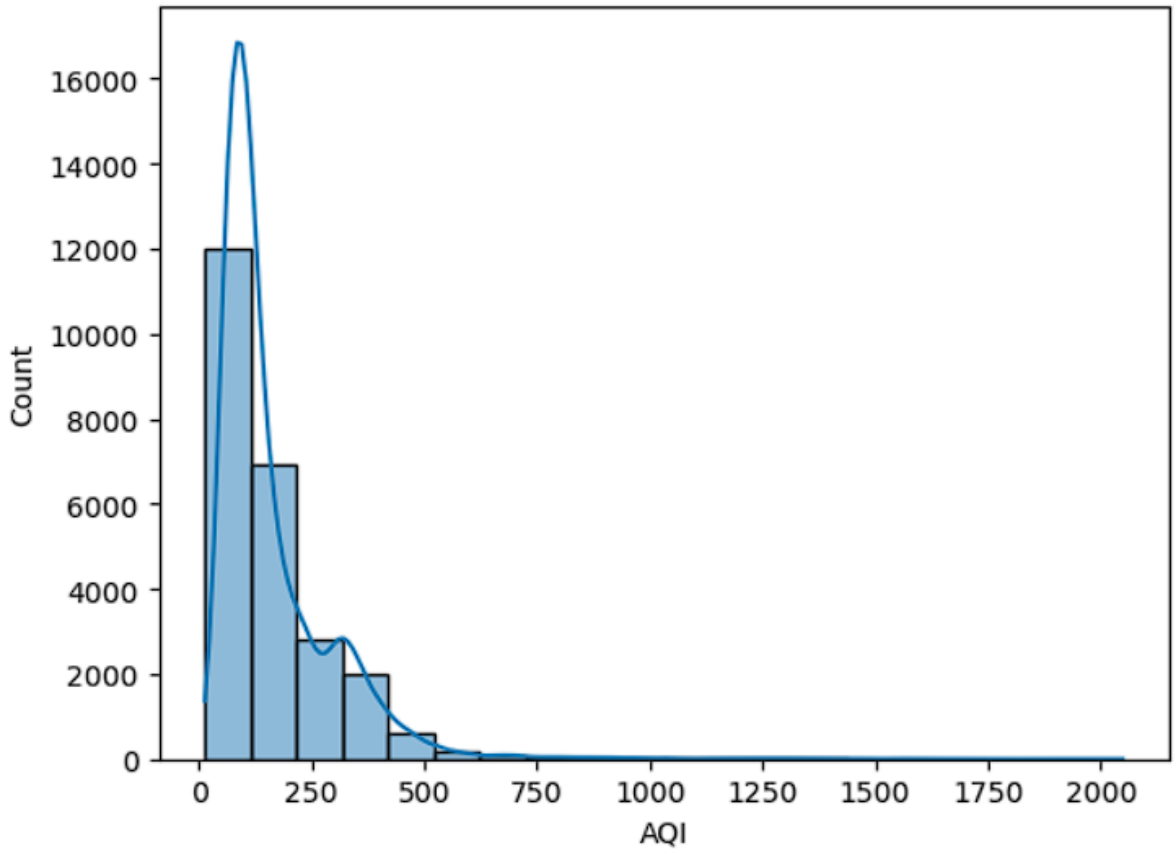
SMARTBRIDGE
Let's Bridge the Gap
a Veranda Enterprise

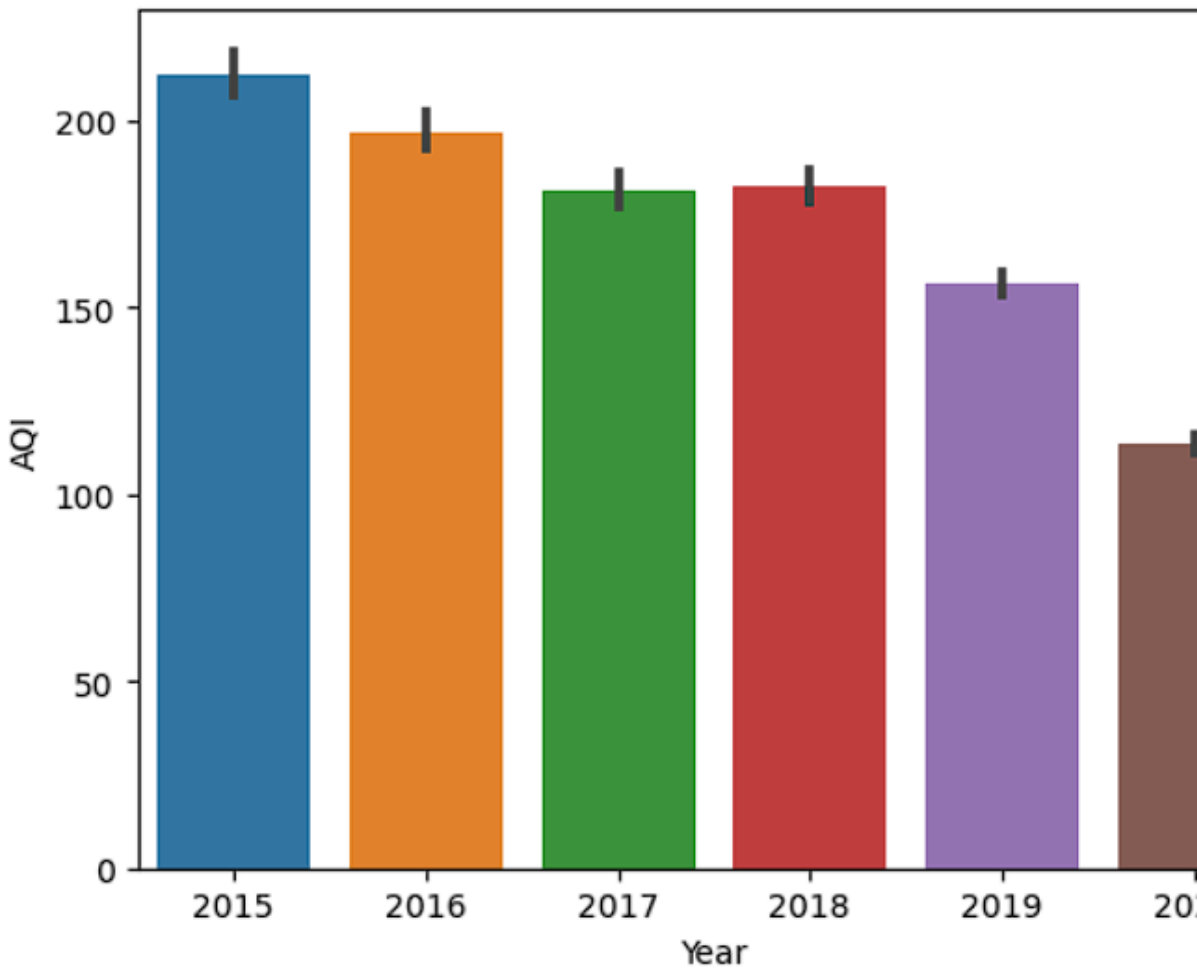
## Data Collection and Preprocessing Phase

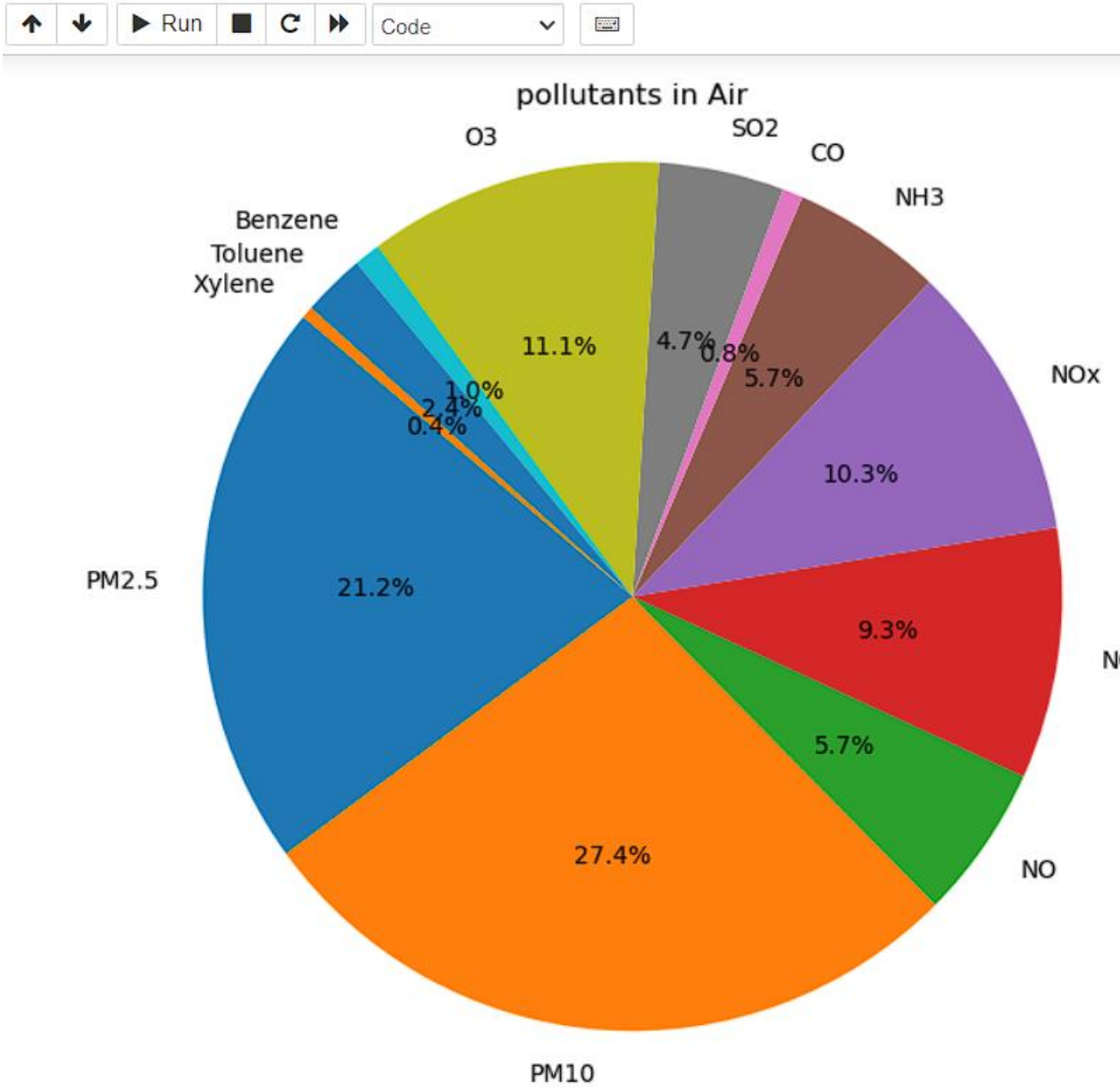
| | |
|---|---|
| Date | 07 JULY 2024 |
| Team ID | 739763 |
| Project Title | Air Quality Index Analyzer using machine learning |
| Maximum Marks | 6 Marks |

## Data Exploration and Preprocessing Report

Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis

| Section | Description |
|---|---|
| Data Overview | Basic statistics, dimensions, and structure of the data.<br>`data_city.describe()`<br>PM2.5 PM10 NO NO2 NOx NH3<br>count 24933.000000 18391.000000 25949.000000 25946.000000 25346.000000 19203.000000<br>mean 67.450578 118.127103 17.574730 28.560659 32.309123 23.483476<br>std 64.661449 90.605110 22.785846 24.474746 31.646011 25.684275<br>min 0.040000 0.010000 0.020000 0.010000 0.000000 0.010000<br>25% 28.820000 56.255000 5.630000 11.750000 12.820000 8.580000<br>50% 48.570000 95.680000 9.890000 21.690000 23.520000 15.850000<br>75% 80.590000 149.745000 19.950000 37.620000 40.127500 30.020000<br>max 949.990000 1000.000000 390.680000 362.210000 467.630000 352.890000 |
| Univariate Analysis | Exploration of individual variables (mean, median, mode, etc.).<br>  |
| Bivariate Analysis | Relationships between two variables (correlation, scatter plots). |

| | |
|---|---|
| |  |
| Multivariate Analysis | Patterns and relationships involving multiple variables. |

| | |
|---|---|
| |  |
| Outliers and Anomalies | Identification and treatment of outliers. |

26]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

def handle_outliers(df):
    # Plot boxplots before handling outliers
    plt.figure(figsize=(15, 10))
    df.boxplot(rot=90)
    plt.title('Boxplot Before Handling Outliers')
    plt.show()

    for column in df.columns:
        if pd.api.types.is_numeric_dtype(df[column]):
            Q1 = df[column].quantile(0.25)
            Q3 = df[column].quantile(0.75)
            IQR = Q3 - Q1
            lower_bound = Q1 - 1.5 * IQR
            upper_bound = Q3 + 1.5 * IQR

            # Cap the outliers
            df[column] = np.where(df[column] < lower_bound, lower_bound,
                                  np.where(df[column] > upper_bound, upper_bound, df[

    # Plot boxplots after handling outliers
    plt.figure(figsize=(15, 10))
    df.boxplot(rot=90)
    plt.title('Boxplot After Handling Outliers')
    plt.show()

    return df
```

**Data Preprocessing Code Screenshots**

Loading Data

Code to load the dataset into the preferred

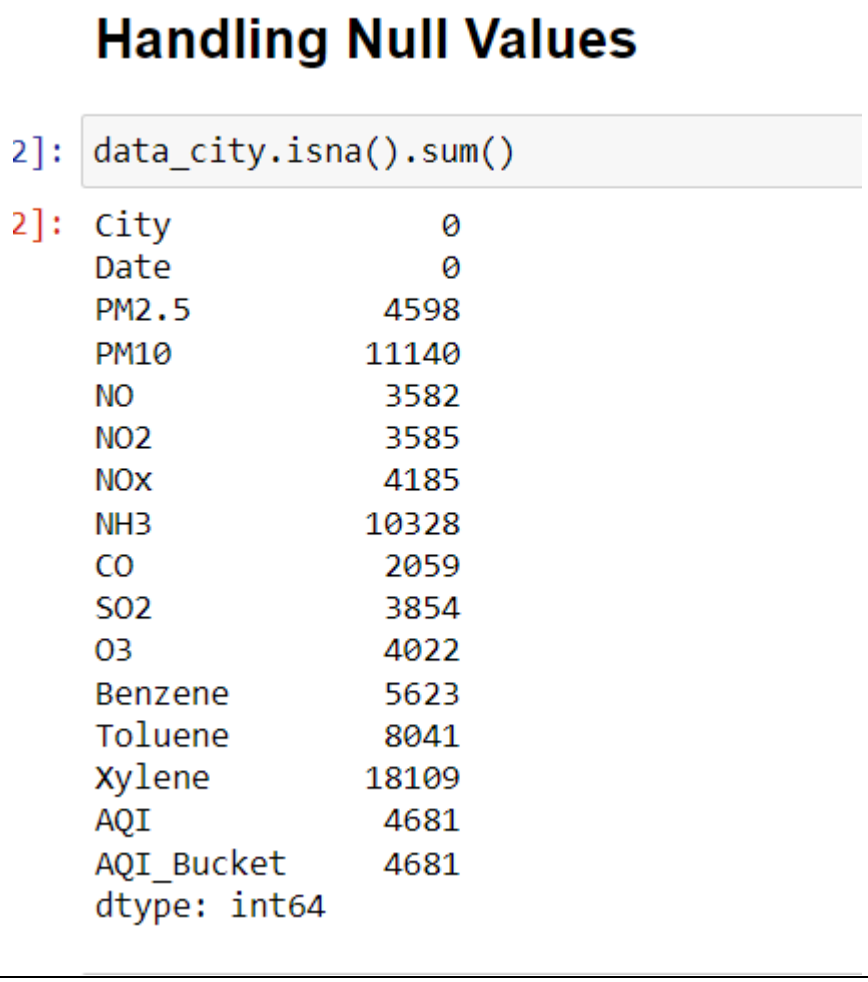
```
In [48]: X=data_city.drop('AQI',axis=1)
         y=data_city['AQI']

In [49]: X
```

Out[49]:

| | City | PM2.5 | PM10 | NO | NO2 | NOx | NH3 | CO | SO2 | C |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 34.515 | 154.750 | 0.92 | 18.22 | 17.15 | 8.975 | 0.92 | 27.640 | 85.63 |
| **1** | 0 | 25.830 | 226.235 | 0.97 | 15.69 | 16.46 | 9.095 | 0.97 | 24.550 | 34.06 |
| **2** | 0 | 36.205 | 72.125 | 17.40 | 19.30 | 29.70 | 6.880 | 2.86 | 29.070 | 30.70 |
| **3** | 0 | 25.830 | 226.235 | 1.70 | 18.48 | 17.97 | 9.085 | 1.70 | 18.590 | 36.08 |
| **4** | 0 | 54.440 | 72.125 | 22.10 | 21.42 | 37.76 | 7.915 | 2.86 | 29.545 | 39.31 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **29526** | 25 | 15.020 | 50.940 | 7.68 | 25.06 | 19.54 | 12.470 | 0.47 | 8.550 | 23.30 |
| **29527** | 25 | 24.380 | 74.090 | 3.42 | 26.06 | 16.53 | 11.990 | 0.52 | 12.720 | 30.14 |
| **29528** | 25 | 22.910 | 65.730 | 3.45 | 29.53 | 18.33 | 10.710 | 0.48 | 8.420 | 30.96 |
| **29529** | 25 | 16.640 | 49.970 | 4.05 | 29.26 | 18.80 | 10.030 | 0.52 | 9.840 | 28.30 |
| **29530** | 25 | 15.000 | 66.000 | 0.40 | 26.85 | 14.05 | 5.200 | 0.59 | 2.100 | 17.05 |

29531 rows × 15 columns

| | |
|---|---|
| Handling Missing Data | Code for identifying and handling missing values.<br>  |
| Data Transformation | Code for transforming variables (scaling, normalization). |
| Feature Engineering | Code for creating new features or modifying existing ones. |

In [48]:
```
X=data_city.drop('AQI',axis=1)
y=data_city['AQI']
```

In [49]:
```
X
```

Out[49]:

| | City | PM2.5 | PM10 | NO | NO2 | NOx | NH3 | CO | SO2 | O3 | Benzene | Toluene | Xylene | Year | Month |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 34.515 | 154.750 | 0.92 | 18.22 | 17.15 | 8.975 | 0.92 | 27.640 | 85.635 | 0.00 | 0.02 | 0.00 | 2015.0 | 1.0 |
| **1** | 0 | 25.830 | 226.235 | 0.97 | 15.69 | 16.46 | 9.095 | 0.97 | 24.550 | 34.060 | 3.68 | 5.50 | 3.77 | 2015.0 | 1.0 |
| **2** | 0 | 36.205 | 72.125 | 17.40 | 19.30 | 29.70 | 6.880 | 2.86 | 29.070 | 30.700 | 6.80 | 16.40 | 2.25 | 2015.0 | 1.0 |
| **3** | 0 | 25.830 | 226.235 | 1.70 | 18.48 | 17.97 | 9.085 | 1.70 | 18.590 | 36.080 | 4.43 | 10.14 | 1.00 | 2015.0 | 1.0 |
| **4** | 0 | 54.440 | 72.125 | 22.10 | 21.42 | 37.76 | 7.915 | 2.86 | 29.545 | 39.310 | 7.01 | 18.89 | 2.78 | 2015.0 | 1.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **29526** | 25 | 15.020 | 50.940 | 7.68 | 25.06 | 19.54 | 12.470 | 0.47 | 8.550 | 23.300 | 2.24 | 12.07 | 0.73 | 2020.0 | 6.0 |
| **29527** | 25 | 24.380 | 74.090 | 3.42 | 26.06 | 16.53 | 11.990 | 0.52 | 12.720 | 30.140 | 0.74 | 2.21 | 0.38 | 2020.0 | 6.0 |
| **29528** | 25 | 22.910 | 65.730 | 3.45 | 29.53 | 18.33 | 10.710 | 0.48 | 8.420 | 30.960 | 0.01 | 0.01 | 0.00 | 2020.0 | 6.0 |
| **29529** | 25 | 16.640 | 49.970 | 4.05 | 29.26 | 18.80 | 10.030 | 0.52 | 9.840 | 28.300 | 0.00 | 0.00 | 0.00 | 2020.0 | 6.0 |
| **29530** | 25 | 15.000 | 66.000 | 0.40 | 26.85 | 14.05 | 5.200 | 0.59 | 2.100 | 17.050 | 0.00 | 0.00 | 0.00 | 2020.0 | 7.0 |

29531 rows × 15 columns