CA01: Exploratory Data Analysis – India Pollution Data

Exploratory Data Analysis is an approach analyzing data sets to summarize their main characteristics such as mean, standard deviation, and count, so on, often with visual methods. It's where the researcher takes a bird's eye view of the data and tries to make some sense of it. It's often the first step in data analysis, implemented before any formal statistical techniques are applied.

India Air Quality Dataset

As always, learning by doing is the best practice to understand deeper. So now, we are going to make our hands dirt by analyzing **India Air Quality Dataset**.

You first need to analyze the structure, domain, and contents of it thoroughly.

Air Pollution

Dataset: Basic Info

What is it about?
This data is released by the Ministry of Environment and Forests and Central Pollution Control Board of India under the National Data Sharing and Accessibility Policy (NDSAP), using this one can explore India's air pollution levels at a more granular scale.

What information it has?
The dataset has 13 columns which are,

- **stn_code**: Station Code

- **sampling_date**: Date of sampling (note how this is formatted)
- **state**: State
- **location**: Location of recording
- **agency**: Agency
- **type**: Type of area
- **so2**: Sulphur dioxide (µg/m3)
- **no2**: Nitrogen dioxide (µg/m3)
- **rspm**: Respirable Suspended Particulate Matter (µg/m3)
- **spm**: Suspended Particulate Matter (µg/m3)
- **location_monitoring_station**: Location of data collection
- **pm2_5**: PSI 2.5 (µg/m3)
- **date**: Date of sampling

Why we need these details?
SPM, RSPM, PM2.5 values are the parameters used to measure the quality of air based on the number of particles present in it. Using these values, we are going to identify the air quality over the period of time in different states of India.

But, how?
This can only be answered once we analyze the data. 😉

Let's start: Import the dataset

You will download this data file (data.csv – 62.5 MB) from the BrightSpace CA01 folder.

Create a new Notebook using the Notebook Template provided with this assignment.

Copy the downloaded data file (data.csv) into the SAME folder as your new Notebook file (.ipynb). Then in your code read the data file like below:

==========

```
# import pandas package

import pandas as pd

# read data into a pandas data-frame called “data” (you can use any name)

data = pd.read_csv(‘data.csv')

# Note that there is no “path” specified in the above read statement, because the data file is in the same folder as this Notebook file
```

==========

Now, display the first few rows with head(), by default head() will return first 5 rows of the dataset, but you can specify any number of rows like *head(10)*.

Check the dataset info

As discussed earlier, the dataset has 13 columns in it. So how many rows are there? Many of the cells are filled with NaN, which is an unknown value and cannot contribute to our analysis. So how many such types of values are there? and how can we get rid of those? let’s find the answer to these questions.

To proceed with EDA, the initial level of investigation of data can be done using several commands,

data.shape

It returns a number of rows and columns in a dataset.

data.isnull().sum()

It returns a number of null values in each column.

data.info()

It returns range, column, number of non-null objects of each column, datatype and memory usage.

data.count()

It results in a number of non null values in each column.

Summarized details

Generate descriptive statistics that summarize the central tendency, dispersion, and shape of a dataset's distribution, excluding `NaN` values.

count: Count number of non-NA/null observations
mean: Mean of the values
std: Standard deviation of the observations
min: Minimum of the values in the object
max: Maximum of the values in the object

Cleansing the dataset

In this step, we need to clean the data by adding and dropping the needed and unwanted data respectively. From the above dataset,

- **Dropping of less valued columns:**
  stn_code, agency, sampling_date, location_monitoring_agency do not add much value to the dataset in terms of information. Therefore, we can **drop** those columns.
- **Changing the types to uniform format:**
  When you see the dataset, you may notice that the *'type'* column has values such as 'Industrial Area' and 'Industrial Areas' — both actually mean the same, so let's remove such type of stuff and make it uniform.
- **Creating a year column**
  To view the trend over a period of time, we need year values for each row and also when you see in most of the values in date column only has 'year' value. So, let's create a new column holding year values.

Handling missing values

The column such as SO2, NO2, rspm, spm, pm2_5 are the ones which contribute much to our analysis. So, we need to remove null from those columns to avoid inaccuracy in the prediction.

Use the `Imputer` from sklearn.preprocessing to fill the missing values in every column with the **mean.**

Now, check the number of null values in each column.

All set! Ready for Data Analysis …

Every preprocessing step are done, let's find out some higher level information from it. As I said earlier, so2, no2, rspm, and spm are the parameters that determine air quality in a particular locality. Now, let's frame a question and get the answer from data.

Which is the state that has higher SO2 content?
Group the data based on states and find the median for so2 content over a period of time, sort it and we will get the states with higher and lower level SO2 content. Provide a graphic visualization by comparing all states in a Bar Chart. Write your observation in Notebook.

Which is the state that has higher NO2 content?
Again the same process, but now for NO2 value, group the data based on states and find the median for NO2 content over a period of time, sort it and we will get the states with higher and lower level NO2 content. Provide a graphic visualization by comparing all states in a Bar Chart. Write your observation in Notebook.

In the same way, generate a graph for rspm and spm values and find the state with max and min rspm and spm value.

What is the yearly trend in a particular state, say 'Andhra Pradesh'?

We have created a new dataframe containing the NO2, SO2, rspm, and spm data regarding state 'Andhra Pradesh' only and group it by 'year'.

Now plot the data, in two graphs: NO2 and SO2; rspm and spm

Did you find anything alarming? What are your exploratory conclusions about air pollution in India? Write your observation in Notebook.

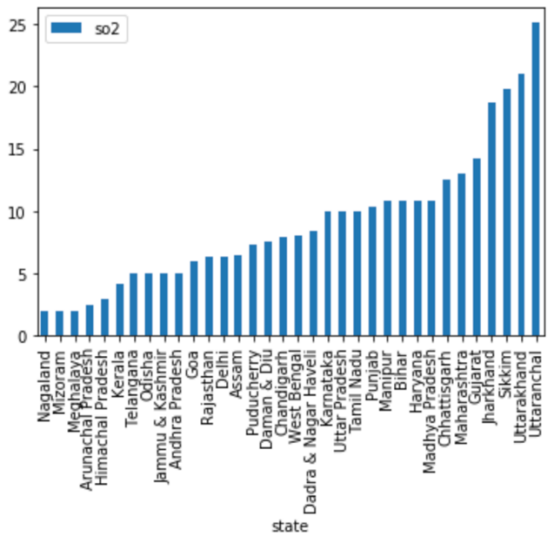
Like this, you can dig data into any deeper levels to find surprising facts. Data analysis is all about unraveling the hidden information.

CODING REQUIREMENTS:

Write all your code using the provided .ipynb template file.

SAMPLE PLOTS (Import and use matplotlib.pyplot)

State wise Pollutant Plot



Year wise Pollutant Comparison Plot