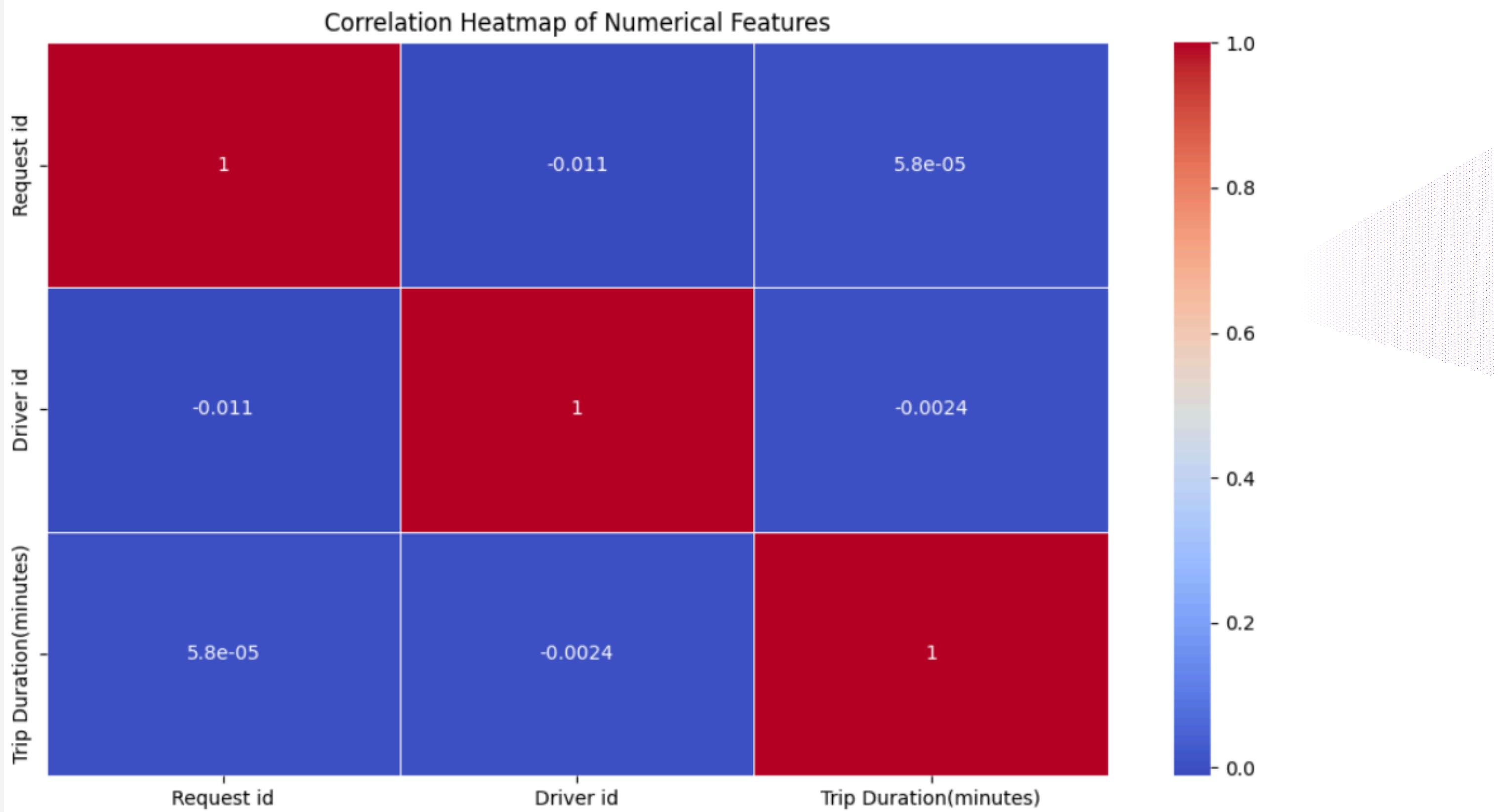
Average Trip Duration by Hour of the Day



A line chart is ideal to capture and compare how the average trip duration changes hourly. It adds a new layer of analysis by focusing on trip time instead of just trip count or status.

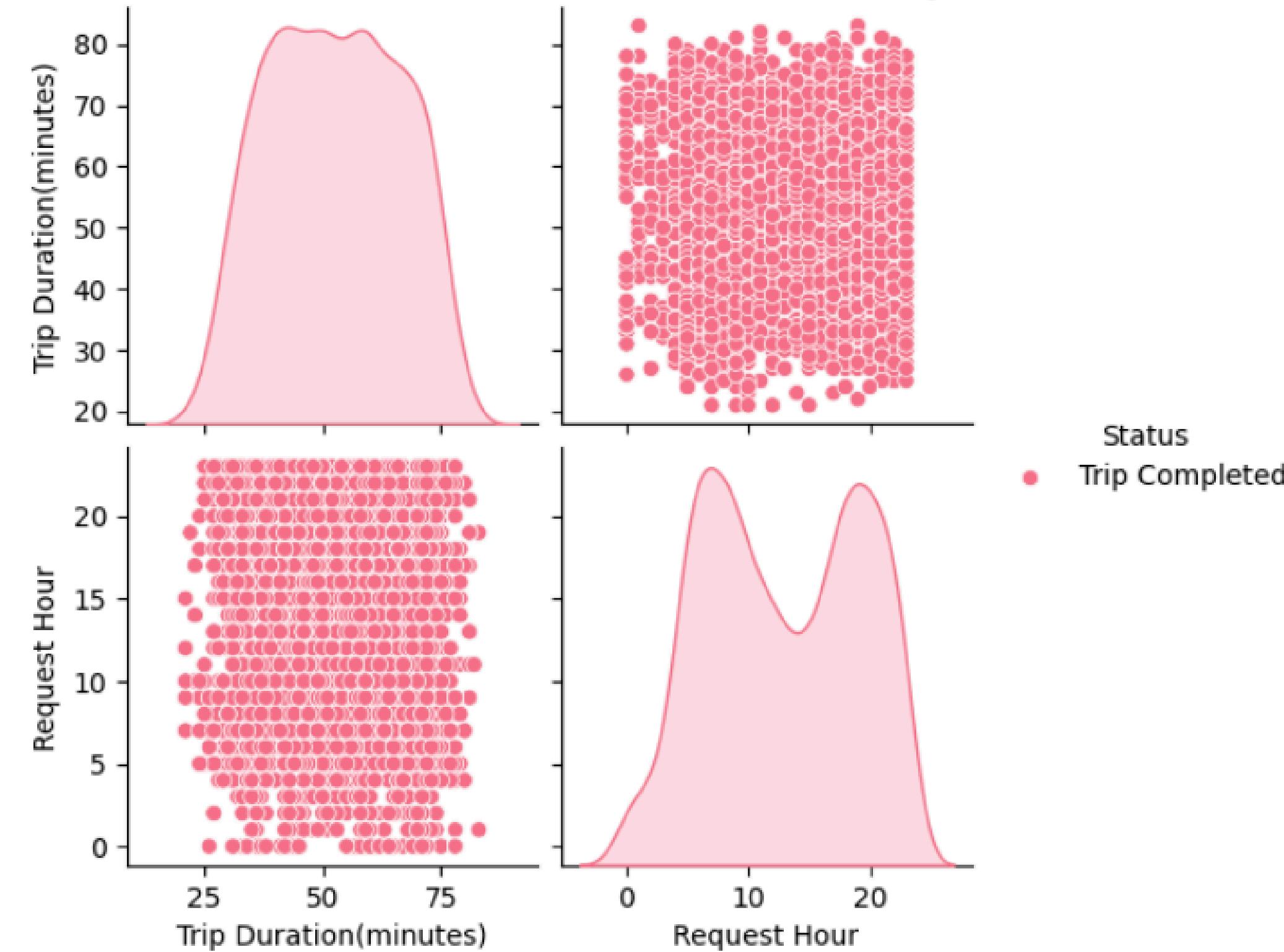


This chart was chosen because cancellations represent a critical business pain point. Bar plots are excellent for showing exact count distributions over categorical variables like time slots. Since Uber operates on tight time-sensitive service windows, understanding cancellation behavior hour by hour provides immense operational insight.

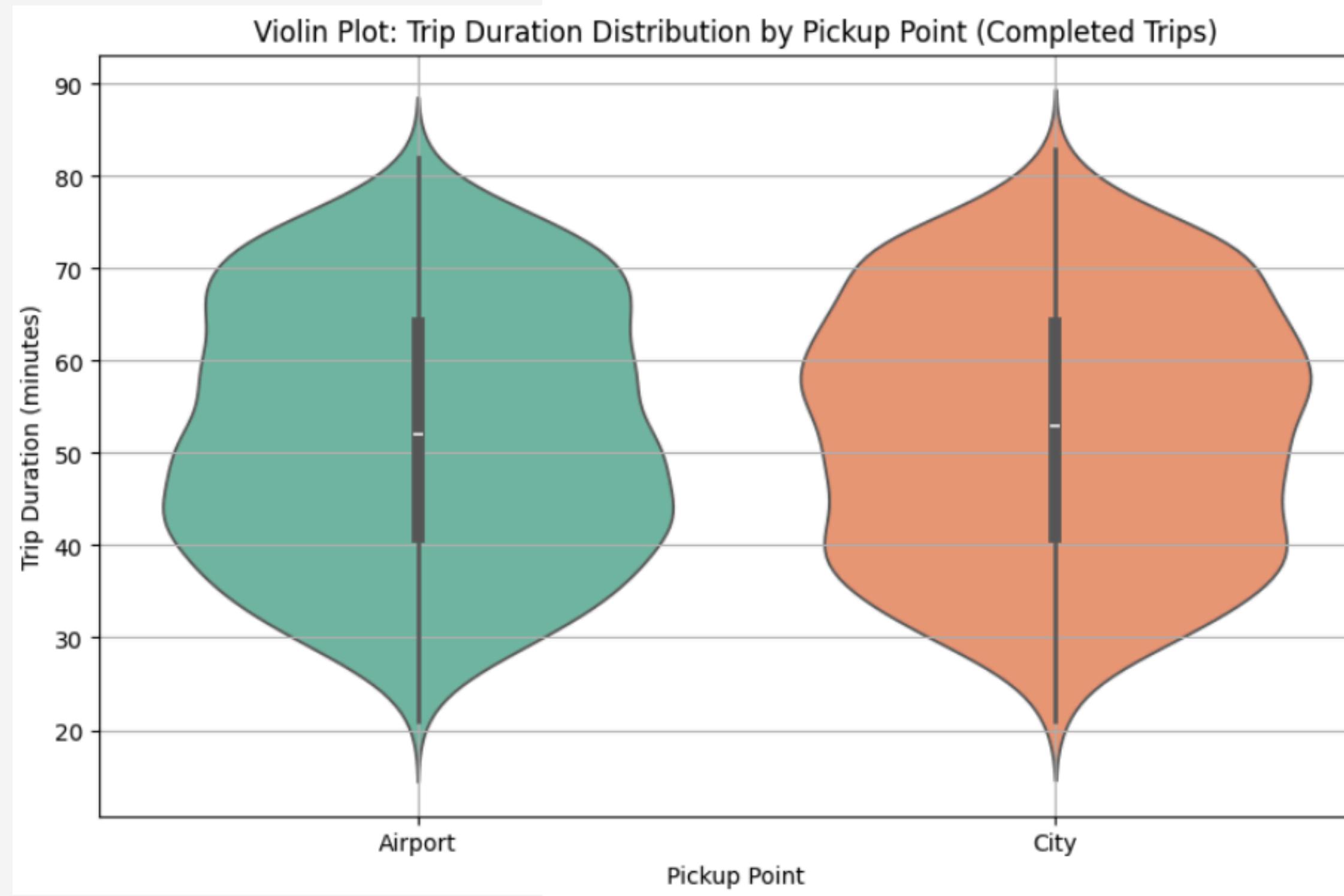


A correlation heatmap was chosen because it helps uncover hidden relationships between multiple numerical features in the dataset—such as Trip Duration (minutes), Request Hour, and possibly Driver ID (if stored as numeric). It visually highlights how closely related any two variables are through color gradients.

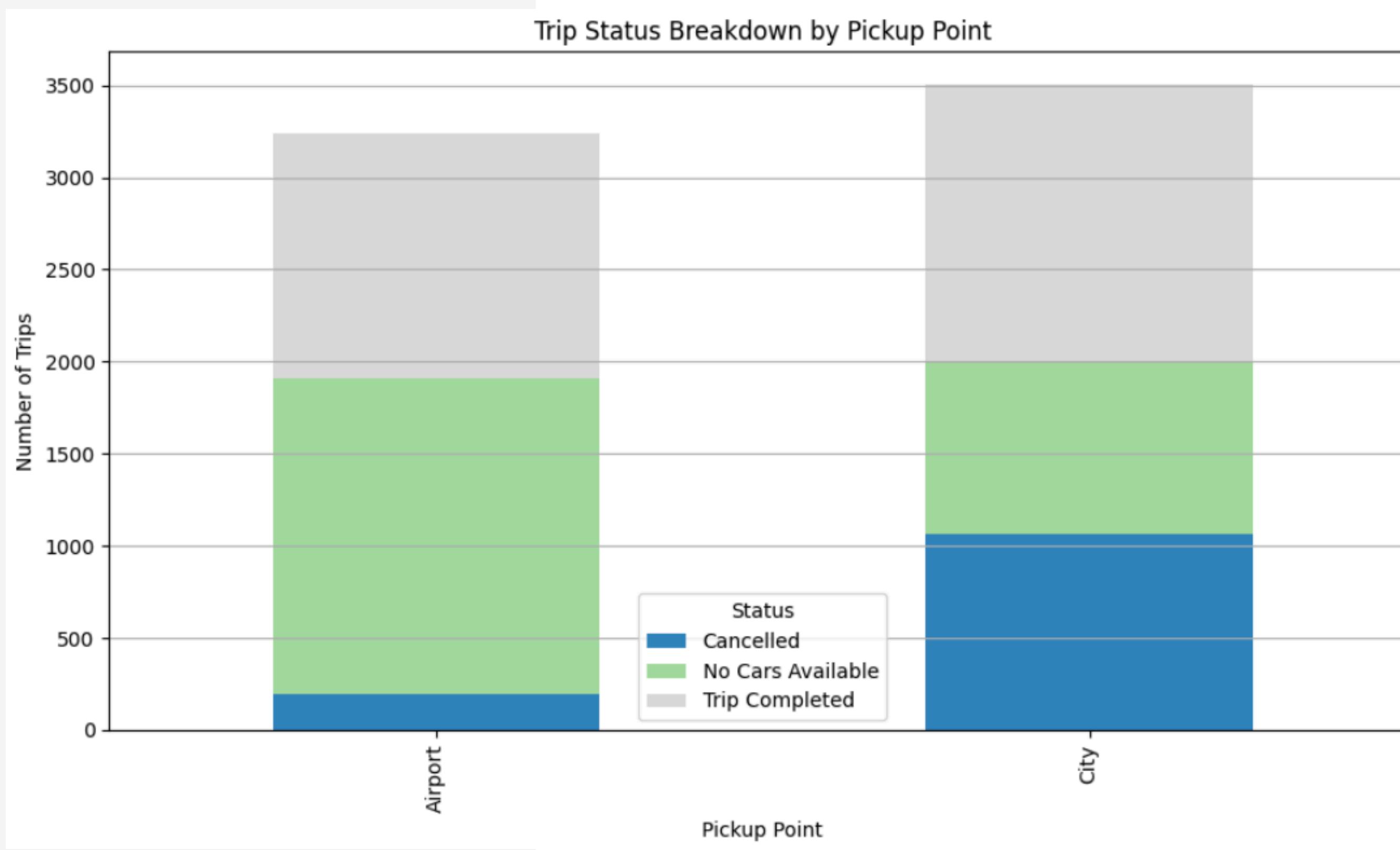
Pair Plot of Numerical Features Colored by Status



A pair plot was selected because it allows us to visualize pairwise relationships between multiple numerical features in one view.



This plot was chosen to explore how trip durations vary between Airport and City pickups in a more detailed way than a box plot. It also shows data density, helping spot where most trips fall and how spread out they are.

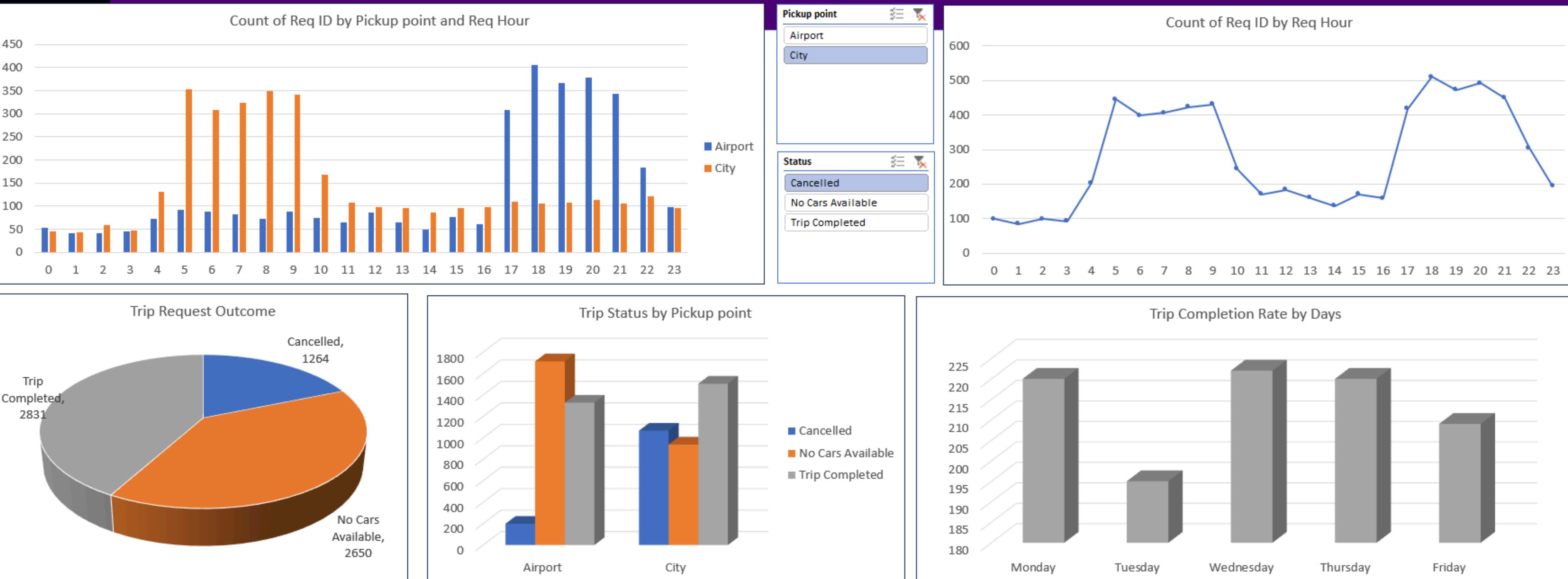


A stacked bar chart clearly shows the proportion of trip statuses (Completed, Cancelled, No Cars Available) for each pickup point (City vs Airport).

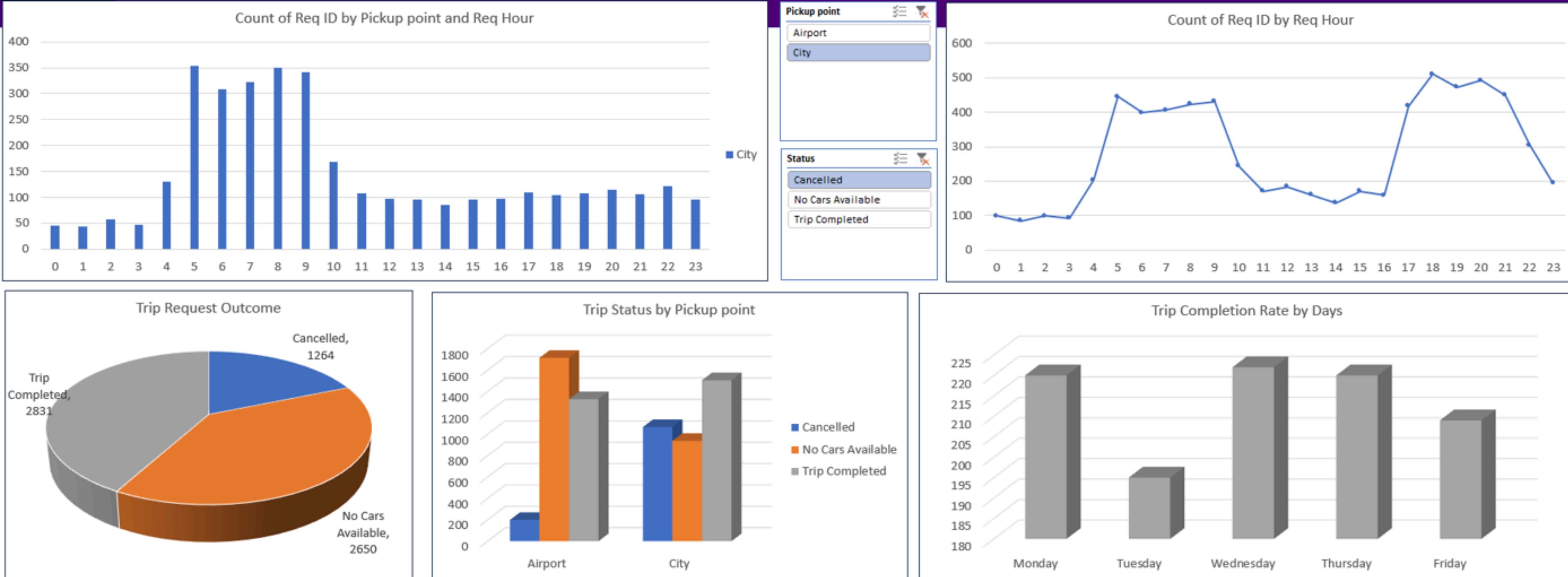
Using Excel

- Visualized trip status (Completed, Cancelled, No Cars Available).
- Analyzed hourly ride demand using bar/line charts.
- Compared requests by weekdays to find busiest days.
- Stacked charts for pickup point vs status to spot service gaps.
- Used slicers (for Day and Pickup Point) to make charts interactive.
- Built insights-driven layout to help improve Uber's operations.

Uber Request Dashboard



Uber Request Dashboard



Using SQL

1) SELECT TOP 10 * FROM [uber dataset];

| | Request_id | Pickup_point | Driver_id | Status | Request_timestamp | Drop_timestamp | Request_Hour | Request_Day | Trip_Duration_minutes |
|----|------------|--------------|-----------|-------------------|-----------------------------|-----------------------------|--------------|-------------|-----------------------|
| 1 | 1 | Airport | 285 | Trip Completed | 2016-07-11 00:20:00.0000000 | 2016-07-11 00:51:00.0000000 | 0 | Monday | 31 |
| 2 | 2 | Airport | NULL | No Cars Available | 2016-07-11 00:23:00.0000000 | NULL | 0 | Monday | NULL |
| 3 | 3 | Airport | 80 | Trip Completed | 2016-07-11 00:24:00.0000000 | 2016-07-11 01:31:00.0000000 | 0 | Monday | 67 |
| 4 | 4 | City | NULL | No Cars Available | 2016-07-11 00:37:00.0000000 | NULL | 0 | Monday | NULL |
| 5 | 5 | Airport | 264 | Trip Completed | 2016-07-11 00:36:00.0000000 | 2016-07-11 01:35:00.0000000 | 0 | Monday | 59 |
| 6 | 6 | City | NULL | No Cars Available | 2016-07-11 00:36:00.0000000 | NULL | 0 | Monday | NULL |
| 7 | 7 | Airport | NULL | No Cars Available | 2016-07-11 00:30:00.0000000 | NULL | 0 | Monday | NULL |
| 8 | 8 | City | NULL | No Cars Available | 2016-07-11 00:40:00.0000000 | NULL | 0 | Monday | NULL |
| 9 | 9 | Airport | 235 | Trip Completed | 2016-07-11 00:45:00.0000000 | 2016-07-11 02:00:00.0000000 | 0 | Monday | 75 |
| 10 | 10 | City | 228 | Trip Completed | 2016-07-11 00:54:00.0000000 | 2016-07-11 01:59:00.0000000 | 0 | Monday | 65 |

2) total ride requests

SELECT COUNT(*) AS Total_Requests FROM [uber dataset];

| | Total_Requests |
|---|----------------|
| 1 | 6745 |

3) count of each ride status

```
SELECT Status, COUNT(*) AS request_Count FROM [uber dataset] GROUP BY Status;
```

| | Status | request_Count |
|---|-------------------|---------------|
| 1 | No Cars Available | 2650 |
| 2 | Trip Completed | 2831 |
| 3 | Cancelled | 1264 |

4) requests by pickup point

```
SELECT [Pickup_point], COUNT(*) AS Total_Requests FROM [uber dataset] GROUP BY [Pickup_point];
```

| | Pickup_point | Total_Requests |
|---|--------------|----------------|
| 1 | City | 3507 |
| 2 | Airport | 3238 |

5) hourly ride requests

```
SELECT Request_Hour, COUNT(*) AS Request_Count FROM [uber dataset] GROUP BY Request_Hour ORDER BY Request_Hour;
```

| | Request_Hour | Request_Count |
|---|--------------|---------------|
| 1 | 0 | 99 |
| 2 | 1 | 85 |
| 3 | 2 | 99 |
| 4 | 3 | 92 |
| 5 | 4 | 203 |
| 6 | 5 | 445 |

6) cancelled rides by location

```
SELECT Pickup_point, COUNT(*) AS Cancelled_Trips FROM [uber dataset] WHERE Status='Cancelled' GROUP BY Pickup_point;
```

| | Pickup_point | Cancelled_Trips |
|---|--------------|-----------------|
| 1 | City | 1066 |
| 2 | Airport | 198 |

7) Completed trips and average duration

```
SELECT COUNT(*) AS Completed_Trips, AVG([Trip_Duration_minutes]) AS Avg_Duration FROM [uber dataset] WHERE Status='Trip Completed';
```

| | Completed_Trips | Avg_Duration |
|---|-----------------|--------------|
| 1 | 2831 | 52 |

8) top 5 busiest hours

```
SELECT TOP 5 Request_Hour, COUNT(*) AS Request_Count FROM [uber dataset] GROUP BY Request_Hour ORDER BY Request_Count DESC;
```

| | Request_Hour | Request_Count |
|---|--------------|---------------|
| 1 | 18 | 510 |
| 2 | 20 | 492 |
| 3 | 19 | 473 |
| 4 | 21 | 449 |
| 5 | 5 | 445 |

9) Status-wise breakdown for each pickup point

```
SELECT Pickup_point, Status, COUNT(*) AS Total FROM [uber dataset] GROUP BY Pickup_point, Status ORDER BY Pickup_point, Status;
```

| | Pickup_point | Status | Total |
|---|--------------|-------------------|-------|
| 1 | Airport | Cancelled | 198 |
| 2 | Airport | No Cars Available | 1713 |
| 3 | Airport | Trip Completed | 1327 |
| 4 | City | Cancelled | 1066 |
| 5 | City | No Cars Available | 937 |
| 6 | City | Trip Completed | 1504 |

10) longest 5 trips completed

SELECT TOP 5 * FROM [uber dataset] WHERE Status = 'Trip Completed' ORDER BY [Trip_Duration_minutes] DESC;

| | Request_id | Pickup_point | Driver_id | Status | Request_timestamp | Drop_timestamp | Request_Hour | Request_Day | Trip_Duration_minutes |
|---|------------|--------------|-----------|----------------|-----------------------------|-----------------------------|--------------|-------------|-----------------------|
| 1 | 2417 | City | 77 | Trip Completed | 2016-07-12 19:59:00.0000000 | 2016-07-12 21:22:00.0000000 | 19 | Tuesday | 83 |
| 2 | 4049 | City | 55 | Trip Completed | 2016-07-14 01:03:00.0000000 | 2016-07-14 02:26:00.0000000 | 1 | Thursday | 83 |
| 3 | 5976 | Airport | 221 | Trip Completed | 2016-07-15 11:10:00.0000000 | 2016-07-15 12:32:00.0000000 | 11 | Friday | 82 |
| 4 | 510 | City | 100 | Trip Completed | 2016-07-11 09:40:00.0000000 | 2016-07-11 11:01:00.0000000 | 9 | Monday | 81 |
| 5 | 2169 | Airport | 239 | Trip Completed | 2016-07-12 17:30:00.0000000 | 2016-07-12 18:51:00.0000000 | 17 | Tuesday | 81 |

11) Day-Wise Cancellation Rate

SELECT Request_Day, COUNT(CASE WHEN Status = 'Cancelled' THEN 1 END) * 100.0 / COUNT(*) AS Cancellation_Percentage
FROM [uber dataset] GROUP BY Request_Day ORDER BY Cancellation_Percentage DESC;

| | Request_Day | Cancellation_Percentage |
|---|-------------|-------------------------|
| 1 | Wednesday | 20.194465220643 |
| 2 | Monday | 19.166057059253 |
| 3 | Thursday | 18.625277161862 |
| 4 | Tuesday | 18.362662586074 |
| 5 | Friday | 17.378711078928 |

12) Drivers with Highest Number of Completed Trips

SELECT [Driver_id], COUNT(*) AS Completed_Trips FROM [uber dataset] WHERE Status = 'Trip Completed' AND [Driver_id] IS NOT NULL
GROUP BY [Driver_id] ORDER BY Completed_Trips DESC;

| | Driver_id | Completed_Trips |
|---|-----------|-----------------|
| 1 | 22 | 16 |
| 2 | 184 | 15 |
| 3 | 233 | 15 |
| 4 | 176 | 14 |

13) Requests That Failed (Cancelled or No Cars)

SELECT * FROM [uber dataset] WHERE Status IN ('Cancelled', 'No Cars Available');

| | Request_id | Pickup_point | Driver_id | Status | Request_timestamp | Drop_timestamp | Request_Hour | Request_Day | Trip_Duration_minutes |
|----|------------|--------------|-----------|-------------------|-----------------------------|----------------|--------------|-------------|-----------------------|
| 1 | 2 | Airport | NULL | No Cars Available | 2016-07-11 00:23:00.0000000 | NULL | 0 | Monday | NULL |
| 2 | 4 | City | NULL | No Cars Available | 2016-07-11 00:37:00.0000000 | NULL | 0 | Monday | NULL |
| 3 | 6 | City | NULL | No Cars Available | 2016-07-11 00:36:00.0000000 | NULL | 0 | Monday | NULL |
| 4 | 7 | Airport | NULL | No Cars Available | 2016-07-11 00:30:00.0000000 | NULL | 0 | Monday | NULL |
| 5 | 8 | City | NULL | No Cars Available | 2016-07-11 00:40:00.0000000 | NULL | 0 | Monday | NULL |
| 6 | 12 | City | NULL | No Cars Available | 2016-07-11 01:08:00.0000000 | NULL | 1 | Monday | NULL |
| 7 | 14 | Airport | NULL | No Cars Available | 2016-07-11 01:10:00.0000000 | NULL | 1 | Monday | NULL |
| 8 | 15 | City | NULL | No Cars Available | 2016-07-11 01:15:00.0000000 | NULL | 1 | Monday | NULL |
| 9 | 16 | Airport | NULL | No Cars Available | 2016-07-11 01:09:00.0000000 | NULL | 1 | Monday | NULL |
| 10 | 17 | Airport | NULL | No Cars Available | 2016-07-11 01:16:00.0000000 | NULL | 1 | Monday | NULL |
| 11 | 18 | City | 201 | Cancelled | 2016-07-11 01:16:00.0000000 | NULL | 1 | Monday | NULL |
| 12 | 19 | City | NULL | No Cars Available | 2016-07-11 01:14:00.0000000 | NULL | 1 | Monday | NULL |
| 13 | 20 | Airport | NULL | No Cars Available | 2016-07-11 01:23:00.0000000 | NULL | 1 | Monday | NULL |
| 14 | 22 | Airport | NULL | No Cars Available | 2016-07-11 01:40:00.0000000 | NULL | 1 | Monday | NULL |

14) Repeated Requests from Same Timestamp

SELECT [Request_timestamp], COUNT(*) AS Duplicate_Count FROM [uber dataset] GROUP BY [Request_timestamp]
HAVING COUNT(*) > 1;

| | Request_timestamp | Duplicate_Count |
|---|-----------------------------|-----------------|
| 1 | 2016-07-12 14:30:00.0000000 | 2 |
| 2 | 2016-07-11 14:16:00.0000000 | 2 |
| 3 | 2016-07-11 19:59:00.0000000 | 2 |
| 4 | 2016-07-12 08:57:00.0000000 | 3 |
| 5 | 2016-07-14 17:49:00.0000000 | 2 |
| 6 | 2016-07-12 18:53:00.0000000 | 3 |
| - | 2016-07-12 22:04:00.0000000 | 2 |

15) Find number of requests from 'City'

```
SELECT COUNT(*) AS City_Requests FROM [uber dataset] WHERE [Pickup_point] = 'City';
```

| | Results | Messages |
|---|---------------|----------|
| | City_Requests | |
| 1 | 3507 | |

16) Show all trips completed by driver 285

```
SELECT * FROM [uber dataset] WHERE [Driver_id] = 285 AND Status = 'Trip Completed';
```

| | Results | Messages | | | | | | | |
|---|------------|--------------|-----------|----------------|-----------------------------|-----------------------------|--------------|-------------|-----------------------|
| | Request_id | Pickup_point | Driver_id | Status | Request_timestamp | Drop_timestamp | Request_Hour | Request_Day | Trip_Duration_minutes |
| 1 | 1 | Airport | 285 | Trip Completed | 2016-07-11 00:20:00.0000000 | 2016-07-11 00:51:00.0000000 | 0 | Monday | 31 |
| 2 | 1923 | City | 285 | Trip Completed | 2016-07-12 11:00:00.0000000 | 2016-07-12 12:00:00.0000000 | 11 | Tuesday | 60 |
| 3 | 3804 | Airport | 285 | Trip Completed | 2016-07-13 21:00:00.0000000 | 2016-07-13 21:37:00.0000000 | 21 | Wednesday | 37 |
| 4 | 4876 | Airport | 285 | Trip Completed | 2016-07-14 17:46:00.0000000 | 2016-07-14 18:23:00.0000000 | 17 | Thursday | 37 |
| 5 | 5428 | Airport | 285 | Trip Completed | 2016-07-15 02:28:00.0000000 | 2016-07-15 03:04:00.0000000 | 2 | Friday | 36 |
| 6 | 6452 | City | 285 | Trip Completed | 2016-07-15 20:07:00.0000000 | 2016-07-15 20:48:00.0000000 | 20 | Friday | 41 |
| 7 | 6703 | City | 285 | Trip Completed | 2016-07-15 22:51:00.0000000 | 2016-07-15 23:37:00.0000000 | 22 | Friday | 46 |

17) Show all requests between 6 AM and 9 AM

```
SELECT * FROM [uber dataset] WHERE Request_Hour BETWEEN 6 AND 9;
```

| | Results | Messages | | | | | | | |
|---|------------|--------------|-----------|-------------------|-----------------------------|-----------------------------|--------------|-------------|-----------------------|
| | Request_id | Pickup_point | Driver_id | Status | Request_timestamp | Drop_timestamp | Request_Hour | Request_Day | Trip_Duration_minutes |
| 1 | 194 | City | NULL | No Cars Available | 2016-07-11 06:05:00.0000000 | NULL | 6 | Monday | NULL |
| 2 | 195 | City | 297 | Trip Completed | 2016-07-11 06:05:00.0000000 | 2016-07-11 06:52:00.0000000 | 6 | Monday | 47 |
| 3 | 196 | Airport | 24 | Cancelled | 2016-07-11 06:07:00.0000000 | NULL | 6 | Monday | NULL |
| 4 | 197 | City | 202 | Trip Completed | 2016-07-11 06:00:00.0000000 | 2016-07-11 06:49:00.0000000 | 6 | Monday | 49 |
| 5 | 198 | Airport | 249 | Trip Completed | 2016-07-11 06:03:00.0000000 | 2016-07-11 06:44:00.0000000 | 6 | Monday | 41 |
| 6 | 199 | City | NULL | No Cars Available | 2016-07-11 06:07:00.0000000 | NULL | 6 | Monday | NULL |
| 7 | 200 | City | 36 | Cancelled | 2016-07-11 06:10:00.0000000 | NULL | 6 | Monday | NULL |
| 8 | 201 | City | 76 | Trip Completed | 2016-07-11 06:09:00.0000000 | 2016-07-11 07:20:00.0000000 | 6 | Monday | 71 |

18) Find all requests made on Monday

SELECT * FROM [uber dataset] WHERE Request_Day = 'Monday';

| | Request_id | Pickup_point | Driver_id | Status | Request_timestamp | Drop_timestamp | Request_Hour | Request_Day | Trip_Duration_minutes |
|---|------------|--------------|-----------|-------------------|-----------------------------|-----------------------------|--------------|-------------|-----------------------|
| 1 | 1 | Airport | 285 | Trip Completed | 2016-07-11 00:20:00.0000000 | 2016-07-11 00:51:00.0000000 | 0 | Monday | 31 |
| 2 | 2 | Airport | NULL | No Cars Available | 2016-07-11 00:23:00.0000000 | NULL | 0 | Monday | NULL |
| 3 | 3 | Airport | 80 | Trip Completed | 2016-07-11 00:24:00.0000000 | 2016-07-11 01:31:00.0000000 | 0 | Monday | 67 |
| 4 | 4 | City | NULL | No Cars Available | 2016-07-11 00:37:00.0000000 | NULL | 0 | Monday | NULL |
| 5 | 5 | Airport | 264 | Trip Completed | 2016-07-11 00:36:00.0000000 | 2016-07-11 01:35:00.0000000 | 0 | Monday | 59 |
| 6 | 6 | City | NULL | No Cars Available | 2016-07-11 00:36:00.0000000 | NULL | 0 | Monday | NULL |
| 7 | 7 | Airport | NULL | No Cars Available | 2016-07-11 00:30:00.0000000 | NULL | 0 | Monday | NULL |
| 8 | 8 | City | NULL | No Cars Available | 2016-07-11 00:40:00.0000000 | NULL | 0 | Monday | NULL |
| 9 | 9 | Airport | 235 | Trip Completed | 2016-07-11 00:45:00.0000000 | 2016-07-11 02:00:00.0000000 | 0 | Monday | 75 |

19) Show requests where Trip Duration > 60 minutes

SELECT * FROM [uber dataset] WHERE [Trip_Duration_minutes] > 60;

| | Request_id | Pickup_point | Driver_id | Status | Request_timestamp | Drop_timestamp | Request_Hour | Request_Day | Trip_Duration_minutes |
|---|------------|--------------|-----------|----------------|-----------------------------|-----------------------------|--------------|-------------|-----------------------|
| 1 | 3 | Airport | 80 | Trip Completed | 2016-07-11 00:24:00.0000000 | 2016-07-11 01:31:00.0000000 | 0 | Monday | 67 |
| 2 | 9 | Airport | 235 | Trip Completed | 2016-07-11 00:45:00.0000000 | 2016-07-11 02:00:00.0000000 | 0 | Monday | 75 |
| 3 | 10 | City | 228 | Trip Completed | 2016-07-11 00:54:00.0000000 | 2016-07-11 01:59:00.0000000 | 0 | Monday | 65 |
| 4 | 33 | Airport | 170 | Trip Completed | 2016-07-11 02:17:00.0000000 | 2016-07-11 03:31:00.0000000 | 2 | Monday | 74 |
| 5 | 34 | City | 78 | Trip Completed | 2016-07-11 02:20:00.0000000 | 2016-07-11 03:34:00.0000000 | 2 | Monday | 74 |
| 6 | 47 | City | 124 | Trip Completed | 2016-07-11 03:22:00.0000000 | 2016-07-11 04:35:00.0000000 | 3 | Monday | 73 |
| 7 | 48 | City | 150 | Trip Completed | 2016-07-11 03:14:00.0000000 | 2016-07-11 04:19:00.0000000 | 3 | Monday | 65 |

20) Count total requests for each day

SELECT Request_Day, COUNT(*) AS Total_Requests FROM [uber dataset] GROUP BY Request_Day;

| | Request_Day | Total_Requests |
|---|-------------|----------------|
| 1 | Wednesday | 1337 |
| 2 | Monday | 1367 |
| 3 | Friday | 1381 |
| 4 | Thursday | 1353 |
| 5 | Tuesday | 1307 |

21) Get all trips that don't have a driver assigned

SELECT * FROM [uber dataset] WHERE [Driver_id] IS NULL;

| | Request_id | Pickup_point | Driver_id | Status | Request_timestamp | Drop_timestamp | Request_Hour | Request_Day | Trip_Duration_minutes |
|---|------------|--------------|-----------|-------------------|-----------------------------|----------------|--------------|-------------|-----------------------|
| 1 | 2 | Airport | NULL | No Cars Available | 2016-07-11 00:23:00.0000000 | NULL | 0 | Monday | NULL |
| 2 | 4 | City | NULL | No Cars Available | 2016-07-11 00:37:00.0000000 | NULL | 0 | Monday | NULL |
| 3 | 6 | City | NULL | No Cars Available | 2016-07-11 00:36:00.0000000 | NULL | 0 | Monday | NULL |
| 4 | 7 | Airport | NULL | No Cars Available | 2016-07-11 00:30:00.0000000 | NULL | 0 | Monday | NULL |
| 5 | 8 | City | NULL | No Cars Available | 2016-07-11 00:40:00.0000000 | NULL | 0 | Monday | NULL |
| 6 | 12 | City | NULL | No Cars Available | 2016-07-11 01:08:00.0000000 | NULL | 1 | Monday | NULL |
| 7 | 14 | Airport | NULL | No Cars Available | 2016-07-11 01:10:00.0000000 | NULL | 1 | Monday | NULL |
| 8 | 15 | City | NULL | No Cars Available | 2016-07-11 01:15:00.0000000 | NULL | 1 | Monday | NULL |
| 9 | 16 | Airport | NULL | No Cars Available | 2016-07-11 01:09:00.0000000 | NULL | 1 | Monday | NULL |

22) Get total completed trips on each weekday

SELECT Request_Day, COUNT(*) AS Completed_Trips FROM [uber dataset] WHERE Status = 'Trip Completed' GROUP BY Request_Day;

| | Request_Day | Completed_Trips |
|---|-------------|-----------------|
| 1 | Monday | 551 |
| 2 | Wednesday | 551 |
| 3 | Friday | 561 |
| 4 | Thursday | 530 |
| 5 | Tuesday | 562 |

23) Show top 3 hours with least number of requests

SELECT TOP 3 Request_Hour, COUNT(*) AS Request_Count FROM [uber dataset] GROUP BY Request_Hour ORDER BY Request_Count ASC;

| | Request_Hour | Request_Count |
|---|--------------|---------------|
| 1 | 1 | 85 |
| 2 | 3 | 92 |
| 3 | 0 | 99 |

24) Get all requests where status is not 'Trip Completed'

SELECT * FROM [uber dataset] WHERE Status != 'Trip Completed';

| | Request_id | Pickup_point | Driver_id | Status | Request_timestamp | Drop_timestamp | Request_Hour | Request_Day | Trip_Duration_minutes |
|---|------------|--------------|-----------|-------------------|-----------------------------|----------------|--------------|-------------|-----------------------|
| 1 | 2 | Airport | NULL | No Cars Available | 2016-07-11 00:23:00.0000000 | NULL | 0 | Monday | NULL |
| 2 | 4 | City | NULL | No Cars Available | 2016-07-11 00:37:00.0000000 | NULL | 0 | Monday | NULL |
| 3 | 6 | City | NULL | No Cars Available | 2016-07-11 00:36:00.0000000 | NULL | 0 | Monday | NULL |
| 4 | 7 | Airport | NULL | No Cars Available | 2016-07-11 00:30:00.0000000 | NULL | 0 | Monday | NULL |
| 5 | 8 | City | NULL | No Cars Available | 2016-07-11 00:40:00.0000000 | NULL | 0 | Monday | NULL |
| 6 | 12 | City | NULL | No Cars Available | 2016-07-11 01:08:00.0000000 | NULL | 1 | Monday | NULL |

25) Show all requests that happened after 6 PM

SELECT * FROM [uber dataset] WHERE Request_Hour >= 18;

| | Request_id | Pickup_point | Driver_id | Status | Request_timestamp | Drop_timestamp | Request_Hour | Request_Day | Trip_Duration_minutes |
|---|------------|--------------|-----------|-------------------|-----------------------------|-----------------------------|--------------|-------------|-----------------------|
| 1 | 866 | Airport | NULL | No Cars Available | 2016-07-11 18:00:00.0000000 | NULL | 18 | Monday | NULL |
| 2 | 874 | Airport | NULL | No Cars Available | 2016-07-11 18:04:00.0000000 | NULL | 18 | Monday | NULL |
| 3 | 877 | Airport | NULL | No Cars Available | 2016-07-11 18:02:00.0000000 | NULL | 18 | Monday | NULL |
| 4 | 878 | City | 128 | Trip Completed | 2016-07-11 18:09:00.0000000 | 2016-07-11 18:49:00.0000000 | 18 | Monday | 40 |
| 5 | 879 | Airport | NULL | No Cars Available | 2016-07-11 18:02:00.0000000 | NULL | 18 | Monday | NULL |
| 6 | 880 | Airport | NULL | No Cars Available | 2016-07-11 18:01:00.0000000 | NULL | 18 | Monday | NULL |

26) List distinct pickup points

SELECT DISTINCT [Pickup_point] FROM [uber dataset];

| | Pickup_point |
|---|--------------|
| 1 | City |
| 2 | Airport |

27) Count trips with missing drop timestamps

```
SELECT COUNT(*) AS Missing_Drop_Times FROM [uber dataset] WHERE [Drop_timestamp] IS NULL;
```

| | Missing_Drop_Times |
|---|--------------------|
| 1 | 3914 |

28) Display all requests sorted by Trip Duration (descending)

```
SELECT * FROM [uber dataset] ORDER BY [Trip_Duration_minutes] DESC;
```

| | Request_id | Pickup_point | Driver_id | Status | Request_timestamp | Drop_timestamp | Request_Hour | Request_Day | Trip_Duration_minutes |
|---|------------|--------------|-----------|----------------|-----------------------------|-----------------------------|--------------|-------------|-----------------------|
| 1 | 2417 | City | 77 | Trip Completed | 2016-07-12 19:59:00.0000000 | 2016-07-12 21:22:00.0000000 | 19 | Tuesday | 83 |
| 2 | 4049 | City | 55 | Trip Completed | 2016-07-14 01:03:00.0000000 | 2016-07-14 02:26:00.0000000 | 1 | Thursday | 83 |
| 3 | 5976 | Airport | 221 | Trip Completed | 2016-07-15 11:10:00.0000000 | 2016-07-15 12:32:00.0000000 | 11 | Friday | 82 |
| 4 | 6042 | Airport | 273 | Trip Completed | 2016-07-15 13:09:00.0000000 | 2016-07-15 14:30:00.0000000 | 13 | Friday | 81 |
| 5 | 5086 | Airport | 113 | Trip Completed | 2016-07-14 19:41:00.0000000 | 2016-07-14 21:02:00.0000000 | 19 | Thursday | 81 |
| 6 | 6623 | City | 291 | Trip Completed | 2016-07-15 21:51:00.0000000 | 2016-07-15 23:12:00.0000000 | 21 | Friday | 81 |
| 7 | 2553 | Airport | 273 | Trip Completed | 2016-07-12 21:23:00.0000000 | 2016-07-12 22:44:00.0000000 | 21 | Tuesday | 81 |
| 8 | 2201 | City | 105 | Trip Completed | 2016-07-12 11:10:00.0000000 | 2016-07-12 12:21:00.0000000 | 11 | Wednesday | 81 |

29) Count requests for each combination of Day and Pickup Point

```
SELECT Request_Day, [Pickup_point], COUNT(*) AS Total_Requests FROM [uber dataset] GROUP BY Request_Day, [Pickup_point] ORDER BY Request_Day, [Pickup_point];
```

| | Request_Day | Pickup_point | Total_Requests |
|----|-------------|--------------|----------------|
| 1 | Friday | Airport | 629 |
| 2 | Friday | City | 752 |
| 3 | Monday | Airport | 661 |
| 4 | Monday | City | 706 |
| 5 | Thursday | Airport | 601 |
| 6 | Thursday | City | 752 |
| 7 | Tuesday | Airport | 684 |
| 8 | Tuesday | City | 623 |
| 9 | Wednesday | Airport | 663 |
| 10 | Wednesday | City | 674 |

30) Calculate average trip duration by pickup point

```
SELECT [Pickup_point], AVG([Trip_Duration_minutes]) AS Avg_Trip_Duration FROM [uber dataset] WHERE [Trip_Duration_minutes] IS NOT NULL  
GROUP BY [Pickup_point];
```

| | Pickup_point | Avg_Trip_Duration |
|---|--------------|-------------------|
| 1 | City | 52 |
| 2 | Airport | 52 |

31) Show requests where Trip Duration is NULL (not completed)

```
SELECT * FROM [uber dataset] WHERE [Trip_Duration_minutes] IS NULL;
```

| | Request_id | Pickup_point | Driver_id | Status | Request_timestamp | Drop_timestamp | Request_Hour | Request_Day | Trip_Duration_minutes |
|----|------------|--------------|-----------|-------------------|-----------------------------|----------------|--------------|-------------|-----------------------|
| 1 | 2 | Airport | NULL | No Cars Available | 2016-07-11 00:23:00.0000000 | NULL | 0 | Monday | NULL |
| 2 | 4 | City | NULL | No Cars Available | 2016-07-11 00:37:00.0000000 | NULL | 0 | Monday | NULL |
| 3 | 6 | City | NULL | No Cars Available | 2016-07-11 00:36:00.0000000 | NULL | 0 | Monday | NULL |
| 4 | 7 | Airport | NULL | No Cars Available | 2016-07-11 00:30:00.0000000 | NULL | 0 | Monday | NULL |
| 5 | 8 | City | NULL | No Cars Available | 2016-07-11 00:40:00.0000000 | NULL | 0 | Monday | NULL |
| 6 | 12 | City | NULL | No Cars Available | 2016-07-11 01:08:00.0000000 | NULL | 1 | Monday | NULL |
| 7 | 14 | Airport | NULL | No Cars Available | 2016-07-11 01:10:00.0000000 | NULL | 1 | Monday | NULL |
| 8 | 15 | City | NULL | No Cars Available | 2016-07-11 01:15:00.0000000 | NULL | 1 | Monday | NULL |
| 9 | 16 | Airport | NULL | No Cars Available | 2016-07-11 01:09:00.0000000 | NULL | 1 | Monday | NULL |
| 10 | 17 | Airport | NULL | No Cars Available | 2016-07-11 01:16:00.0000000 | NULL | 1 | Monday | NULL |
| 11 | 18 | City | 201 | Cancelled | 2016-07-11 01:16:00.0000000 | NULL | 1 | Monday | NULL |
| 12 | 19 | City | NULL | No Cars Available | 2016-07-11 01:14:00.0000000 | NULL | 1 | Monday | NULL |
| 13 | 20 | Airport | NULL | No Cars Available | 2016-07-11 01:23:00.0000000 | NULL | 1 | Monday | NULL |
| 14 | 22 | Airport | NULL | No Cars Available | 2016-07-11 01:40:00.0000000 | NULL | 1 | Monday | NULL |
| 15 | 23 | City | NULL | No Cars Available | 2016-07-11 01:38:00.0000000 | NULL | 1 | Monday | NULL |
| 16 | 24 | Airport | NULL | No Cars Available | 2016-07-11 01:48:00.0000000 | NULL | 1 | Monday | NULL |