public transport impact

June 22, 2025

[ ]:

```
Project Title: Public Transport Impact on Urban Air Pollution

# Problem Framing & Hypothesis
Objective: Assess the relationship between public transport usage and pollution␣
  ↪(PM2.5/PM10).
Goal Analyze pollution trends and compare them with transport data.
KPI: PM2.5 levels, monthly averages, correlation with ridership
Hypothesis: Cities with higher public transport ridership show lower pollution␣
  ↪levels.
```

[24]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import statsmodels.api as sm
```

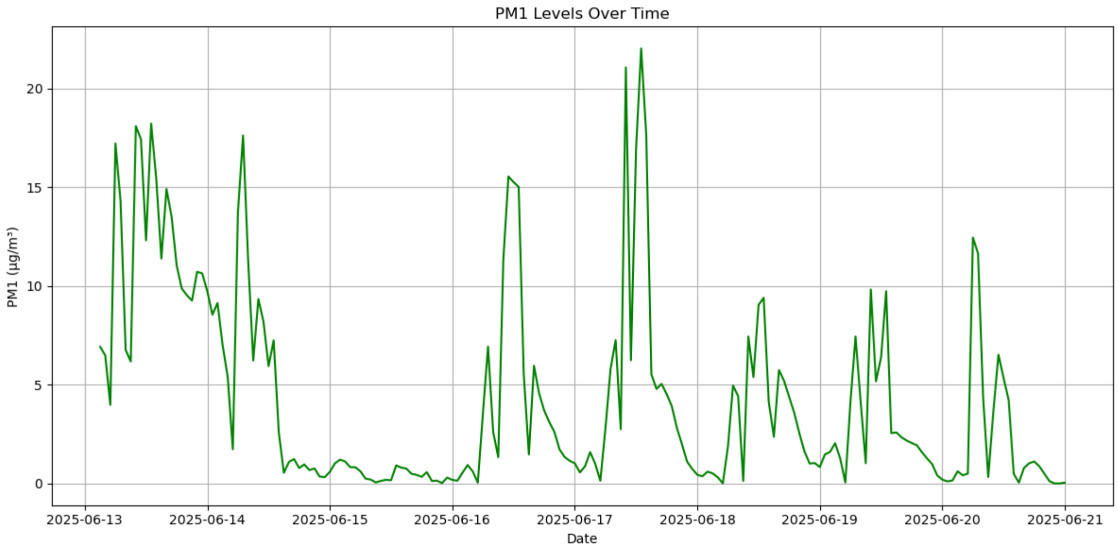
[25]:

```
df = pd.read_csv("openaq_location_4720578_measurments (1).csv")
df.columns = df.columns.str.strip().str.lower()
df['datetimeutc'] = pd.to_datetime(df['datetimeutc'])

df = df[df['parameter'] == 'pm1']
df = df[['datetimeutc', 'value']].dropna()
```

[26]:

```
# Descriptive Analysis
# Line chart of PM1 over time
plt.figure(figsize=(12, 6))
plt.plot(df['datetimeutc'], df['value'], color='green')
plt.title("PM1 Levels Over Time")
plt.xlabel("Date")
plt.ylabel("PM1 (µg/m³)")
plt.grid(True)
plt.tight_layout()
plt.show()
```

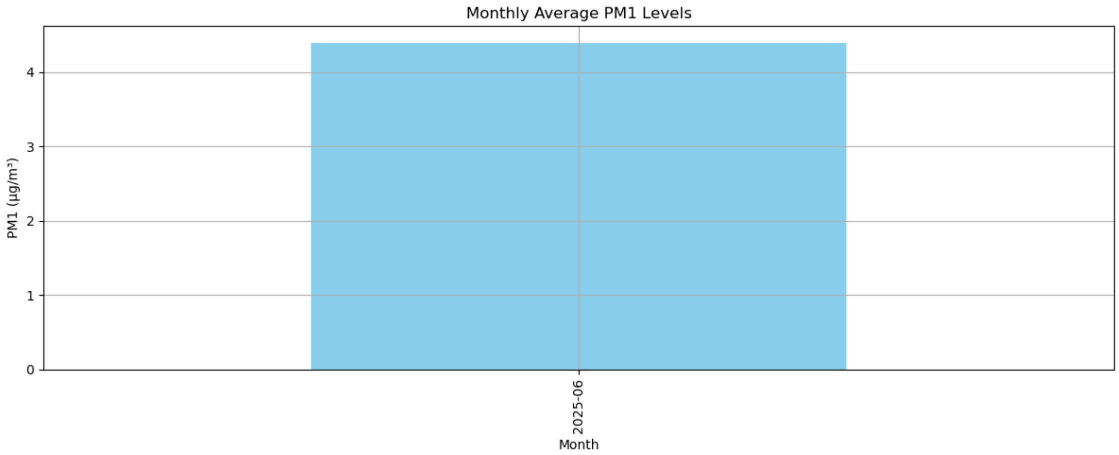


[27]:

```
# Monthly averages
df['month'] = df['datetimeutc'].dt.to_period('M')
monthly_avg = df.groupby('month')['value'].mean()

monthly_avg.plot(kind='bar', figsize=(12, 5), color='skyblue')
plt.title("Monthly Average PM1 Levels")
plt.ylabel("PM1 (µg/m³)")
plt.xlabel("Month")
plt.grid(True)
plt.tight_layout()
plt.show()
```

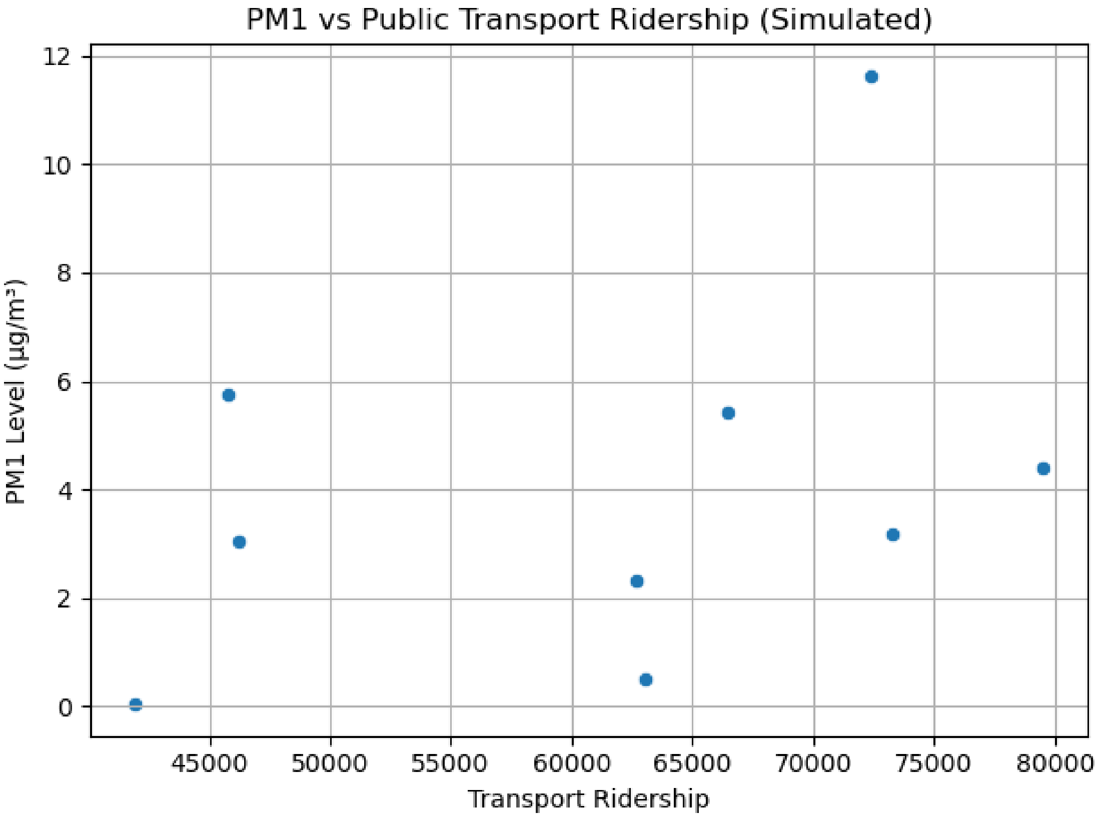
```
C:\Users\neeli\AppData\Local\Temp\ipykernel_161944\3678348255.py:2: UserWarning:
Converting to PeriodArray/Index representation will drop timezone information.
  df['month'] = df['datetimeutc'].dt.to_period('M')
```



[28]:

```
# Diagnostic Analysis: Simulated Transport Data
# Create fake public transport ridership data
df_daily = df.resample('D', on='datetimeutc').mean().dropna().reset_index()
df_daily['transport_ridership'] = np.random.randint(30000, 80000,␣
 ↪size=len(df_daily))

# Scatter Plot
sns.scatterplot(data=df_daily, x='transport_ridership', y='value')
plt.title("PM1 vs Public Transport Ridership (Simulated)")
plt.xlabel("Transport Ridership")
plt.ylabel("PM1 Level (µg/m³)")
plt.grid(True)
plt.tight_layout()
plt.show()
```



[30]:

```
# Inferential Analysis: Linear Regression
X = df_daily[['transport_ridership']]
y = df_daily['value']
```

```
X = sm.add_constant(X)

model = sm.OLS(y, X).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  value   R-squared:                       0.149
Model:                            OLS   Adj. R-squared:                  0.027
Method:                 Least Squares   F-statistic:                     1.225
Date:                Sun, 22 Jun 2025   Prob (F-statistic):              0.305
Time:                        19:39:01   Log-Likelihood:                -22.695
No. Observations:                   9   AIC:                             49.39
Df Residuals:                       7   BIC:                             49.79
Df Model:                           1
Covariance Type:            nonrobust
================================================================================
=======
                          coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
-------
const                  -2.0070      5.572     -0.360      0.729     -15.184
11.170
transport_ridership  9.857e-05   8.91e-05      1.107      0.305      -0.000
0.000
==============================================================================
Omnibus:                        3.228   Durbin-Watson:                   1.465
Prob(Omnibus):                  0.199   Jarque-Bera (JB):                1.342
Skew:                           0.944   Prob(JB):                        0.511
Kurtosis:                       2.890   Cond. No.                     3.06e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 3.06e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

```
C:\ProgramData\anaconda3\New folder\anaconda\Lib\site-
packages\scipy\stats\_axis_nan_policy.py:531: UserWarning: kurtosistest only
valid for n>=20 … continuing anyway, n=9
  res = hypotest_fun_out(*samples, **kwds)
```

[31]:

```
# Correlation
corr = df_daily['value'].corr(df_daily['transport_ridership'])
print("Correlation between PM1 and Transport Ridership:", round(corr, 3))
```

```
Correlation between PM1 and Transport Ridership: 0.386
```

[34]:

```
print("""
 Recommendations:
1. Increase frequency and accessibility of public transport to lower pollution.
2. Promote electric or non-polluting transit options.
3. Monitor high PM1 days to introduce 'no car' or 'green' days.
""")
```

```
 Recommendations:
1. Increase frequency and accessibility of public transport to lower pollution.
2. Promote electric or non-polluting transit options.
3. Monitor high PM1 days to introduce 'no car' or 'green' days.
```

[35]:

```
# Summary Output
print("Summary:")
print("Average PM1 Level:", round(df['value'].mean(), 2))
print("Max PM1 Level:", round(df['value'].max(), 2))
print("Min PM1 Level:", round(df['value'].min(), 2))
```

```
Summary:
Average PM1 Level: 4.39
Max PM1 Level: 22.04
Min PM1 Level: 0.0
```

[ ]:

```

```