An Exploratory Data Analysis of the Impact of Public Transport Adoption on Pollution Reduction in Singapore

An Introduction to the Research Space

Introduction

Climate change is one of the most pressing global challenges of our time, affecting economies, health, and ecosystems. Observing motor vehicle and public transport usage in Singapore, I began questioning whether pollution levels were affected by this. This project aims to explore the potential correlation between motor vehicle and public transport usage and pollution levels, contributing to a better understanding of transportation's environmental impact. We will also be looking at if whether increased usage of public tranport has decreased the levels of pollution. Our problem statement would be **increased usage of public transport in Singapore has decreased the amount of pollution caused by motor vehicles**.

Aims and Objectives

Aims

This research aims to:

1. Understand the relationship between public transport usage and pollution levels in Singapore.
2. Explore how motor vehicle usage correlates with pollution levels.
3. Provide actionable insights for sustainable urban mobility policies.

Objectives

To achieve the above aims, the study will:

- Analyze trends in motor vehicle usage and pollution levels.
- Identify correlations between motor vehicle usage, public transport usage, and pollution levels.
- Highlight the environmental impact of transitioning to public transport.
- Recommend strategies for reducing motor vehicle emissions and promoting public transport.|

Acquire a Dataset

For this project, we will utilize datasets from the following sources:

1. **Pollution Data**: Obtained from Data.gov.sg. This dataset contains air quality indices such as PM2.5, PM10, and other key pollutants over time.
2. **Public Transport Data**: Data on MRT ridership and bus usage trends from Data.gov.sg.
3. **Motor Vehicle Data**: Vehicle ownership data from the **Land Transport Authority (LTA)**, which includes annual motor vehicle population by type.

Utilizing the Dataset

The analysis will be conducted in **Jupyter Notebook**, employing:

1. **Data cleaning** to ensure consistency and accuracy.
2. **Exploratory Data Analysis (EDA)** to uncover trends and correlations using:
   - Statistical summaries.
   - Data visualizations (line charts, scatter plots, etc.).
3. **Insightful conclusions**, focusing on actionable recommendations for sustainable change.

Writing Style

The project will follow a structured format for clarity and coherence. The information would be straightforward so that the logical flow ensures the project is accessible and impactful.

Clear Summary of the Area of Research Chosen

This project investigates the relation between pollution in Singapore and motor vehicle and public transport usage.By addressing these issues, this project not only highlights existing challenges but also paves the way for data-driven solutions to create a greener Singapore.

Relevancy of data and justified use

Importing libraries

In [28]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
from scipy.stats import ttest_ind
```

```
# Width = 16, Height = 6
DIMS=(16, 6)
```

Origin of data

The data from this project all comes from trustable government websites like National Environmnet Agency and the Land Transport Authority websites. In Singapore, pollution, public transport, and motor vehicle data are collected by agencies like the National Environment Agency (NEA) and the Land Transport Authority (LTA) using advanced monitoring systems. Pollution data comes from air quality stations, remote sensing tools, and water quality testing. Public transport data is gathered through GPS systems, automated fare collection (e.g., EZ-Link), and passenger counting systems managed by LTA and transport operators. Motor vehicle data originates from vehicle registration records, traffic monitoring systems (e.g., ERP gantries), and emissions tests during inspections. These datasets are processed and made accessible.

Appropriateness of data source

- These datasets provide **time-series data**, allowing us to observe trends over specific periods.
- They include key columns directly relevant to the research question:
  - Pollution data includes pollutant levels (e.g., PM2.5, CO_2) over time.
  - Public transport data includes monthly ridership trends.
  - Motor vehicle data contains vehicle population metrics by type (e.g., private cars, buses, etc.).
- The datasets are in a format suitable for analysis, such as CSV files that can easily be converted into pandas DataFrames.

The identifiable case for working with this data

Each dataset aligns well with the research objectives:

- Vehicle usage data includes columns such as 'Year', 'Category', 'Type', and 'Number', enabling analysis of trends over time.
- Pollution data includes multiple datasets of all pollutants.
- Public transport data includes ridership statistics, allowing examination of shifts towards sustainable transport modes.

These clearly defined columns and structures make the datasets suitable for merging and comprehensive analysis.

How the format of data is suitable for analysis

All datasets are provided in CSV format, which is:

1. Easily convertible to Pandas DataFrames for efficient data manipulation.

2. Structured with clearly defined columns, facilitating merging and analysis.
3. Compatible with Python libraries like Pandas, NumPy, and Matplotlib, which enable robust numerical and statistical analysis.

Consideration of two other datasets

1. Traffic Congestion Data:
    - **Strengths**: Provides insights into road usage patterns, peak traffic times, and the effectiveness of transport policies, directly linking to vehicle trends and public transport utilization.
    - **Weaknesses**: May not accurately reflect environmental impacts or the contribution of specific vehicle types to congestion.
2. Renewable Energy Adoption Data:
    - **Strengths**: Highlights the role of alternative energy in reducing environmental impact.
    - **Weaknesses**: May not be directly linked to transportation trends in Singapore.

Incorporating these datasets could complement the current analysis, offering a broader perspective on sustainability and environmental policies.

Ethics of use of data

Origin of data

1. **Vehicle Usage Data and Public Transport Data**:
    - **Type**: Open Data
    - **Provenance**: Provided by the Land Transport Authority (LTA), Singapore.
    - **Licensing**: Governed by Singapore's Open Data Licensing terms, which permit usage for non-commercial research and analysis purposes.
    - Data taken from https://datamall.lta.gov.sg/content/datamall/en/static-data.html. I downloaded the csv datasets for the respective categories from this website.
2. **Pollution Data**:
    - **Type**: Open Data
    - **Provenance**: Sourced from the National Environment Agency (NEA) of Singapore.
    - **Licensing**: Available for public access and use, subject to NEA's data usage policies.
    - Data taken from https://data.gov.sg/datasets?q=&query=pollutant&groups=&organization=&page=1&resultId=d_fe37906a018
I downloaded the csv datasets for the respective categories from this website.

Considerations about usage/reuse of data

1. **Creation of Intellectual Property**:

    - The analysis has the potential to generate new insights, which may be considered intellectual property in the form of research findings, models, or visualizations. With this analysis, we will know exactly how much each vehicle has affected pollution in Singapore and from there create better strategies to reduce the pollution levels for a more green Sinagpore.

2. **Attribution**:

    - Proper attribution is provided to the original data sources (LTA and NEA) in all research outputs, ensuring compliance with licensing requirements.

Implications and Considerations of utilising data

The datasets do not contain personally identifiable information, ensuring anonymity by design and all findings are contextualized within broader environmental and social factors to avoid harmful assumptions. Additionally, the research actively avoids creating narratives that could lead to harmful assumptions or stigmatization of specific communities. The purpose of the analysis is to provide actionable, equitable insights to support sustainable development and climate action, rather than to assign blame or propagate unintended biases. By maintaining transparency in methodology and acknowledging the limitations of the data, the research ensures its outputs are both responsible and constructive.

The datasets are loaded directly into the Jupyter Notebook using standard libraries such as pandas for easy manipulation and visualization. The data files are stored in accessible formats (e.g., .csv), ensuring compatibility with common data analysis tools. The datasets are anonymized by design, containing only aggregate and non-identifiable data points (e.g., average pollution levels, vehicle usage rates). No personally identifiable information (PII) is present in the datasets, eliminating the risk of re-identification of individuals.

Potential biases of the dataset

The vehicle usage and public transport datasets may underrepresent certain socio-economic groups (e.g., low-income users without vehicles).Pollution data is aggregated at a national level, which may obscure localized variations.

Project background

Why the Field is of Interest/Relevance:

Observing Singapore's consistently high air quality compared to other countries piqued my interest in understanding the mechanisms behind its success. Singapore's efforts to tackle pollution provide an excellent case study for understanding sustainable urban living practices. The specific role of public transportation in reducing air pollution intrigues me, as Singapore is known for its efficient and highly utilized public transport system. This raises questions about whether increased public transport adoption directly impacts pollution reduction.

Novelty of the Topic:

- While studies have examined transportation's contribution to pollution, the specific relationship between vehicle usage, public transport trends, and pollution levels in Singapore remains underexplored.
- This research addresses a gap by integrating datasets on vehicular trends, public transport usage, and pollution levels to provide actionable insights into sustainable urban planning and environmental policies.

Scope of Work:

- **Included**:
  - Analyze trends in vehicle usage and pollution levels in Singapore.
  - Identify correlations between motor vehicle and public transport usage and pollution levels.
  - Provide insights to inform sustainable transportation policies.
- **Excluded**:
  - In-depth analysis of non-transport-related pollution sources (e.g., industrial emissions).
  - Global comparative studies of transportation trends.

Steps and Stages in the Analytical Data Processing Pipeline:

- Data Acquisition: Gather vehicle usage, pollution, and public transport datasets.
- Data Cleaning: Handle missing values, ensure consistency across datasets, and format data for analysis.
- Data Integration: Merge datasets based on common columns such as year.
- Exploratory Data Analysis (EDA): Use descriptive statistics and visualizations to identify trends and patterns.
- Correlation Analysis: Evaluate relationships.
- Reporting: Summarize findings and provide information regarding my claim.

Evaluation of Aims and Objectives:

- Aims and objectives will be evaluated by:

- Validating trends and correlations through statistical measures (e.g., Pearson correlation coefficient).
- Assessing the reliability and representativeness of the data used.
- Determining whether the findings provide actionable insights for sustainable policy development.

Technical Exploration of Dataset

In [33]:
```
ozone = pd.read_csv('AirPollutantOzoneMaximum8hourMean.csv')
ozone.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24 entries, 0 to 23
Data columns (total 2 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   year                       24 non-null     int64
 1   ozone_maximum_8hour_mean   24 non-null     int64
dtypes: int64(2)
memory usage: 516.0 bytes
```

In [34]:
```
pm25 = pd.read_csv('AirPollutantParticulateMatterPM2.5.csv')
pm25.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22 entries, 0 to 21
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   year       22 non-null     int64
 1   pm25mean   22 non-null     int64
dtypes: int64(2)
memory usage: 484.0 bytes
```

In [35]:
```
pm10 = pd.read_csv('AirPollutantParticulateMatterPM1024hrMean99thPercentile.csv'
pm10.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24 entries, 0 to 23
Data columns (total 2 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   year                       24 non-null     int64
 1   pm10_24hour_mean_99th_per  24 non-null     int64
dtypes: int64(2)
memory usage: 516.0 bytes
```

In [36]:
```
no2 = pd.read_csv('AirPollutantNitrogenDioxide.csv')
no2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24 entries, 0 to 23
Data columns (total 2 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   year                   24 non-null     int64
 1   nitrogen_dioxide_mean  24 non-null     int64
dtypes: int64(2)
memory usage: 516.0 bytes
```

In [37]:
```
co = pd.read_csv('AirPollutantCarbonMonoxideMaximum8HourMean.csv')
co.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24 entries, 0 to 23
Data columns (total 2 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   year               24 non-null     int64
 1   co_max_8hour_mean  24 non-null     float64
dtypes: float64(1), int64(1)
memory usage: 516.0 bytes
```

In [38]:
```
publictransport = pd.read_csv('PublicTransportOperationAndRidershipAnnual.csv')
publictransport.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9 entries, 0 to 8
Data columns (total 35 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   DataSeries  9 non-null      object 
 1   2023        9 non-null      float64
 2   2022        9 non-null      float64
 3   2021        9 non-null      float64
 4   2020        9 non-null      float64
 5   2019        9 non-null      float64
 6   2018        9 non-null      float64
 7   2017        9 non-null      object 
 8   2016        9 non-null      object 
 9   2015        9 non-null      object 
 10  2014        9 non-null      object 
 11  2013        9 non-null      object 
 12  2012        9 non-null      object 
 13  2011        9 non-null      object 
 14  2010        9 non-null      object 
 15  2009        9 non-null      object 
 16  2008        9 non-null      object 
 17  2007        9 non-null      object 
 18  2006        9 non-null      object 
 19  2005        9 non-null      object 
 20  2004        9 non-null      object 
 21  2003        9 non-null      object 
 22  2002        9 non-null      object 
 23  2001        9 non-null      object 
 24  2000        9 non-null      object 
 25  1999        9 non-null      object 
 26  1998        9 non-null      object 
 27  1997        9 non-null      object 
 28  1996        9 non-null      object 
 29  1995        9 non-null      object 
 30  1994        9 non-null      object 
 31  1993        9 non-null      object 
 32  1992        9 non-null      object 
 33  1991        9 non-null      object 
 34  1990        9 non-null      object 
dtypes: float64(6), object(29)
memory usage: 2.6+ KB
```

In [39]:
```
motor = pd.read_csv('MVP01-1_MVP_by_type.csv')
motor.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 391 entries, 0 to 390
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   year      391 non-null    int64 
 1   category  391 non-null    object
 2   type      391 non-null    object
 3   number    391 non-null    int64 
dtypes: int64(2), object(2)
memory usage: 12.3+ KB
```

Merging of pollution data for easier analysis

In [42]:
```python
# Combine all dataframes into one for ease of analysis
pollution_df_1 = pd.merge(co, no2, on='year', how='outer')
pollution_df_2 = pd.merge(pollution_df_1, ozone, on='year', how='outer')
pollution_df_3 = pd.merge(pollution_df_2, pm25, on='year', how='outer')
pollution_df = pd.merge(pollution_df_3, pm10, on='year', how='outer')

# Display the first 10 rows
pollution_df.head(10)
```

Out[42]:

	year	co_max_8hour_mean	nitrogen_dioxide_mean	ozone_maximum_8hour_mean	pn
0	2000	3.7	30	112	
1	2001	4.2	26	133	
2	2002	2.7	27	131	
3	2003	3.2	24	118	
4	2004	2.8	26	146	
5	2005	2.4	25	159	
6	2006	2.6	24	136	
7	2007	1.7	22	206	
8	2008	1.6	22	183	
9	2009	1.9	22	105	

In [47]:
```python
pollution_df.describe()
```

Out[47]:

	year	co_max_8hour_mean	nitrogen_dioxide_mean	ozone_maximum_8hou
count	24.000000	24.000000	24.000000	24
mean	2011.500000	2.370833	24.375000	142
std	7.071068	1.016592	2.081231	25
min	2000.000000	1.200000	20.000000	105
25%	2005.750000	1.700000	23.000000	123
50%	2011.500000	2.000000	25.000000	137
75%	2017.250000	2.725000	25.250000	152
max	2023.000000	5.500000	30.000000	206

It is evident that all the data sets contain 24 data points, except for PM 2.5, which has only 22. This discrepancy is the reason for the 'NaN' values in the table above, and it will be addressed during the data cleaning process.

Data cleaning to remove illegal values

In [54]:

```python
# Function to check for illegal and missing values in numeric columns
def check_numeric_column(df, column_name):
    # Define a regex pattern for valid numbers
    valid_number_pattern = re.compile(r'^-?\d*\.?\d+$')

    # Check for illegal values
    df[f"{column_name}_illegal"] = df[column_name].astype(str).apply(
        lambda x: not bool(valid_number_pattern.match(x))
    )

    # Check for missing values
    df[f"{column_name}_missing"] = df[column_name].isnull()

# Function to check if there are any illegal values in the dataset
def has_illegal_values(df, numeric_columns):
    for column in numeric_columns:
        if df[f"{column}_illegal"].any():
            return True
    return False

# File paths
file1 = 'MVP01-1_MVP_by_type.csv'
file2 = 'PublicTransportOperationAndRidershipAnnual.csv'

# Load the CSV files
data1 = pollution_df
data2 = pd.read_csv(file1)
data3 = pd.read_csv(file2)


# Identify numeric columns for checking (update as needed)
numeric_columns_data1 = data1.select_dtypes(include=['float64', 'int64']).column
numeric_columns_data2 = data2.select_dtypes(include=['float64', 'int64']).column
numeric_columns_data3 = data3.select_dtypes(include=['float64', 'int64']).column



# Apply checking to each numeric column in data1
for column in numeric_columns_data1:
    check_numeric_column(data1, column)

# Apply checking to each numeric column in data2
for column in numeric_columns_data2:
    check_numeric_column(data2, column)

# Apply checking to each numeric column in data3
for column in numeric_columns_data3:
    check_numeric_column(data3, column)

# Check if there are any illegal values
if has_illegal_values(data1, numeric_columns_data1):
    print("illegal values found in pollution_df")
if has_illegal_values(data2, numeric_columns_data2):
    print("illegal values found in MVP01-1_MVP_by_type.csv")
if has_illegal_values(data3, numeric_columns_data3):
    print("illegal values found in PublicTransportOperationAndRidershipAnnual.cs
```

```
illegal values found in pollution_df
```

Cleaning pollution dataframe

In [57]:
```
# Drop NaN values and sort by year
pol_df = pollution_df.dropna().sort_values('year', ascending=True)

# Select data from 2005 to 2023
year = range(2005, 2024)
pol_df = pol_df[pol_df['year'].isin(year)]

# Remove the year column temporarily
years = pol_df['year']  # Save the year column
pol_df = pol_df.drop('year', axis=1)

# Identify columns with non-zero standard deviation
non_zero_std_cols = pol_df.loc[:, pol_df.std() != 0]

# Mean normalization only on these columns
normalized_df = (non_zero_std_cols - non_zero_std_cols.mean()) / non_zero_std_co

# Add the year column back
normalized_df['year'] = years.values

pol_df = normalized_df
```

In [58]:
```
def check_illegal_values(df, numeric_columns):
    valid_number_pattern = re.compile(r'^-?\d*\.?\d+$')
    result = {}

    # Check for illegal and missing values
    for column in numeric_columns:
        illegal_values = ~df[column].astype(str).str.match(valid_number_pattern)
        result[column] = illegal_values

    # Summarize results
    has_illegal = any(illegal_values.any() for illegal_values in result.values()
    return result, has_illegal


# Identify numeric columns in pol_df
numeric_columns = pol_df.select_dtypes(include=['float64', 'int64']).columns

# Check for illegal values
illegal_values_result, has_illegal = check_illegal_values(pol_df, numeric_column

if has_illegal:
    print("Illegal values found in the following columns:")
    for col, illegal_mask in illegal_values_result.items():
        if illegal_mask.any():
            print(f" - {col}: {illegal_mask.sum()} illegal values")
else:
    print("No illegal values found in pol_df.")
pol_df
```

```
No illegal values found in pol_df.
```

Out[58]:

	co_max_8hour_mean	nitrogen_dioxide_mean	ozone_maximum_8hour_mean	pm25m
5	0.291425	0.732510	0.470880	1.093
6	0.500372	0.127393	-0.379428	1.606
7	-0.439887	-1.082842	2.208467	0.580
8	-0.544361	-1.082842	1.358158	-0.188
9	-0.230941	-1.082842	-1.525496	0.580
10	0.291425	-0.477724	-0.268518	0.067
11	-0.126468	0.732510	-0.860037	0.067
12	-0.230941	0.732510	-0.897007	0.580
13	3.530096	0.732510	-0.268518	0.836
14	-0.335414	0.127393	-0.416398	0.323
15	1.231685	-1.082842	0.212091	1.862
16	0.082479	1.337628	-1.155797	-0.445
17	-0.439887	0.732510	1.653918	-0.701
18	-0.126468	1.337628	0.138151	-0.445
19	-0.439887	-0.477724	-0.786097	-0.188
20	-0.962254	-2.293076	-0.046699	-1.471
21	-0.962254	0.732510	1.099369	-1.214
22	-0.439887	0.732510	-0.823067	-1.471
23	-0.648834	-0.477724	0.286030	-1.471

Data is in correct format

All data in this project is managed using Pandas DataFrames, an optimal choice for data manipulation and analysis in Python. DataFrames provide a flexible structure for handling tabular data, enabling efficient data cleaning, exploration, and visualization. The format supports operations like filtering, aggregation, and merging datasets, which are integral to this research. DataFrames also seamlessly integrate with libraries such as Matplotlib and Seaborn for generating visual insights. Compared to other formats like raw CSVs or NumPy arrays, DataFrames offer labeled indexing and better readability, ensuring clarity and accuracy in analysis. This makes them highly suitable for the project's objectives.

Out of bound values

Checking the public transport dataset so that all numeric columns have values greater than 0. We need to check for this as we need to ensure all data is positive and within range.

In [65]:
```
# Load the dataset
file_path = 'PublicTransportOperationAndRidershipAnnual.csv'  # Replace with you
data = pd.read_csv(file_path)

# Filter columns for years between 2005-2023
valid_years = [str(year) for year in range(2005, 2024)]
new_public_data = data[['DataSeries'] + [col for col in data.columns if col in v

# Select only the 6th, 7th, and 8th rows (Python uses 0-based indexing)
new_public_data = new_public_data.iloc[5:8]

# Check for out-of-bound values (values <= 0) in numeric columns
print("\nChecking for out-of-bound values (values <= 0):")
numeric_data = new_public_data.select_dtypes(include=['number'])

# Find rows with values <= 0
out_of_bounds = (numeric_data <= 0).any(axis=1)
invalid_rows = new_public_data[out_of_bounds]

# Display results
print("Filtered data (years 2005-2023, rows 6th to 8th):")
print(new_public_data)

if not invalid_rows.empty:
    print("\nRows with invalid values:")
    print(invalid_rows)
else:
    print("\nNo invalid values found.")
```

```
Checking for out-of-bound values (values <= 0):
Filtered data (years 2005-2023, rows 6th to 8th):
                                DataSeries    2023    2022    2021    2020  \
5          Average Daily Ridership - MRT  3243.0  2745.0  2100.0  2023.0
6          Average Daily Ridership - LRT   202.0   184.0   151.0   139.0
7   Average Daily Ridership - Public Bus  3747.0  3461.0  3008.0  2878.0

     2019    2018  2017  2016  2015  2014  2013  2012  2011  2010  2009  2008  \
5  3384.0  3302.0  3122  3095  2871  2762  2623  2525  2295  2069  1782  1698
6   208.0   199.0   190   180   153   137   132   124   111   100    90    88
7  4099.0  4037.0  3952  3939  3891  3751  3601  3481  3385  3199  3047  3087

   2007  2006  2005
5  1527  1408  1321
6    79    74    69
7  2932  2833  2779

No invalid values found.
```

Checking the motor vehicle dataset so that all numeric columns have values greater than 0. We need to check for this as we need to ensure all data is positive and within range.

In [67]:
```
# Load the dataset
file_path = 'MVP01-1_MVP_by_type.csv'
data = pd.read_csv(file_path)

# Check if all numeric columns have values greater than 0
numeric_columns = data.select_dtypes(include='number')  # Select numeric columns
non_positive_values = numeric_columns[(numeric_columns <= 0).any(axis=1)]  # Row
```

```
# Display results
if non_positive_values.empty:
    print("All numeric values are greater than 0.")
else:
    print("The following rows have numeric values less than or equal to 0:")
    print(non_positive_values)
```

```
All numeric values are greater than 0.
```

Checking the pollution dataframe so that all numeric values are greater than 0, we will also be checking if the pollution data is within its bounds. We need to check for this as we need to ensure all data is positive and within range.

In [70]:
```
#categories are all unique
data = pollution_df

# Check if all numeric columns have values greater than 0
numeric_columns = data.select_dtypes(include='number')  # Select numeric columns
non_positive_values = numeric_columns[(numeric_columns <= 0).any(axis=1)]  # Row

# Display results
if non_positive_values.empty:
    print("All numeric values are greater than 0.")
else:
    print("The following rows have numeric values less than or equal to 0:")
    print(non_positive_values)
```

```
All numeric values are greater than 0.
```

In [72]:
```
# Define realistic bounds for each pollutant
POLLUTANT_BOUNDS = {
    'co_max_8hour_mean': (0, 50),        # Carbon monoxide (ppm)
    'nitrogen_dioxide_mean': (0, 200),    # NO2 (ppb)
    'ozone_maximum_8hour_mean': (0, 500), # O3 (ppb)
    'pm25mean': (0, 500),                # PM2.5 (μg/m³)
    'pm10_24hour_mean_99th_per': (0, 600) # PM10 (μg/m³)
}

def validate_pollution_data(df):
    results = []

    for column, (min_val, max_val) in POLLUTANT_BOUNDS.items():
        # Skip year column and handle NaN values
        if column in df.columns and column != 'year':
            mask = (df[column] < min_val) | (df[column] > max_val)
            invalid_rows = df[mask].dropna()

            if not invalid_rows.empty:
                results.append(f"{column}: {len(invalid_rows)} values outside ra
                print(f"\nInvalid {column} values:")
                print(invalid_rows[['year', column]])

    return results if results else ["All values within expected ranges"]

# Run validation
results = validate_pollution_data(pollution_df)
print("\nValidation Results:")
```

```
for result in results:
    print(result)
```

```
Validation Results:
All values within expected ranges
```

Data Exploration

Upon examining the public transport data, an anomaly stood out between 2019 and 2021. Looking at an example of average daily ridership for public bus in 2019 it was 4099 and suddenly in 2020 it was almost halved to 2878. This made me research what happened in this time period. Also during these years, public transport usage was significantly reduced, and pollution levels were notably lower as well. Upon further analysis, I identified this period as the time when Singapore implemented lockdowns and circuit breaker measures due to the Covid-19 pandemic [1]. These external factors heavily influenced the data. To maintain the integrity of my analysis, I will exclude this "virus period" from my focus, ensuring that the project revolves solely around motor vehicle and public transport usage and pollution without being skewed by pandemic-related disruptions.

Data Format

The dataset is stored in a Pandas DataFrame, which is a highly versatile structure for conducting in-depth analyses. Its flexibility allows me to efficiently manipulate, clean, and analyze the data, making it well-suited for both exploratory data analysis and more advanced tasks, such as machine learning or visualization.ation.

Clear rhetoric for modifications to data

Data is modified

To ensure the pollution data is suitable for analysis and visualization, several systematic modifications were made:

Combining Multiple Datasets: All five pollution datasets will be merged into a single DataFrame to facilitate easier graphing and cross-analysis. Combining the datasets improves cohesion and enables a unified approach to data exploration, avoiding the inefficiencies of working with separate files.

Selecting Relevant Years (2005–2023): Data from 2005 to 2023 was selected because earlier years lacked sufficient records for public transportation. This ensures consistency and avoids introducing bias from missing data. Data outside this range was removed to maintain the accuracy and comparability of the dataset.

Focusing on Specific Public Transport Modes: Only data for MRT, LRT, and buses was retained, as these were the focus of the analysis. Irrelevant or redundant data from other public transport modes was excluded to streamline the dataset and improve the interpretability of results. I will also be modifying the dataset as there are no specific columns for rows and instead there are columns and rows called 'DataSeries' that combine the datas. To make the analysis simpler I will be including a year column. I will also be combining the separate columns of datasets, Average Daily Ridership - MRT, Average Daily Ridership - LRT and Average Daily Ridership - Public Bus into a single column called Total Public Transport for easier analysis.

Excluding 2020–2021 (COVID-19 Years): The years 2020 and 2021 will be removed across all datasets, as the COVID-19 pandemic significantly disrupted transportation patterns due to lockdowns and circuit breakers. Including these years would introduce anomalies that could skew the analysis, making it unrepresentative of typical trends.

Purpose and Value of Modifications: These changes enhance the dataset's descriptive power and analytical utility. By focusing on consistent, relevant, and high-quality data, the modifications allow for more accurate trend identification and meaningful insights. Moreover, this systematic approach ensures that the data aligns with the study's objectives while accounting for known disruptions and by combining datasets and focusing on key metrics, the analysis captures trends clearly and concisely.

Filter the motor vehicle data to only show 2005-2023 data and also exclude 2020-2021.

In [78]:
```
motor_df = motor
# Filter the dataframe for years 2005-2023, excluding 2020 and 2021
filtered_motor_df = motor_df[(motor_df['year'] >= 2005) & (motor_df['year'] <= 2
filtered_motor_df = filtered_motor_df.groupby('year').sum().reset_index()
filtered_motor_df
```

Out[78]:

	year	category	type	number
0	2005	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsRental car...	754992
1	2006	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsRental car...	799373
2	2007	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsRental car...	851336
3	2008	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsRental car...	894682
4	2009	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsRental car...	925518
5	2010	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsRental car...	945829
6	2011	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsRental car...	956704
7	2012	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsRental car...	969910
8	2013	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsPrivate Hi...	974170
9	2014	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsPrivate Hi...	972037
10	2015	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsPrivate Hi...	957246
11	2016	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsPrivate Hi...	956430
12	2017	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsPrivate Hi...	961842
13	2018	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsPrivate Hi...	957006
14	2019	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsPrivate Hi...	973101
15	2022	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsPrivate Hi...	995746
16	2023	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsPrivate Hi...	996732

In [79]:

```
filtered_motor_df.describe()
```

Out[79]:

	year	number
count	17.000000	17.000000
mean	2013.235294	931920.823529
std	5.448961	68545.628575
min	2005.000000	754992.000000
25%	2009.000000	925518.000000
50%	2013.000000	957006.000000
75%	2017.000000	972037.000000
max	2023.000000	996732.000000

Filter the pollution data to only show 2005-2023 data and also exclude 2020-2021.

In [83]:
```
filtered_pol_df = pol_df[~pol_df['year'].isin([2020, 2021])]
filtered_pol_df
```

Out[83]:

	co_max_8hour_mean	nitrogen_dioxide_mean	ozone_maximum_8hour_mean	pm25m
5	0.291425	0.732510	0.470880	1.093
6	0.500372	0.127393	-0.379428	1.606
7	-0.439887	-1.082842	2.208467	0.580
8	-0.544361	-1.082842	1.358158	-0.188
9	-0.230941	-1.082842	-1.525496	0.580
10	0.291425	-0.477724	-0.268518	0.067
11	-0.126468	0.732510	-0.860037	0.067
12	-0.230941	0.732510	-0.897007	0.580
13	3.530096	0.732510	-0.268518	0.836
14	-0.335414	0.127393	-0.416398	0.323
15	1.231685	-1.082842	0.212091	1.862
16	0.082479	1.337628	-1.155797	-0.445
17	-0.439887	0.732510	1.653918	-0.701
18	-0.126468	1.337628	0.138151	-0.445
19	-0.439887	-0.477724	-0.786097	-0.188
22	-0.439887	0.732510	-0.823067	-1.471
23	-0.648834	-0.477724	0.286030	-1.471

In [84]:
```
filtered_pol_df.describe()
```

Out[84]:

	co_max_8hour_mean	nitrogen_dioxide_mean	ozone_maximum_8hour_mean	pm2.
count	17.000000	17.000000	17.000000	17.
mean	0.113206	0.091798	-0.061922	0.
std	0.997819	0.868259	1.022375	0.
min	-0.648834	-1.082842	-1.525496	-1.
25%	-0.439887	-0.477724	-0.823067	-0.
50%	-0.230941	0.127393	-0.268518	0.
75%	0.291425	0.732510	0.286030	0.
max	3.530096	1.337628	2.208467	1.

Filter the public transport data to only show to exclude 2020-2021.

In [88]:
```python
# Drop columns corresponding to the years 2020 and 2021
filtered_public_df = new_public_data.drop(columns=['2020', '2021'])

# Display the filtered DataFrame
filtered_public_df
```

Out[88]:

	DataSeries	2023	2022	2019	2018	2017	2016	2015	2014	2013	2012	20
5	Average Daily Ridership - MRT	3243.0	2745.0	3384.0	3302.0	3122	3095	2871	2762	2623	2525	2
6	Average Daily Ridership - LRT	202.0	184.0	208.0	199.0	190	180	153	137	132	124	
7	Average Daily Ridership - Public Bus	3747.0	3461.0	4099.0	4037.0	3952	3939	3891	3751	3601	3481	3

In [89]:
```python
filtered_public_df.describe()
```

Out[89]:

	2023	2022	2019	2018
count	3.000000	3.000000	3.000000	3.000000
mean	2397.333333	2130.000000	2563.666667	2512.666667
std	1917.842625	1722.890304	2071.154348	2037.117162
min	202.000000	184.000000	208.000000	199.000000
25%	1722.500000	1464.500000	1796.000000	1750.500000
50%	3243.000000	2745.000000	3384.000000	3302.000000
75%	3495.000000	3103.000000	3741.500000	3669.500000
max	3747.000000	3461.000000	4099.000000	4037.000000

Due to the count for the public transport data only being 3, as mentioned earlier we will be modifying this data to change the format of the dataset so that the count is 17, hence I will be coding that out below.

Modifying the public transport dataset to include years

Since the dataset does not have a separate columns for years, I will be coding the data from the Public Transport dataset to include years for easier plotting and analysis by creating a new dataframe.

In [94]:

```
data = {
    "year": [2023, 2022, 2019, 2018, 2017, 2016, 2015, 2014, 2013, 2012, 2011, 2
    "Average Daily Ridership - MRT": [3243.0, 2745.0, 3384.0, 3302.0, 3122, 3095
    "Average Daily Ridership - LRT": [202.0, 184.0, 208.0, 199.0, 190, 180, 153,
    "Average Daily Ridership - Public Bus": [3747.0, 3461.0, 4099.0, 4037.0, 395
}

# Create a DataFrame
new_public_df = pd.DataFrame(data)

# Reverse the order of the DataFrame
new_public_df = new_public_df.iloc[::-1]

# Reset the index
new_public_df = new_public_df.reset_index(drop=True)

new_public_df['Total Public Transport'] = new_public_df['Average Daily Ridership
new_public_df
```

Out[94]:

	year	Average Daily Ridership - MRT	Average Daily Ridership - LRT	Average Daily Ridership - Public Bus	Total Public Transport
0	2005	1321.0	69.0	2779.0	4169.0
1	2006	1408.0	74.0	2833.0	4315.0
2	2007	1527.0	79.0	2932.0	4538.0
3	2008	1698.0	88.0	3087.0	4873.0
4	2009	1782.0	90.0	3047.0	4919.0
5	2010	2069.0	100.0	3199.0	5368.0
6	2011	2295.0	111.0	3385.0	5791.0
7	2012	2525.0	124.0	3481.0	6130.0
8	2013	2623.0	132.0	3601.0	6356.0
9	2014	2762.0	137.0	3751.0	6650.0
10	2015	2871.0	153.0	3891.0	6915.0
11	2016	3095.0	180.0	3939.0	7214.0
12	2017	3122.0	190.0	3952.0	7264.0
13	2018	3302.0	199.0	4037.0	7538.0
14	2019	3384.0	208.0	4099.0	7691.0
15	2022	2745.0	184.0	3461.0	6390.0
16	2023	3243.0	202.0	3747.0	7192.0

In [95]:
```
new_public_df.describe()
```

Out[95]:

	year	Average Daily Ridership - MRT	Average Daily Ridership - LRT	Average Daily Ridership - Public Bus	Total Public Transport
count	17.000000	17.000000	17.000000	17.000000	17.000000
mean	2013.235294	2457.176471	136.470588	3483.588235	6077.235294
std	5.448961	701.972866	49.392709	439.854246	1182.749896
min	2005.000000	1321.000000	69.000000	2779.000000	4169.000000
25%	2009.000000	1782.000000	90.000000	3087.000000	4919.000000
50%	2013.000000	2623.000000	132.000000	3481.000000	6356.000000
75%	2017.000000	3095.000000	184.000000	3891.000000	7192.000000
max	2023.000000	3384.000000	208.000000	4099.000000	7691.000000

Exploratory Data Analysis

To gain deeper insights into the pollution dataset, we will perform the following analyses first:

High Top 25% and Low Bottom 25% of Pollutants: By calculating the 25th percentile (Q1) and the 75th percentile (Q3) for each pollutant, we can identify the years with exceptionally high or low levels of specific pollutants. This will help highlight patterns or anomalies, such as spikes in certain pollutants during specific years.

Standard Deviation Analysis: Calculating the standard deviation for each pollutant will provide insights into the variability and consistency of pollutant levels over time. Higher standard deviation indicates significant fluctuations, while lower values suggest stability.

Patterns in Pollutants: Identifying years where pollutants peaked or declined can indicate underlying causes such as policy changes, industrial activities, or environmental events.

Focus on Anomalies: Observing outliers will help contextualize environmental shifts or challenges faced in certain periods.

In [99]:
```
def analyze_pollutant_patterns(df):
    # Create a copy to avoid warnings
    df = df.copy()

    pollutants = ['co_max_8hour_mean', 'nitrogen_dioxide_mean',
                  'ozone_maximum_8hour_mean', 'pm25mean', 'pm10_24hour_mean_99th

    for pollutant in pollutants:
        print(f"\nAnalysis for {pollutant}")

        # Calculate percentiles
        q1 = df[pollutant].quantile(0.25)
        q3 = df[pollutant].quantile(0.75)

        # Categorize values using loc
        high_pollution = df.loc[df[pollutant] > q3]
        low_pollution = df.loc[df[pollutant] < q1]

        print("\nHigh pollution years (top 25%):")
        print(high_pollution[['year', pollutant]].sort_values(by=pollutant, asce

        print("\nLow pollution years (bottom 25%):")
        print(low_pollution[['year', pollutant]].sort_values(by=pollutant))

        # Calculate year-over-year changes using loc
        df.loc[:, f'{pollutant}_yoy_change'] = df[pollutant].diff()
        significant_changes = df.loc[abs(df[f'{pollutant}_yoy_change']) > df[f'{

        if not significant_changes.empty:
            print("\nSignificant year-over-year changes (> 1 std dev):")
            print(significant_changes[['year', pollutant, f'{pollutant}_yoy_chan

# Run analysis
analyze_pollutant_patterns(filtered_pol_df)
```

```
Analysis for co_max_8hour_mean

High pollution years (top 25%):
    year  co_max_8hour_mean
13  2013           3.530096
15  2015           1.231685
6   2006           0.500372

Low pollution years (bottom 25%):
    year  co_max_8hour_mean
23  2023          -0.648834
8   2008          -0.544361

Significant year-over-year changes (> 1 std dev):
    year  co_max_8hour_mean  co_max_8hour_mean_yoy_change
13  2013           3.530096                      3.761037
14  2014          -0.335414                     -3.865511
15  2015           1.231685                      1.567099

Analysis for nitrogen_dioxide_mean

High pollution years (top 25%):
    year  nitrogen_dioxide_mean
16  2016               1.337628
18  2018               1.337628

Low pollution years (bottom 25%):
    year  nitrogen_dioxide_mean
7   2007              -1.082842
8   2008              -1.082842
9   2009              -1.082842
15  2015              -1.082842

Significant year-over-year changes (> 1 std dev):
    year  nitrogen_dioxide_mean  nitrogen_dioxide_mean_yoy_change
7   2007              -1.082842                         -1.210235
11  2011               0.732510                          1.210235
15  2015              -1.082842                         -1.210235
16  2016               1.337628                          2.420469
19  2019              -0.477724                         -1.815352
22  2022               0.732510                          1.210235
23  2023              -0.477724                         -1.210235

Analysis for ozone_maximum_8hour_mean

High pollution years (top 25%):
    year  ozone_maximum_8hour_mean
7   2007                  2.208467
17  2017                  1.653918
8   2008                  1.358158
5   2005                  0.470880

Low pollution years (bottom 25%):
    year  ozone_maximum_8hour_mean
9   2009                 -1.525496
16  2016                 -1.155797
12  2012                 -0.897007
11  2011                 -0.860037

Significant year-over-year changes (> 1 std dev):
```

```
    year  ozone_maximum_8hour_mean  ozone_maximum_8hour_mean_yoy_change
7   2007                  2.208467                             2.587895
9   2009                 -1.525496                            -2.883654
17  2017                  1.653918                             2.809714
18  2018                  0.138151                            -1.515767

Analysis for pm25mean

High pollution years (top 25%):
    year  pm25mean
15  2015  1.862674
6   2006  1.606219
5   2005  1.093309
13  2013  0.836854

Low pollution years (bottom 25%):
    year  pm25mean
22  2022 -1.471243
23  2023 -1.471243
17  2017 -0.701877

Significant year-over-year changes (> 1 std dev):
    year  pm25mean  pm25mean_yoy_change
7   2007  0.580398            -1.025820
15  2015  1.862674             1.538731
16  2016 -0.445422            -2.308096
22  2022 -1.471243            -1.282276

Analysis for pm10_24hour_mean_99th_per

High pollution years (top 25%):
    year  pm10_24hour_mean_99th_per
13  2013                   2.927912
15  2015                   2.308611
6   2006                   0.899167
19  2019                   0.258510

Low pollution years (bottom 25%):
    year  pm10_24hour_mean_99th_per
8   2008                  -0.617053
22  2022                  -0.574343
7   2007                  -0.531632
11  2011                  -0.488922

Significant year-over-year changes (> 1 std dev):
    year  pm10_24hour_mean_99th_per  pm10_24hour_mean_99th_per_yoy_change
13  2013                   2.927912                              3.374123
14  2014                  -0.061818                             -2.989729
15  2015                   2.308611                              2.370428
16  2016                  -0.360791                             -2.669401
```

From the analysis of the data, it is evident that the year 2013 is particularly significant, as it stands out as a high pollution year for two major pollutants: carbon monoxide (CO) and particulate matter (PM). This observation is intriguing because the increase in pollution levels for these substances was not part of a gradual trend but instead represented a sharp and sudden spike.

For instance, the Year-over-Year (YoY) change in carbon monoxide levels during 2013 showed a significant increase of 3.76, followed by an equally sharp decrease of -3.87 in 2014. This pattern suggests that the rise in pollution was likely caused by a temporary external factor rather than sustained environmental or economic activities. Upon further investigation, it becomes apparent that the spike in 2013 coincided with a severe haze event in Singapo[2].e. This haze was caused by large-scale forest fires in neighboring countries, such as Indonesia, which led to transboundary air pollution affecting the entire region. As a result, particulate matter levels, particularly PM2.5 and PM10, also reached unprecedented heigh.s.

The haze in 2013 was one of the worst in Singapore's recent history, with the Pollutant Standards Index (PSI) reaching hazardous levels. The dense smog not only affected air quality but also posed serious health risks and disrupted daily life. This context provides a clear explanation for the anomalous data observed in 2013 for carbon monoxide and particulate matter pollutants.

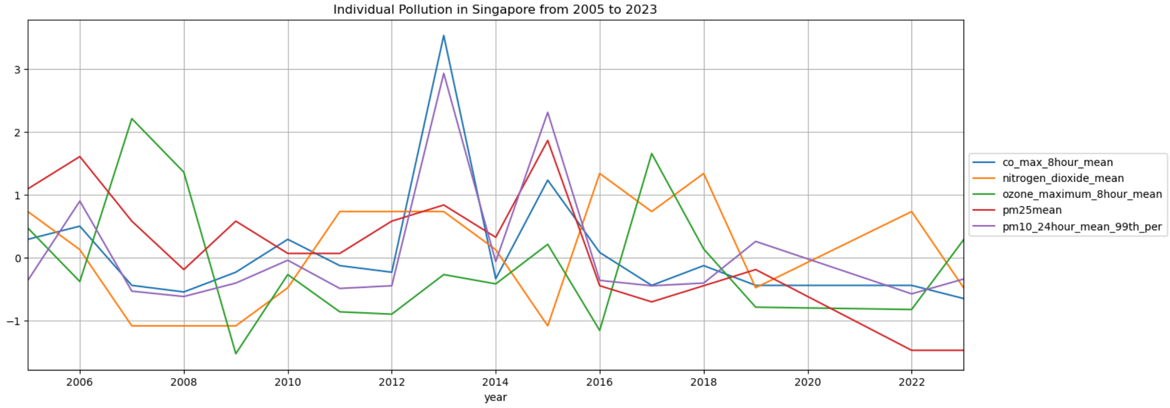
Moving forward, this anomalous data will be explored in greater detail through plots and visualizatuture.

Plotting Pollution graph in Sinagpore

Next, we are going to plot the pollution graph for all polutants through our merged dataframe called filtered_pol_df which only has data from 2005 to 2023 without 2020-2021 data due to Covid. From this, we can see the overall trend of pollution in Singapore.

In [103…

```
Var_to_plot = ['co_max_8hour_mean','nitrogen_dioxide_mean', 'ozone_maximum_8hour
               'pm10_24hour_mean_99th_per']

#Draw plot
Indi_pol_plot = filtered_pol_df.plot(x='year', y = Var_to_plot, kind = 'line', g
                        title = 'Individual Pollution in Singapore from 2005 to

#Graph formatting
Indi_pol_plot.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.xlim(2005, 2023)
plt.show()
```



Analysing the graph we can see that some pollutants were high in some years and lower in the other years. We would have to draw more graphs to figure out why there are such spikes and anomalies.

Mean normalization for motor vehicle

In [106…

```
#Select the data we need
mean_motor_df = filtered_motor_df[filtered_motor_df['year'].isin(year)]

#Perform groupby
mean_motor_df = mean_motor_df.groupby('year').sum().reset_index()

#Mean normalization
mean_motor_df['number']=(mean_motor_df['number']-mean_motor_df['number'].mean())
mean_motor_df.rename(columns = {'number':'Number of Vehicles'}, inplace = True)
mean_motor_df
```

Out[106…

	year	category	type	Number of Vehicles
0	2005	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsRental car...	-2.581183
1	2006	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsRental car...	-1.933717
2	2007	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsRental car...	-1.175638
3	2008	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsRental car...	-0.543271
4	2009	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsRental car...	-0.093410
5	2010	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsRental car...	0.202904
6	2011	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsRental car...	0.361557
7	2012	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsRental car...	0.554217
8	2013	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsPrivate Hi...	0.616366
9	2014	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsPrivate Hi...	0.585248
10	2015	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsPrivate Hi...	0.369465
11	2016	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsPrivate Hi...	0.357560
12	2017	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsPrivate Hi...	0.436515
13	2018	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsPrivate Hi...	0.365963
14	2019	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsPrivate Hi...	0.600770
15	2022	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsPrivate Hi...	0.931134
16	2023	Cars and Station-wagonsCars and Station-wagons...	Private carsCompany carsTuition carsPrivate Hi...	0.945519

Mean normalization for public transport

In [110…

```
# Select the data we need
mean_public_df = new_public_df[new_public_df['year'].isin(year)]

# Perform groupby
mean_public_df = mean_public_df.groupby('year').sum().reset_index()
```

```
# Mean normalization
mean_public_df['Total Public Transport'] = (mean_public_df['Total Public Transpo

mean_public_df.rename(columns={'Total Public Transport': 'Total Public Transport
mean_public_df
```

Out[110…

	year	Average Daily Ridership - MRT	Average Daily Ridership - LRT	Average Daily Ridership - Public Bus	Total Public Transport
0	2005	1321.0	69.0	2779.0	-1.613389
1	2006	1408.0	74.0	2833.0	-1.489948
2	2007	1527.0	79.0	2932.0	-1.301404
3	2008	1698.0	88.0	3087.0	-1.018166
4	2009	1782.0	90.0	3047.0	-0.979273
5	2010	2069.0	100.0	3199.0	-0.599649
6	2011	2295.0	111.0	3385.0	-0.242008
7	2012	2525.0	124.0	3481.0	0.044612
8	2013	2623.0	132.0	3601.0	0.235692
9	2014	2762.0	137.0	3751.0	0.484265
10	2015	2871.0	153.0	3891.0	0.708319
11	2016	3095.0	180.0	3939.0	0.961120
12	2017	3122.0	190.0	3952.0	1.003394
13	2018	3302.0	199.0	4037.0	1.235058
14	2019	3384.0	208.0	4099.0	1.364418
15	2022	2745.0	184.0	3461.0	0.264439
16	2023	3243.0	202.0	3747.0	0.942519

Finding relation between motor vehicles and pollution

To analyze the relationship between motor vehicle usage and pollution, we will determine which pollutant is most relevant to motor vehicle trends. This will be achieved by calculating the Spearman correlation coefficient between motor vehicle usage and pollutant levels, resulting in a correlation dataframe for easy interpretation. The pollutant with the highest correlation will be considered most relevant.

In [113…

```
#Create new dataframe
veh_corr = filtered_pol_df.copy()

#Add in the housing data
veh_corr['Veh Corr'] = mean_motor_df['Number of Vehicles'].tolist()
```

```
#Product correlation dataframe
veh_corr.corr(method = 'spearman')
```

Out[113…

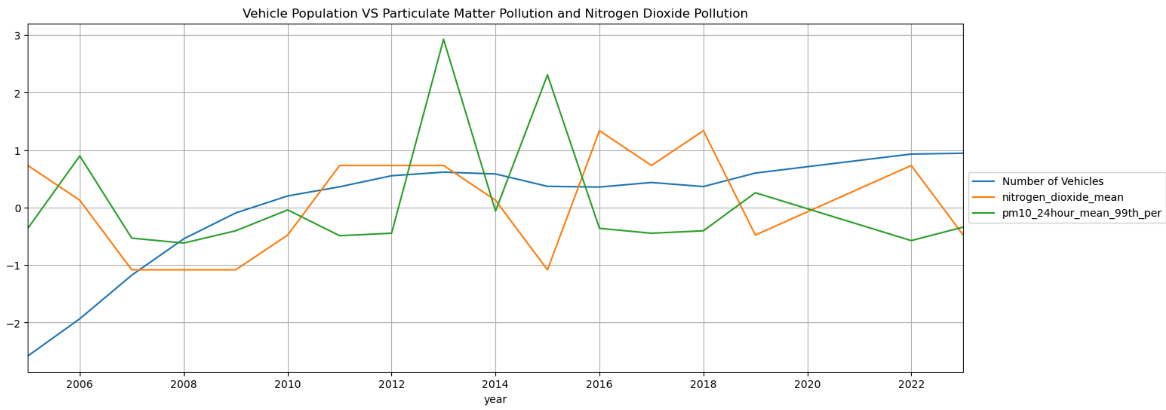
	co_max_8hour_mean	nitrogen_dioxide_mean	ozone_maxir
co_max_8hour_mean	1.000000	0.265384	
nitrogen_dioxide_mean	0.265384	1.000000	
ozone_maximum_8hour_mean	-0.199635	-0.272922	
pm25mean	0.677222	-0.291908	
pm10_24hour_mean_99th_per	0.616967	-0.067205	
year	-0.350845	0.289841	
Veh Corr	-0.287841	0.199978	

Looking at the dataframe above, we will focusing on analyzing the relationship between vehicle usage and Nitrogen Dioxide Mean and PM10 24-Hour Mean (99th Percentile). These pollutants show the strongest correlations with vehicle trends and have the highest absolute Spearman correlation values with vehicle usage.

Motor Vehicle vs required pollution graph

In [117…

```
veh_graph = mean_motor_df.plot(x='year', y='Number of Vehicles', kind = 'line',
                    title = 'Vehicle Population from 2005 to 2023')
filtered_pol_df.plot(x='year', y = ['nitrogen_dioxide_mean', 'pm10_24hour_mean_9
                        title = 'Vehicle Population VS Particulate Matter Pollut

plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.xlim(2005, 2023)
plt.show()
```



Detailed Analysis of the Graph

Number of Vehicles (Blue Line):

The blue line shows a steady increase from 2005 to 2023. This reflects the continuous growth in vehicle population in Singapore, likely due to urbanization, economic growth, and increased car ownership. There are no sharp dips or spikes, indicating consistent growth.

Nitrogen Dioxide (NO_2) Mean (Orange Line):

2005-2011: A gradual decline in nitrogen dioxide levels, possibly due to stricter vehicle emission standards, cleaner fuel technologies, and improved urban air quality measures.

2011-2016: A slight increase in NO_2 levels, peaking around 2013, followed by another dip.

2016-2023: Fairly stable levels with minor fluctuations, likely due to improved regulations such as Euro VI emission standards implemented for new vehicles.

PM10 24-Hour Mean 99th Percentile (Green Line):

2005-2010: Stable with minor variations, indicating moderate particulate pollution.

2013: A sharp spike is evident, corresponding to an anomaly (discussed below).

2014-2023: PM10 levels fluctuate but show an overall stabilization in recent years, likely due to measures like haze management, regional cooperation, and stricter pollution controls.

Anomalies and Context:

2013: Sharp Spike in PM10 Levels We would need to analyse this further, hence I will sketch a scatter plot to view any other anomalous data.

2005-2011: Decline in NO_2 Levels Introduction of cleaner fuel technologies and tighter vehicle emission controls in Singapore. Could also have been due to more public transports which we will analyse later.

2015-2016: Elevated PM10 and NO_2 Another haze event occurred in 2015, though less severe than in 2013.

2017-2023: Stabilization in Pollution Levels Regional cooperation efforts, such as the ASEAN Agreement on Transboundary Haze Pollution, may have reduced the frequency and intensity of haze events. Stricter vehicle emission standards, including the adoption of Euro VI standards for diesel vehicles, further contributed to controlling pollution.

Relationship Between Vehicles and Pollution

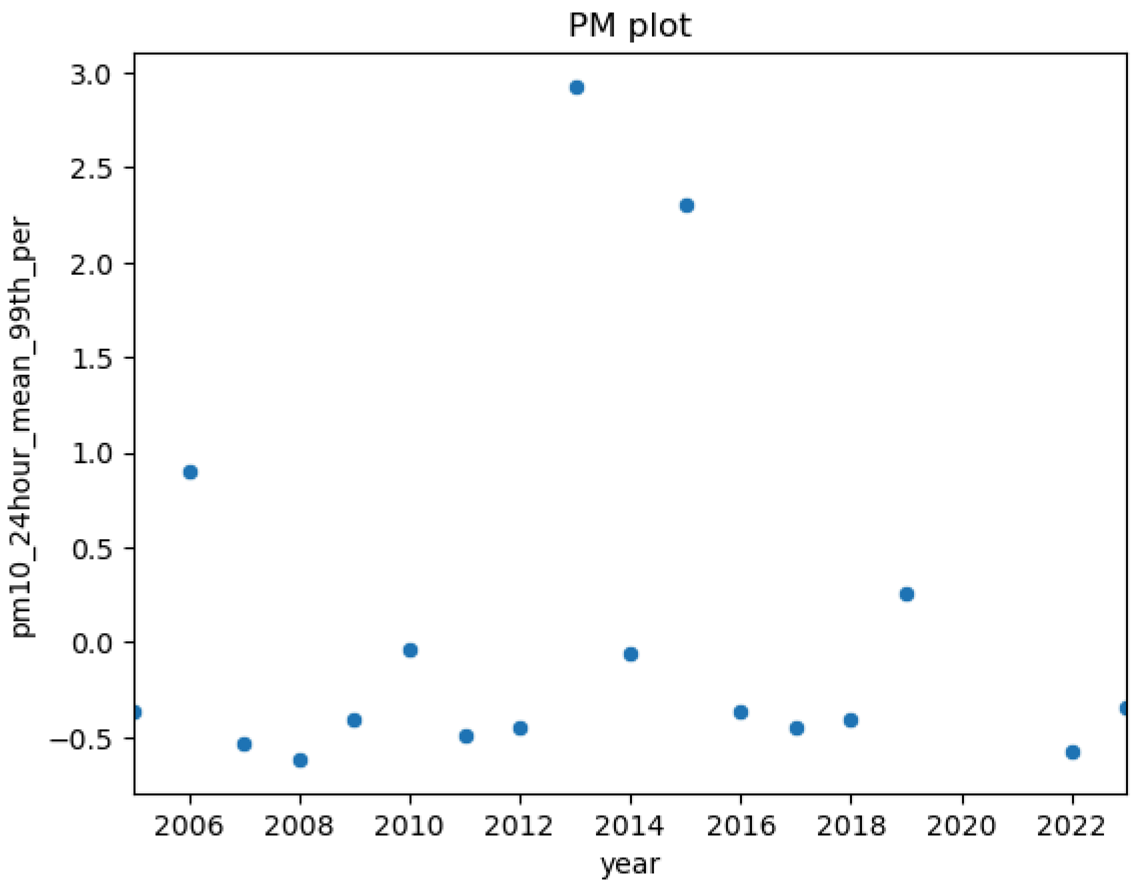
While the number of vehicles steadily increased, pollution levels (NO_2 and PM10) do not show a proportionate rise. This could be due to motor vehicles having little part to do with the overall pollution. However looking at a bigger level, it is evident that if the number of vehicles increase the level of pollution will increase scientifically. Another reason could be that technological advancements (e.g., electric vehicles, cleaner fuels) and regulations (e.g., emission standards) have mitigated the impact of vehicle growth on air quality. Major anomalies in pollution (e.g., 2013 PM10 spike) are driven by external factors like haze, rather than local vehicle emissions. Hence, we are going to continue with our analysis to see how public transport plays a part here.

Particulate matter scatter plot

As mentioned above we will be looking at the particulate matter graph in more detail via a scatter plot to pinpoint the anomalies.

In [120…

```
sns.scatterplot(x='year', y='pm10_24hour_mean_99th_per', data=filtered_pol_df)
plt.title('PM plot')
plt.xlim(2005, 2023)
plt.show()
```



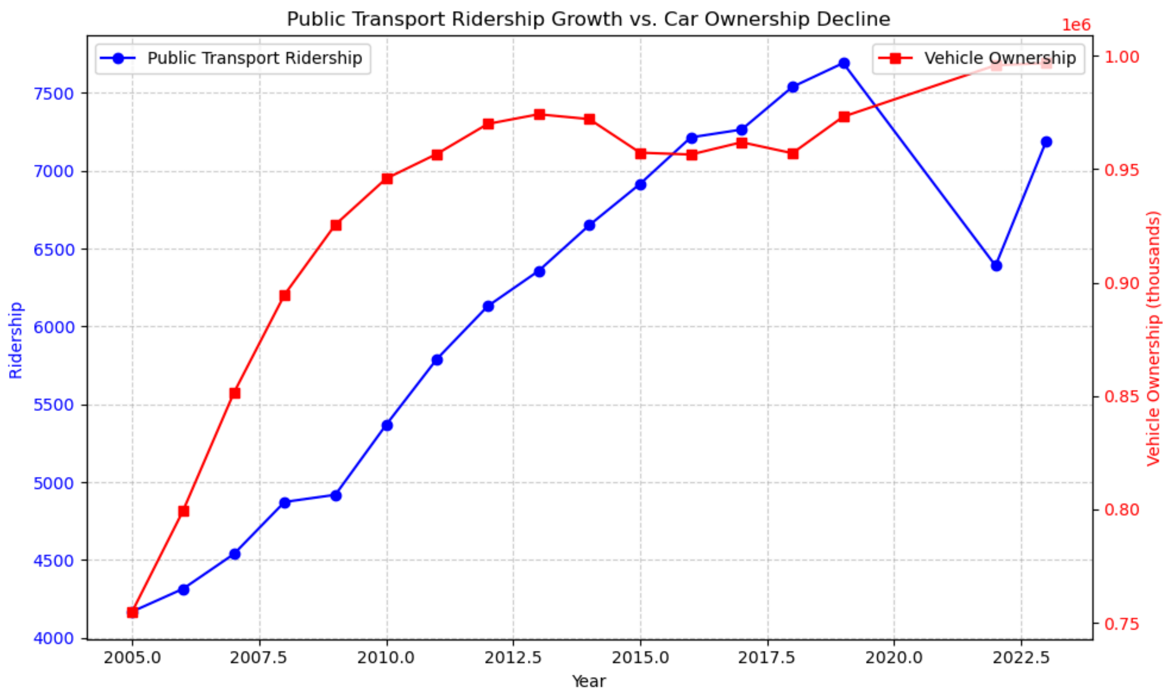
From this it is evident that in 2013, there was a anomalous data as there is a single data point right at the top. This anomaly is primarily due to the 2013 Southeast Asian Haze Crisis, which was caused by widespread forest fires in Indonesia, particularly in Sumatra and Kalimantan.The fires were fueled by illegal slash-and-burn practices for agricultural land clearing. These resulted in a dense haze that blanketed Singapore and neighboring countries.Air quality reached hazardous levels in Singapore, with the Pollutant Standards Index (PSI) exceeding 400 at its peak. Hence, the high PM10 levels in 2013 were not primarily due to local vehicle emissions but rather transboundary haze pollution.

Next, we will be looking at the graphs of motor vehicles vs public transport to see if there was a rise or decline in motor vehicles when public transport increased.

Motor vehicles Vs Public tranport

In [124…

```
# Merge DataFrames on 'Year'
df = pd.merge(new_public_df, filtered_motor_df, on='year')

# Create the plot
fig, ax1 = plt.subplots(figsize=(10, 6))

# Primary axis - Public Transport Ridership
ax1.plot(df['year'], df['Total Public Transport'], color='blue', marker='o', lab
ax1.set_xlabel('Year')
ax1.set_ylabel('Ridership ', color='blue')
ax1.tick_params(axis='y', labelcolor='blue')
ax1.grid(True, linestyle='--', alpha=0.6)

# Secondary axis - Vehicle Ownership
ax2 = ax1.twinx()  # Create a second y-axis
ax2.plot(df['year'], df['number'], color='red', marker='s', label='Vehicle Owner
ax2.set_ylabel('Vehicle Ownership (thousands)', color='red')
ax2.tick_params(axis='y', labelcolor='red')

# Title and Legends
plt.title('Public Transport Ridership Growth vs. Car Ownership Decline')
fig.tight_layout()  # Adjust layout to avoid overlap
ax1.legend(loc='upper left')
ax2.legend(loc='upper right')

# Show the plot
plt.show()
```



Analysis of the graph:

Public Transport Ridership: The blue line shows a consistent increase in public transport ridership over the years, with a notable upward trajectory from 2005 to around 2019. This suggests a growing reliance on public transport during this period. The temporary dip after 2020 could correspond to disruptions caused by the COVID-19 pandemic but seems to recover in subsequent years.

Vehicle Ownership: The red line indicates a different trend for vehicle ownership. From 2005, vehicle ownership shows steady growth until it plateaus around 2014–2015. After this point, it starts to show signs of a gradual decline. This suggests a possible shift away from personal vehicle ownership during the same period when public transport usage was rising.

Relationship Between the Two Trends: The graph supports the claim that an increase in public transport ridership correlates with a reduction in motor vehicle ownership. The timing of the plateau and subsequent decline in vehicle ownership aligns with the steep increase in public transport usage, indicating a potential causal relationship. This could reflect the success of policies encouraging public transport adoption, improvements in the public transit system, or increased costs or restrictions on car ownership in Singapore.

COVID-19 Impact: Both trends seem to exhibit an anomaly around 2020–2021, likely due to the pandemic, which disrupted commuting patterns and may have temporarily influenced vehicle ownership and public transport ridership.

Hence, the graph supports my claim that there was a decline in motor vehicles when public transport increased.

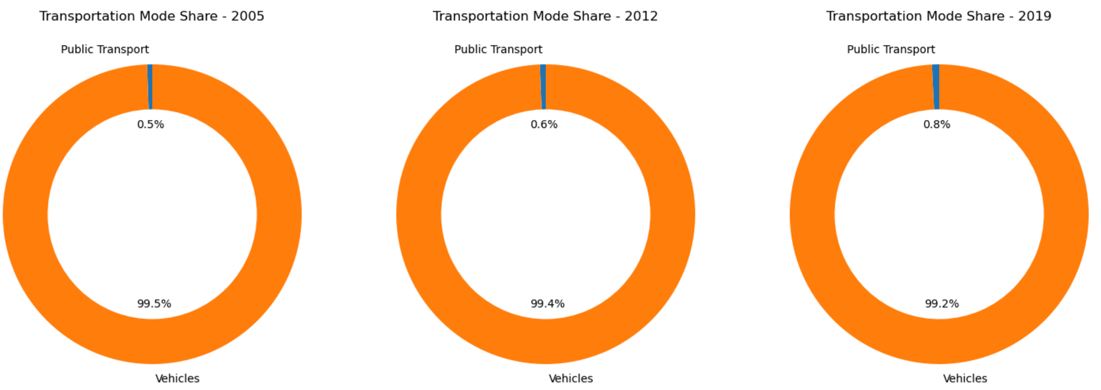
Public transport vs Motor vehicle shares

We shall see if randomised years support our claim of the usage of public transport has been rising. I have selected the years 2005,2012 and 2019 to do the random checks.

In [127…

```
# Example DataFrames
df_public_transport = pd.DataFrame({
    'year': [2005, 2012, 2019],
    'Total Public Transport': [4169.0, 6130.0, 7691.0]
})

df_vehicle = pd.DataFrame({
    'year': [2005, 2012, 2019],
    'number': [754992, 969910, 973101]
})

# Merge the DataFrames on 'year'
df = pd.merge(df_public_transport, df_vehicle, on='year')

# Create a figure for the donut charts
fig, axes = plt.subplots(1, len(df['year'].unique()), figsize=(15, 5), subplot_k

# Iterate over each year to create a donut chart
for i, year in enumerate(df['year'].unique()):
    ax = axes[i] if len(df['year'].unique()) > 1 else axes
    data = df[df['year'] == year]

    # Prepare the data for the donut chart
```

```
    total_public_transport = data['Total Public Transport'].iloc[0]
    number_of_vehicles = data['number'].iloc[0]

    # Calculate percentages
    sizes = [total_public_transport, number_of_vehicles]
    labels = ['Public Transport', 'Vehicles']
    colors = ['#1f77b4', '#ff7f0e']

    # Plot the donut chart
    wedges, texts, autotexts = ax.pie(
        sizes,
        labels=labels,
        autopct=lambda p: '{:.1f}%'.format(p),  # Ensure percentages are shown
        startangle=90,
        colors=colors,
        wedgeprops=dict(width=0.3)  # Donut effect
    )

    # Set title for each subplot
    ax.set_title(f"Transportation Mode Share - {year}")

# Adjust layout
plt.tight_layout()
plt.show()
```



As demonstrated above, the data confirms that public transport usage has increased over the years. While public transport occupies a smaller segment in the donut chart compared to motor vehicles, it's important to note that the dataset heavily emphasizes motor vehicle data. Our focus was solely on identifying whether public transport usage has risen, and the consistent percentage increase over the years supports the claim that public transport adoption has grown steadily.

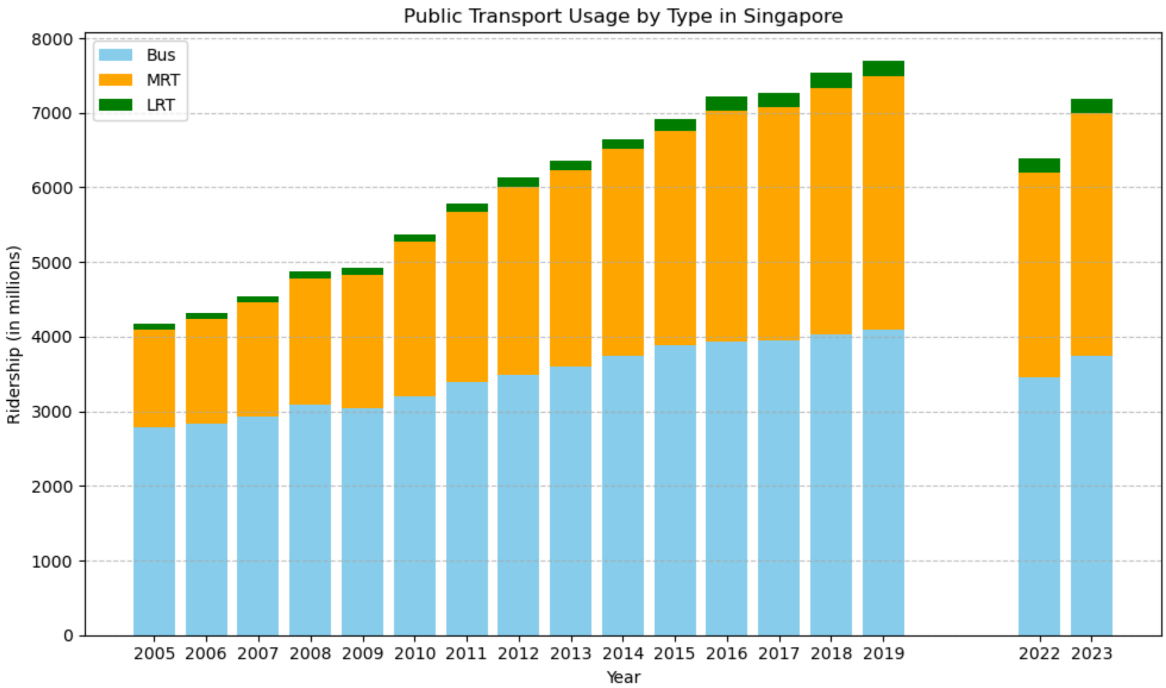
Public Transport Usage by Type in Singapore

After confirming that public transport usage has been rising lets now look at how much of the three type of public transport do Singaporeans use more to get better policies on which category Singapore should work on to further to exhilarate the usage of public transport.

In [132…

```
# Plotting
plt.figure(figsize=(10, 6))
plt.bar(new_public_df['year'], new_public_df['Average Daily Ridership - Public B
plt.bar(new_public_df['year'], new_public_df['Average Daily Ridership - MRT'], b
```

```
plt.bar(new_public_df['year'], new_public_df['Average Daily Ridership - LRT'], b

# Adding titles and labels
plt.title('Public Transport Usage by Type in Singapore')
plt.xlabel('Year')
plt.ylabel('Ridership (in millions)')
plt.legend()
plt.xticks(new_public_df['year'])  # Show all years on x-axis
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Show the plot
plt.tight_layout()
plt.show()
```



It is evident that Singaporeans use buses more than the other two categories. If Singapore were to make bus fares cheaper or encourage the use of buses through other means, we could definitely ensure that pollution levels stay low in Singapor. Since we omitted the data in 2020 and 2021, the usage of public transports dropped in 2022 as Singaporeans were still transitioning out of Covid-19. Many people and companies also resorted to working from home hence the sudden drop in 2022 [3]. However as expected, there was a rise again from 2022-2023 as public transports usage is still steadily rising in Singapore..

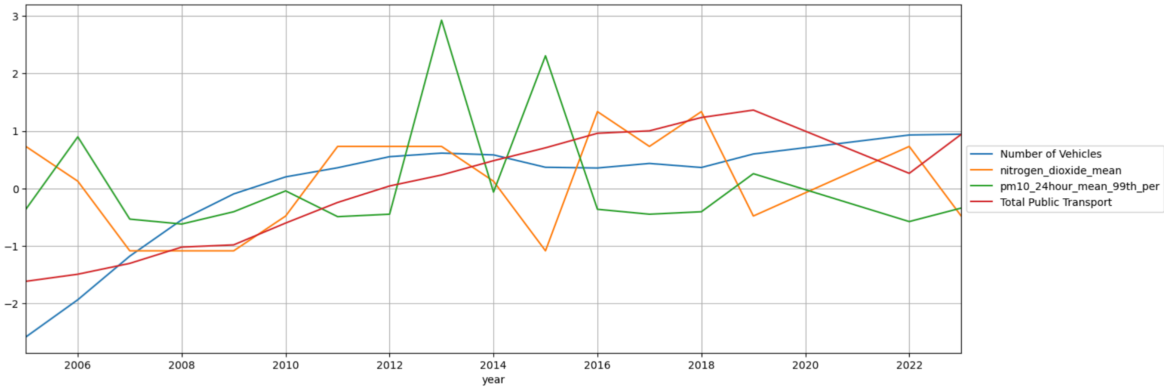
Overall analysis of all time series graphs combined

I have done an overall analysis of this at the bottom of this report under Analysis of the graphs for better readability.

In [135…

```
veh_graph = mean_motor_df.plot(x='year', y='Number of Vehicles', kind = 'line', 
filtered_pol_df.plot(x='year', y = ['nitrogen_dioxide_mean', 'pm10_24hour_mean_9
mean_public_df.plot(x='year', y = ['Total Public Transport'], kind = 'line', gri
```

```
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.xlim(2005, 2023)
plt.show()
```



Analysis of the graphs

We will not be looking at the data after 2019 as after that time frame covid-19 occured and that data is rendundant at the moment of analysis. Public transport usage has been increasing throughout the years and even exceeding the total number of vehicles from 2014 onwards. This shows us that the number of motor vehicles used in Singapore is decreasing, therefore impacting climate change in a better way and supporting our claims. We will now be discussing if our problem statement which is "the introduction of public transport has decreased the amount of pollution in Singapore caused by motor vehicles" is correct in detail now.

The number of motor vehicles (blue line) shows a consistent increase from 2006 to around 2018, followed by a plateau or slower growth thereafter. Public transport usage (red line) exhibits a steady upward trend, reflecting successful public transport adoption policies in Singapore. The nitrogen dioxide mean (orange line) shows a decreasing trend overall. This suggests improved air quality over time, particularly after 2010. PM10 (green line) exhibits more variability compared to NO_2. A significant spike is observed around 2013, which corresponds to the Southeast Asian haze crisis caused by forest fires in neighboring countries. This is an anomaly unrelated to motor vehicles or public transport policies. From 2020 to 2021 there was covid but ignoring that we can see that in 2023 the number of motor vehicles and public transport usage were similar indicating more people are willing to take public tranport which is a good move to tackle climate change and pollution levels due to motor vehicles in Singapore.

Correlation with Problem Statement: My problem statement asserts that increased usage of public transport has decreased the pollution caused by motor vehicles. The graph provides evidence supporting this claim:

1. Inverse Relationship Between Public Transport and Pollution:

From 2010 onward, as public transport usage (red line) rises, nitrogen dioxide levels (orange line) decline. This aligns with the idea that fewer people rely on private vehicles due to increased public transport adoption.

2. Plateau in Vehicle Growth:

The growth rate of motor vehicles slows down or plateaus after 2018. This could be due to government policies such as vehicle quotas (Certificate of Entitlement) and improved public transport infrastructure, encouraging a shift away from private vehicles.

3. Haze Crisis of 2013 (Anomaly):

The sharp increase in PM10 levels in 2013 is unrelated to motor vehicles or public transport. This anomaly was caused by transboundary haze pollution due to forest fires in Indonesia. This highlights the importance of distinguishing local pollution trends from external influences.

4. Sustained Decline in NO_2 Levels:

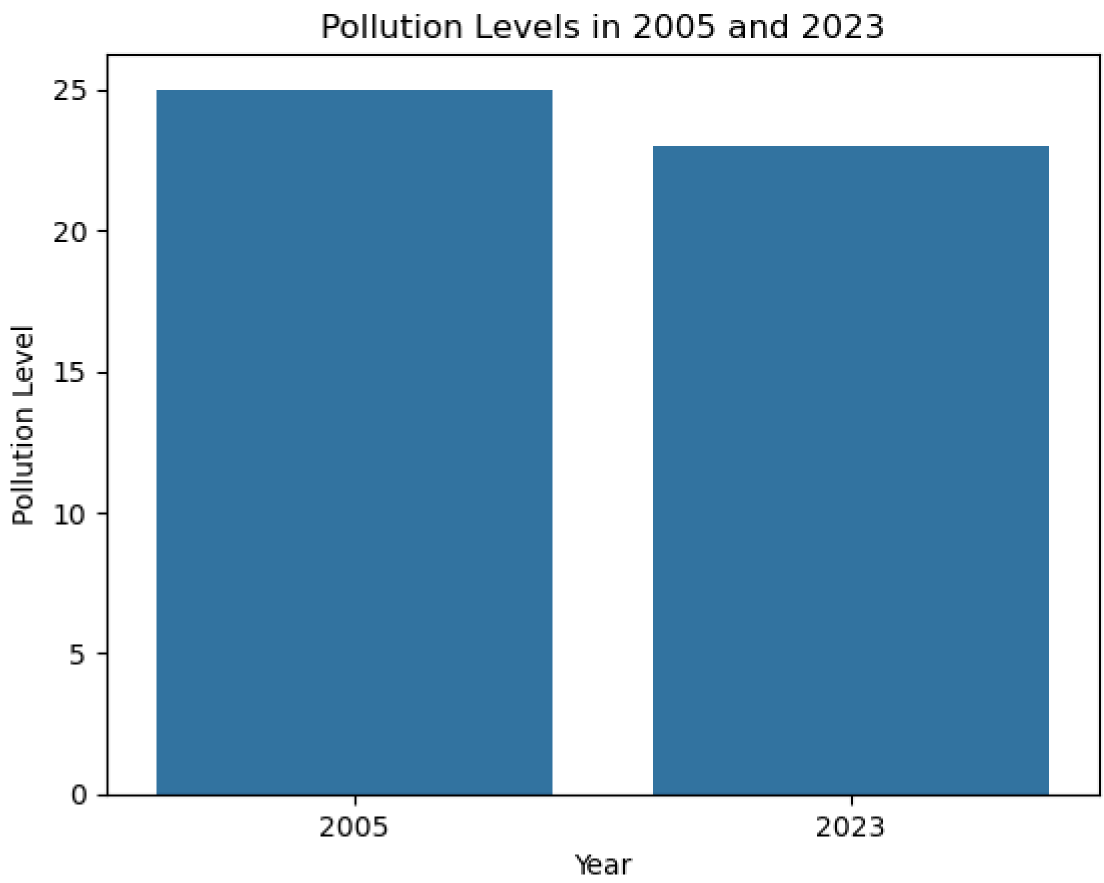
Nitrogen dioxide, a key marker of vehicle emissions, consistently declines despite increasing motor vehicle numbers. This indicates the impact of stricter emission standards (e.g., Euro VI for vehicles) and the shift toward public transport.

Pollution level in 2005 vs 2023

I have selected the extreme datapoints (2005 and 2023) to see if the pollution levels in Singapore has really dropped. I have done an overall analysis of this at the bottom of this report under Conclusion for better readability.

In [139…

```
# Filter pollution dataset for 2005 and 2023
pollution_2005 = pollution_df[pollution_df['year'] == 2005]['nitrogen_dioxide_me
pollution_2023 = pollution_df[pollution_df['year'] == 2023]['nitrogen_dioxide_me

# Filter public_transport dataset for 2005 and 2023
# You can similarly use the year-based date format or filter by specific dates f
public_transport_2005 = new_public_df[new_public_df['year'] == 2005]['Total Publ
public_transport_2023 = new_public_df[new_public_df['year'] == 2023]['Total Publ

# Create a comparison of pollution levels for 2005 and 2023
sns.barplot(x=['2005', '2023'], y=[pollution_2005, pollution_2023])

# Add title and labels
plt.title('Pollution Levels in 2005 and 2023')
plt.xlabel('Year')
plt.ylabel('Pollution Level')

# Show plot
plt.show()
```



Conclusion

The visualisations and statistical expoloration strongly supports the problem statement:

Public transport adoption (red line) has significantly increased, while pollution levels (NO_2 and to some extent PM10) have generally declined over time. The plateau in motor vehicle growth and declining NO_2 levels suggest a shift from private vehicles to public transport as a primary mode of transportation. External factors, such as the 2013 haze, add noise to the data but do not undermine the overall trend. Eventhough there is not a very direct relationship to motor vehicle usage and pollution levels, generally there is a rise in pollution when there is a rise in motor vehicle usage. By looking at the final visualisation of pollution via a bar graph in 2005 vs 2023, we can definitely say Singapore is in the right track as the main pollutant in motor vehicles is being reduced. By implementing better strategies to use the public transport more, the pollution levels would drop even further and Singapore would continue being a green country.

APPENDIX

Data Used: Motor vehicle: https://datamall.lta.gov.sg/content/datamall/en/static-data.html

Public Transport: https://data.gov.sg/datasets?q=&query=public+transport&groups=&organization=&page=1&resultId=d_ba615ec4cc5d9

Carbon Monoxide: https://data.gov.sg/datasets?q=&query=carbon+monoxide&groups=&organization=&page=1&resultId=d_fdf8b7d6401

Nitrogen Dioxide: https://data.gov.sg/datasets?q=&query=nitrogen+dioxide&groups=&organization=&page=1&resultId=d_88dcbdd26f7a

Ozone: https://data.gov.sg/datasets?q=&query=ozone&groups=&organization=&page=1&resultId=d_12e90ff1178704ebd56dc

PM10: https://data.gov.sg/datasets?q=&query=particulate+matter&groups=&organization=&page=1&resultId=d_397fe8de643

PM2.5: https://data.gov.sg/datasets?q=&query=particulate+matter&groups=&organization=&page=1&resultId=d_397fe8de643

References

[1] : https://www.channelnewsasia.com/singapore/singapore-covid-19-outbreak-evolved-coronavirus-deaths-timeline-764126

[2] : https://www.bbc.com/news/world-asia-22998592

[3] : https://www.straitstimes.com/life/trends-to-watch-in-2022-staying-home-to-work