

# Logistics & Transportation- Fleet Performance & Delivery Efficiency

## Data Cleaning & Modelling

1. Fix missing fuel consumption values (use avg. per vehicle type).

Found Value "0" in "Fuel\_Consumed\_L" Column > Found Average value > Replaced it

The screenshot illustrates the steps taken to clean the 'Fuel\_Consumed\_L' column in the Power BI Data Editor.

**Top Left:** A screenshot of the Power BI formula bar. It shows the formula `= List.Average(#"Changed Type"[Fuel_Consumed_L])` with a result of `89.529`. Above the formula bar are three buttons: Statistics, Standard, and Scientific, with Statistics selected. Below the formula bar is a dropdown menu with options: Sum, Minimum, Maximum, Median, Average (which is highlighted), Standard Deviation, Count Values, and Count Distinct Values.

**Middle Left:** A screenshot of the Power BI context menu for the 'Fuel\_Consumed\_L' column. The 'Replace Values...' option is highlighted in yellow. Other options include Copy, Remove, Remove Other Columns, Duplicate Column, Add Column From Examples..., Remove Duplicates, Remove Errors, Change Type, Transform, Group By..., Fill, Unpivot Columns, Unpivot Other Columns, Unpivot Only Selected Columns, Rename..., Move, Drill Down, and Add as New Query.

**Bottom Right:** A screenshot of the 'Replace Values' dialog box. It has two input fields: 'Value To Find' containing '0' and 'Replace With' containing '89.52'. At the bottom right are 'OK' and 'Cancel' buttons.

Blue arrows indicate the flow from the formula bar to the context menu, and from the context menu to the 'Replace Values' dialog box.

## 2.Relate Trips with Vehicle Master.

We have two table called **Trip** and **Vehicle Master** and each table has a common column called **Vehicle\_ID**.

# Trip ID                              Vehicle ID

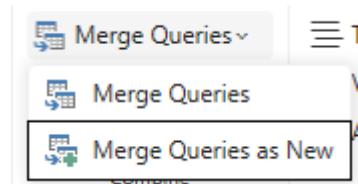
Vehicle_ID	Trip_ID	Date	Origin	Destination	Distance_m	Duration_h	On-Time
1	1001	2023-01-01	Mumbai	Bangalore	1200	10:00	Yes
2	1001	2023-01-01	Mumbai	Pune	350	03:30	Yes
3	1001	2023-01-01	VGS	Mumbai	340	04:30	No
4	1001	2023-01-01	VGS	Pune	340	04:30	No
5	1001	2023-01-01	VGS	Bangalore	1200	10:00	Yes
6	1001	2023-01-01	VGS	Chennai	1200	10:00	Yes
7	1001	2023-01-01	VGS	Surat	700	08:00	Yes
8	1001	2023-01-01	VGS	Pune	340	04:30	No
9	1009	2023-01-01	VGS	Delhi	1200	10:00	Yes
10	1002	2023-01-01	VGS	Hyderabad	1200	10:00	Yes
11	1002	2023-01-01	VGS	Bangalore	1200	10:00	Yes
12	1002	2023-01-01	VGS	Kolkata	1200	10:00	Yes
13	1002	2023-01-01	VGS	Delhi	1200	10:00	Yes
14	1002	2023-01-01	VGS	Chennai	1200	10:00	Yes
15	1002	2023-01-01	VGS	Kolkata	1200	10:00	Yes
16	1002	2023-01-01	VGS	Mumbai	1200	10:00	Yes
17	1002	2023-01-01	VGS	Pune	1200	10:00	Yes
18	1002	2023-01-01	VGS	Bangalore	1200	10:00	Yes
19	1002	2023-01-01	VGS	Surat	1200	10:00	Yes
20	1002	2023-01-01	VGS	Mumbai	1200	10:00	Yes
21	1002	2023-01-01	VGS	Pune	1200	10:00	Yes
22	1002	2023-01-01	VGS	Bangalore	1200	10:00	Yes
23	1002	2023-01-01	VGS	Chennai	1200	10:00	Yes
24	1002	2023-01-01	VGS	Kolkata	1200	10:00	Yes
25	1002	2023-01-01	VGS	Delhi	1200	10:00	Yes
26	1002	2023-01-01	VGS	Chennai	1200	10:00	Yes
27	1007	2023-01-01	VGS	Pune	1200	10:00	Yes
28	1009	2023-01-01	VGS	Pune	1200	10:00	Yes
29	1009	2023-01-01	VGS	Mumbai	1200	10:00	Yes
30	1009	2023-01-01	VGS	Kolkata	1200	10:00	Yes
31	1009	2023-01-01	VGS	Hyderabad	1200	10:00	Yes
32	1009	2023-01-01	VGS	Bangalore	1200	10:00	Yes
33	1009	2023-01-01	VGS	Chennai	1200	10:00	Yes
34	1009	2023-01-01	VGS	Delhi	1200	10:00	Yes
35	1009	2023-01-01	VGS	Pune	1200	10:00	Yes
36	1009	2023-01-01	VGS	Bangalore	1200	10:00	Yes

Vehicle_ID	Type	Capacity_kg	Maintenance_Cost
1	VGS	8424	5615
2	VGZ	2970	2770
3	VGS	1207	2427
4	VGS	8009	3009
5	VGS	9941	17753
6	VGS	6953	3914
7	VGS	2598	12887

# Vehicle\_ID Column

We're going to relate a both table using “Merge queries”



I Used **Left Outer** for this joins

X

## Merge

Select tables and matching columns to create a merged table.

Trip\_ID (Trip\_Data)

Trip_ID	Vehicle_ID	Driver_ID	Origin	Destination	Distance_km	Fuel_Consumed_L	Delivery_Status
T001	V04	D01	Delhi	Pune	1173	108.42	On-Time
T002	V06	D08	Mumbai	Bangalore	1727	161.33	On-Time
T003	V06	D08	Mumbai	Pune	1459	154.7	On-Time
T004	V04	D09	Hyderabad	Pune	382	26.6	On-Time
T005	V06	D09	Hyderabad	Pune	1173	108.42	On-Time

Vehicle\_ID (Vehicle\_Master)

Vehicle_ID	Vehicle_Type	Capacity_kg	Maintenance_Cost
V01	Van	8424	5633
V02	Truck	1970	6776
V03	Truck	1207	18031
V04	Mini-Truck	8803	9033
V05	Truck	9941	17751

Join Kind

Left Outer (all from first, matching from second)

Use fuzzy matching to perform the merge

▷ Fuzzy matching options

✓ The selection matches 50 of 50 rows from the first table.

OK Cancel

Delivery\_Date

Vehicle\_ID (Vehicle\_Master)

Search Columns to Expand A Z

Expand Aggregate

(Select All Columns)

Vehicle\_ID

Vehicle\_Type

Capacity\_kg

Maintenance\_Cost

Use original column name as prefix

OK Cancel

Renamed as **Vehicle\_Report**

## DAX Measures

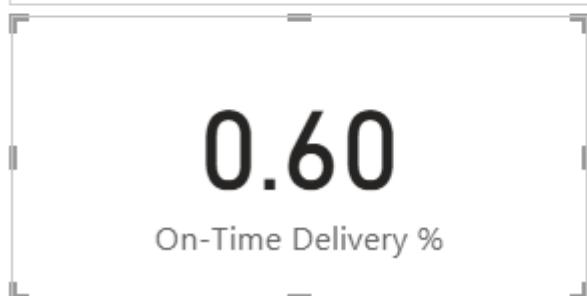
### 1. Fuel Efficiency = Distance / Fuel Consumed

```
1 Fuel_Efficiency = DIVIDE(SUM(Vehicle_Report[Distance_km]),SUM('Trip_ID_(Trip_Data)'[Fuel_Consumed_L]),0)
```



### 2. On-Time Delivery % = On-Time Trips / Total Trips

```
1 On-Time_Delivery % =
2 DIVIDE(
3     CALCULATE(
4         COUNTROWS('Vehicle_Report'),
5         'Vehicle_Report'[Delivery_status] = "On-Time"
6     ),
7     COUNTROWS('Vehicle_Report'),
8     0
9 )
```



3.Cost per km = (Fuel Cost + Maintenance Cost) / Distance

```
1 Fuel Cost = 85.50
```

85.50

Fuel Cost

```
1 Cost per km =
2 DIVIDE(
3     [Fuel Cost] + SUM('Vehicle_Report'[Maintenance_Cost]),
4     SUM('Vehicle_Report'[Distance_km]),
5     0
6 )
```

9.82

Cost per km

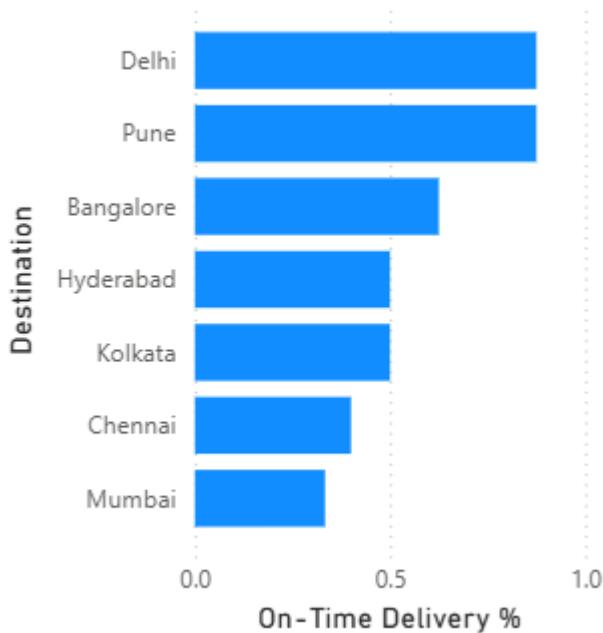
85.50

Fuel Cost

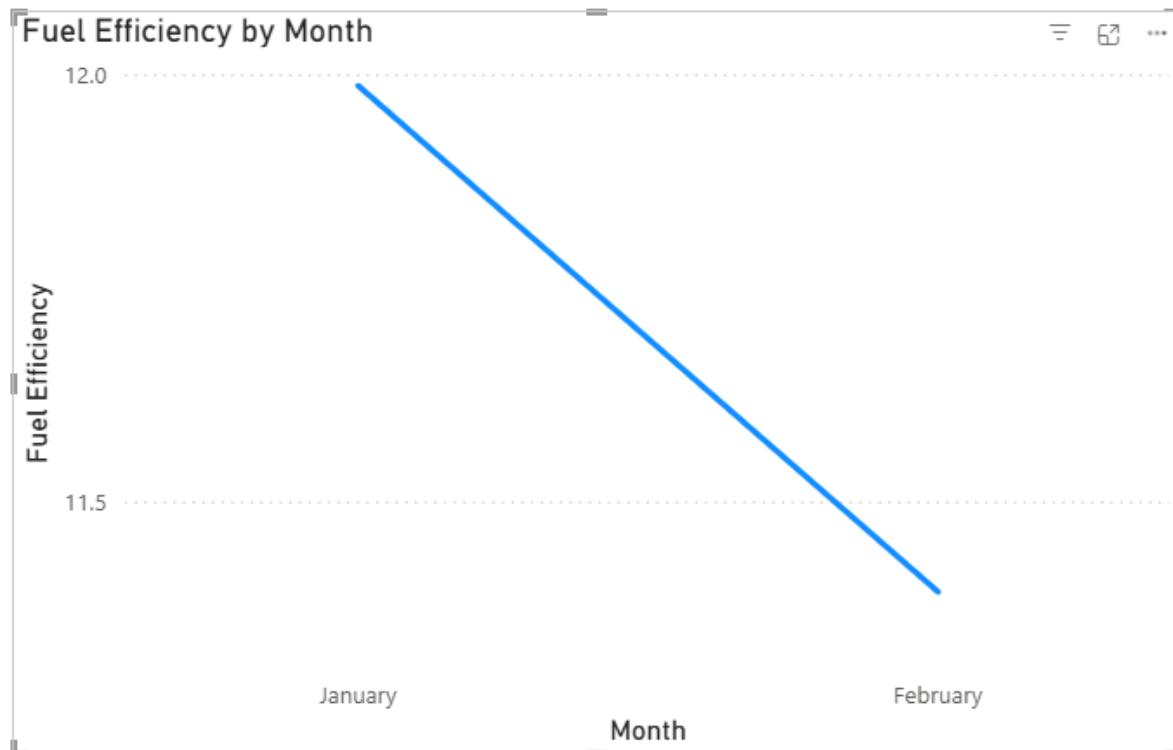
## Visualization

### 1. Bar chart: On-Time Delivery % by Route.

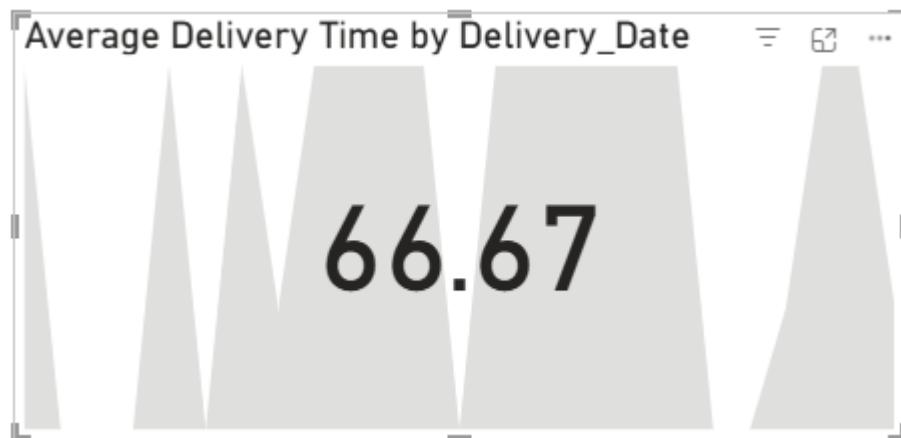
On-Time Delivery % by Destination



### 2. Line chart: Fuel Efficiency trend by month.



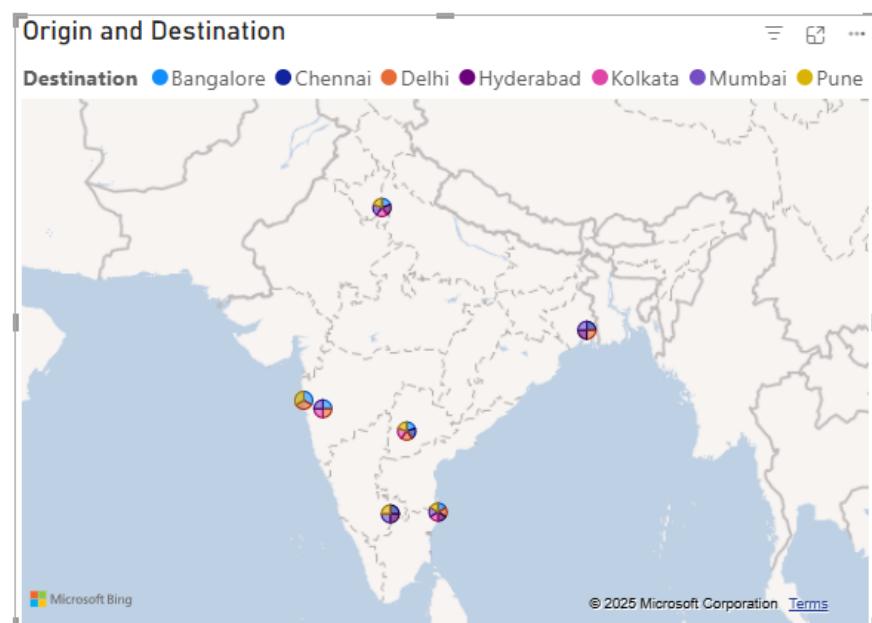
### 3. KPI cards: Avg. Delivery Time, Cost per km.



Cost per km by Destination



### 4. Map visual: Delivery performance by route (Origin → Destination).



## A transport operations dashboard to optimize routes and fleet usage.

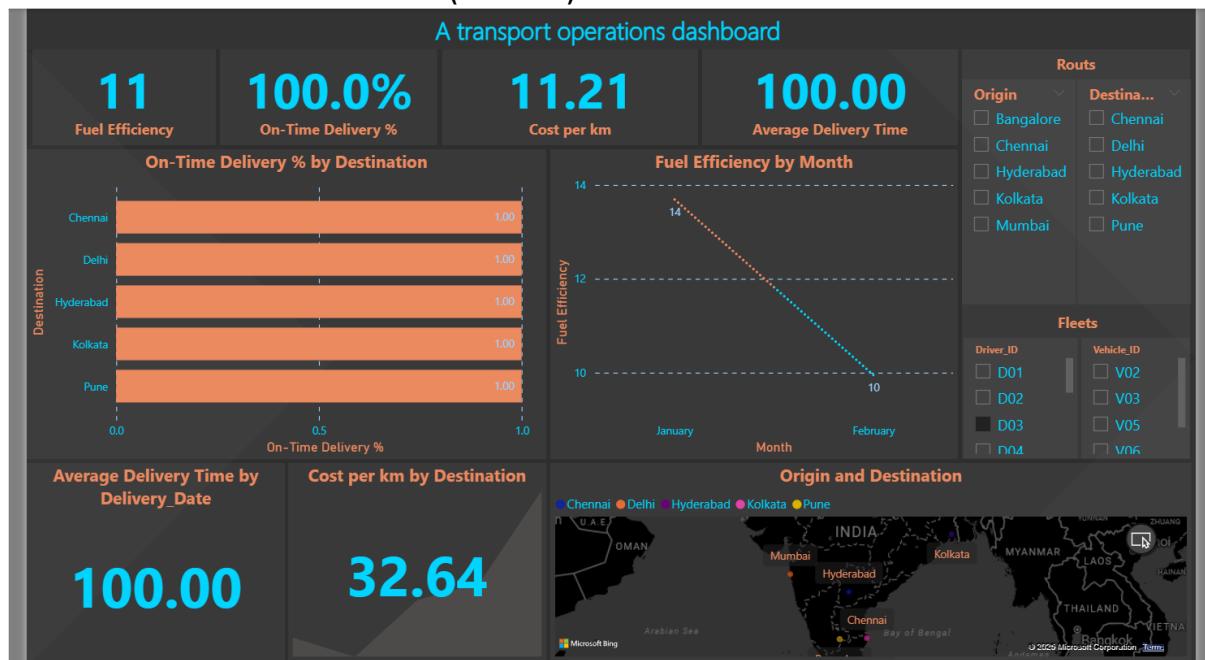
### Dashboard view 1 : Normal



### Dashboard view 2 : Based on Rout (Delhi to Pune)



### Dashboard view 3 : Based on Fleet (Driver ID)



### Mobile Dashboard View: 1

