

# *AQI Analysis*

## Index

---

<u>S. No</u>	<u>Topic</u>	<u>Page No.</u>
1.	Certificate	
2.	Acknowledgement	
3.	About Project and Software used	
4.	AQI Analysis (Introduction)	
5.	Python Overview	
6.	Source Code	
7.	Output	
8.	Bibliography	

AIM:

To Develop: *AQI Analysis*

Front End:

Python 3.11

Back End:

CSV Files

Operating System:

Windows 11

Minimum Hardware and Software Requirements:

Operating System:

x86 64-bit CPU (Intel i7 7700)

8 GB RAM.

76 GB free disk space.

Open Source Software being used:

1. Python 3.9
  - a. Pandas
  - b. Matplotlib

## Pandas:

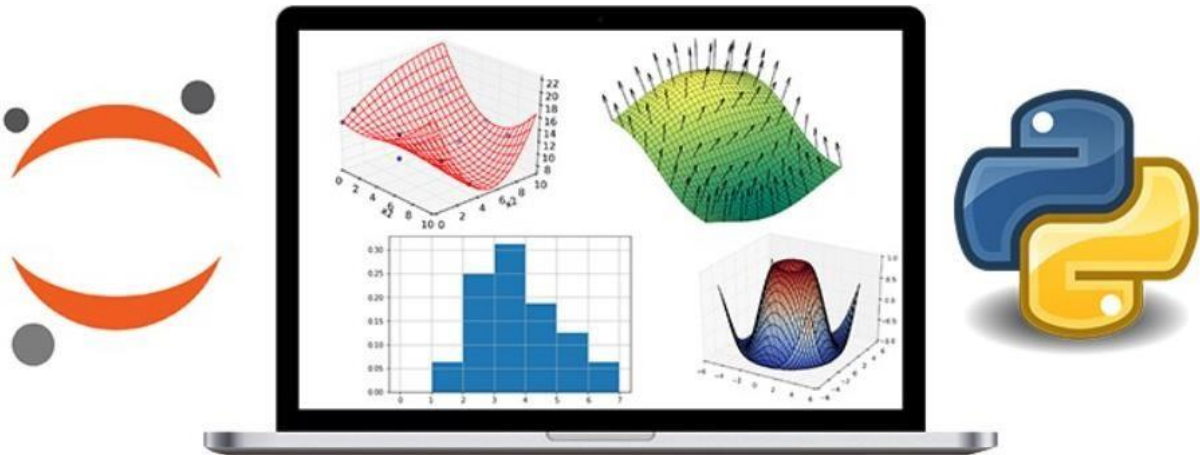


Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the **NumPy library**. Pandas is fast and it has high performance & productivity for users. It is developed by **Wes McKinney**. It is built on the Numpy package and its key data structure is called the **DataFrame**. DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables.

### Advantages :

- ✓ Fast and efficient for manipulating and analyzing data.
- ✓ Data from different file objects can be loaded.
- ✓ Easy handling of missing data (represented as NaN) in floating point as well as non-floating point data
- ✓ Size mutability: columns can be inserted and deleted from DataFrame and higher dimensional objects

# Matplotlib:



Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

- ✓ Create publication quality plots.
- ✓ Make interactive figures that can zoom, pan, update.
- ✓ Customize visual style and layout.
- ✓ Export to many file formats.
- ✓ Embed in JupyterLab and Graphical User Interfaces.
- ✓ Use a rich array of third-party packages built on Matplotlib.

## AQI Analysis



### Introduction

The Air Quality Index (AQI) is a numerical scale used worldwide to assess and communicate the quality of air in a specific region. It measures how clean or polluted the air is and what associated health effects might be of concern for people. The AQI primarily tracks the levels of major pollutants in the atmosphere, such as particulate matter (PM<sub>2.5</sub> and PM<sub>10</sub>), ozone (O<sub>3</sub>), nitrogen dioxide (NO<sub>2</sub>), sulfur dioxide (SO<sub>2</sub>), carbon monoxide (CO), and other harmful substances.

Each pollutant is assigned a score on the AQI scale, typically ranging from 0 to 500, where:

0-50: Good (Air quality is considered satisfactory, and air pollution poses little or no risk.)

51-100: Moderate (Air quality is acceptable, but some pollutants may be

a concern for a very small number of people sensitive to air pollution.)

101-150: Unhealthy for Sensitive Groups (Members of sensitive groups may experience health effects, but the general public is less likely to be affected.)

151-200: Unhealthy (Everyone may begin to experience health effects; members of sensitive groups may experience more serious effects.)

201-300: Very Unhealthy (Health alert: everyone may experience more serious health effects.)

301-500: Hazardous (Health warning of emergency conditions: the entire population is more likely to be affected.)

In this project, we will analyze AQI data for various cities to understand pollution trends, examine seasonal variations, and visualize the impact of different pollutants. The analysis will help identify key factors affecting air quality and how it varies across different regions and times of the year.

## **ABOUT THE PROJECT:**

### **Menu System:**

As soon as the project is executed, it loads a Main Menu, where there are 4 options as under:

#### **MAIN MENU**

- 1- Data Analysis
- 2- Data Manipulation
- 3- Data Visualization
- 4- Exit

## 1- Data Analysis

1. Read Complete CSV File
2. Display Data in DataFrame Format
3. Display Top Records
4. Display Bottom Records
5. Display Unique Cities
6. Filter by City
7. Filter by Season
8. Calculate Average AQI by City
9. Sort by AQI
10. Display Summary Statistics
11. Display Correlation Matrix
12. Find Maximum AQISeason wise chart

## 2- Manipulation

1. Add a Column
2. Add a Row
3. Export Filtered Data to

## 3- Visualization

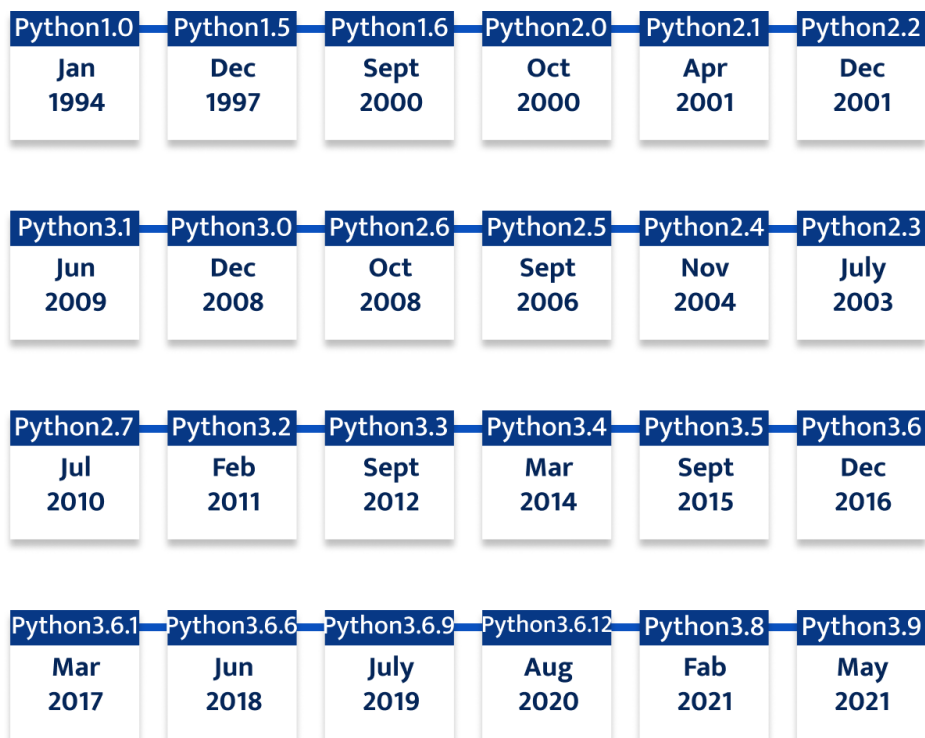
1. Visualize AQI Distribution
2. Seasonal AQI Trend Line
3. City-Wise AQI Comparison
4. AQI Category Pie Chart
5. Heatmap of AQI by Month and City

## Python Introduction

### Python :

- ✓ Python is an interpreted, object-oriented, an open-source, high-level programming language.
- ✓ It is developed by Guido Van Rossum in 1991 at the National Research Institute of Mathematics, Netherlands.
- ✓ It is general purpose programming language that can be used effectively to build any kind of program.
- ✓ It is considered to be high versatile programming language because it supports multiple models of programming, such as: OOP, Functional, Imperative and Procedural.
- ✓ There are a vast number of libraries and packages available in Python that developers can freely use to create feature-rich tools and software for various applications.

### Versions of Python:





## Advantages of Python :

### ✓ Easy to code :

Python is very easy to learn the language as compared to other languages like C, C++, C#, Java, etc. Python is a very developer-friendly language which means that anyone and everyone can learn to code it in a couple of hours or days.

### ✓ Free and Open Source :

Python is freely available on its official website [www.python.org](http://www.python.org). The open-source means, "Anyone can download its source code without paying any money." If he/she wants then change the code and contribute updated code towards Python Community.

### ✓ Highly Portable:

Highly portable means "Cross Platform Language" i.e. "Write once, run anywhere". For example, if we have python code for windows and want to run same code on different platforms such as Linux, Unix, and Mac then we do not need to change it, we can run this code on any platform.

### ✓ Large Standard Library :

Python comes inbuilt with a large number of libraries that can be imported at any instance and be used in a specific program. The presence of libraries also makes sure that you do not have to write your own code for every single thing, just import the library and use the existing module and functions. Some of the common libraries are Pandas, Numpy, Django, Flask, Kivy etc.

### ✓ GUI Programming :

Graphical User Interface (GUI) is one of the important feature of Python, because it has the ability to add flair to code and make the results more visual. Python has support for a wide range of modules like PyQt5, PyQt4, wxPython, or Tk for creating GUI applications. With the help of these modules you can easily create any desktop application.

### ✓ Dynamically Typed :

In Python, we don't need to specify the data-type of the variable. When we assign some value to the variable, it automatically allocates the memory to the variable at run time. Even you can change the value of that variable with another data type. For example, a=15 (a store integer value), on next line you will write like a="Study" (now a store string value).

## WHAT CAN YOU DO WITH PYTHON :



## WHO USES PYTHON :



## WHERE CAN I CODE PYTHON :



**IDLE**



**Spyder**



**Pycharm**



**eric**



**Vin**



**PyDev**



**Atom**



**Jupyter**



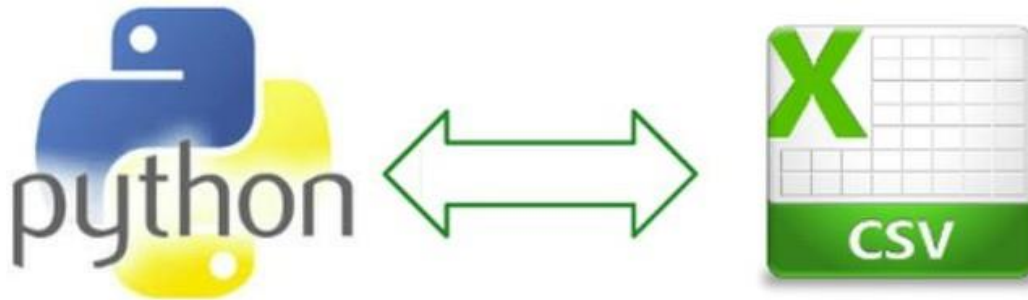
**Anaconda**



**Thonny**



## CSV Files



CSV (Comma Separated Values) is a simple file format used to store tabular data, such as a spreadsheet or database. A CSV file stores tabular data (numbers and text) in plain text. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format.

For working CSV files in Python, there is an inbuilt module called csv.

# Comma Separated Values (C.S.V) File

## AQI\_Analysis\_Seasonal\_Trend.csv (Excel View):

	A	B	C	D	E	F	G	H	I	J	K
1	City	Season	Month	AQI	Main_Pollutant	Temperature_C	Humidity_%	Wind_Speed_kph	AQI_Category		
2	Delhi	Spring	3	119	NO2	-4.1	45	3.3	Unhealthy		
3	Faridabad	Spring	4	388	O3	-9.5	14	26.5	Hazardous		
4	Noida	Spring	12	72	PM2.5	37.9	55	25.4	Moderate		
5	Gurugram	Winter	3	478	SO2	20.7	81	19.2	Hazardous		
6	Noida	Summer	9	357	SO2	26.3	75	17.4	Hazardous		
7	Faridabad	Summer	8	83	SO2	31.5	86	12.4	Moderate		
8	Faridabad	Winter	3	37	PM2.5	34.8	48	8.1	Good		
9	Noida	Winter	1	278	O3	27	90	13.3	Hazardous		
10	Sonipat	Fall	1	153	PM2.5	-3.2	61	28.8	Very Unhealthy		
11	Ghaziabad	Spring	3	113	PM2.5	-6.8	33	14.5	Unhealthy		
12	Noida	Winter	11	163	PM10	3.7	66	22.4	Very Unhealthy		
13	Najafgarh	Summer	4	103	O3	19.9	21	3.7	Unhealthy		
14	Najafgarh	Spring	1	240	NO2	26.6	84	13.1	Hazardous		
15	Gurugram	Spring	5	431	PM10	14.3	79	8.4	Hazardous		
16	Hisaar	Summer	7	346	SO2	15.1	85	16.7	Hazardous		
17	Noida	Fall	12	169	SO2	28.7	45	3.3	Very Unhealthy		
18	Ghaziabad	Summer	9	147	NO2	28.3	71	18.2	Unhealthy		
19	Noida	Fall	4	402	O3	23.9	57	22.8	Hazardous		
20	Delhi	Fall	10	156	PM2.5	34.1	42	3.6	Very Unhealthy		
21	Gurugram	Fall	7	487	CO	35.8	60	9.7	Hazardous		
22	Gurugram	Spring	9	296	CO	2.3	11	6.3	Hazardous		
23	Hisaar	Fall	9	89	NO2	3.2	90	13.8	Moderate		
24	Hisaar	Fall	8	370	NO2	2.7	61	16.9	Hazardous		

## AQI\_Analysis\_Seasonal\_Trend.csv (Notepad View):

```

City,Season,Month,AQI,Main_Pollutant,Temperature_C,Humidity_%,Wind_Speed_kph,AQI_Category
Delhi,Spring,3,119,NO2,-4.1,45,3.3,Unhealthy
Faridabad,Spring,4,388,O3,-9.5,14,26.5,Hazardous
Noida,Spring,12,72,PM2.5,37.9,55,25.4,Moderate
Gurugram,Winter,3,478,SO2,20.7,81,19.2,Hazardous
Noida,Summer,9,357,SO2,26.3,75,17.4,Hazardous
Faridabad,Summer,8,83,SO2,31.5,86,12.4,Moderate
Faridabad,Winter,3,37,PM2.5,34.8,48,8.1,Good
Noida,Winter,1,278,O3,27,90,13.3,Hazardous
Sonipat,Fall,1,153,PM2.5,-3.2,61,28.8,Very Unhealthy
Ghaziabad,Spring,3,113,PM2.5,-6.8,33,14.5,Unhealthy
Noida,Winter,11,163,PM10,3.7,66,22.4,Very Unhealthy
Najafgarh,Summer,4,103,O3,19.9,21,3.7,Unhealthy
Najafgarh,Spring,1,240,NO2,26.6,84,13.1,Hazardous
Gurugram,Spring,5,431,PM10,14.3,79,8.4,Hazardous
Hisaar,Summer,7,346,SO2,15.1,85,16.7,Hazardous
Hisaar,Fall,8,370,NO2,2.7,61,16.9,Hazardous
  
```

Ln 1, Col 1    4,758 characters    100%    Windows (CRLF)    UTF-8

## **CODING**

```
AQI-Analysis.py - C:/Users/verma/OneDrive/Documents/AQI-Analysis.py (3.11.0)
File Edit Format Run Options Window Help

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the data
df = pd.read_csv('AQI_Analysis_Seasonal_Trends.csv')

# Menu
while True:
    print("\n--- AQI Data Analysis Menu ---")
    print("1. Data Analysis:")
    print("2. Data Manipulation:")
    print("3. Data Visualization:")
    print("4. Exit:")
    choice=input("Enter your choice: ")
    if choice == '1':
        print("1. Read Complete CSV File")
        print("2. Display Data in DataFrame Format")
        print("3. Display Top Records")
        print("4. Display Bottom Records")
        print("5. Display Unique Cities")
        print("6. Filter by City")
        print("7. Filter by Season")
        print("8. Calculate Average AQI by City")
        print("9. Sort by AQI")
        print("10. Display Summary Statistics")
        print("11. Display Correlation Matrix")
        print("12. Find Maximum AQI")
        da=input("Enter option : ")
        if da == '1':
            print(df)
        elif da == '2':
            print(df.head())
        elif da == '3':
            print(df.head(int(input("Enter number of top records to display: "))))
        elif da == '4':
            print(df.tail(int(input("Enter number of bottom records to display: "))))
```

```

elif da == '5':
    print(df['City'].unique())
elif da == '6':
    city = input("Enter city name: ")
    print(df[df['City'] == city])
elif da == '7':
    season = input("Enter season: ")
    print(df[df['Season'] == season])
elif da == '8':
    print(df.groupby('City')['AQI'].mean())
elif da == '9':
    order = input("Enter 'asc' for ascending or 'desc' for descending: ")
    print(df.sort_values('AQI', ascending=(order == 'asc')))
elif da == '10':
    print(df.describe())
elif da == '11':
    correlation = df.corr()
    sns.heatmap(correlation, annot=True, cmap='coolwarm')
    plt.title('Correlation Matrix')
    plt.show()
elif da == '12':
    max_aqi = df['AQI'].max()
    max_record = df[df['AQI'] == max_aqi]
    print("Maximum AQI Record:\n", max_record)
if choice == '2':
    print("1. Add a Column")
    print("2. Add a Row")
    print("3. Export Filtered Data to CSV")
    dm=input("Enter option : ")
    if dm == '1':
        df['New_Column'] = " "
        print("New column added.")
    elif dm == '2':
        new_row = {'City': 'Example City', 'Season': 'Winter', 'Month': 1, 'AQI': 50,
                    'Main_Pollutant': 'PM2.5', 'Temperature_C': 10, 'Humidity_%': 50,
                    'Wind_Speed_kph': 10, 'AQI_Category': 'Good'}
        df.loc[len(df)] = new_row
        print("New row added.")

```

```

elif dm == '3':
    city_name = input("Enter city name for filtered data: ")
    filtered_df = df[df['City'] == city_name]
    filtered_df.to_csv(f'{city_name}_AQI_data.csv', index=False)
    print(f"Filtered data saved to {city_name}_AQI_data.csv")
if choice == '3':
    print("1. Visualize AQI Distribution")
    print("2. Seasonal AQI Trend Line")
    print("3. City-Wise AQI Comparison")
    print("4. AQI Category Pie Chart")
    print("5. Heatmap of AQI by Month and City")
    dv=input("Enter option : ")
    if dv == '1':
        plt.hist(df['AQI'], bins=30, color='skyblue')
        plt.xlabel('AQI')
        plt.ylabel('Frequency')
        plt.title('AQI Distribution')
        plt.show()
    elif dv == '2':
        df.groupby('Season')['AQI'].mean().plot(kind='line', marker='o', color='purple')
        plt.xlabel('Season')
        plt.ylabel('Average AQI')
        plt.title('Seasonal AQI Trend')
        plt.show()
    elif dv == '3':
        df.groupby('City')['AQI'].mean().plot(kind='bar', color='orange')
        plt.xlabel('City')
        plt.ylabel('Average AQI')
        plt.title('City-Wise AQI Comparison')
        plt.show()
    elif dv == '4':
        df['AQI_Category'].value_counts().plot(kind='pie', autopct='%1.1f%%', colors=['green', 'yellow', 'red'])
        plt.title('AQI Categories')
        plt.show()
    elif dv == '5':
        pivot_table = df.pivot_table(values='AQI', index='Month', columns='City')
        sns.heatmap(pivot_table, cmap='coolwarm', annot=True)
        plt.title('Heatmap of AQI by Month and City')
        plt.show()

if choice == '4':
    break
else:
    print("Invalid choice. Please enter a number between 1 and 21.")

```



## *Screenshots /Output*

--- AQI Data Analysis Menu ---

1. Data Analysis:
2. Data Manipulation:
3. Data Visualization:
4. Exit:

Enter your choice: 1

1. Read Complete CSV File
2. Display Data in DataFrame Format
3. Display Top Records
4. Display Bottom Records
5. Display Unique Cities
6. Filter by City
7. Filter by Season
8. Calculate Average AQI by City
9. Sort by AQI
10. Display Summary Statistics
11. Display Correlation Matrix
12. Find Maximum AQI

Enter option :

Enter option : 2

	City	Season	Month	...	Humidity_%	Wind_Speed_kph	AQI_Category
0	Delhi	Spring	3	...	45	3.3	Unhealthy
1	Faridabad	Spring	4	...	14	26.5	Hazardous
2	Noida	Spring	12	...	55	25.4	Moderate
3	Gurugram	Winter	3	...	81	19.2	Hazardous
4	Noida	Summer	9	...	75	17.4	Hazardous

[5 rows x 9 columns]

To Display top records :

1. Read Complete CSV File
2. Display Data in DataFrame Format
3. Display Top Records
4. Display Bottom Records
5. Display Unique Cities
6. Filter by City
7. Filter by Season
8. Calculate Average AQI by City
9. Sort by AQI
10. Display Summary Statistics
11. Display Correlation Matrix
12. Find Maximum AQI

Enter option : 3

Enter number of top records to display: 7

	City	Season	Month	...	Humidity_%	Wind_Speed_kph	AQI_Category
0	Delhi	Spring	3	...	45	3.3	Unhealthy
1	Faridabad	Spring	4	...	14	26.5	Hazardous
2	Noida	Spring	12	...	55	25.4	Moderate
3	Gurugram	Winter	3	...	81	19.2	Hazardous
4	Noida	Summer	9	...	75	17.4	Hazardous
5	Faridabad	Summer	8	...	86	12.4	Moderate
6	Faridabad	Winter	3	...	48	8.1	Good

[7 rows x 9 columns]

To Display bottom records

Enter your choice: 1

1. Read Complete CSV File
2. Display Data in DataFrame Format
3. Display Top Records
4. Display Bottom Records
5. Display Unique Cities
6. Filter by City
7. Filter by Season
8. Calculate Average AQI by City
9. Sort by AQI
10. Display Summary Statistics
11. Display Correlation Matrix
12. Find Maximum AQI

Enter option : 4

Enter number of bottom records to display: 3

	City	Season	Month	...	Humidity_%	Wind_Speed_kph	AQI_Category
97	Noida	Winter	12	...	13	12.2	Hazardous
98	Delhi	Fall	1	...	47	9.3	Hazardous
99	Sonipat	Winter	4	...	88	1.9	Hazardous

[3 rows x 9 columns]

Invalid choice. Please enter a number between 1 and 21.

To display all cities :

1. Read Complete CSV File
2. Display Data in DataFrame Format
3. Display Top Records
4. Display Bottom Records
5. Display Unique Cities
6. Filter by City
7. Filter by Season
8. Calculate Average AQI by City
9. Sort by AQI
10. Display Summary Statistics
11. Display Correlation Matrix
12. Find Maximum AQI

Enter option : 5

['Delhi' 'Faridabad' 'Noida' 'Gurugram' 'Sonipat' 'Ghaziabad' 'Najafgarh'  
'Hisaar' 'Mathura' 'Agra']

## Display Data by City

1. Read Complete CSV File
2. Display Data in DataFrame Format
3. Display Top Records
4. Display Bottom Records
5. Display Unique Cities
6. Filter by City
7. Filter by Season
8. Calculate Average AQI by City
9. Sort by AQI
10. Display Summary Statistics
11. Display Correlation Matrix
12. Find Maximum AQI

Enter option : 6

Enter city name: Delhi

	City	Season	Month	...	Humidity_%	Wind_Speed_kph	AQI_Category
0	Delhi	Spring	3	...	45	3.3	Unhealthy
18	Delhi	Fall	10	...	42	3.6	Very Unhealthy
26	Delhi	Spring	7	...	82	27.8	Unhealthy
28	Delhi	Fall	7	...	43	14.3	Hazardous
35	Delhi	Spring	9	...	40	29.8	Unhealthy
36	Delhi	Summer	3	...	44	20.3	Hazardous
41	Delhi	Summer	12	...	54	26.5	Unhealthy
57	Delhi	Fall	10	...	40	20.3	Moderate
60	Delhi	Fall	7	...	69	26.5	Moderate
78	Delhi	Fall	12	...	29	12.7	Hazardous
98	Delhi	Fall	1	...	47	9.3	Hazardous

Display data by season

1. Read Complete CSV File
2. Display Data in DataFrame Format
3. Display Top Records
4. Display Bottom Records
5. Display Unique Cities
6. Filter by City
7. Filter by Season
8. Calculate Average AQI by City
9. Sort by AQI
10. Display Summary Statistics
11. Display Correlation Matrix
12. Find Maximum AQI

Enter option : 7

Enter season: Winter

	City	Season	Month	...	Humidity_%	Wind_Speed_kph	AQI_Category
3	Gurugram	Winter	3	...	81	19.2	Hazardous
6	Faridabad	Winter	3	...	48	8.1	Good
7	Noida	Winter	1	...	90	13.3	Hazardous
10	Noida	Winter	11	...	66	22.4	Very Unhealthy
24	Hisaar	Winter	4	...	80	4.1	Moderate
29	Agra	Winter	6	...	11	5.7	Very Unhealthy
33	Gaziabad	Winter	2	...	81	18.4	Moderate
34	Sonipat	Winter	10	...	73	2.0	Hazardous
47	Faridabad	Winter	12	...	30	19.2	Unhealthy
48	Najafgarh	Winter	3	...	52	1.2	Hazardous
52	Gaziabad	Winter	10	...	32	4.9	Very Unhealthy
...	...	...	...	...	...	...	...
64	Gurugram	Winter	3	...	84	27.1	Unhealthy
71	Agra	Winter	6	...	44	4.2	Hazardous
77	Agra	Winter	6	...	19	24.3	Hazardous
79	Noida	Winter	10	...	81	28.3	Hazardous
82	Agra	Winter	5	...	87	27.7	Hazardous
83	Najafgarh	Winter	11	...	25	1.2	Hazardous
85	Agra	Winter	10	...	29	29.2	Good
87	Agra	Winter	4	...	81	13.6	Moderate
88	Gurugram	Winter	9	...	70	28.1	Very Unhealthy
94	Hisaar	Winter	8	...	72	9.9	Hazardous
95	Gurugram	Winter	12	...	61	13.8	Very Unhealthy
97	Noida	Winter	12	...	13	12.2	Hazardous
99	Sonipat	Winter	4	...	88	1.9	Hazardous

Calculate Average AQI

1. Read Complete CSV File
2. Display Data in DataFrame Format
3. Display Top Records
4. Display Bottom Records
5. Display Unique Cities
6. Filter by City
7. Filter by Season
8. Calculate Average AQI by City
9. Sort by AQI
10. Display Summary Statistics
11. Display Correlation Matrix
12. Find Maximum AQI

Enter option : 8

City

Agra	210.600000
Delhi	202.454545
Faridabad	216.500000
Gaziabad	156.125000
Gurugram	319.454545
Hisaar	266.500000
Mathura	270.200000
Najafgarh	231.857143
Noida	271.000000
Sonipat	267.777778

Name: AQI, dtype: float64

## Data Manipulation

### --- AQI Data Analysis Menu ---

1. Data Analysis:
2. Data Manipulation:
3. Data Visualization:
4. Exit:

Enter your choice: 2

1. Add a Column
2. Add a Row
3. Export Filtered Data to CSV

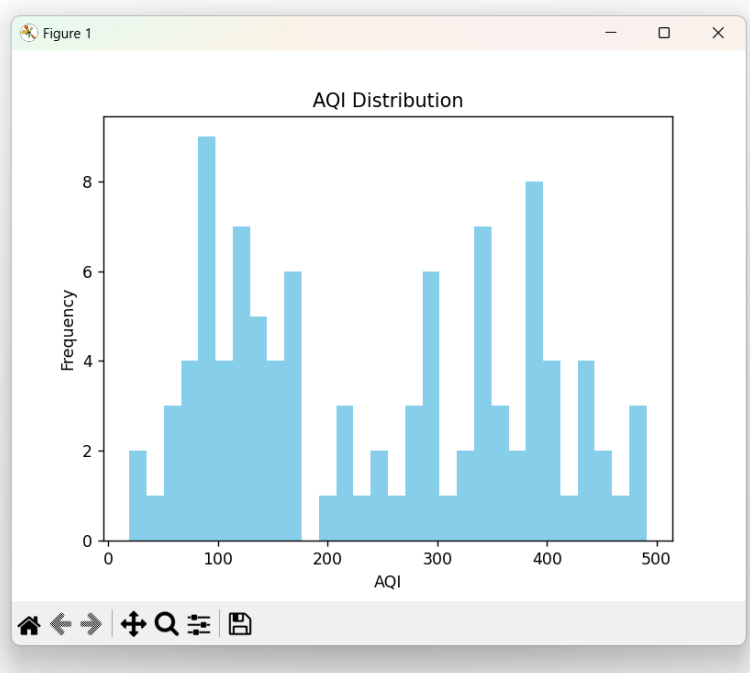
Enter option : 1

New column added.

## AQI Distribution

### --- AQI Data Analysis Menu ---

1. Data Analysis:
  2. Data Manipulation:
  3. Data Visualization:
  4. Exit:
- Enter your choice: 3
1. Visualize AQI Distribution
  2. Seasonal AQI Trend Line
  3. City-Wise AQI Comparison
  4. AQI Category Pie Chart
  5. Heatmap of AQI by Month and City
- Enter option : 1



--- AQI Data Analysis Menu ---

1. Data Analysis:

2. Data Manipulation:

3. Data Visualization:

4. Exit:

Enter your choice: 3

1. Visualize AQI Distribution

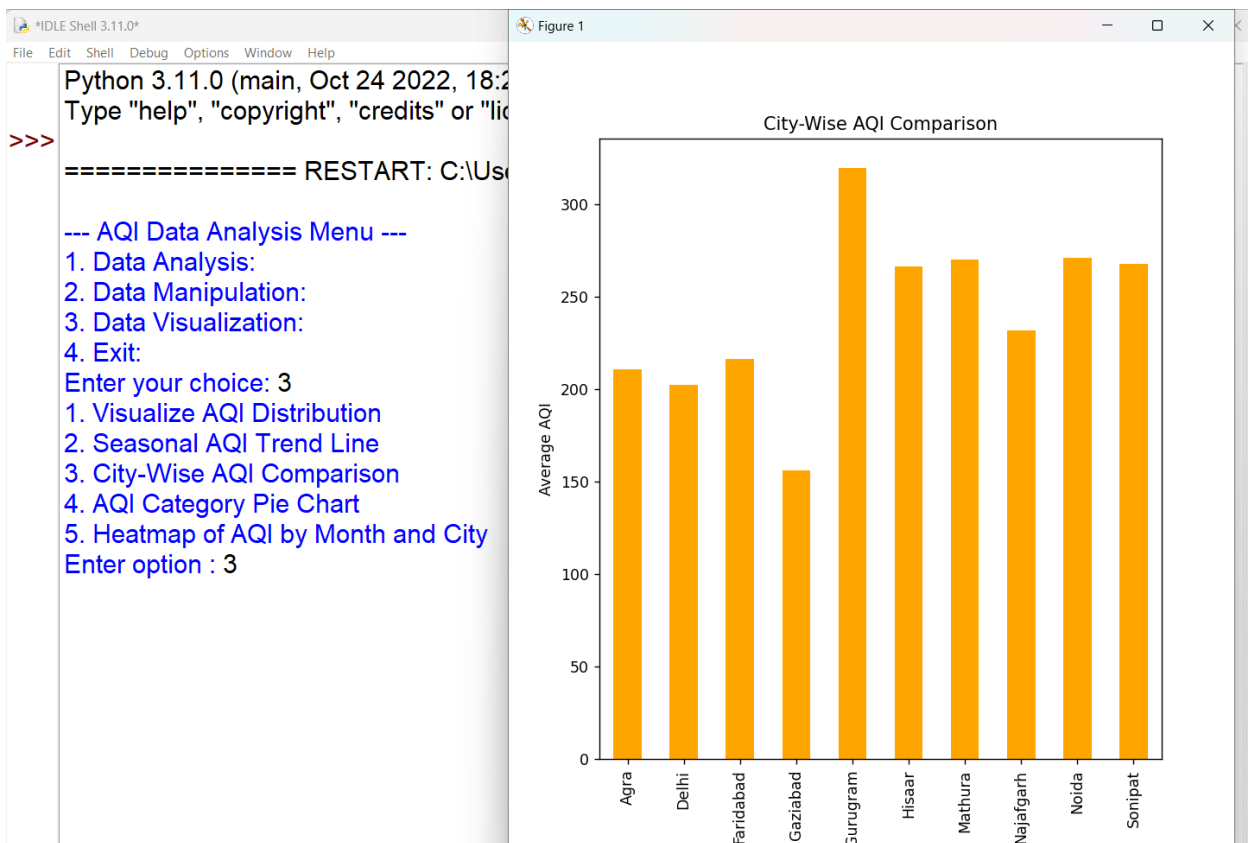
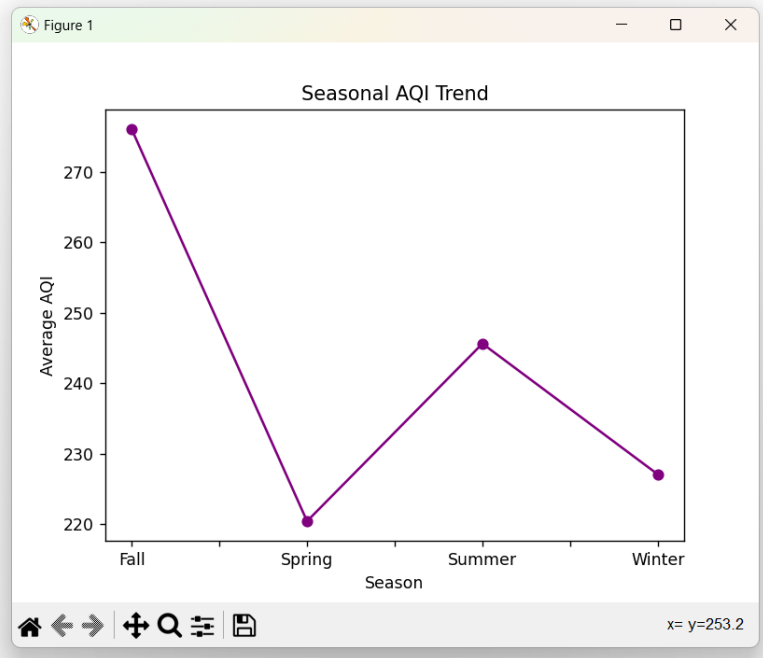
2. Seasonal AQI Trend Line

3. City-Wise AQI Comparison

4. AQI Category Pie Chart

5. Heatmap of AQI by Month and City

Enter option : 2





--- AQI Data Analysis Menu ---

1. Data Analysis:

2. Data Manipulation:

3. Data Visualization:

4. Exit:

Enter your choice: 3

1. Visualize AQI Distribution

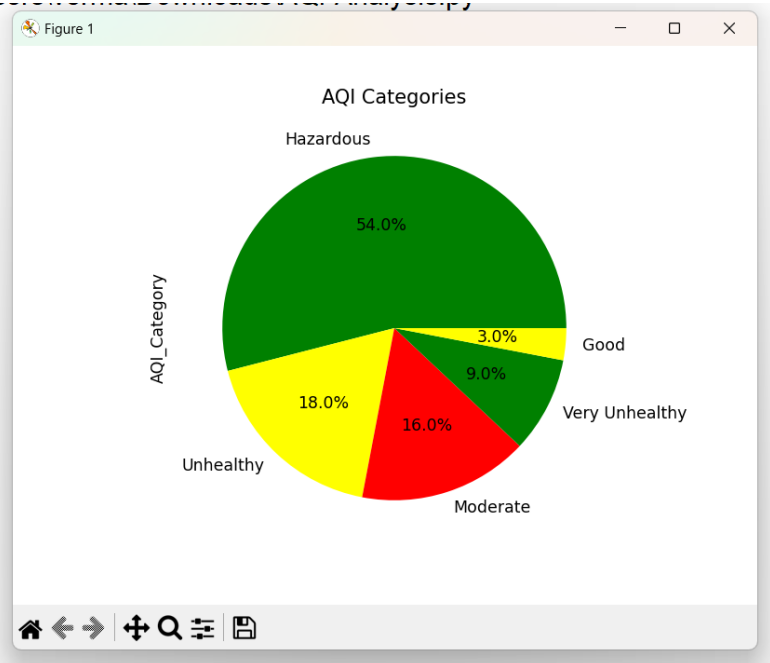
2. Seasonal AQI Trend Line

3. City-Wise AQI Comparison

4. AQI Category Pie Chart

5. Heatmap of AQI by Month and City

Enter option : 4



# Bibliography

- *Informatics Practices Class 12 by Preeti Arora.*
- *Study Trigger Mobile App*  
(<https://play.google.com/store/apps/details?id=com.studytrigger>) for Python Introduction, uses and other theory.
- <https://stackoverflow.com/> (for error resolving)
- <https://www.tutorialspoint.com/>
- <https://www.geeksforgeeks.org/>