



London, or as the Romans called it "Londonium"! Home to [over 8.5 million residents](#) who speak over [300 languages](#). While the City of London is a little over one square mile (hence its nickname "The Square Mile"), Greater London has grown to encompass 32 boroughs spanning a total area of 606 square miles!



Given the city's roads were originally designed for horse and cart, this area and population growth has required the development of an efficient public transport system! Since the year 2000, this has been through the local government body called **Transport for London**, or *TfL*, which is managed by the London Mayor's office. Their remit covers the London Underground, Overground, Docklands Light Railway (DLR), buses, trams, river services (clipper and [Emirates Airline cable car](#)), roads, and even taxis.

The Mayor of London's office make their data available to the public [here](#). In this project, you will work with a slightly modified version of a dataset containing information about public transport journey volume by transport type.

The data has been loaded into a **Snowflake** database called `TFL` with a single table called `JOURNEYS`, including the following data:

## TFL.JOURNEYS

Column	Definition	Data type
<code>MONTH</code>	Month in number format, e.g., <code>1</code> equals January	<code>INTEGER</code>
<code>YEAR</code>	Year	<code>INTEGER</code>
<code>DAYS</code>	Number of days in the given month	<code>INTEGER</code>
<code>REPORT_DATE</code>	Date that the data was reported	<code>DATE</code>
<code>JOURNEY_TYPE</code>	Method of transport used	<code>VARCHAR</code>
<code>JOURNEYS_MILLIONS</code>	Millions of journeys, measured in decimals	<code>FLOAT</code>

Note that in *Snowflake* all databases, tables, and columns are **upper case** by default.

You will execute SQL queries to answer three questions, as listed in the instructions.

London Public Transport DataFrame as most\_popu

```
-- most_popular_transport_types
SELECT
    JOURNEY_TYPE,
    SUM(JOURNEYS_MILLIONS) AS TOTAL_JOURNEYS_MILLIONS
FROM TFL.JOURNEYS
GROUP BY JOURNEY_TYPE
ORDER BY TOTAL_JOURNEYS_MILLIONS DESC;
```

index	...	↑↓	JOURNEY_TYPE
		0	Bus
		1	Underground & DLR
		2	Overground
		3	TfL Rail
		4	Tram
		5	Emirates Airline

Rows: 6

[Expand](#)

London Public Transport DataFrame as emirates.

```
-- emirates_airline_popularity
SELECT
    MONTH,
    YEAR,
    ROUND(JOURNEYS_MILLIONS, 2) AS ROUNDED_JOURNEYS_MILLIONS
FROM TFL.JOURNEYS
WHERE JOURNEY_TYPE = 'Emirates Airline'
    AND JOURNEYS_MILLIONS IS NOT NULL
ORDER BY ROUNDED_JOURNEYS_MILLIONS DESC
LIMIT 5;
```

index	...	↑↓	MONTH	...	↑↓	YEAR	...	↑↓	R
		0			5			2012	
		1			6			2012	
		2			4			2012	
		3			5			2013	
		4			5			2015	

Rows: 5

[Expand](#)

London Public Transport DataFrame as least\_po

```
-- least_popular_years_tube
SELECT
    YEAR,
    JOURNEY_TYPE,
    SUM(JOURNEYS_MILLIONS) AS TOTAL_JOURNEYS_MILLIONS
FROM TFL.JOURNEYS
WHERE JOURNEY_TYPE LIKE '%Underground%'
GROUP BY YEAR, JOURNEY_TYPE
ORDER BY TOTAL_JOURNEYS_MILLIONS ASC
LIMIT 5;
```

index	...	↑↓	YEAR	...	↑↓	JOURNEY_TYPE
		0	2020			Underground & DLR
		1	2021			Underground & DLR
		2	2022			Underground & DLR
		3	2010			Underground & DLR
		4	2011			Underground & DLR

Rows: 5

[Expand](#)

How likely are you to recommend DataLab to a friend or co-worker?



Not at all likely

0

1

2

3

4

5

6

7

8

9

10

Extremely likely

powered by InMoment