Name: Abhinav Swaminathan Class: D15C Roll No: 01

Experiment 1

Aim: Introduction to Data science and Data preparation using Pandas steps.

Theory:

Data science is the study of data that helps us derive useful insight for business decision making. Data Science is all about using tools, techniques, and creativity to uncover insights hidden within data. It combines math, computer science, and domain expertise to tackle real-world challenges in a variety of fields.

<u>Data science involves these key steps:</u>

- **Data Collection:** Gathering raw data from various sources, such as databases, sensors, or user interactions.
- **Data Cleaning:** Ensuring the data is accurate, complete, and ready for analysis.
- **Data Analysis:** Applying statistical and computational methods to identify patterns, trends, or relationships.
- **Data Visualization:** Creating charts, graphs, and dashboards to present findings clearly.
- **Decision-Making:** Using insights to inform strategies, create solutions, or predict outcomes.

Dataset Overview:

The dataset consists of air pollution readings across various cities in India over the last five years. Below are the key attributes:

- **City:** The city where pollution data was recorded.
- **Date:** The timestamp of the measurement.
- **PM2.5 & PM10:** Particulate matter concentration. (The numeric figure represents the diameter in micro-meter)
- **NO, NO2, NOx, NH3:** nitrogen-based pollutants.
- **CO, SO2, O3:** Harmful environmental pollutants.
- **Benzene, Toluene, Xylene:** Hazardous air pollutants, usually generated by industries and power plants.
- **AQI:** Air Quality Index representing overall pollution level.

- **AQI_Bucket:** Categorized pollution levels (Good, Moderate, Poor, etc.).

Problem Statement:

The objective is to analyze air pollution trends across Indian cities and identify key pollutants affecting air quality. Since the dataset provides information over cities of the past 5 years, we can use this information to predict air quality of a particular region in the future.

- Understanding variations in AQI across cities and time periods.
- Identifying major pollutants contributing to poor air quality.
- Visualizing trends, drawing meaningful conclusions and attempting future analysis from the dataset.

Code:

Loading the Dataset

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats

# Load dataset
df = pd.read_csv('Data.csv')
```

Basic Dataset Information

df.shape(): This returns a tuple indicating the number of rows and columns in the DataFrame.

df.info(): This prints "dataset info" and displays the DataFrame's structure including data types and non-null counts.

df.describe(): This prints "dataset description" and shows summary statistics like mean, standard deviation, and percentiles for numerical columns

```
print("Dataset Shape:", df.shape)
print("\nDataset Info:")
df.info()
print("\nDataset Description:")
print(df.describe())
```

```
Dataset Shape: (29531, 16)
```

.

```
Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29531 entries, 0 to 29530
Data columns (total 16 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   City        29531 non-null  object
 1   Date        29531 non-null  object
 2   PM2.5       24933 non-null  float64
 3   PM10        18391 non-null  float64
 4   NO          25949 non-null  float64
 5   NO2         25946 non-null  float64
 6   NOx         25346 non-null  float64
 7   NH3         19203 non-null  float64
 8   CO          27472 non-null  float64
 9   SO2         25677 non-null  float64
 10  O3          25509 non-null  float64
 11  Benzene     23908 non-null  float64
 12  Toluene     21490 non-null  float64
 13  Xylene      11422 non-null  float64
 14  AQI         24850 non-null  float64
 15  AQI_Bucket  24850 non-null  object
dtypes: float64(13), object(3)
memory usage: 3.6+ MB
```

```
Dataset Description:
              PM2.5          PM10            NO           NO2           NOx  \
count  24933.000000  18391.000000  25949.000000  25946.000000  25346.000000
mean      67.450578    118.127103     17.574730     28.560659     32.309123
std       64.661449     90.605110     22.785846     24.474746     31.646011
min        0.040000      0.010000      0.020000      0.010000      0.000000
25%       28.820000     56.255000      5.630000     11.750000     12.820000
50%       48.570000     95.680000      9.890000     21.690000     23.520000
75%       80.590000    149.745000     19.950000     37.620000     40.127500
max      949.990000   1000.000000    390.680000    362.210000    467.630000

                NH3            CO           SO2            O3       Benzene  \
count  19203.000000  27472.000000  25677.000000  25509.000000  23908.000000
mean      23.483476      2.248598     14.531977     34.491430      3.280840
std       25.684275      6.962884     18.133775     21.694928     15.811136
min        0.010000      0.000000      0.010000      0.010000      0.000000
25%        8.580000      0.510000      5.670000     18.860000      0.120000
50%       15.850000      0.890000      9.160000     30.840000      1.070000
75%       30.020000      1.450000     15.220000     45.570000      3.080000
max      352.890000    175.810000    193.860000    257.730000    455.030000

            Toluene        Xylene           AQI
count  21490.000000  11422.000000  24850.000000
mean       8.700972      3.070128    166.463581
std       19.969164      6.323247    140.696585
min        0.000000      0.000000     13.000000
25%        0.600000      0.140000     81.000000
50%        2.970000      0.980000    118.000000
75%        9.150000      3.350000    208.000000
max      454.850000    170.370000   2049.000000
```

Removing Duplicate Entries

```
df = df.drop_duplicates()
```

Creating Dummy Variables (One-Hot Encoding) for AQI Bucket:

This creates dummy data out of “AQI Bucket” for the various severity levels of pollution. This helps to convert categorical data to numerical data and helps in analysis in the algorithm

We use **df.head()** to verify this.

```
[11] df = pd.get_dummies(df, columns=['AQI_Bucket'], drop_first=True)
```

```
print(df.head(10))
```

```
           City        Date  PM2.5    PM10    NO    NO2    NOx    NH3    CO  \
2123  Amaravati  25-11-2017  81.40  124.50  1.44  20.50  12.08  10.72  0.12
2124  Amaravati  26-11-2017  78.32  129.06  1.26  26.00  14.85  10.28  0.14
2125  Amaravati  27-11-2017  88.76  135.32  6.60  30.85  21.77  12.91  0.11
2126  Amaravati  28-11-2017  64.18  104.09  2.56  28.07  17.01  11.42  0.09
2127  Amaravati  29-11-2017  72.47  114.84  5.23  23.20  16.59  12.25  0.16
2128  Amaravati  30-11-2017  69.80  114.86  4.69  20.17  14.54  10.95  0.12
2129  Amaravati  01-12-2017  73.96  113.56  4.58  19.29  13.97  10.95  0.10
2130  Amaravati  02-12-2017  89.90  140.20  7.71  26.19  19.87  13.12  0.10
2131  Amaravati  03-12-2017  87.14  130.52  0.97  21.31  12.12  14.36  0.15
2132  Amaravati  04-12-2017  84.64  125.00  4.02  26.98  17.58  14.41  0.18

        SO2      O3  Benzene  Toluene  Xylene    AQI  AQI_Bucket_Moderate  \
2123  15.24  127.09     0.20     6.50    0.06  184.0                 True
2124  26.96  117.44     0.22     7.95    0.08  197.0                 True
2125  33.59  111.81     0.29     7.63    0.12  198.0                 True
2126  19.00  138.18     0.17     5.02    0.07  188.0                 True
2127  10.55  109.74     0.21     4.71    0.08  173.0                 True
2128  14.07  118.09     0.16     3.52    0.06  165.0                 True
2129  13.90  123.80     0.17     2.85    0.04  191.0                 True
2130  19.37  128.73     0.25     2.79    0.07  191.0                 True
2131  11.41  114.80     0.23     3.82    0.04  227.0                False
2132   9.84  112.41     0.31     3.53    0.09  168.0                 True

      AQI_Bucket_Poor  AQI_Bucket_Satisfactory  AQI_Bucket_Severe  \
2123            False                    False              False
2124            False                    False              False
2125            False                    False              False
2126            False                    False              False
2127            False                    False              False
2128            False                    False              False
2129            False                    False              False
2130            False                    False              False
2131             True                    False              False
2132            False                    False              False

      AQI_Bucket_Very Poor
2123                 False
2124                 False
2125                 False
2126                 False
2127                 False
```

Identifying Outliers manually using the Standardization Approach (Z-Score Method)

To identify outliers manually we use the standardization approach (z score method). We find mean and standard deviation of the vehicle weight and calculate its z score; if it's less than -3 or greater than 3 means it's an outlier.

```
#By Z-score method
mean_aqi = df['AQI'].mean()
std_aqi = df['AQI'].std()

print (f"Mean of AQI: {mean_aqi}")
print (f"Standard Deviation of AQI: {std_aqi}")

df['Z_Score'] = (df['AQI'] - mean_aqi) / std_aqi
print(df[['AQI', 'Z_Score']])

# Identify outliers based on the Z-score
outliers =df[df['Z_Score'].abs() > 3]
print (outliers)
```

```
Mean of AQI: 138.48802144412798
Standard Deviation of AQI: 91.64490404411067
         AQI   Z_Score
2123   184.0  0.496612
2124   197.0  0.638464
2125   198.0  0.649376
2126   188.0  0.540259
2127   173.0  0.376584
...      ...       ...
29523   86.0 -0.572733
29524   77.0 -0.670938
29525   47.0 -0.998288
29526   41.0 -1.063758
29527   70.0 -0.747319

[5969 rows x 2 columns]
```

```
       AQI_Bucket_Moderate  AQI_Bucket_Poor  AQI_Bucket_Satisfactory  \
3308                 False            False                    False
4265                 False            False                    False
10229                False            False                    False
10230                False            False                    False
10521                False            False                    False
...                    ...              ...                      ...
14880                False            False                    False
14881                False            False                    False
14994                False            False                    False
14995                False            False                    False
25531                False            False                    False

       AQI_Bucket_Severe  AQI_Bucket_Very Poor   Z_Score
3308                True                 False  3.540971
4265                True                 False  3.704647
10229               True                 False  3.639176
10230               True                 False  3.442766
10521               True                 False  3.159062
...                  ...                   ...       ...
14880               True                 False  3.802852
14881               True                 False  3.966527
14994               True                 False  3.311826
14995               True                 False  4.053820
25531               True                 False  3.388208
```

Normalizing AQI using Min-Max Scaling

We normalize the data across the AQI on a scale of 0 to 1.

```
[25] min_aqi = df['AQI'].min()
     max_aqi = df['AQI'].max()
     df['AQI_normalized'] = (df['AQI'] - min_aqi) / (max_aqi - min_aqi)
```

```
print(df [['AQI', 'AQI_normalized']])
```

```
         AQI  AQI_normalized
2123   184.0        0.246177
2124   197.0        0.266055
2125   198.0        0.267584
2126   188.0        0.252294
2127   173.0        0.229358
...      ...             ...
29523   86.0        0.096330
29524   77.0        0.082569
29525   47.0        0.036697
29526   41.0        0.027523
29527   70.0        0.071865

[5969 rows x 2 columns]
```

Conclusion

This experiment focused on preparing and analyzing air pollution data in India by addressing common data quality issues. Missing values were managed through replacement and removal techniques, while duplicate entries were eliminated to ensure data integrity. Outliers in AQI values were identified using the Z-score method, and Min-Max scaling was applied to normalize the data for better comparison. These preprocessing steps helped create a more structured and reliable dataset, which will allow for meaningful analysis of pollution trends.