

AIR QUALITY REPORT



Mohammad Adnan Sherwani

8860414

Multivariate Statistics STAT 8030

Project 1

Air is what keeps humans alive. Monitoring it and understanding its quality is of immense importance to our well-being. Around the world, about 3 million people die every year due to air quality problems. Many studies have shown that atmospheric pollutants have a certain correlation with lung cancer and cardiovascular disease. . The air monitoring items mainly include PM2.5, PM10, CO, NO2, SO2, and O3 (''two dusts and four gases").

The data has been made publicly available by the Central Pollution Control Board: <https://cpcb.nic.in/> which is the official portal of Government of India.



The dataset used is hourly air quality data (2015 - 2020) with weather data combined/merge externally <https://www.kaggle.com/datasets/hiteshsoneji/historical-weather-data-for-indian-cities?select=delhi.csv>

We'll use one station (RK Puram in Delhi) and compare it with the hourly weather data.

The air quality dataset for this project is collected from the CPCD repository i.e., publicly available by the Central Pollution Control Board: <https://cpcb.nic.in/> which is the official portal of Government of India. The dataset is available in XLSX format. It is downloaded and imported to the project. The dataset contains data of average hourly responses of different elements in the air for nearly 5 years from April 2015 to December 2019. The second dataset i.e., the hourly weather data consist of several datasets of temperature in Delhi for the relevant time period including other observations such as rainfall, minimum temperature, maximum temperature etc. Using Python for combining/merging the data. The total dataset consists of 34380 rows and 37 columns. The main objective to achieve from the dataset is the role of temperature inversion which causes pollution level increase or decrease due to the phenomena where the warmer air rises and acts as a lid trapping the colder air close to the ground. Pollution, including that from road traffic is also trapped, so **the air layer closest to the ground becomes more and more polluted**.

*Attributes:*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Attributes* | *Data Types* | | *Description* | |
| *From* | *datetime64*[ns] | | Timestamp’YYYY’-‘MM’-‘DD’‘HH’:’MM’ | |
| PM2.5 | float64 | | mg/m3 Fine Particulate Matter 2.5 | |
| PM10 | float64 | | mg/m3 Particulate Matter 10 | |
| NO | float64 | | Nitrogen Oxide mg/m3 | |
| NO2 | float64 | | Nitrogen Dioxide mg/m3 | |
| NOx | float64 | | Nitric X-Oxide ppb | |
| SO2 | | float64 | | Sulphur Oxide mg/m3 |
| CO | | float64 | | Carbon Monoxide mg/m3 |
| Ozone | | float64 | | Ozone mg/m3 |
| Year | | int64 | | Year |
| Month | | int64 | | Month |
| Day | | int64 | | Day |
| Hour | | int64 | | Hour |
| WeekDay | | object | | Week |
| maxtempC | | int64 | | Maximum Temperature |
| mintempC | | int64 | | Minimum Temperature |
| sunHour | | float64 | | Time from sunrise to sunset |
| uvIndex | | int64 | | Ultraviolet rays index |
| uvIndex.1 | | int64 | |  |
| moonrise | | object | | Time moonrise |
| moonset | | object | | Time Moonset |
| sunrise | | object | | Sun rise time |
| sunset | | object | | Sun set time |
| DewPointC | | int64 | | Dew |
| FeelsLikeC | | int64 | | Temperature Celsius |
| HeatIndexC | | int64 | | Heat Index |
| WindChillC | | int64 | | Wind Index |
| WindGustKmph | | int64 | | Wind Speed KMPH |
| cloudcover | | int64 | | Cloud |
| humidity | | int64 | | Humidity |
| precipMM | | float64 | | Precipitation |
| pressure | | int64 | | Pressure |
| tempC | | int64 | | Temperatureq |
| visibility | | int64 | | Visibility |
| winddirDegree | | int64 | | Wind Degree |
| windspeedKmph | | int64 | | Wind Speed |
| AQI\_VALUE | | int32 | | AQI |
| Date | | object | | Date ‘YYYY’-‘MM’-‘DD’ |
|  | |  | |  |

**Data Cleaning:**

The data is cleaned with Python using the software Jupyter Notebook.

Process:

1. Changing datatypes from char to datetime/double/int
2. Replacing str ‘None’ characters with the NaN variable for NA observations
3. Filtering all NaN observations in temperature, PM10, PM2.5
4. Removing columns with MORE THAN 5% of NaN values using drop method.
5. Extracting Year, Month, Week, Day from the date timestamp.
6. Filter weather data
7. Merge weather data with air quality data.
8. Using groupby, creating dataframe for daily observations
9. Creating New columns for better understanding of the Air Quality Index(AQI)

Note:

1. The AQI calculation uses 7 measures: **PM2.5, PM10, SO2, NOx, NH3, CO and O3**.
2. For **PM2.5, PM10, SO2, NOx and NH3** the average value in last 24-hrs is used with the condition of having at least 16 values.
3. For **CO and O3** the maximum value in last 8-hrs is used.
4. Each measure is converted into a Sub-Index based on pre-defined groups.
5. Sometimes measures are not available due to lack of measuring or lack of required data points.
6. Final AQI is the maximum Sub-Index with the condition that at least one of PM2.5 and PM10 should be available and at least three out of the seven should be available.

With the help of CBCR website <https://app.cpcbccr.com/ccr_docs/How_AQI_Calculated.pdf>

and <https://www.airnow.gov/sites/default/files/2020-05/aqi-technical-assistance-document-sept2018.pdf>

I was able to extract AQI from the data provided by CPCB.

With different models used, we came to certain conclusion that the Air Quality increases with the decrease in temperature. Here, the roll of temperature inversion plays a factor for increased AQI.

The total sun hours in a day also proves the point the as the number of hours of day > the air quality will better than the air quality with a smaller number of sun hours.