LAB REPORT
ON

GET ACQUAINTED WITH DATA SCIENCE TOOLS AND PERFORM STATISTICAL ANALYSIS

FOUNDATION OF DATA SCIENCE

BY
SHANDES BAASNET
PUR080BCT084

To
ER. SUJAN KARKI

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

PURWANCHAL CAMPUS
DHARAN, NEPAL

Lab 1

1. Get acquainted with data science tools and perform statistical analysis

1.1 Objective

-To familiarize yourself with essential tools and libraries used in data science.

-To learn to describe data using descriptive statistics.

1.2 Theory

- **Python**

Python is a high-level and widely used programming language for data science due to its simplicity and rich ecosystem of libraires. Its popular libraries include Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, etc. Python is used for data manipulation, visualization, machine learning, and automation.

- **Anaconda**

Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment.

- **Jupyter Note Book**

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations,and narrative text. Its uses include data cleaning and transformation, numerical simulation, statistical modelling, data visualization,machine learning, and much more.

- **NumPy**

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. NumPy, stands for Numerical Python, is used for the manipulation of elements of numerical array data.

- **SciPy**

SciPy is a scientific computation library that uses NumPy underneath. SciPy stands for Scientific Python. It provides more utility functions for optimization, stats and signal processing. Like NumPy, SciPy is open source so we can use it freely. SciPy, stands for Scientific Python, is used for numerical computations in Python. Both these packages provide extended functionality to work with Python.

- **Stats models**

Stats models is a popular library in Python that enables us to estimate and analyse various statistical models. It is built on numeric and scientific libraries like NumPy and SciPy. It includes various models of linear regression like ordinary least squares, generalized least squares, weighted least squares, etc

- **Pandas**

Pandas are really powerful. They provide you with a huge set of important commands and features which are used to easily analyse your data. We can use Pandas to perform various tasks like filtering your data according to certain conditions, or segmenting and segregating the data according to preference, etc.

- **Matplotlib**

Matplotlib is Python library used for creation static, interactive, and animated visualizations. It provides a variety of plotting options, such as line plots, bar charts, scatter plots, and histograms. Matplotlib is highly customizable and serves as the foundation for other visualization libraries like Seaborn.

- **Seaborn**

Seaborn is a Python library built on Matplotlib, designed to simplify the creation of visually appealing and informative statistical graphics. It includes features like heatmaps, violin plots, pair plots, and categorical plots. Seaborn integrates seamlessly with Pandas, making it ideal for exploratory data analysis.

- **Scikit-learn**

Scikit learn is a powerful machine- learning in Python that provide stools for supervised and unsupervised learning, such as regression, classification, clustering, and dimensionality reduction. It also includes utilities for model selection, pre-processing, and evaluation. Scikit-learn is built on NumPy, SciPy, and Matplotlib.

- **Google Colab**

Google Colab is a cloud-based platform for running Python code, offering free GPU and TPU resources. It supports popular data science libraries like TensorFlow, PyTorch, Pandas, and NumPy. Google Colab is commonly used for machine learning, data analysis, and collaborative projects.

Descriptive statistics

Descriptive statistics is a branch of statistics that focuses on summarizing and organizing data to uncover patterns, relationships, and trends. It provides a foundation for data analysis by offering methods to describe and present data meaningfully.

Types of Descriptive statistics:

- **Measures of Central Tendency: Central tendency provides a single value that represents the centre or typical value of a dataset.**

Mean (Average): The sum of all data points divided by the total number of points.
Median: The middle value of an ordered dataset. It is less sensitive to outliers.
Mode: The value that appears most frequently in a dataset.

- **Measures of Dispersion (Spread): Dispersion measures how spread out the data values are around the central tendency.**

Range: The difference between the maximum and minimum values.
Variance: The average squared deviation from the mean.
Standard Deviation: The square root of the variance, showing how data points deviate from the mean.
Inter-quartile Range (IQR): The range of the middle 50% of the data, calculated as Q3 - Q1

- **Measures of Shape: These metrics describe the distribution and symmetry of the data.**

Skewness: Measures asymmetry. Positive skew indicates a longer tail on the right; negative skew indicates a longer tail on the left.
Kurtosis: Measures the "tailed ness" of the distribution. High kurtosis means heavy tails; low kurtosis means light tails.

1.3 Installing Process

Download Anaconda:

Step1: Search "Anaconda Download" in Browser.
Step 2: Provide you email to get link for downloading or you can skip registration to Download
Step 3: Click skip registration to go to download page. Click 64 bit install according to your OS.
Step 4: Open the downloaded exe file and install anaconda on your pc according to your preferences.
Step 5: After installing anaconda scroll down and search Jupyter notebook, and if not installed then install it and launch it
Step 6: Jupyter notebook will open on your browser. Then click on new and click Python3
Step 7: Write your code and run it. Shortcut is (SHIFT+ENTER).

Installing Packages:

Step 1: Go to the folder annoconda on your start menu and open "Annoconda Prompt".
Step 2: Installing
->Confirm Python is installed correctly, by typing "python -V".
->Confirm conda is installed properly, by typing "condda -V".
->You can install packages using pip or conda "pip(conda) install<package_name>"

- **For installing particular version of a package**

"pip install <package_name>=package_version"

- **For installing multiple packages all at once**

"pip install <package_name1> <package_name2> <package_name3>"

- **For updating package "pip install --upgrade <package_name>"**

- **For installing libraries for data-science**

"pip install numpy scipy stats models pandas seaborn matplotlib scikit-learn"
OR
"conda install numpy scipy statsmodels pandas seaborn matplotlib scikit-learn"

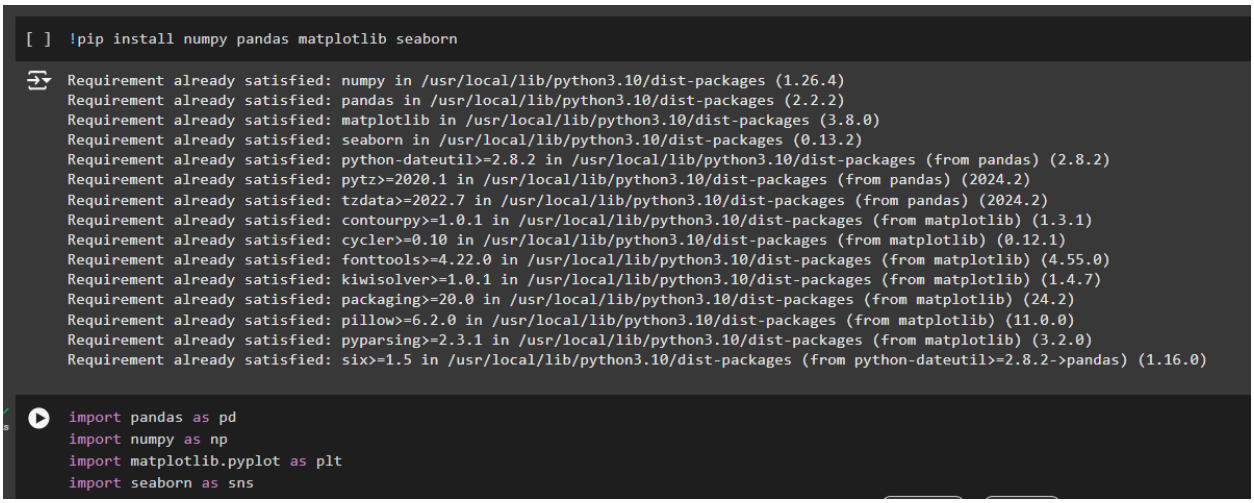
Access Google Colab

1. Visit https://colab.research.google.com/ and sign in with your Google account.
2. Create a new notebook by clicking on "New Notebook".

In this lab we will be using Google Colab.

1.4 Colab File:

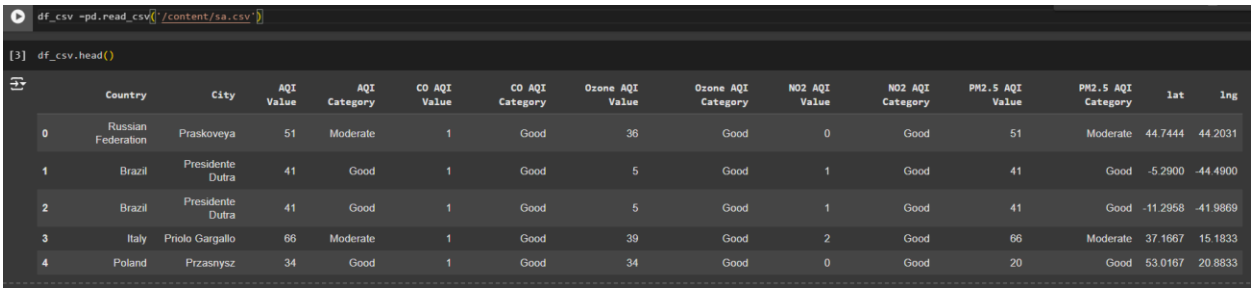
Screenshots of the data is below:



```
[ ] !pip install numpy pandas matplotlib seaborn

Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (1.26.4)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (2.2.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (3.8.0)
Requirement already satisfied: seaborn in /usr/local/lib/python3.10/dist-packages (0.13.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (4.55.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.7)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (24.2)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (11.0.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (3.2.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)

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



```
df_csv =pd.read_csv('/content/sa.csv')

[3] df_csv.head()
```

	Country	City	AQI Value	AQI Category	CO AQI Value	CO AQI Category	Ozone AQI Value	Ozone AQI Category	NO2 AQI Value	NO2 AQI Category	PM2.5 AQI Value	PM2.5 AQI Category	lat	lng
0	Russian Federation	Praskoveya	51	Moderate	1	Good	36	Good	0	Good	51	Moderate	44.7444	44.2031
1	Brazil	Presidente Dutra	41	Good	1	Good	5	Good	1	Good	41	Good	-5.2900	-44.4900
2	Brazil	Presidente Dutra	41	Good	1	Good	5	Good	1	Good	41	Good	-11.2958	-41.9869
3	Italy	Priolo Gargallo	66	Moderate	1	Good	39	Good	2	Good	66	Moderate	37.1667	15.1833
4	Poland	Przasnysz	34	Good	1	Good	34	Good	0	Good	20	Good	53.0167	20.8833

```
df_csv.shape
```

```
(16695, 14)
```

```
[6] df_csv.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16695 entries, 0 to 16694
Data columns (total 14 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Country              16393 non-null  object 
 1   City                 16695 non-null  object 
 2   AQI Value            16695 non-null  int64  
 3   AQI Category         16695 non-null  object 
 4   CO AQI Value         16695 non-null  int64  
 5   CO AQI Category      16695 non-null  object 
 6   Ozone AQI Value      16695 non-null  int64  
 7   Ozone AQI Category   16695 non-null  object 
 8   NO2 AQI Value        16695 non-null  int64  
 9   NO2 AQI Category     16695 non-null  object 
 10  PM2.5 AQI Value      16695 non-null  int64  
 11  PM2.5 AQI Category   16695 non-null  object 
 12  lat                  16695 non-null  float64
 13  lng                  16695 non-null  float64
dtypes: float64(2), int64(5), object(7)
memory usage: 1.8+ MB
```

```
df_csv.isnull()
```

	Country	City	AQI Value	AQI Category	CO AQI Value	CO AQI Category	Ozone AQI Value	Ozone AQI Category	NO2 AQI Value	NO2 AQI Category	PM2.5 AQI Value	PM2.5 AQI Category	lat	lng
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
16690	False	False	False	False	False	False	False	False	False	False	False	False	False	False
16691	False	False	False	False	False	False	False	False	False	False	False	False	False	False
16692	False	False	False	False	False	False	False	False	False	False	False	False	False	False
16693	False	False	False	False	False	False	False	False	False	False	False	False	False	False
16694	False	False	False	False	False	False	False	False	False	False	False	False	False	False

16695 rows × 14 columns

```
df_csv.isnull().sum()
```

	0
Country	302
City	0
AQI Value	0
AQI Category	0
CO AQI Value	0
CO AQI Category	0
Ozone AQI Value	0
Ozone AQI Category	0
NO2 AQI Value	0
NO2 AQI Category	0
PM2.5 AQI Value	0
PM2.5 AQI Category	0
lat	0
lng	0

dtype: int64

```
[9] df_csv.describe()
```

	AQI Value	CO AQI Value	Ozone AQI Value	NO2 AQI Value	PM2.5 AQI Value	lat	lng
count	16695.000000	16695.000000	16695.000000	16695.000000	16695.000000	16695.000000	16695.000000
mean	62.998682	1.342138	31.767355	3.819647	59.821324	30.267148	-3.944485
std	43.091971	2.371379	22.839343	5.880677	43.208298	22.947398	73.037148
min	7.000000	0.000000	0.000000	0.000000	0.000000	-54.801900	-171.750000
25%	38.500000	1.000000	20.000000	0.000000	34.000000	16.515450	-75.180000
50%	52.000000	1.000000	29.000000	2.000000	52.000000	38.815800	5.643100
75%	69.000000	1.000000	38.000000	5.000000	69.000000	46.683300	36.275000
max	500.000000	133.000000	222.000000	91.000000	500.000000	70.767000	178.017800

```
[10] df_csv.describe(include='all')
```

	Country	City	AQI Value	AQI Category	CO AQI Value	CO AQI Category	Ozone AQI Value	Ozone AQI Category	NO2 AQI Value	NO2 AQI Category	PM2.5 AQI Value	PM2.5 AQI Category	lat	lng
count	16393	16695	16695.000000	16695	16695.000000	16695	16695.000000	16695	16695.000000	16695	16695.000000	16695	16695.000000	16695.000000
unique	174	14229	NaN	6	NaN	3	NaN	5	NaN	2	NaN	6	NaN	NaN
top	United States of America	Santa Cruz	NaN	Good	NaN	Good	NaN	Good	NaN	Good	NaN	Good	NaN	NaN
freq	3954	17	NaN	7708	NaN	16691	NaN	15529	NaN	16684	NaN	7936	NaN	NaN
mean	NaN	NaN	62.998682	NaN	1.342138	NaN	31.767355	NaN	3.819647	NaN	59.821324	NaN	30.267148	-3.944485
std	NaN	NaN	43.091971	NaN	2.371379	NaN	22.839343	NaN	5.880677	NaN	43.208298	NaN	22.947398	73.037148
min	NaN	NaN	7.000000	NaN	0.000000	NaN	0.000000	NaN	0.000000	NaN	0.000000	NaN	-54.801900	-171.750000
25%	NaN	NaN	38.500000	NaN	1.000000	NaN	20.000000	NaN	0.000000	NaN	34.000000	NaN	16.515450	-75.180000
50%	NaN	NaN	52.000000	NaN	1.000000	NaN	29.000000	NaN	2.000000	NaN	52.000000	NaN	38.815800	5.643100
75%	NaN	NaN	69.000000	NaN	1.000000	NaN	38.000000	NaN	5.000000	NaN	69.000000	NaN	46.683300	36.275000
max	NaN	NaN	500.000000	NaN	133.000000	NaN	222.000000	NaN	91.000000	NaN	500.000000	NaN	70.767000	178.017800

```
[11] df_csv.describe(include='object')
```

	Country	City	AQI Category	CO AQI Category	Ozone AQI Category	NO2 AQI Category	PM2.5 AQI Category
count	16393	16695	16695	16695	16695	16695	16695
unique	174	14229	6	3	5	2	6
top	United States of America	Santa Cruz	Good	Good	Good	Good	Good
freq	3954	17	7708	16691	15529	16684	7936

```
[12] df_csv.columns
```

```
Index(['Country', 'City', 'AQI Value', 'AQI Category', 'CO AQI Value',
       'CO AQI Category', 'Ozone AQI Value', 'Ozone AQI Category',
       'NO2 AQI Value', 'NO2 AQI Category', 'PM2.5 AQI Value',
       'PM2.5 AQI Category', 'lat', 'lng'],
      dtype='object')
```

```
[14] df_csv.dropna()
```

	Country	City	AQI Value	AQI Category	CO AQI Value	CO AQI Category	Ozone AQI Value	Ozone AQI Category	NO2 AQI Value	NO2 AQI Category	PM2.5 AQI Value	PM2.5 AQI Category	lat	lng
0	Russian Federation	Praskoveya	51	Moderate	1	Good	36	Good	0	Good	51	Moderate	44.7444	44.2031
1	Brazil	Presidente Dutra	41	Good	1	Good	5	Good	1	Good	41	Good	-5.2900	-44.4900
2	Brazil	Presidente Dutra	41	Good	1	Good	5	Good	1	Good	41	Good	-11.2958	-41.9869
3	Italy	Priolo Gargallo	66	Moderate	1	Good	39	Good	2	Good	66	Moderate	37.1667	15.1833
4	Poland	Przasnysz	34	Good	1	Good	34	Good	0	Good	20	Good	53.0167	20.8833
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
16690	United States of America	Highland Springs	54	Moderate	1	Good	34	Good	5	Good	54	Moderate	37.5516	-77.3285
16691	Slovakia	Martin	71	Moderate	1	Good	39	Good	1	Good	71	Moderate	49.0650	18.9219
16692	Slovakia	Martin	71	Moderate	1	Good	39	Good	1	Good	71	Moderate	36.3385	-88.8513
16693	France	Sceaux	50	Good	1	Good	20	Good	5	Good	50	Good	48.7786	2.2906
16694	United States of America	Westerville	71	Moderate	1	Good	44	Good	2	Good	71	Moderate	40.1241	-82.9210

16393 rows × 14 columns

```
[16] df_csv['AQI Value'].unique()
```

```
array([ 51,  41,  66,  34,  54,  64,  68,  59,  55,  72,  28, 154,  67,
        62,  31,  56,  77,  44,  30,  79,  61,  32,  29, 247,  45,  36,
       124,  60,  47,  37,  58,  89,  52,  38,  88,  49, 203,  35,  90,
        48,  19, 155,  46, 142, 166,  27,  23, 170,  22, 133,  73,  50,
        25,  81,  65,  53,  57, 163, 126,  63,  86, 112, 121,  69,  20,
        75, 168, 103, 307,  42, 102, 125,  92, 143, 187,  98, 156, 105,
        80,  26, 107,  76, 104,  39, 500, 117,  17,  21, 151,  24, 152,
        13,  93, 153, 179,  94, 444,  18,  40, 182, 356, 226,  16,  43,
       101, 175,  33, 169, 176, 171,  74, 118,  70, 138,  95,  78, 150,
       116,  84, 130,  14, 159, 160,  12,  15,  82, 180, 194, 206, 157,
        87, 114,  99, 111, 301, 217,  85, 144, 174,  83, 291,  71, 109,
       165, 128, 264,  97, 148, 137, 108, 421,  96, 201, 120, 204, 186,
       198,  91, 162, 279, 164, 106, 131, 196, 100, 185, 181, 132, 178,
       173, 236, 224, 113, 145, 140, 167, 188, 240, 158, 122, 202, 295,
       260, 123, 119, 110, 161, 172, 316, 184, 190, 149, 134, 191, 135,
       127, 244,  10, 205, 136, 147, 207, 320, 192, 193, 220, 281,  11,
       189, 212, 129, 183, 177, 200, 324, 290, 265, 275, 146, 230, 195,
       115, 242, 305, 141,   9, 321, 274, 248, 329, 384, 358, 197,   7,
       235, 355, 277, 222, 219, 249, 210, 256, 245, 233, 234, 232, 225,
       283, 386, 139, 296, 209, 215, 208, 269, 211, 365, 425, 199,   8,
       372, 325, 266, 310, 377, 438, 214, 254, 216, 213, 237, 323, 243,
       392, 270, 328, 285, 267, 262, 251, 253, 252])
```

```
[17] df_csv['AQI Value'].value_counts()
```

AQI Value	count
50	413
52	374
35	366
51	359
54	352
...	...
234	1
232	1
225	1
283	1
252	1

282 rows × 1 columns

dtype: int64

```
[22] mean= df_csv['AQI Value'].mean()
     print('Mean =',mean)
```

```
Mean = 62.99868224019168
```

```
[23] median = df_csv['AQI Value'].median()
     print('Median =', median)
```

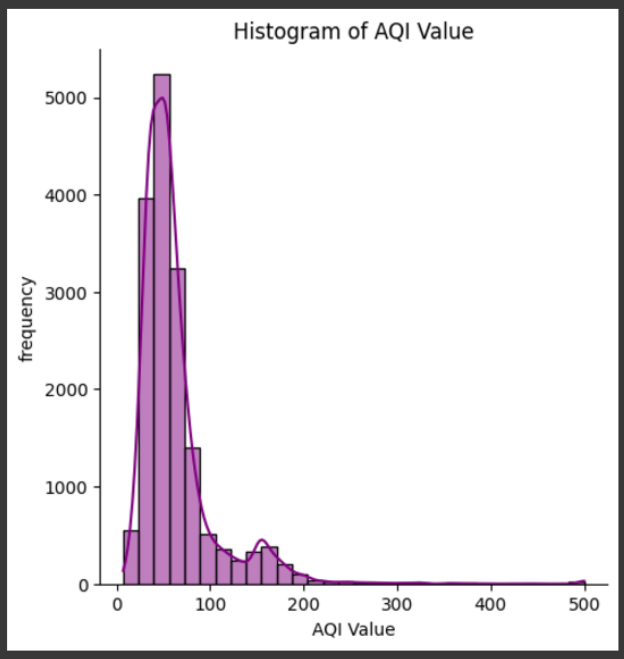
```
Median = 52.0
```

```
[24] mode = df_csv['AQI Value'].mode
     print('Mode =', mode)
```

```
Mode = <bound method Series.mode of 0        51
1        41
2        41
3        66
4        34
         ..
16690    54
16691    71
16692    71
16693    50
16694    71
Name: AQI Value, Length: 16695, dtype: int64>
```

HERE, mean > median > mode

```
sns.displot(df_csv['AQI Value'], kde= True, bins= 30, kind='hist', color ='purple')
plt.title('Histogram of AQI Value')
plt.xlabel('AQI Value')
plt.ylabel('frequency')
plt.show()
```



```
[32] min = df_csv['AQI Value'].min()
     print('Minimum value = ', min)
```

```
Minimum value =  7
```

```
[33] max = df_csv['AQI Value'].max()
     print('Maximum value = ', max)
```

```
Maximum value =  500
```

```
[35] range = max- min
     print('Range = ', range)
```

```
Range =  493
```

```
[37] variance = df_csv['AQI Value'].var()
     print("Variance = ", variance)
```

```
Variance =  1856.9179328506536
```

```
[38] sd= df_csv['AQI Value'].std()
     print('Standard Deviation =',sd)
```

```
Standard Deviation = 43.09197063085713
```

```
[40] q1= df_csv['AQI Value'].quantile(0.25)
     print('Q1 =',q1)
```

```
Q1 = 38.5
```

```
[41] q2= df_csv['AQI Value'].quantile(0.5)
     print('Q2 =',q2)
```

```
Q2 = 52.0
```

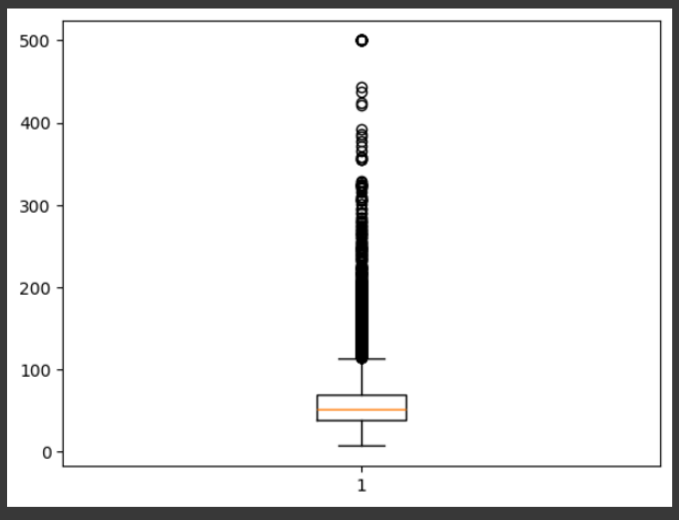
```
[42] q3= df_csv['AQI Value'].quantile(0.75)
     print('Q3 =',q3)
```

```
Q3 = 69.0
```

```
[44] irq=q3-q1
     print("Inner quartile range =",irq;)
```

```
Inner quartile range = 30.5
```

```
[45] plt.boxplot(df_csv['AQI Value'])
     plt.show()
```



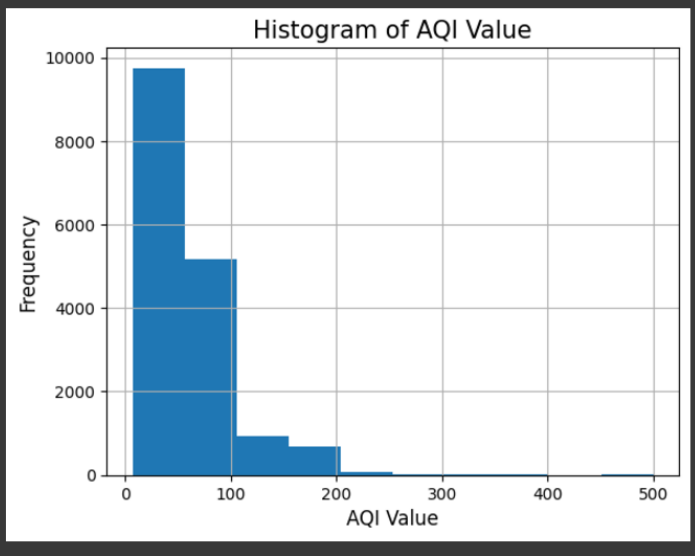

```
[ ] skew = df_csv['AQI Value'].skew()
    print('Skewness = ', skew)
```

```
Skewness =  3.5842884193221445
```

```
[48] kurt = df_csv['AQI Value'].kurt()
     print('Kurtosis = ',kurt)
```

```
Kurtosis =  22.83451926518874
```

```
plt.hist(df_csv['AQI Value'])
plt.title('Histogram of AQI Value', fontsize=15)
plt.xlabel('AQI Value', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.grid(True)
plt.show()
```



1.5 Conclusion

Hence, Anaconda Navigator and Jupyter Notebook, along with Python packages such as NumPy, Pandas, Matplotlib, and Seaborn, have been successfully downloaded and installed. Also, these data science packages were explored in Google Colab to perform descriptive statistical analysis.