

REPORT TITLE - UBER TRIP ANALYSIS

DEVELOPER NAME - VIJAY LAXMI ARUP MITRA

DATE OF CREATION -13/08/2025

Github Link - [Mitravijay-Viji/Uber-Analysis](#)

Power BI Project: Uber Trip Analysis

Overview: This Power BI project presents an interactive and visually rich analysis of Uber trip data, designed to uncover key insights into ride patterns, operational efficiency, and revenue trends. By leveraging advanced data modelling and dynamic visuals, the dashboard enables stakeholders to explore trip volumes, fare performance, and distance metrics across various time frames and locations.

The analysis covers essential dimensions such as total trips, average fare per trip, trip distances, peak travel hours, and location-based pickup and drop-off patterns. Customized and dynamic measures and DAX calculations enhance accuracy and allow for deeper exploration, including identifying the farthest trips, busiest periods, and high-demand areas.

Project Goals

With an intuitive layout, color-coded KPIs, and interactive filters, this dashboard empowers decision-makers to:

- Monitor overall performance at a glance.
- Identify trends in customer demand and travel behaviour.
- Optimize pricing and resource allocation based on data-driven insights.

Ultimately, the Uber Trip Analysis dashboard transforms raw trip data into actionable intelligence, helping improve service quality, operational planning, and strategic growth.

This project consists of 3 pages

1. Overview Analysis
2. Time Analysis
3. Details

Data Scope:

1/06/2024-30/06/2024

Steps covered to complete this project:

1. Data Cleaning & Processing in Power BI: In this project, I started by importing raw Uber Trip Analysis data into Power BI. Before I could visualize and analyse the data, cleaning and preparation of data is required. This involved identifying and handling missing values, correcting data types, and removing duplicates. For this purpose I used Power BI's data transformation capabilities, particularly Power Query.

2. Power BI Dashboard Setup: Once the data was cleaned and processed, I set up the foundation for the Uber Trip analytics dashboard. This included creating a new Power BI report and adding various visuals, which would collectively form the dashboard.

3. Import Data in Power BI: The next step was to import the cleaned Uber Trip Analysis into Power BI. This involved connecting to the data source (e.g., Excel) and loading the data into the Power BI environment.

4. Power Query in Power BI: Power Query is a crucial tool in this project. I used it to transform and shape the data further. This included creating custom columns and custom tables to derive insights.

5. Data Modelling: This Step is very important to extract desired results. In this project, two data table has been used, Trip details and Location Table. I have created relationship between Trip location and location tables based on Pick up and drop up location Id from trip table and location id from location table. Now I have created 2 relationships. Now one relationship needs to be active and another one inactive as you know in power BI one cannot create 2 active relationships in data modelling in the same table. You can make Inactive relationship active when you need to create any visual and write measure and DAX function. I have created DAX function for the same.

I have created a new table Called calendar column using calendar function =Calendar Table
= CALENDAR(MIN('Trip Details'[Pickup Date]),MAX('Trip Details'[Pickup Date]))

Data Modelling: After creating Calendar table I modelled it with Trip detail table, on pickup date column which shows one to many relationships. One to Many because in the calendar table date reflection is once so it is automatically a primary key and in the trip details table one date is repeated many times as per the trip.

This data model type is star schema model where we have one fact table which is Trip details and two dimensions table, one is location table and another one is calendar table.

6. DAX in Power BI: DAX (Data Analysis Expressions) played a significant role in the project. I have used DAX formulas to create calculated columns and measures. I have explained all the DAX in the visual. Refer all the pages for the same.

7. Measures and Calculations in Power BI: Within Power BI, DAX is used to define measures and calculations. Measures are dynamic calculations that respond to user

interactions and slicer selections. This allowed us to provide real-time insights based on user needs.

Few Important dynamic measures used in this dashboard is as follows:

Dynamic Measures = {

```
("Total Bookings", NAMEOF('Trip Details'[Total Bookings]), 0),  
("Total Booking Value", NAMEOF('Trip Details'[Total Booking Value]), 1),  
("Total Trip Distance", NAMEOF('Trip Details'[Total Trip Distance Measure]), 2)
```

How to Make a Dynamic Title if you are using Dynamic Measure?

```
Dynamic Title = IF('Dynamic Measures'[Dynamic Measures Order]=0, "Total  
Bookings",IF('Dynamic Measures'[Dynamic Measures Order]=1,"Total Booking  
Value",IF('Dynamic Measures'[Dynamic Measures Order]=2, "Total Trip Distance",  
"Other")))
```

Dynamic Measure for Time Analysis—In the dashboard , in the ribbon section go to modelling, click on parameters, select field option, select all the fields which you want to created measure, like I selected these three measures:

- ✓ Total Bookings
- ✓ Total Booking Value
- ✓ Total Trip Distance

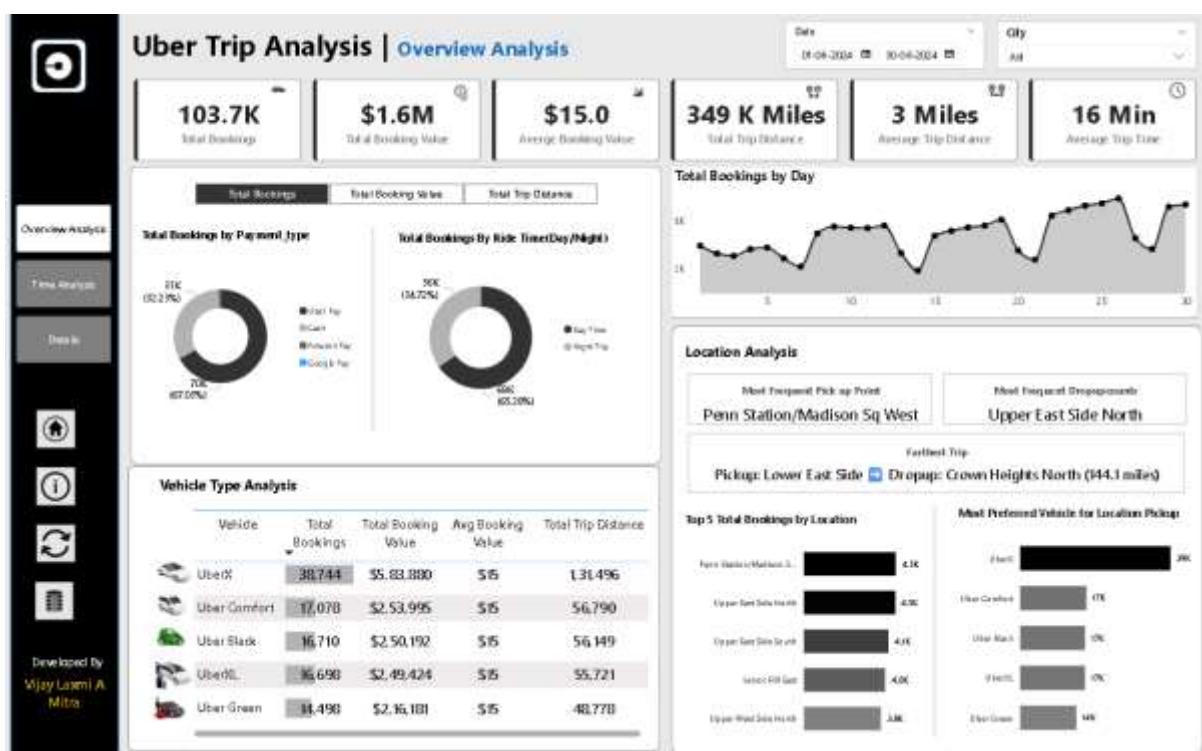
8. Charts in Power BI: Charts and visuals are at the heart of data visualization. I have used various charts like, Donut chart, Area chart, Stacked bar Chart, Matrix table etc. to represent Uber Trip Analysis metrics visually. This will help Stakeholders to understand trends and patterns.

9. Filters and Slicers in Power BI: To enhance interactivity, I implemented few filters to analyse the data according to date, city, pickup location, drop location, Vehicle type and payment type. These tools allowed users to select specific criteria, locations, time periods, or any other relevant data points. The visuals automatically adjusted to reflect the user's selections.

10. Dashboard in Power BI: The dashboard itself is a collection of interconnected visuals, charts, and tables. It presented a comprehensive view of Uber Trip Analysis, enabling stakeholders to explore and gain insights at a glance.

🚗 Uber Trip Analysis – Overview Page Documentation

This Overview Analysis page serves as a high-level summary of Uber ride trends, performance metrics, and customer behaviour. It is structured to give decision-makers quick, interactive insights.



Top KPI Cards

Visual Type: Card(New)

Visuals:

- Total Bookings
- Total Booking Value
- Average Booking Value
- Total Trip Distance
- Average Trip Distance

- Average Trip Time

Purpose: To highlight the most important summary metrics at a glance.

DAX Measures Used:

DAX:

Total Bookings = COUNT('Trip Details'[Trip ID])

Total Booking Value = SUM('Trip Details'[fare_amount])+SUM('Trip Details'[Surge Fee])

Average Booking Value = DIVIDE([Total Booking Value],[Total Bookings],BLANK())

Total Trip Distance = VAR TotalMiles=SUM('Trip Details'[trip_distance])/1000 RETURN
CONCATENATE(FORMAT(TotalMiles,"0")," K Miles")

Why the calculations is in Miles: Since this data covers the international trip, there trip distance is calculated in miles.

Purpose of the Measure:

This measure is used to display the total trip distance in thousands of miles, formatted as a user-friendly label — for example, "349 K Miles" instead of showing a long number like 349,000.

Explanation of Each Part:

1. VAR Total Miles = SUM('Trip Details'[trip_distance]) / 1000

- This sums up all the values in the [trip_distance] column from the 'Trip Details' table.
- Then, it divides the result by 1000 to convert the value into thousands of miles (K Miles).

2. FORMAT(TotalMiles, "0")

- This formats the TotalMiles variable to display no decimal places. So, 349.6 becomes "350".

3. CONCATENATE(..., " K Miles")

- The formatted number is then combined with the text string " K Miles" to produce a readable label, such as:

"350 K Miles"

Average Trip Distance = Var Avgmiles=ROUND(AVERAGE('Trip Details'[trip_distance]),0)
 RETURN CONCATENATE(Avgmiles," Miles")

Average Trip Time = Var AvgTime=AVERAGEX('Trip Details', DATEDIFF('Trip Details'[Pickup Time],'Trip Details'[Drop Off Time],MINUTE)) RETURN
 CONCATENATE(FORMAT(AvgTime,"0")," Min")

What This Measure Does:

This measure calculates the average trip duration in minutes across all Uber trips and formats it as a readable label like "16 Min".

Breakdown of the DAX Logic:

1. DATEDIFF('Pickup Time', 'Drop Off Time', MINUTE)

- This calculates the duration of each trip in minutes.
- For example, if a trip starts at 10:00 AM and ends at 10:20 AM, the result is 20.

2. AVERAGEX('Trip Details', DATEDIFF(...))

- This is the core part: it loops through every row in the 'Trip Details' table, calculates each trip duration using DATEDIFF, and then returns the average of all those durations.
- The result is the average trip time across all rides in minutes.

Why not just use AVERAGE(DATEDIFF(...))?

Because DATEDIFF(...) is a row-wise calculation, and you can't use AVERAGE() directly on a row context.

That's why we need AVERAGEX, which:

- Takes a table (in this case, 'Trip Details')
- Iterates row by row
- Applies the expression (DATEDIFF(...))
- Then averages the result

3. VAR AvgTime =

- The result of the AVERAGEX calculation is stored in a variable called AvgTime.
- This keeps the formula cleaner, more readable, and avoids re-calculation.

4. FORMAT(AvgTime, "0")

- Rounds the average duration to whole minutes.

- For example, 15.4 becomes "15".

5. CONCATENATE(..., " Min")

- Appends the text " Min" to the numeric value for display.
- Final result: "16 Min" (as shown in your dashboard).

Why AVERAGEX is Used (and Not Just AVERAGE)

Function

AVERAGE() Works only on a single column of numbers

AVERAGEX() Iterates row by row through a table, evaluates an expression, and then averages the result

In this case, I am not averaging a column — I am averaging the result of a row-by-row calculation (DATEDIFF(...)). So here AVERAGEX function is used.

In Simple Terms:

"Go through every trip, calculate how many minutes it lasted, then give me the average of all those durations — and display it as a nice label like '16 Min'."

Creation Steps:

1. Insert Card (New) visuals.
2. Place corresponding DAX measures into each card.
3. Place icons as per the visual context (e.g., clock icon for trip time)
4. Responsive to date and city filters

Formatting:

- Color: White text on dark background
- Layout format: Arrangement-single row, Style is card, space between cards 15Px
- Callout Values Format: Font-Segoe UI Bold, font size-25, Color-Black, Alignment-Centre
- Card shape-Rounded Rectangle, Rounded Corners-5PX, background color-white, Border Color- #666666, width-1PX, Switch On Accent Bar, Color-Black, Width-5PX

Note: I have created a separate table for named 'Calendar' with three columns namely, Date, Day Name and Day Number. I have created three measures for these three columns to extract the desired results.

1. Date= CALENDAR(MIN('Trip Details'[Pickup Date]),MAX('Trip Details'[Pickup Date]))
2. Day Name = FORMAT('Calender Table'[Date], "ddd")
3. Day Num = WEEKDAY('Calender Table'[Date],2)

Donut Chart – Total Bookings by Payment Type

Visual Type: Donut Chart

Purpose: To analyse the distribution of bookings by payment methods (Uber Pay, Cash, Google Pay, etc.)

Fields Used:

- Legend: Payment_Type
- Values: Dynamic Measures

Dynamic Measures: I created dynamic Measure for Time Analysis—In the dashboard.

Steps to create Dynamic Measures

1. In the ribbon section go to modelling
2. Click on parameters
3. Select field option
4. Select all the fields which you want to created measure, like I selected these three measures:
 - ✓ Total Bookings
 - ✓ Total Booking Value
 - ✓ Total Trip Distance

Creation Steps:

1. Add Donut Chart visual.
2. Drag Payment Type to the Legend.
3. Use dynamic measure (It includes Total Bookings, Total Booking Value & Total Trip Distance) as the Value.

Formatting:

- Set the visual container size, height:269, width:300
- Kept the Legend on, Positioned the same in the centre right, font for the legend is segoe UI Semibold, size is 8, color of legend text is #666666
- Set the Legend Slices color.

- Kept the detail label On, Position selected is "Outside", font is segoe UI Semibold, size is 9, color of value is #000000, Display unis selected is Thousand & Value decimal places selected is 0.
- Background color is off, Tooltips option is On
- Label Contents selected is data value, percentage of total
- Responsive to filters

Donut Chart – Total Bookings by Ride Time (Day/Night)

Visual Type: Donut Chart

Purpose: To understand whether users prefer day or night rides.

Fields Used:

- Legend: Trip Type Day/Night measure
- Values: Dynamic Measures

DAX used to calculate Ride Time:

Trip Type Day/Night) = Var hourofday=HOUR('Trip Details'[Pickup Time]) RETURN
IF(hourofday>=17 || hourofday<6,"Night Trip","Day Time")

Step-by-Step Explanation:

- VAR hourofday = HOUR('Trip Details'[Pickup Time])**
 - This extracts the hour (0 to 23) from the Pickup Time column.
 - For example:
 - 08:30 AM → 8
 - 06:15 PM → 18
- IF(hourofday >= 17 || hourofday < 6, "Night Trip", "Day Time")**
 - Checks whether the hour is 5 PM (17) or later, or before 6 AM:
 - If true, it's considered a "Night Trip"
 - If false, it's considered "Day Time"

Purpose:

This categorizes each trip into Day Time or Night Trip based on the time of pickup, allowing for time-based analysis such as:

- Comparing number of trips in day vs night

- Identifying peak night travel hours
- Analysing fare trends by time of day

⌚ Logic Summary:

Hour Range Label

6 AM to 4:59 PM --Day Time

5 PM to 5:59 AM --Night Trip

Formatting:

- I have copied the same formatting used in the Payment Type visual
- Consistent donut size with Payment Type chart
- Responsive to filters

Line Chart – Total Bookings by Day

Visual Type: Line Chart

Purpose: To show daily trends in bookings, identifying peaks or dips.

Fields Used:

- X-axis: Date and Date from Calender Table
- Values: Total Bookings

Steps:

1. Add Line Chart.
2. Drag Date field to Axis.
3. Use Total Bookings measure on Y-axis.

Formatting:

- Grey line with black markers
- Tooltip: Date and booking count
- X-axis formatted to show only day numbers
- I have kept the both the gridlines off

- Shade area is On, Area Transparency is 80%
- Formatted X & Y Axis values With font as Segoe UI Semibold, size-9 & Color as grey.
- Slicer synced with date range filter above

Slicers – Date and City

Visual Type: Slicer

Purpose: To filter the entire dashboard based on a selected date range or city.

Fields Used:

- Calender table date
- City field

Formatting:

- Between style for Date slicer
 - Dropdown style for City
 - For City slicer Multi-Select selection is used
 - Background is kept ON
 - Syncs with all visuals on the page
-

Table – Vehicle Type Analysis

Visual Type: Matrix Table

Purpose: To analyse key stats for each vehicle type, such as UberX, Uber Comfort Uber green, etc.

Columns:

- Vehicle Type
- Total Bookings
- Total Booking Value
- Average Booking Value
- Total Trip Distance

Formatting:

- Vehicle icons beside names (custom images added)

- Layout and Style presets- Style used as Default, Layout is Compact
- Kept Horizontal Grid open
- Kept Border Position as Top, border color blue and Width as 1
- Value font is Segoe UI Semibold, size is 14, Text color and alternate text color is black, background color is white
- Formatted column and row header- font is Segoe UI Semibold, size is 12, text color is black, background color is white, kept the alignment centre.
- Aligned columns as centre
- Kept the tooltips On
- Border separators for readability
- Created Measure for the grid title: Title for grid = "Vehicle Type Analysis"

Location Analysis

Visuals:

- Most Frequent Pickup Point
- Most Frequent Dropup Point
- Farthest Trip(Pickup & Drop up)

Visual Type: Card(New)

Purpose: To find the pickup location that appears most frequently in the 'Trip Details' table based on the count of 'Trip ID'.

DAX Examples

Most Frequent Pickup Point = VAR PickPoint=TOPN(1,SUMMARIZE('Trip Details','Location Table'[Location],"PickPoint", COUNT('Trip Details'[Trip ID])),[PickPoint],DESC) RETURN CONCATENATEX(PickPoint,'Location Table'[Location],",")

Breakdown of the DAX Code:

Explanation by Components:

1. SUMMARIZE(...)

SUMMARIZE('Trip Details','Location Table'[Location], "PickPoint", COUNT('Trip Details'[Trip ID]))

- Groups the data by Location Table[Location].
 - Adds a new column called PickPoint that stores the count of trips (Trip ID) for each location.
2. TOPN(1, ..., [PickPoint], DESC)
- From the summarized table, it returns only the row with the highest count (most frequent pickup location).
 - [PickPoint] DESC ensures it's sorted in descending order to pick the top one.

3. CONCATENATEX(...)

CONCATENATEX(PickPoint, 'Location Table'[Location], ",")

- Converts the result of the top row(s) into a text string.
- In this case, returns the name of the most frequent pickup location as text.
- Even though only one location is expected (due to TOPN(1,...)), CONCATENATEX handles it gracefully, allowing for possible ties.

Result:

A single location name (or multiple if tied) as a text value showing the most frequent pickup point based on trip count.

Most Frequent Dropup Point

```
Most Frequent Dropupcounts = VAR Dropupcounts=ADDCOLUMNS(SUMMARIZE('Trip Details','Location Table'[Location]), "Dropupcount", CALCULATE(COUNT('Trip Details'[Trip ID]),USERELATIONSHIP('Trip Details'[DOLocationID],'Location Table'[LocationID]))) VAR Rankeddropupcounts=ADDCOLUMNS(Dropupcounts,"Rank",RANKX(Dropupcounts,[Drop upcount],,DESC,Dense)) VAR Topdropup=FILTER(Rankeddropupcounts,[Rank]=1) RETURN CONCATENATEX(Topdropup,'Location Table'[Location], ", ")
```

Purpose: To identify the location where drop-offs occurred most frequently (i.e., the most common drop-off point) in the 'Trip Details' table, using a relationship through DOLocationID.

Step-by-Step Explanation:

- Create Drop-off Counts Table
- SUMMARIZE groups the data by Location Table[Location].
- ADDCOLUMNS adds a new column Dropup count: This counts how many trips had that location as the drop-off point.
- USERELATIONSHIP is used to activate an inactive relationship between 'Trip Details'[DOLocationID] and 'Location Table'[LocationID] specifically for this calculation.

Rank the Drop-off Counts—`VAR Rankeddropupcounts = ADDCOLUMNS(Dropupcounts, "Rank", RANKX(Dropupcounts, [Dropupcount], , DESC, DENSE))`

5. A new column Rank to the table, ranking each location based on its drop-off count.
6. RANKX is used: Rankings are in descending order (highest drop count = rank 1).
7. DENSE ranking ensures no gaps in rank (e.g., 1, 2, 3 instead of 1, 2, 4).

Filter for Top Drop-off Location(s): `VAR Topdropup = FILTER(Rankeddropupcounts, [Rank] = 1)`-Filters the ranked table to keep only the location(s) with the highest drop-up count (i.e., rank = 1).

Return Result as Text: `RETURN CONCATENATEX(Topdropup, 'Location Table'[Location], ", ")`: Returns the name(s) of the top drop-off location(s) as a comma-separated string. If there's a tie, multiple names will be listed.

Farthest Trip (Pickup & Drop up)

Purpose:

To return a text summary of the longest trip (by distance) from the dataset, showing:

- Pickup Location
- Drop-off Location
- Distance in miles

DAX:

```
VAR Maxdistance=MAX('Trip Details'[trip_distance]) VAR
PickupLocation=LOOKUPVALUE('Location Table'[Location],'Location
Table'[LocationID],CALCULATE(SELECTEDVALUE('Trip Details'[PULocationID]),'Trip
Details'[trip_distance]=Maxdistance)) VAR DropupLocation=LOOKUPVALUE('Location
Table'[Location],'Location Table'[LocationID],CALCULATE(SELECTEDVALUE('Trip
Details'[DOLocationID]),'Trip Details'[trip_distance]=Maxdistance)) RETURN "Pickup: " &
PickupLocation & " ➔ Dropup: " & DropupLocation & " (" & FORMAT(Maxdistance, "0.0") & " miles)"
```

✿ Step-by-Step Explanation:

Get the Maximum Trip Distance

`VAR Maxdistance = MAX('Trip Details'[trip_distance])`

- Finds the maximum value in the trip_distance column.

- This represents the farthest trip taken.

Get the Pickup Location Name

```
VAR PickupLocation =  
    LOOKUPVALUE(  
        'Location Table'[Location],  
        'Location Table'[LocationID],  
        CALCULATE(  
            SELECTEDVALUE('Trip Details'[PULocationID]),  
            'Trip Details'[trip_distance] = Maxdistance  
        )  
    )
```

- CALCULATE + SELECTEDVALUE filters the data to only the trip(s) where the distance equals the maximum.
- Extracts the corresponding pickup location ID.
- LOOKUPVALUE then retrieves the location name from the Location Table.

Get the Drop-off Location Name

```
VAR DropupLocation =  
    LOOKUPVALUE(  
        'Location Table'[Location],  
        'Location Table'[LocationID],  
        CALCULATE(  
            SELECTEDVALUE('Trip Details'[DOLocationID]),  
            'Trip Details'[trip_distance] = Maxdistance  
        )  
    )
```

- Similar to the previous step, but this one fetches the drop-off location ID and its location name.

Return Final String

RETURN

```
"Pickup: " & PickupLocation &  
" Dropup: " & DropupLocation &  
" (" & FORMAT(Maxdistance, "0.0") & " miles)"
```

- Constructs a readable text summary.
- Includes both locations and the distance (formatted to one decimal place).
- To get the arrow in the dax: shortcut key is windows + dot

Formatting:

- Kept Layout arrangement as Single Row, Style as cards and space between cards are 15PX.
- Kept Call out values font as Segoe UI Semibold, size-15 & Font Color as Black.
- Label font as Segoe UI Semibold, size-10 & Font Color as Black #605E5C, Position is Above Value, Space between label and callout value is 8PX
- Kept the border option On, Color is #B3B0AD and width is 1PX
- Icon used for Farthest Trip
- Created measure for the title: Title for Location = "Location Analysis"

Bar Chart – Top 5 Total Bookings by Location

Visual Type: Stacked Bar Chart

Purpose: To display the top locations with the highest number of bookings.

Fields:

- Y Axis: Location
- X Axis: Total Bookings

Steps:

- Sort the chart by descending total bookings
- Top 5 locations only (filter applied)

Formatting:

- Y Axis formatting: font as Segoe UI Semibold, size-8 & Font Color as #605E5C and Maximum width is 47%
- Kept X Axis Value off
- Kept Data labels on
- Kept Title on : font as Segoe UI Semibold, size-11 & Font Color as #252423
- Tooltip shows location name and booking count
- Visual interaction enabled with slicers and other charts

Bar Chart – Most Preferred Vehicle by Pickup Location

Visual Type: Stacked Bar Chart

Purpose: To show which vehicle type is most popular as per pickup location.

Fields:

- Y Axis: Vehicle
- X Axis: PULocationID(Count)

Formatting:

- Kept the formatting same as Top 5 Total Bookings by location
- Category labels aligned left
- Visual interaction enabled with slicers and other charts

⌚ Page Formatting & Interactions

- Theme: Professional grey-white with black and blue highlights
- Navigation Pane: Left sidebar with icons and bookmarks (Overview, Time Analysis, Details)
- Data Details represented by I-icon, created bookmark on the same to show data details

Trip Details Table

- The Trip Details table is the primary fact table in the Uber Analysis Dashboard. It stores detailed records of individual Uber rides and is used for analysing ride activity, revenue trends, vehicle demand, and time-based patterns.

Description

- Trip ID**- A unique identifier for each trip. Used to track individual rides in the dataset.
- Pickup Time**- Timestamp indicating when the passenger was picked up. Enables time-based analysis such as peak hours and trip durations.
- DropOff Time**- Timestamp indicating when the trip ended. Used to calculate trip duration and completion patterns.
- Passenger Count**- Number of passengers on the ride. Useful for identifying ride-sharing behaviour and vehicle utilization.
- Trip Distance**- Total distance of the trip in miles. Helps assess fare calculation, trip categorization (short/long), and operational efficiency.
- PULocation ID (Pickup Location ID)**- Numeric code representing the pickup location. Connected to the Location Table for area name mapping.
- DOLocation ID (Drop- Location ID)**- Numeric code representing the drop location. Linked to the Location Table for meaningful location analysis.
- Payment Type**- Mode of payment used (e.g., Credit Card, Cash, Wallet, Google Pay etc). Supports financial analysis and user behaviour segmentation.
- Fare Amount**- The base fare charged for the trip, excluding additional fees. Important for analysing revenue and pricing strategies.
- Surge Fee**- Additional fee charged during high-demand periods. Helps track surge pricing trends and demand spikes.
- Vehicle Type**- Specifies the type of Uber service used (e.g., UberX, UberXL, Uber Black, Uber Green etc). Enables vehicle-level demand and service preference analysis.

Location Table

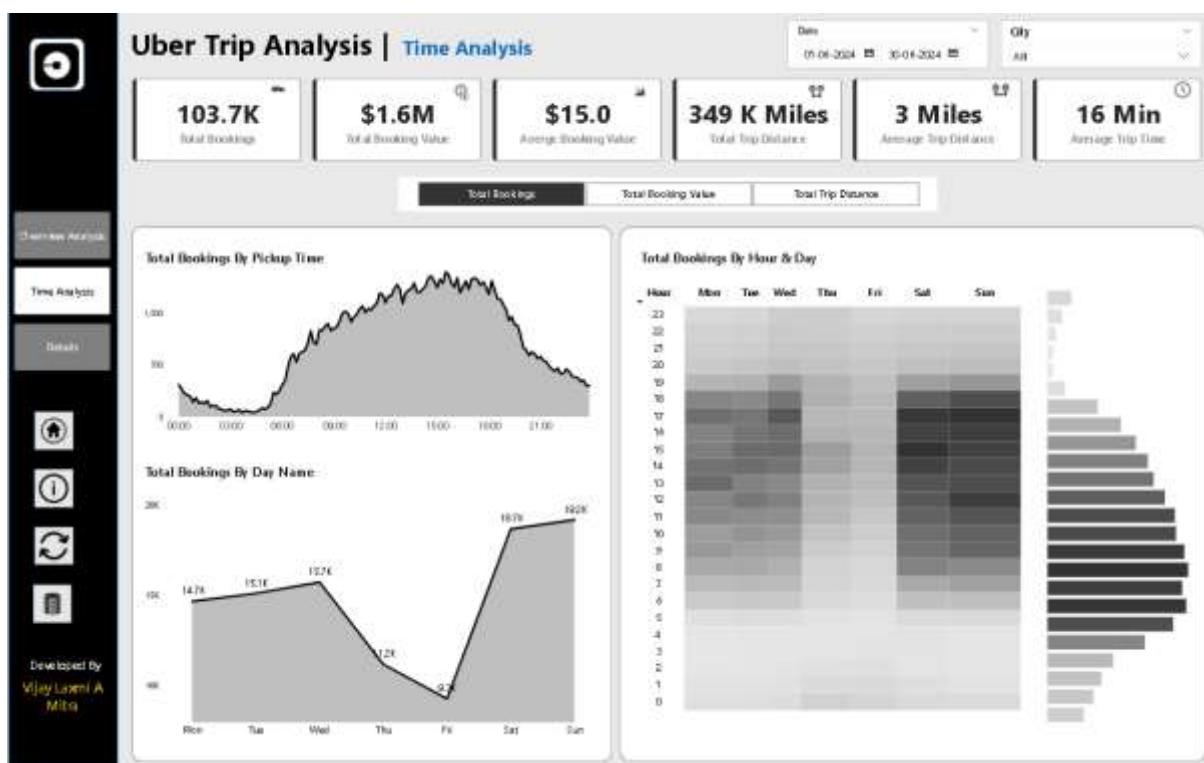
The Location Table is a dimension table that decodes numerical location IDs into readable location names and corresponding cities. It is essential for mapping pickup and drop-off points in visualizations and supports geographic analysis.

Description

- Location ID**- Unique identifier for each location. Acts as a foreign key in trip data to link geographic context.
- Location**- Name of the neighbourhood or area associated with the location ID. Used for pickup/drop-off visualizations.

- **City-** The city in which the location exists. Helps with city-level segmentation and geographic filtering.
 - **Tooltips:** Customized for all charts to show relevant detail on hover
 - **Icons:** Used for metrics, farthest trip, and navigation
 - **Bookmarks:** Used for tab switching/navigation
 - **Interactions:** Clicking on charts filters other visuals (cross-filtering enabled)
-

TIME ANALYSIS



⌚ Uber Trip Analysis – Time Analysis Page Documentation

The Time Analysis page helps identify when Uber trips are most in demand, both by time of day and day of the week. This view supports operational planning and demand

forecasting by highlighting peak and off-peak hours, day-wise patterns, and booking volume distribution.

Top KPI Cards

Visual Type: Card (New)

Visuals Included:

- Total Bookings
- Total Booking Value
- Average Booking Value
- Total Trip Distance
- Average Trip Distance
- Average Trip Time

Purpose:

To quickly provide key performance indicators for time-based trip analysis.

DAX Measures Used (Same as Overview Page)

Formatting used (Same as Overview page). I just duplicate the overview page and renamed the page as "Time Analysis".

Slicers – Date & City

Visual Type: Slicer

Purpose:

To filter all visuals based on selected date range and city.

Fields Used:

- Date from Calender Table
- City (text column)

Formatting:

- Between style for Date slicer
- Dropdown style for City
- For City slicer Multi-Select selection is used
- Background is kept ON

- Syncs with all visuals on the page

Visual: Total Bookings by Pickup Time

Visual Type: Area Chart

Purpose:

To visualize trip demand throughout the day. Helps identify peak operating hours.

Fields Used:

- X-axis: Pickup Hour -HH-MM-SS(0–23)
- Y-axis: Dynamic Measures

DAX :

Pickup Hour(HH MM SS) = TIME(HOUR('Trip Details'[Pickup Time]),MINUTE('Trip Details'[Pickup Time]),SECOND('Trip Details'[Pickup Time]))

Steps to Build:

1. Create a calculated column for pickup hour.
2. Use Area Chart visual.
3. Drag Pickup Hour to Axis and Dynamic Measures columns to Y-Axis.

Formatting:

- Filled area in grey for a clean look
- Kept values Font as Seoge UI semibold, font size as 9, Color code- #605E5C
- Kept both the gridlines off
- Background is kept on.
- Shaded Area is on , Area Transparency is 75%
- Data labels and series labels are kept off
- Smoothed line for better readability
- Tooltip shows hour and number of bookings
- Used Dynamic Title using DAX function 1: Title for By Pickup Time = SELECTEDVALUE('Dynamic Measures'[Dynamic Title]) & " By Pickup Time"

Visual: Total Bookings by Day Name

Visual Type: Area Chart

Purpose:

To compare booking volume across different days of the week.

Fields Used:

- X-axis: Day Name from Calender Tabl (Mon–Sun)
- Y-axis: Dynamic Measures

DAX Column:

DAX:

Day Name = FORMAT('Calender Table'[Date], "ddd")

Steps to Build:

1. Create Day Name column.
2. Use Area Chart visual.
3. Drag Day Name to X-axis, Dynamic Measures in Y-axis
4. Sort by Day Name in the Ascending Order.

Formatting:

- Grey shaded area with black border line
- Data labels enabled (e.g., "18.7K" for Saturday)
- Used Dynamic Title using DAX Function: Title for By Day Name = SELECTEDVALUE('Dynamic Measures'[Dynamic Title]) & " By Day Name"
- Rest of the formatting is same as "Pick hour" visual

Visual: Total Bookings by Hour & Day

Visual Type: Matrix Table with Conditional Formatting (Heatmap Style)

Purpose: To visualize when demand is highest throughout the week — combining hours and weekdays into a time grid.

Fields Used:

- Rows: Pickup Hour (0–23)
- Columns: Day Name (Mon–Sun)
- Values: Dynamic Measures

Steps to Build:

1. Add Matrix visual.
2. Rows → Pickup Hour
3. Columns → Day Name
4. Values → Dynamic Measures which includes Total Booking, Total Booking Value & Total Trip Distance

DAX Used:

(Same Pickup Hour and Day Name as above)

Formatting:

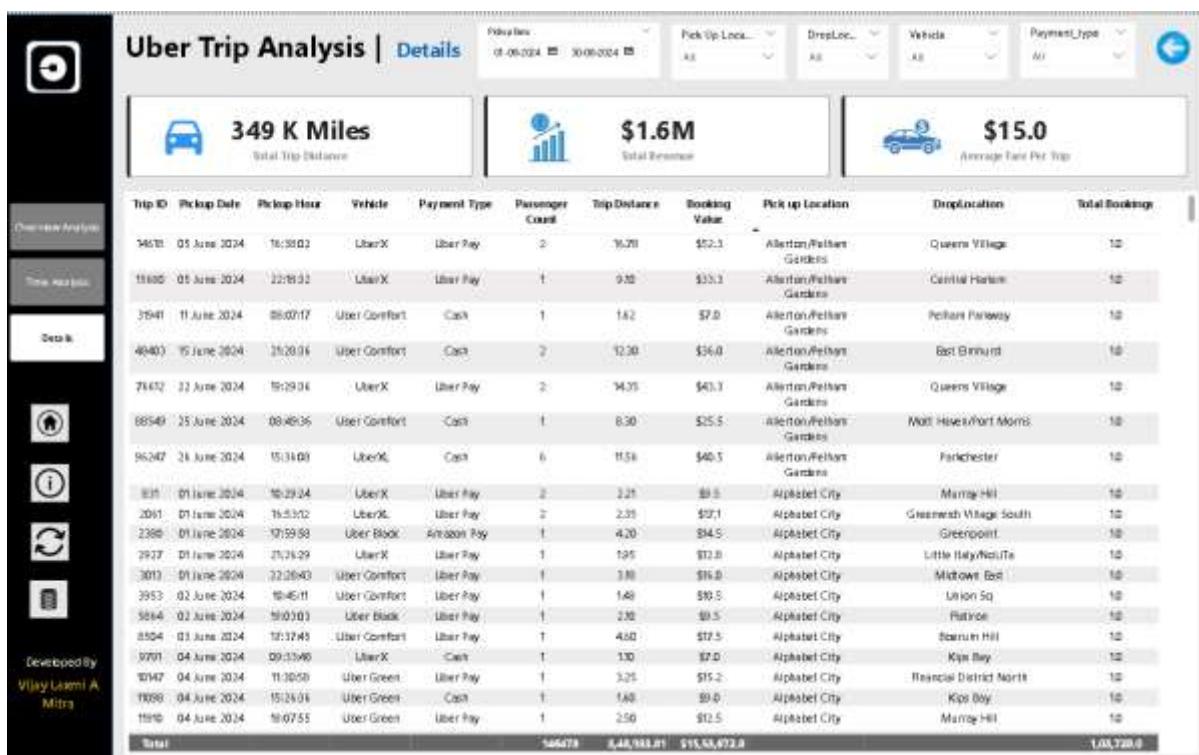
- Conditional formatting with gradient for Total Booking- Format Style- Gradients, Field- Total Bookings, Minimum(Lowest Value) color code: #E6E6E6, Center(Middle Value) color code: #B3B3B3, Maximum(Highest Value) color code: #B3B3B3
- Conditional formatting with gradient for Total Booking Value- Format Style- Gradients, Field- Total Booking Value, Minimum(Lowest Value) color code: # A0A7D8, Center(Middle Value) color code: #717BC5, Maximum(Highest Value) color code: #414FB1
- Conditional formatting with gradient for Total Trip Distance- Format Style- Gradients, Field- Total Trip Distance, Minimum(Lowest Value) color code: #E669B9, Center(Middle Value) color code: #A8337D, Maximum(Highest Value) color code: #702254
- Layout & Style Presets: Style is Default, Layout is Compact, Indentation is 10
- Border applied for All, color code is #E6E6E6 & Width is 1 Px
- In the values section, Font is Seoge UI Semibold, font Size is 11, Text Color and Alternate text color code is #252423, background color is white and alignment is centre.
- Column Header - Font is Seoge UI Semibold, font Size is 10, Text Color and Alternate text color code is #252423
- RowHeader - Font is Seoge UI Semibold, font Size is 10, Text Color code is #252423 and Background color is white.
- Column grand total and row grand total is off.
- Both the Gridlines are off.

- Background is off, Tooltip is On.
- Interactive: Hover to reveal exact booking count

Page Formatting & Interactions

- **Layout Style:** Symmetrical with visuals split into left and right panels
- **Color Theme:** Canvas Background color code is #CCCCCC, Transparency is 56%
- **Interactions:**
 - Clicking on slicers filters all visuals
 - Matrix and area charts interact with each other
- **Bookmarks:**
 - Navigation panel (left) connects to “Overview” and “Details” pages via bookmarks
- **Tooltips:**
 - All visuals have tooltips enabled for deeper insights
- **Visual Titles:** Clear and descriptive Dynamic titles

Details Page



Purpose of the Page

This page is the "Details" view of Uber Analysis dashboard. It provides a record-level breakdown of all trips for the selected time period, letting users drill down into individual ride data while also showing key KPIs at the top for quick reference.

Page Layout

Top Navigation & Filters

- **Date Range Filter** – Allows selection of a specific date range (currently 01–06–2024 to 30–06–2024).
- **Pick Up Location Filter** – Dropdown to filter trips by pickup area.
- **Drop-off Location Filter** – Dropdown to filter trips by drop location.
- **Vehicle Filter** – Lets you filter trips by vehicle type (UberX, Uber Black, Uber Comfort, etc.).
- **Payment Type Filter** – Lets you filter by payment method (Cash, Uber Pay, Amazon Pay, etc.).
- **Navigation Icon (Top Right)** – Likely used to go back to the overview page.

KPI Cards (Top Row)

Three large KPI cards provide high-level metrics based on applied filters:

1. **Total Trip Distance** – Shows total miles travelled (e.g., 349K Miles).
2. **Total Revenue** – Displays total booking value (e.g., \$1.6M).
3. **Average Fare Per Trip** – Displays average revenue per ride (e.g., \$15.0).

Main Data Table

This central table visual lists trip-level details with the following columns:

- **Trip ID** – Unique identifier for each ride.
- **Pickup Date** – Date when the trip started.
- **Pickup Hour** – Exact start time of the trip.

- **Vehicle** – Category of Uber ride (e.g., Uber Black).
- **Payment Type** – Mode of payment used by the passenger.
- **Passenger Count** – Number of passengers for that ride.
- **Trip Distance** – Distance travelled during the trip.
- **Booking Value** – Fare/booking value for the ride.
- **Pick Up Location** – Starting neighbourhood/location.
- **Drop Location** – Destination neighbourhood/location.
- **Total Bookings** – Number of bookings for the given Trip ID (here, mostly 1.0 per row).

At the bottom of the table, a totals row aggregates:

- **Total Trips:** 146,478
- **Total Trip Distance:** 348,933.81
- **Total Booking Value:** \$15,53,672.8
- **Total Bookings:** 1,03,728

Left Navigation Panel

- **Overview Analysis** – Links to the summary dashboard.
- **Time Analysis** – Links to time-based trend visualizations.
- **Details** – Current page (highlighted).
- **Home Icon** – It returns to the home page (Bookmark is created to show the home page of this page).
- **Info Icon** – It opens a page description.(Book Mark is created to show the complete details of the trip)
- **Refresh Icon** – It Clears the filters used and accordingly it refreshes the visuals.
- **Data Icon** – It is linked to a website
- **Developer Credit** – *Developed By Vijay Laxmi A Mitra.*

Key Functional Use

- **Audit & Verification:** Check individual trip data for accuracy.
- **Filter-Driven Analysis:** Narrow down trips by date, location, vehicle type, or payment method.
- **Performance Tracking:** See how total distance, revenue, and fare averages change with filters.
- **Pattern Identification:** Spot high-value trips, popular routes, and preferred payment modes.