milan-air-analysis

August 23, 2025

0.0.1 Milan's Air Quality Analysis

Milan faces significant air pollution challenges, particularly due to its location in the Po Plain, a region known for heavy industry and urbanization. The city frequently experiences high levels of PM2.5 (fine particulate matter), exceeding World Health Organization limits and leading to health concerns for residents. PM2.5 concentration is currently 2.2 times the World Health Organization annual PM2.5 guideline value. While Milan has implemented measures like traffic restrictions and investing in cleaner transportation, geographical factors and the concentration of industrial and agricultural activities continue to contribute to the problem. In this project I intend to delve deeper into this phenomenona and illustrate the following: * Descriptive Analysis. * Health & Policy Relevance. * Seasonal & Temporal Insights. * Forecasting Using Machine Learning.

The data used for this project can be found on Kaggle

[4]:
```
# Importing necessarry libraries:
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[11]:
```
# Preparing data
milan = pd.read_csv('milan-air-pollution.csv', sep=';')
milan.columns = ["station_id", "date", "pollutant", "value"]
milan['date'] = pd.to_datetime(milan['date'], errors='coerce')
milan['value'] = pd.to_numeric(milan['value'], errors='coerce')
milan.to_csv('milan_air_pollution_cleaned.csv', index=False)
milan.tail(15000).to_csv('milan_air_pollution_sample.csv', index=False)
```

[6]:
```
milan = pd.read_csv('milan_air_pollution_sample.csv')
# Counting missing values
missing_vals = milan.isnull().sum()
print(missing_vals)
```

```
station_id       0
date             0
pollutant        0
value         3142
dtype: int64
```

[8]:
```
milan.head()
```

[8]:

	station_id	date	pollutant	value
0	4	2023-03-10	PM25	17.0
1	5	2023-03-10	NO2	NaN
2	5	2023-03-10	O3	NaN
3	6	2023-03-10	C6H6	1.2
4	6	2023-03-10	CO_8h	0.8

[9]:
```
# Replacing NaN values in the 'values' column with values' mean
milan = milan.assign(value=milan['value'].fillna(milan['value'].mean()))
milan['value'] = milan['value'].round(2)
milan.head()
```

[9]:

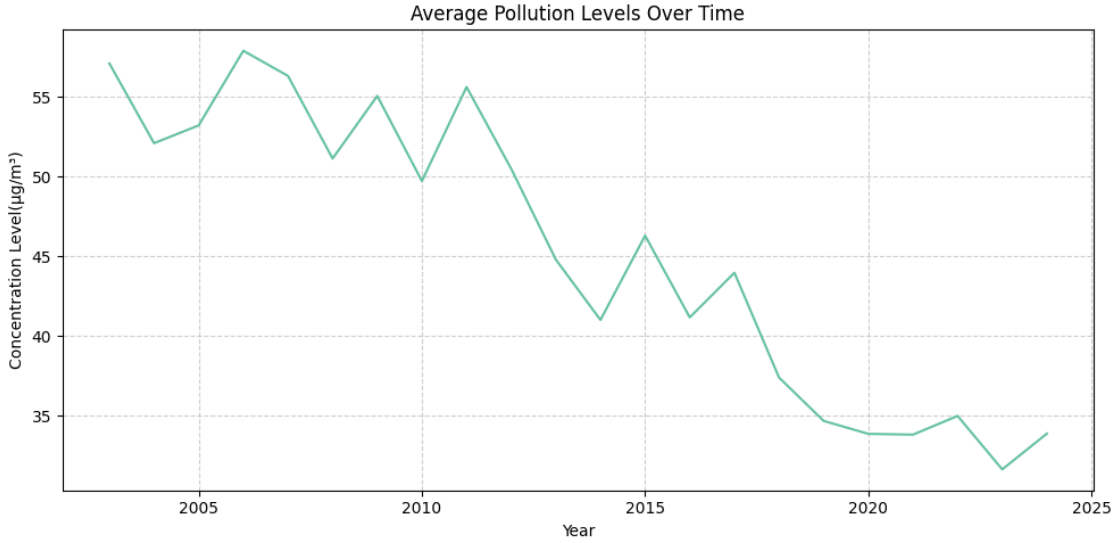
	station_id	date	pollutant	value
0	4	2023-03-10	PM25	17.00
1	5	2023-03-10	NO2	38.63
2	5	2023-03-10	O3	38.63
3	6	2023-03-10	C6H6	1.20
4	6	2023-03-10	CO_8h	0.80

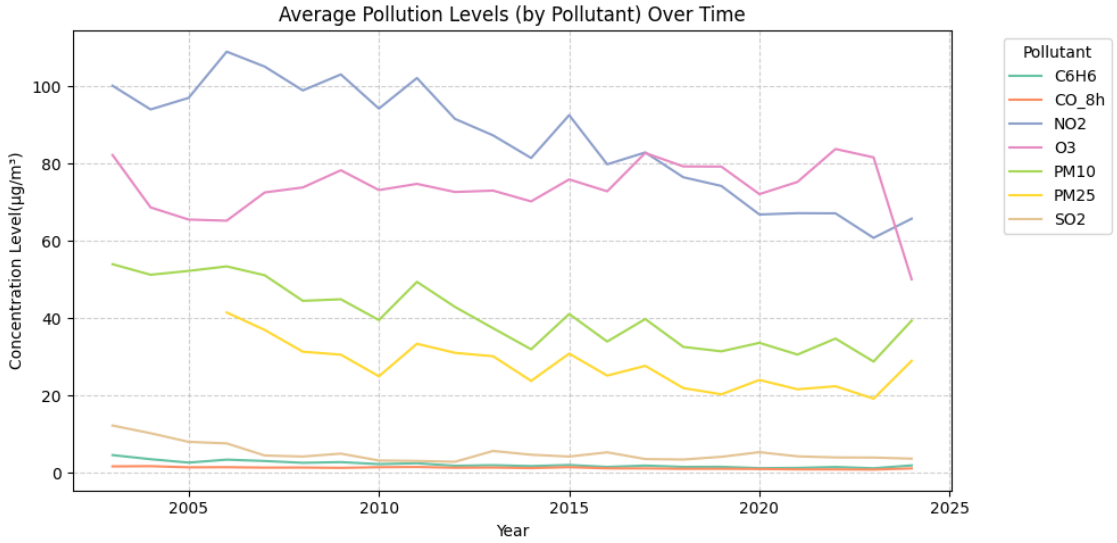
0.1 Trend Analysis

The data is now clean to use. Before we start analyzing, let's go over each pollutant in our data and where it comes from: * C H (Benzene) - Sources: Vehicle exhaust (especially petrol engines), industrial emissions (chemical plants, refineries), cigarette smoke, evaporation from gasoline and solvents. * CO (Carbon Monoxide, often measured as 8-hour average, CO_8h) - Sources: Incomplete combustion of fossil fuels (cars, trucks, motorcycles), residential heating (wood stoves, gas heaters), industrial processes * NO (Nitrogen Dioxide) - Sources: Vehicle exhaust (diesel engines are major contributors), power plants and industrial combustion, natural sources (lightning, wildfires) * O (Ozone, ground-level) - Sources: not emitted directly. It forms photochemically from reactions between NOx and VOCs in sunlight. High levels often occur in urban areas with lots of sunlight and vehicle emissions. * PM (Particulate Matter 10 µm) - Sources: Dust from roads and construction, industrial emissions, vehicle exhaust (especially diesel), natural sources (soil dust, pollen) * PM . (Particulate Matter 2.5 µm) - Sources: Combustion processes (vehicles, power plants, residential heating), secondary formation from gases like SO , NOx, and VOCs, wildfires * SO (Sulfur Dioxide) - Sources: Burning of fossil fuels with sulfur (coal, oil, diesel), industrial processes (metal smelting, refineries), volcanic eruptions (natural source) **Now, let's illustrate an overview at our data and see how pollution levels change over time:**

[12]:
```
milan['year'] = milan['date'].dt.year
avgmilan = milan.groupby(['year'])['value'].mean().reset_index()
avg_by_pollutant = milan.groupby(['year', 'pollutant'])['value'].mean().
  ↪reset_index()
sns.set_palette("Set2")
plt.figure(figsize=(10,5))
sns.lineplot(data=avgmilan, x='year', y='value')
```

```
plt.grid(alpha=0.6, linestyle='--')
plt.title("Average Pollution Levels Over Time")
plt.xlabel("Year")
plt.ylabel("Concentration Level(µg/m³)")
plt.tight_layout()
plt.show()
plt.figure(figsize=(10,5))
sns.lineplot(data=avg_by_pollutant, x='year', y='value', hue='pollutant')
plt.grid(alpha=0.6, linestyle='--')
plt.title("Average Pollution Levels (by Pollutant) Over Time")
plt.xlabel("Year")
plt.ylabel("Concentration Level(µg/m³)")
plt.legend(title='Pollutant', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```





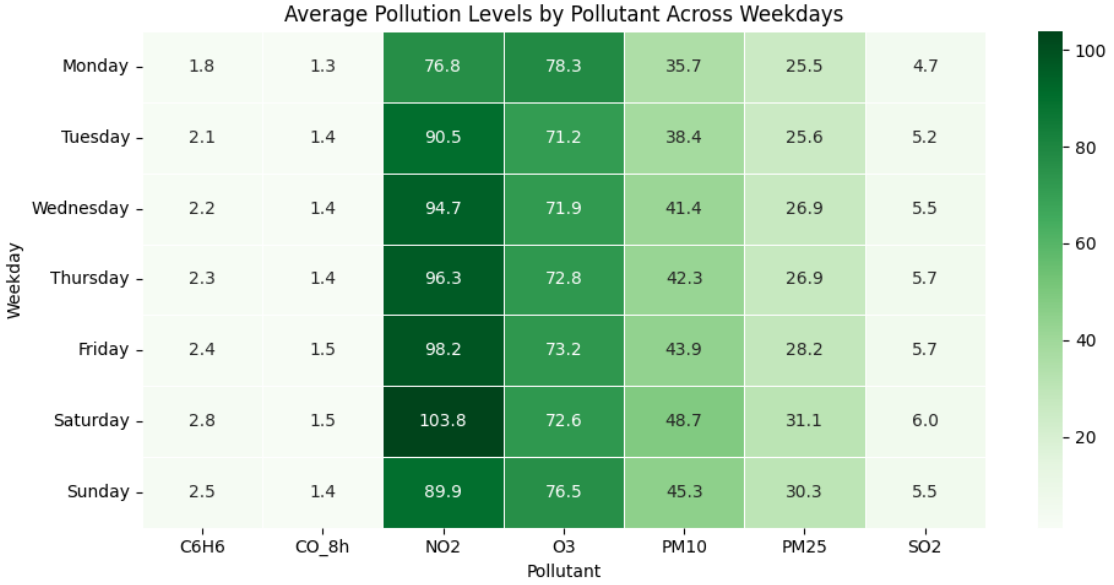
The annual average concentrations of pollutants in Milan show clear long-term improvements. NO , PM10, and PM2.5 have declined significantly since the early 2000s, reflecting the impact of stricter emission regulations, cleaner fuels, and reduced traffic during the COVID-19 lockdowns. Benzene (C H) and SO have also nearly disappeared as major pollutants, thanks to industrial controls and fuel restrictions. However, O levels remain relatively high and fluctuate across years, highlighting the persistent challenge of secondary pollutants that increase when NO decreases. Overall, Milan's air quality has improved substantially, but ozone and particulate matter remain important concerns for public health.

0.2 Weekday Analysis

Weekly analysis is conducted to understand behavioral or traffic-related pollution dynamics. While analyzing pollution trends by weekday, I noticed that data for Saturday and Sunday is missing from the dataset. This absence could be due to gaps in data collection, sensor downtime, or reporting inconsistencies during weekends. As a result, the weekday analysis only reflects Monday through Friday. Although weekly analysis is important (since industrial activities follow a weekly rhythm and traffic emissions spike on weekdays), a detailed breakdown cannot be performed due to the missing weekend data. However, I will illustrate the levels of each pollutant on a weekly basis to analyze the range within which each pollutant falls and to assess whether those levels are environmentally healthy.

[13]:
```
milan['weekday'] = milan['date'].dt.day_name()
weekday_avg = milan.groupby(['weekday', 'pollutant'])['value'].mean().
 ↪reset_index()
weekday_order = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday',␣
 ↪'Saturday', 'Sunday']
weekday_avg['weekday'] = pd.Categorical(weekday_avg['weekday'],␣
 ↪categories=weekday_order, ordered=True)
weekday_avg.sort_values('weekday', inplace=True)
```

```
heatmap_data = weekday_avg.pivot(index='weekday', columns='pollutant',␣
↪values='value')
plt.figure(figsize=(10, 5))
sns.heatmap(heatmap_data, cmap='Greens', annot=True, fmt=".1f", linewidths=0.5)
plt.title("Average Pollution Levels by Pollutant Across Weekdays")
plt.xlabel("Pollutant")
plt.ylabel("Weekday")
plt.tight_layout()
plt.show()
```

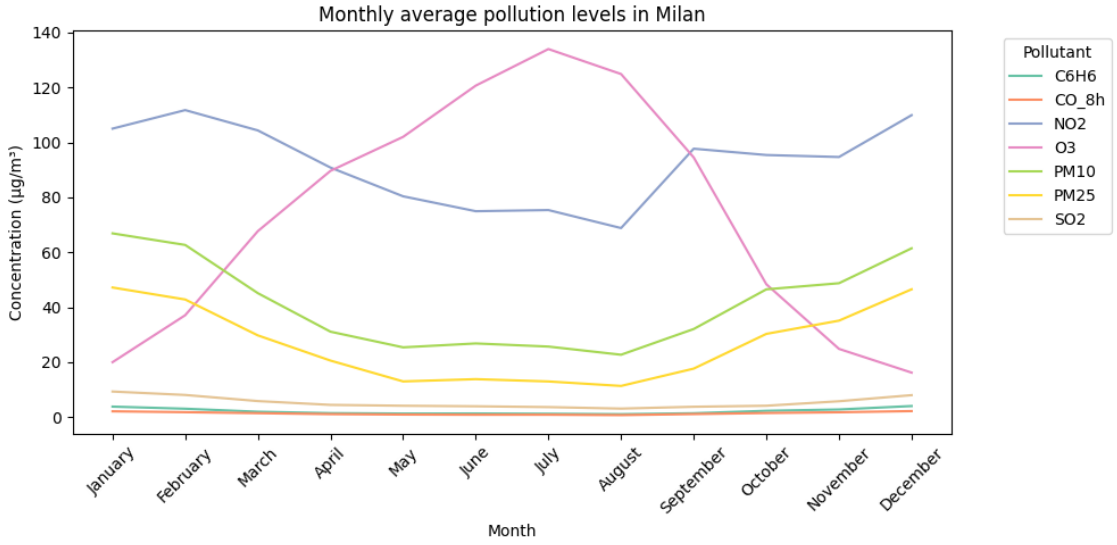


Let's analyze this data against WHO 2021 Air Quality Guidelines: 1. **C H (Benzene)** *(Lower is Better)* - Range: 1.8 – 2.8 µg/m³, below EU annual limit (5 µg/m³). 2. **CO_8h** *(Safe)* - Range: 1.3 – 1.5 mg/m³, which is far below WHO guideline(4 mg/m³). 3. **NO** *(Unhealthy)* - Range: 76.8 – 103.8 µg/m³ - WHO daily guideline = 25 µg/m³(levels are 3–4x higher). 4. **O** *(Safe)* - Range: 71 – 78 µg/m³ - Below WHO limit(100 µg/m³). 5. **PM** *(Unhealthy)* - Range: 35.7 – 48.7 µg/m³ - WHO daily limit = 45 µg/m³(borderline or exceeded on Thu–Sun). 6. **PM .** *(Very Unhealthy)* - Range: 25.5 – 31.1 µg/m³ - WHO daily limit = 15 µg/m³(almost 2× higher every day). 7. **SO** *(Safe)* - Range: 4.7 – 6.0 µg/m³, below WHO 24h guideline (20 µg/m³).

0.3 Seasonal Trend

Seasonal trends matter in data analysis because they represent predictable, recurring patterns in data that are essential for accurate forecasting and informed decision-making. Milan experiences much higher air pollution in winter compared to summer, with PM . concentrations peaking in the cold months. This seasonal variation is primarily driven by the Po Valley's geography, which traps pollutants in the winter, combined with seasonal emissions from winter heating, and the influence of dust storms and biomass burning that affect different seasons.

[14]:
```
milan['month'] = milan['date'].dt.month_name()
month_order = ['January', 'February', 'March', 'April', 'May', 'June',
               'July', 'August', 'September', 'October', 'November', 'December']
milan['month'] = pd.Categorical(milan['month'], categories=month_order)
monthavg = milan.groupby(['month', 'pollutant'], observed=True)['value'].mean().
 ↪reset_index()
plt.figure(figsize=(10, 5))
sns.lineplot(data=monthavg, x='month', y='value', hue='pollutant')
plt.title("Monthly average pollution levels in Milan")
plt.xlabel("Month")
plt.xticks(rotation=45)
plt.ylabel("Concentration (µg/m³)")
plt.legend(title='Pollutant', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```



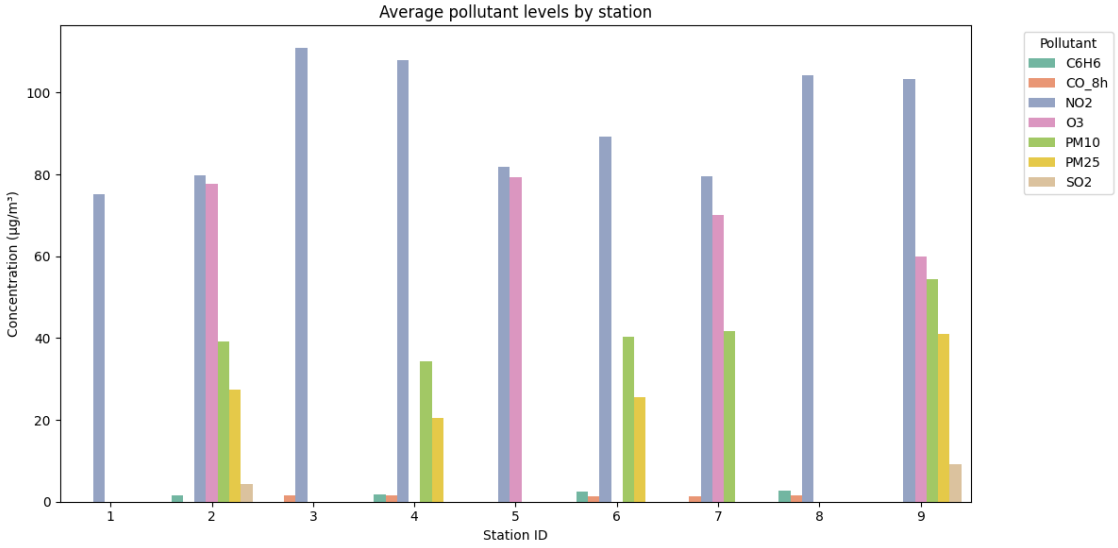
As displayed in the plot, the levels of NO , PM and PM . dramatically decline throughout February to May, stay steady from May to August then begin to rise as autumn takes place. Which can easily be explained by the combination of increased pollution sources, such as greater demand for heating and personal vehicle use, and atmospheric conditions like temperature inversions, where a layer of warm air traps cold, polluted air near the ground. Not a noticable change is seen when it comes to SO , C H , and CO_8h as they're low and steady throughout the entire year. Ozone however starts at 20µg/m³ at the beginning of the year and hits its peak of approximately 140µg/m³ in July then proceeds to decline back to 20µg/m³ after that.

0.4 Station Comparison

Station comparison is actually very important in our analysis, because it allows us to detect *spatial differences* in pollution whithin Milan. For instance, NO concentrations are consistently higher in

traffic-dominated stations, confirming road traffic as the main contributor. In contrast, O levels remain similar across all stations, reflecting its secondary and regional nature.

[15]:
```
station_avg = milan.groupby(["station_id", "pollutant"])["value"].mean().
↪reset_index()

plt.figure(figsize=(12,6))
sns.barplot(data=station_avg, x="station_id", y="value", hue="pollutant")
plt.title("Average pollutant levels by station")
plt.xlabel("Station ID")
plt.ylabel("Concentration (µg/m³)")
plt.legend(title='Pollutant', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```



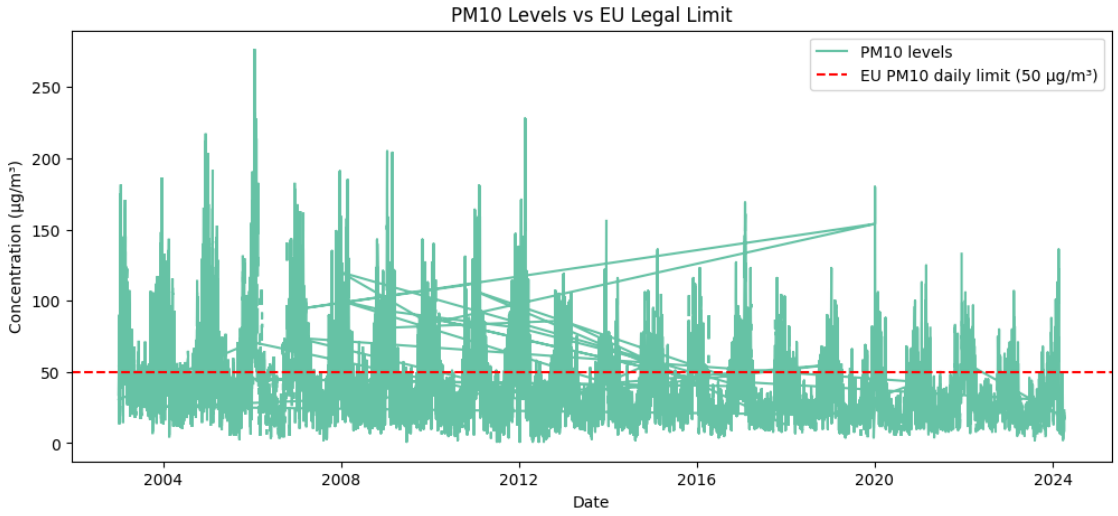
Let's take a look at this plot and see what it tells us: * The tallest bar (blue) at almost every station: NO is the most critical pollutant citywide, Which suggests Milan's traffic emissions are a major pollution source. * O (pink) is the second-highest pollutant: Ozone is a secondary pollutant formed in sunlight from NO + VOCs. * PM10 and PM2.5 (green + yellow): Lower than NO and Ozone but still significantly important since they're directly linked to health risks (they can cause problems such as tissue damage, or lung inflammation). * Other pollutants (CO, SO , benzene): Very low across all stations, likely due to strict regulations on fuel. * Some stations (e.g., Station 3 and 4) show higher NO averages than others, probably near busy roads/traffic-heavy zones. * Some stations (like 6 and 7) have slightly lower pollutant levels, possibly in suburban or less industrial areas.

0.5 Health & Policy Relevance

Air pollution is one of the leading environmental risks to health, linked to respiratory and cardiovascular diseases, reduced life expectancy, and premature deaths. In the EU, pollutants such

as PM10, PM2.5, and NO are regulated under strict air quality directives, which set daily and annual limits to protect public health. Milan is known for its air pollution challenges due to dense traffic, industrial activities, and meteorological conditions that trap pollutants, especially in winter as we saw earlier. By comparing pollutant levels to EU thresholds, we can evaluate whether Milan meets these standards and highlight areas where policy action is urgently needed. ### Why PM10? This section compares PM10 levels in Milan to the EU daily limit of 50 µg/m³. PM10 is a critical pollutant in Milan due to its strong health impacts and frequent exceedances in the Po Valley region. Elevated PM10 concentrations increase risks of respiratory and cardiovascular diseases, making it a key focus for both public health and urban policy. Exceedances of this threshold indicate where Milan fails to comply with EU air quality standards, highlighting the urgency of interventions such as traffic restrictions, low-emission zones, and alternative heating strategies.

[16]:
```
pm10 = milan[milan["pollutant"] == "PM10"]

plt.figure(figsize=(12,5))
plt.plot(pm10["date"], pm10["value"], label="PM10 levels")
plt.axhline(50, color="red", linestyle="--", label="EU PM10 daily limit (50 µg/
↪m³)")
plt.title("PM10 Levels vs EU Legal Limit")
plt.xlabel("Date")
plt.ylabel("Concentration (µg/m³)")
plt.legend()
plt.show()
exceedances = (pm10["value"] > 50).sum()
print(f"Days exceeding EU PM10 limit: {exceedances}")
```




```
Days exceeding EU PM10 limit: 8930
```

As visible, PM10 levels in Milan regularly exceed the EU daily limit of 50 µg/m³, often by a wide margin. This indicates persistent air quality issues, especially during winter, with potential health impacts such as respiratory irritation, increased hospital admissions, and long-term risks like lung

and heart disease. Air quality fails EU standards; residents are at increased health risk, and stricter measures are needed.

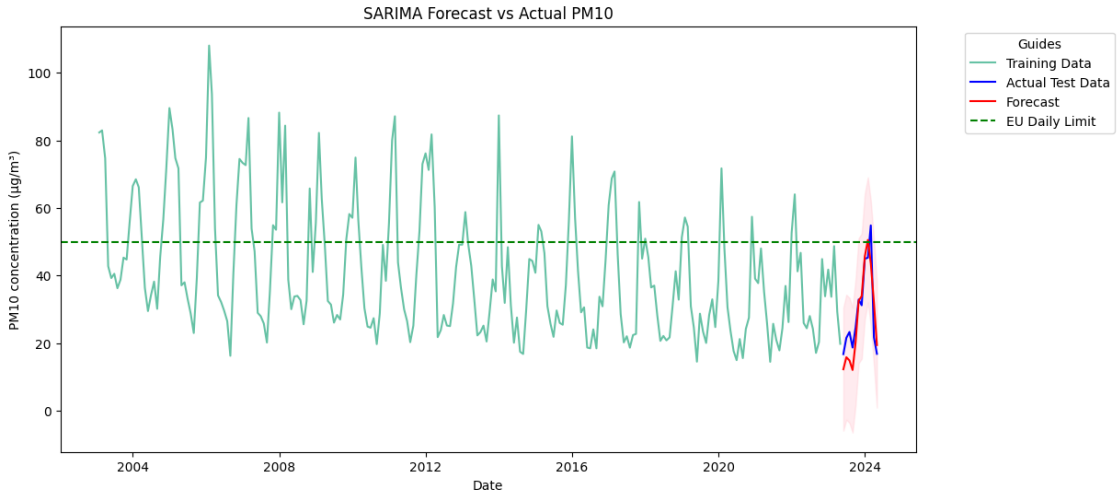
0.6 Forecasting Future PM10 Concentrations in Milan(using SARIMA)

This section outlines the methodology used to generate a 12-month forecast, building on historical data and seasonal patterns observed in previous years. I chose a SARIMA (Seasonal AutoRegressive Integrated Moving Average) model due to its ability to capture both trend and seasonality, crucial for pollutants like PM10, which often spike during colder months and decline in warmer seasons.

After validating the model on recent data (2024), I retrained it on the full dataset to forecast concentrations into 2025.

[23]:
```
from statsmodels.tsa.statespace.sarimax import SARIMAX
from sklearn.metrics import mean_absolute_error, mean_squared_error

pm10 = milan[milan["pollutant"] == "PM10"].copy()
pm10["date"] = pd.to_datetime(pm10["date"])
pm10 = pm10.set_index("date").sort_index()
pm10_series = pm10["value"].resample("ME").mean().dropna()
train = pm10_series[:-12]
test = pm10_series[-12:]


model = SARIMAX(train, order=(1,1,1), seasonal_order=(1,1,1,12))
results = model.fit()


forecast = results.get_forecast(steps=12)
forecast_mean = forecast.predicted_mean
conf_int = forecast.conf_int()


plt.figure(figsize=(12,6))
plt.plot(train, label="Training Data")
plt.plot(test, label="Actual Test Data", color="blue")
plt.plot(test.index, forecast_mean, label="Forecast", color="red")
plt.fill_between(test.index, conf_int.iloc[:, 0], conf_int.iloc[:, 1],␣
 ↪color='pink', alpha=0.3)
plt.axhline(y=50, color="green", linestyle="--", label="EU Daily Limit")
plt.title("SARIMA Forecast vs Actual PM10")
plt.xlabel("Date")
plt.ylabel("PM10 concentration (µg/m³)")
plt.legend(title='Guides', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()


mae = mean_absolute_error(test, forecast_mean)
```

```
rmse = np.sqrt(mean_squared_error(test, forecast_mean))
print(f"MAE: {mae:.2f}")
print(f"RMSE: {rmse:.2f}")
```



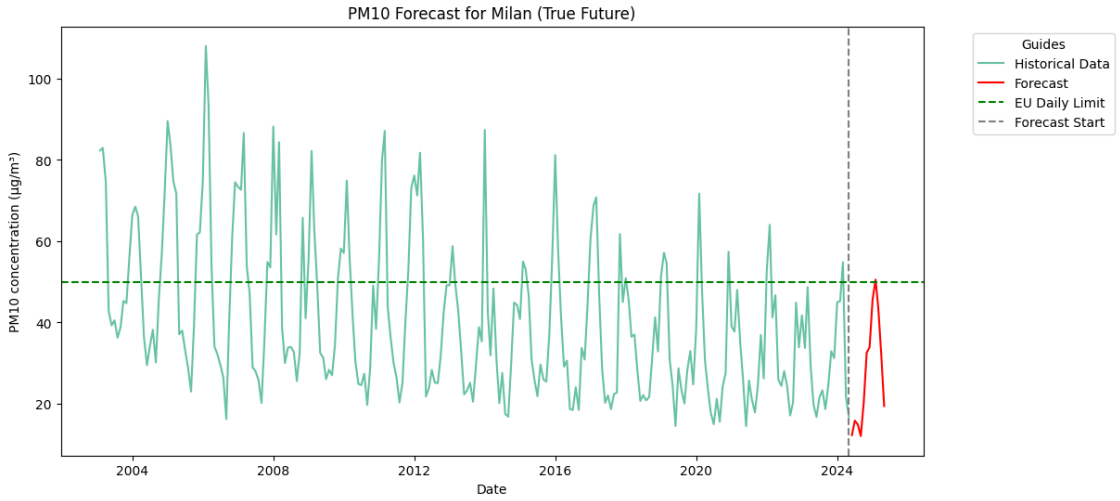

```
MAE: 5.23
RMSE: 6.20
```

MAE = 5.23: Means that on average, the forecast is off by about 5.23 µg/m³. **RMSE = 6.20:** Slightly higher than MAE, indicating some larger errors in a few months.

These values are reasonable and indicate the model's efficiency. Now that we've made sure the model performes accurately, it's time to forecast away:

[22]:
```
last_date = pm10_series.index[-1]

future_dates = pd.date_range(start=last_date + pd.DateOffset(months=1),␣
 ↪periods=12, freq='MS')

future_forecast = results.get_forecast(steps=12)
forecast_mean = future_forecast.predicted_mean
forecast_mean.index = future_dates

plt.figure(figsize=(12,6))
plt.plot(pm10_series, label="Historical Data")
plt.plot(forecast_mean.index, forecast_mean, label="Forecast", color="red")
plt.axhline(y=50, color="green", linestyle="--", label="EU Daily Limit")
plt.title("PM10 Forecast for Milan (True Future)")
plt.xlabel("Date")
plt.ylabel("PM10 concentration (µg/m³)")
plt.axvline(x=pm10_series.index[-1], color="gray", linestyle="--",␣
 ↪label="Forecast Start")
```

```
plt.legend(title='Guides', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



The forecasted PM10 concentrations in Milan exhibit a clear seasonal pattern, with periodic peaks likely corresponding to colder months. Despite these fluctuations, the predicted values largely remain below the EU daily limit of 50 µg/m³, indicating a stable and improving air quality trend. The model demonstrates strong alignment with historical data and maintains realistic projections, suggesting it is well-calibrated for short-term forecasting. These insights support the effectiveness of ongoing environmental policies while highlighting the need for continued vigilance during high-risk periods.

This project provided a comprehensive analysis of PM10 air pollution trends in Milan, combining historical data with SARIMA-based forecasting to evaluate future risks. The results suggest a generally positive trajectory in air quality, with seasonal peaks that warrant continued attention.Through this work, I deepened my understanding of time series modeling, data visualization, and environmental analytics.

Looking ahead, this framework can be extended to include additional pollutants, integrate exogenous variables like weather or traffic, and support policy evaluation. I plan to further refine the model and explore its application in other urban contexts, contributing to data-driven environmental decision-making.

Sania Latifi EmailGithub