

Python Project

The Impact of Climate Change Through Air Quality Analysis

Computer Information System, Cal State LA

CIS 5270 – Business Intelligence

Dr. Shilpa Balan

Presented By

Christopher Lee

Sirisha Mahesh

Table of Contents

Introduction	3
Data Sources.....	6
Data Dictionary	8
Data Cleaning.....	10
Summary Statistics	24
Analysis and Visualizations	35
Bibliography.....	47

Introduction

Climate change is no longer a distant possibility but an urgent reality that is actively reshaping ecosystems, economies, and daily life. The importance of this project, which analyzes global climate and air quality data, lies in its ability to uncover patterns in environmental degradation and public health risks. By drawing from quantitative evidence, this work contributes to a deeper understanding of how pollutants, weather conditions, and human activity intertwine to accelerate climate impacts. It provides critical insight into the causes and consequences of climate change, empowering both researchers and policymakers to act with precision and urgency.

According to the United Nations, the primary driver of current climate change is human activity, especially the emission of greenhouse gases through burning fossil fuels and deforestation. These emissions intensify the Earth's natural greenhouse effect, trapping heat and disrupting planetary climate systems. This in turn leads to rising temperatures, melting glaciers, sea level rise, and increased frequency and intensity of extreme weather events such as droughts, wildfires, and hurricanes (Causes and Effects of Climate Change). However, the more immediate and personal dimension of these changes is air quality which is something this project places at the forefront of its analysis.

As NASA explains, air quality and climate change are deeply interconnected. Pollutants like ground-level ozone and particulate matter not only pose direct health threats but also contribute to warming. For example, black carbon absorbs sunlight and accelerates the melting of ice and snow. On the other hand, rising temperatures exacerbate the formation of ground-level ozone, especially during heat waves (Climate Change). These reinforcing cycles make air quality both a symptom and a driver of climate change, reinforcing the necessity of tracking it over time and across regions.

The value of this project lies in its use of actual environmental data—processed and visualized through Python—to reveal meaningful correlations and trends. For instance, the analysis shows how higher temperatures often coincide with higher Air Quality Index (AQI) values, indicating degraded air quality. It also reveals temporal and geographic variability in pollution patterns. This aligns with NASA’s broader observation that understanding Earth’s climate requires not only high-resolution satellite data, but also local ground-truth analysis that can explain variation at the city or country level (Climate Change).

Furthermore, this project plays an important role in connecting environmental science to public health. Poor air quality is linked to respiratory illnesses, cardiovascular diseases, and premature death. The World Health Organization attributes nearly 7 million deaths annually to air pollution. Climate-sensitive diseases, environmental stress, and displacement due to rising sea levels are increasing frequently. By identifying where and when air pollution levels spike, our analysis supports more informed public health interventions and urban planning strategies.

This work also aligns with global policy goals. The United Nations Sustainable Development Goals (SDGs), especially Goal 13 (Climate Action), emphasize the need for timely data to support mitigation and adaptation strategies. Projects like this not only satisfy academic criteria, but also have real-world applications in sustainability and policy development. As emphasized by the UN, addressing climate change requires both reducing emissions and improving our understanding of ongoing changes (Causes and Effects of Climate Change).

In conclusion, the significance of this project is rooted in its ability to transform data into insight. It bridges the gap between abstract scientific models and lived environmental realities. By contextualizing climate change through air quality which is something we can see, measure, and

feel, work such as this becomes a vital tool in educating stakeholders and informing decision-making. It demonstrates how localized, data-driven storytelling can help us confront a global crisis and push us toward more sustainable, equitable, and informed solutions.

Data Sources

1. <https://www.kaggle.com/datasets/khushikyad001/air-quality-data>

- a. This dataset conveys comprehensive air quality information by highlighting major atmospheric pollutants including carbon monoxide (CO), nitrogen oxides (NO_x and NO₂), ozone (O₃), sulfur dioxide (SO₂), and particulate matter (PM_{2.5} and PM₁₀). At the center is an Air Quality Index (AQI) gauge that spans from green (indicating clean air) to red (indicating hazardous conditions), offering an intuitive snapshot of environmental health.

The data is set against the backdrop of an urban environment, emphasizing the relevance of pollution in urban areas. Additional data elements such as temperature, humidity, wind speed indicators, and pollutant-specific bar charts enrich the context, making it a multi-layered and insightful tool for interpreting air quality data.

2. <https://www.kaggle.com/datasets/hasibalmuzdadid/global-air-pollution-dataset>

- a. This dataset provides geolocated measurements of key air pollutants with direct implications for public health and environmental quality. Nitrogen Dioxide (NO₂), primarily emitted by vehicles and industrial sources, aggravates asthma and respiratory conditions, particularly in children, the elderly, and those with preexisting lung issues. Ground-level Ozone (O₃), formed by the reaction of nitrogen oxides and volatile organic compounds in sunlight, causes coughing, chest pain, and worsens conditions like asthma and bronchitis. It also damages crops and vegetation, especially during the growing season. Carbon Monoxide (CO), a colorless and odorless gas from fossil fuel combustion,

impairs oxygen delivery to vital organs and can cause dizziness, confusion, or death in high concentrations, especially indoors. Particulate Matter, PM10 and PM2.5, consists of airborne particles from sources like vehicle emissions, construction, and combustion. These particles can enter the lungs and bloodstream, leading to cardiovascular and respiratory diseases, and have been classified as carcinogenic. Together, these pollutants form a critical part of environmental monitoring efforts. This dataset enables regional and temporal tracking of air quality trends, supporting research, public health planning, and climate policy.

Data Dictionary

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Date	Time	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	CO(GT)	NOx(GT)	NO2(GT)	O3(GT)	SO2(GT)	PM2.5
2	1/1/2024	0:00	Russian Federation	Praskovey	51	Moderate	1	Good	36	Good	0	Good	51	Moderate	3.807947	172.0268	144.3333	118.1208	1.215679	147.3497
3	1/1/2024	1:00	Brazil	Presidente	41	Good	1	Good	5	Good	1	Good	41	Good	9.512072	241.8243	137.7693	15.32583	1.016178	40.97984
4	1/1/2024	2:00	Italy	Priolo Gar	66	Moderate	1	Good	39	Good	2	Good	66	Moderate	7.34674	228.2881	20.05509	44.37704	24.14091	72.59474
5	1/1/2024	3:00	Poland	Przasnysz	34	Good	1	Good	34	Good	0	Good	20	Good	6.026719	47.01607	184.5919	139.4886	2.435392	134.3397
6	1/1/2024	4:00	France	Punaauiua	22	Good	0	Good	22	Good	0	Good	6	Good	1.644585	45.62559	114.126	95.63477	48.7521	99.00742
7	1/1/2024	5:00	United States	Punta Gorda	54	Moderate	1	Good	14	Good	11	Good	54	Moderate	1.644346	81.18414	73.38138	167.1065	11.88119	149.0214
8	1/1/2024	6:00	Germany	Puttlingen	62	Moderate	1	Good	35	Good	3	Good	62	Moderate	0.675028	108.9613	151.5512	77.7465	24.16057	192.4355
9	1/1/2024	7:00	Belgium	Puurs	64	Moderate	1	Good	29	Good	7	Good	64	Moderate	8.675144	123.1282	52.21573	156.6964	3.995744	27.7266
10	1/1/2024	8:00	Russian Federation	Pyatigorsk	54	Moderate	1	Good	41	Good	1	Good	54	Moderate	6.051039	204.2295	139.0085	81.36302	26.58037	143.3657
11	1/1/2024	9:00	Egypt	Qalyub	142	Unhealthy	3	Good	89	Moderate	9	Good	142	Unhealthy	7.109919	17.94745	8.90253	176.2854	19.61785	49.91711
12	1/1/2024	10:00	China	Qinzhou	68	Moderate	2	Good	68	Moderate	1	Good	58	Moderate	0.303786	11.36714	168.0422	172.8104	3.188341	85.82296
13	1/1/2024	11:00	Netherlands	Raalte	41	Good	1	Good	24	Good	6	Good	41	Good	9.702108	118.1813	90.186	59.91547	17.42109	11.40823
14	1/1/2024	12:00	India	Radaur	158	Unhealthy	3	Good	139	Unhealthy	1	Good	158	Unhealthy	8.341182	209.4519	127.7019	159.5803	20.18036	31.50194
15	1/1/2024	13:00	Pakistan	Radhan	158	Unhealthy	1	Good	50	Good	1	Good	158	Unhealthy	2.202157	58.83717	45.02633	152.7224	49.50615	67.35657
16	1/1/2024	14:00	Republic of Moldova	Radovis	83	Moderate	1	Good	46	Good	0	Good	83	Moderate	1.900067	192.8098	135.7377	38.71179	11.18641	71.68723
17	1/1/2024	15:00	France	Raismes	59	Moderate	1	Good	30	Good	4	Good	59	Moderate	1.915705	78.68861	154.6758	112.748	27.6942	180.4191
18	1/1/2024	16:00	India	Rajgir	154	Unhealthy	3	Good	100	Unhealthy	2	Good	154	Unhealthy	3.111998	265.9397	95.16942	14.01013	49.37846	149.6535
19	1/1/2024	17:00	Italy	Ramacca	55	Moderate	1	Good	47	Good	0	Good	55	Moderate	5.295089	268.8113	123.9277	150.7712	39.80894	194.5757
20	1/1/2024	18:00	United States	Phoenix	72	Moderate	1	Good	4	Good	23	Good	72	Moderate	4.376256	89.88887	91.80886	32.24371	49.67501	121.8189
21	1/1/2024	19:00	India	Phulaban	161	Unhealthy	2	Good	71	Moderate	0	Good	161	Unhealthy	2.983168	69.76813	178.9242	13.67583	48.44662	52.13418
22	1/1/2024	20:00	Poland	Piasieczno	28	Good	1	Good	28	Good	2	Good	28	Good	6.157344	123.9799	100.076	72.63922	24.98744	68.89273
23	1/1/2024	21:00	India	Pimpri	118	Unhealthy	2	Good	30	Good	2	Good	118	Unhealthy	1.480989	72.91894	168.5199	104.2691	35.12242	67.36801
24	1/1/2024	22:00	Brazil	Pindobaci	33	Good	0	Good	10	Good	1	Good	33	Good	2.992232	202.0428	160.4691	154.0301	6.411713	68.66639

Field Name	Description	Example Value	Type
Date	Date of measurement	1/30/2024	Text
Time	Time of measurement	16:00	Text
Country	Country where data was recorded	China	Text
City	City where data was recorded	Kaitong	Text
AQI Value	Overall Air Quality Index value, calculated from multiple pollutants to represent general air pollution level on a scale (typically 0–500).	95	Number
AQI Category	Categorical interpretation of AQI Value (e.g., Good, Moderate, Unhealthy, etc.).	Moderate	Text
CO AQI Value	AQI contribution value specifically from Carbon Monoxide (CO) concentrations.	2	Number
CO AQI Category	Health category derived from CO AQI Value (e.g., Moderate, Unhealthy).	Good	Text
Ozone AQI Value	AQI contribution from Ozone (O ₃) levels.	79	Number
Ozone AQI Category	Ozone-specific health category based on Ozone AQI Value.	Moderate	Text
NO2 AQI Value	AQI contribution from Nitrogen Dioxide (NO ₂) concentrations.	0	Number
NO2 AQI Category	Health impact category derived from NO ₂ levels.	Good	Text
PM2.5 AQI Value	AQI value based on concentrations of fine particulate matter (PM2.5).	95	Number
PM2.5 AQI Category	Air quality category based on PM2.5 AQI value.	Moderate	Text

CO(GT)	Ground Truth (GT) measurement of Carbon Monoxide in mg/m ³ .	4.768604267	Number
NOx(GT)	Ground Truth (GT) measurement of nitrogen oxides (NO + NO ₂) in ppb or µg/m ³ .	77.73751495	Number
NO2(GT)	Ground Truth (GT) measurement of Nitrogen Dioxide in ppb or µg/m ³ .	133.1610221	Number
O3(GT)	Ground Truth (GT) measurement of ground-level Ozone in ppb or µg/m ³ .	119.4784093	Number
SO2(GT)	Ground Truth (GT) measurement of Sulfur Dioxide in ppb or µg/m ³ .	25.16995727	Number
PM2.5	Measured concentration of fine particulate matter (≤2.5 µm) in µg/m ³ .	89.94586349	Number
PM10	Measured concentration of particulate matter (≤10 µm) in µg/m ³ .	172.129641	Number
Climate_Condition	Qualitative weather description (e.g., Good, Moderate, Unhealthy, etc.).	Moderate	Text
Temperature	Ambient air temperature in degrees Celsius (°C).	27.80015693	Number
Humidity	Relative humidity in percentage (%).	32.85552177	Number
Pressure	Atmospheric pressure in hPa or millibars.	991.4972976	Number
WindSpeed	Speed of wind at ground level, typically in meters per second (m/s).	2.077258534	Number
WindDirection	Direction from which wind is coming, measured in degrees (0–360°).	12.58163294	Number
CO_NOx_Ratio	Ratio of Carbon Monoxide to Nitrogen Oxides, indicating combustion quality or traffic intensity.	0.060563307	Number
NOx_NO2_Ratio	Ratio of total nitrogen oxides (NOx) to NO ₂ , used to infer the presence of NO.	0.579434427	Number
Temp_Humidity_Index	Combined metric estimating human-perceived temperature (also known as heat index).	9.133886611	Number
AirQualityIndex	Redundant or consolidated field representing AQI (may match AQI Value).	285.4678628	Number
CO_MA3	Three-day moving average of CO(GT), smoothing short-term fluctuations.	6.130561204	Number
NO2_MA3	Three-day moving average of NO ₂ (GT).	117.7197176	Number
O3_MA3	Three-day moving average of O ₃ (GT).	59.93539639	Number
DayOfWeek	Numerical day of the week	1	Number
Hour	Numerical time of day	16	Number

Data Cleaning

Data cleaning was an essential process to prepare the dataset for statistical analysis and visualization. Three distinct data cleaning techniques were applied to ensure completeness, accuracy, and consistency.

Handling Missing Values

The dataset included missing values in both numerical and categorical fields. For numerical fields such as CO_MA3, NO2_MA3, and O3_MA3, the mean of each column was used to impute missing data. Categorical fields like 'Country', 'City', and 'AQI Category' were filled using the mode. Additionally, missing values in the 'Date' and 'Time' fields were filled using forward fill techniques. This ensured that rows were not discarded unnecessarily, and time-series data continuity was preserved.

Data Type Conversion

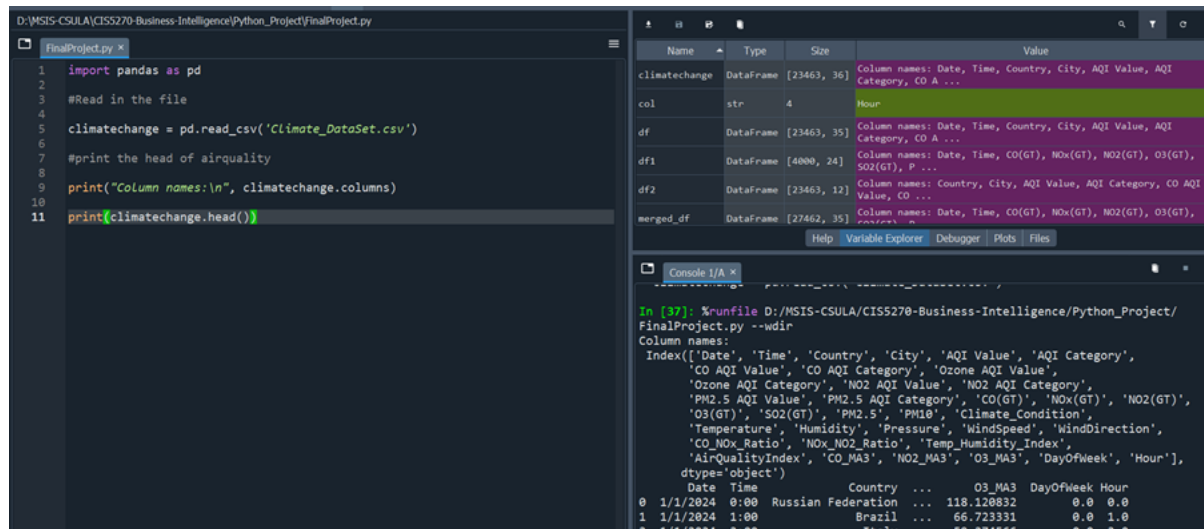
Columns 'Date' and 'Time' were initially of object type and contained inconsistent or unrecognized formats. These were converted to datetime and time formats respectively using pandas datetime parsing with coercion for errors. This step was crucial to allow chronological operations such as grouping by day or hour.

Standardization and Normalization

Standardization was performed on categorical fields to remove case sensitivity, leading/trailing whitespace, and inconsistent labels. Country names were normalized (e.g., 'Usa', 'U.S.A' -> 'USA') and special characters in the 'AQI Category' column were removed. City names were also title-cased for consistency across entries. This allowed more reliable filtering, grouping, and

comparisons during analysis. The following screenshots illustrate the before and after results for each of the three data cleaning categories:

Reading the CSV file



The screenshot displays a Jupyter Notebook environment. On the left, a code editor shows the following Python code:

```
1 import pandas as pd
2
3 #Read in the file
4
5 climatechange = pd.read_csv('Climate_DataSet.csv')
6
7 #print the head of airquality
8
9 print("Column names:\n", climatechange.columns)
10
11 print(climatechange.head())
```

On the right, the Variable Explorer shows the structure of the loaded data:

Name	Type	Size	Value
climatechange	DataFrame	[23463, 36]	Column names: Date, Time, Country, City, AQI Value, AQI Category, CO A ...
col	str	4	Hour
df	DataFrame	[23463, 35]	Column names: Date, Time, Country, City, AQI Value, AQI Category, CO A ...
df1	DataFrame	[4000, 24]	Column names: Date, Time, CO(GT), NOx(GT), NO2(GT), O3(GT), SO2(GT), P ...
df2	DataFrame	[23463, 12]	Column names: Country, City, AQI Value, AQI Category, CO AQI Value, CO ...
merged_df	DataFrame	[27462, 35]	Column names: Date, Time, CO(GT), NOx(GT), NO2(GT), O3(GT), ...

Below the Variable Explorer, the Console shows the output of the code:

```
In [37]: %runfile D:/MSIS-CSULA/CISS270-Business-Intelligence/Python_Project/FinalProject.py --wdir
Column names:
Index(['Date', 'Time', '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', 'CO(GT)', 'NOx(GT)', 'NO2(GT)',
      'O3(GT)', 'SO2(GT)', 'PM2.5', 'PM10', 'Climate Condition',
      'Temperature', 'Humidity', 'Pressure', 'WindSpeed', 'WindDirection',
      'CO_NOx_Ratio', 'NOx_NO2_Ratio', 'Temp_Humidity_Index',
      'AirQualityIndex', 'CO_MA3', 'NO2_MA3', 'O3_MA3', 'DayOfWeek', 'Hour'],
      dtype='object')
Date Time Country ... O3_MA3 DayOfWeek Hour
0 1/1/2024 0:00 Russian Federation ... 118.120832 0.0 0.0
1 1/1/2024 1:00 Brazil ... 66.723331 0.0 1.0
2 1/1/2024 2:00 Brazil ... 66.723331 0.0 2.0
```

Data Cleaning –1 Melt Function in Data Cleaning

Explanation of the Code:

This Python code performs a data transformation on a climate dataset using Pandas. The main intention of this script is to reshape the data from a wide format (where each type of measurement like Temperature, Humidity, and Pressure has its own column) to a long format (where measurement types are stored in a single column with their corresponding values in another column). This transformation is often useful for data analysis, visualization, or feeding data into machine learning models, which typically prefer long-format data.

```

Created on Sat May 10 13:00:37 2025

@author: Sirisha
"""

import pandas as pd

# Read the file
climatechange = pd.read_csv('Climate_DataSet.csv')

# Configure display options for better visibility
pd.set_option('display.max_rows', 1000)
pd.set_option('display.max_columns', 10)
pd.set_option('display.max_colwidth', 100)
pd.set_option('display.width', None)

print("==== BEFORE MELT: Wide Format =====")
print(climatechange.head(10)) # print first 10 rows for brevity
|
# Melt selected columns into long format
climatechange_melt = pd.melt(
    climatechange,
    id_vars=['Date', 'City'], # columns to keep
    value_vars=['Temperature', 'Humidity', 'Pressure'], # columns to unpivot
    var_name='Measurement_Type',
    value_name='Value'
)

# Print the melted DataFrame (Long Format)
print("\n==== AFTER MELT: Long Format =====")
print(climatechange_melt.head(20)) # print first 20 rows for better context

```

Before Data Cleaning Screenshot

Code Does Before the Transformation:

- It reads a CSV file named Climate_DataSet.csv into a Pandas DataFrame.
- It sets display options to make sure the output is readable in the console.
- It prints the first 10 rows of the original dataset, which is in wide format: each row contains a Date, City, and separate columns for Temperature, Humidity, and Pressure.

```

In [102]: %runfile D:/MSIS-CSULA/CIS5270-Business-Intelligence/Python_Project/FinalProject.py --wdir
===== BEFORE MELT: Wide Format =====

```

	Date	Time	Country	City	AQI	Value	...	\
0	1/1/2024	0:00	Russian Federation	Praskoveya	51	...		
1	1/1/2024	1:00	Brazil	Presidente Dutra	41	...		
2	1/1/2024	2:00	Italy	Priolo Gargallo	66	...		
3	1/1/2024	3:00	Poland	Przasnysz	34	...		
4	1/1/2024	4:00	France	Punaauia	22	...		
5	1/1/2024	5:00	United States of America	Punta Gorda	54	...		
6	1/1/2024	6:00	Germany	Puttlingen	62	...		
7	1/1/2024	7:00	Belgium	Puurs	64	...		
8	1/1/2024	8:00	Russian Federation	Pyatigorsk	54	...		
9	1/1/2024	9:00	Egypt	Qalyub	142	...		

	CO_MA3	NO2_MA3	O3_MA3	DayOfWeek	Hour
0	3.807947	144.333317	118.120832	0.0	0.0
1	6.660009	141.051317	66.723331	0.0	1.0
2	6.888920	100.719240	59.274566	0.0	2.0
3	7.628510	114.138771	66.397156	0.0	3.0
4	5.006015	106.257654	93.166802	0.0	4.0
5	3.105216	124.033085	134.076611	0.0	5.0
6	1.321319	113.019508	113.495911	0.0	6.0
7	3.664839	92.382761	133.849793	0.0	7.0
8	5.133737	114.258454	105.268646	0.0	8.0
9	7.278700	66.708905	138.114960	0.0	9.0

[10 rows x 36 columns]

After Data Cleaning Screenshot

Code Does After the Transformation:

- It uses `pd.melt()` to convert the wide-format data into long format.
- The Date and City columns are retained as identifiers (`id_vars`).
- The measurement columns (Temperature, Humidity, Pressure) are "melted" into two columns - `Measurement_Type` (indicates what type of measurement it is), `Value` (holds the actual numeric data).
- It prints the first 20 rows of the transformed Data Frame to show the new structure.

```

===== AFTER MELT: Long Format =====
   Date      City Measurement_Type      Value
0  1/1/2024   Praskoveya      Temperature  28.564580
1  1/1/2024  Presidente Dutra      Temperature   6.793192
2  1/1/2024   Priolo Gargallo      Temperature  24.436552
3  1/1/2024   Przasnysz      Temperature  26.463951
4  1/1/2024   Punaauia      Temperature  10.530331
5  1/1/2024   Punta Gorda      Temperature  16.371354
6  1/1/2024   Puttlingen      Temperature  13.169519
7  1/1/2024    Puurs      Temperature  37.630705
8  1/1/2024   Pyatigorsk      Temperature   9.397699
9  1/1/2024    Qalyub      Temperature   3.249537
10 1/1/2024   Qinzhou      Temperature  11.451678
11 1/1/2024    Raalte      Temperature  25.128207
12 1/1/2024   Radaur      Temperature  26.381321
13 1/1/2024   Radhan      Temperature  36.582579
14 1/1/2024   Radovis      Temperature  34.678952
15 1/1/2024   Raismes      Temperature   7.974423
16 1/1/2024   Rajgir      Temperature  15.482346
17 1/1/2024   Ramacca      Temperature  28.627104
18 1/1/2024   Phoenix      Temperature  36.244178
19 1/1/2024   Phulabani      Temperature  10.172426
d:\msis-csula\cis5270-business-intelligence\python_project\finalproject.py:11: DtypeWarning: Columns (0,1) have mixed types.
Specify dtype option on import or set low_memory=False.
climatechange = pd.read_csv('Climate_DataSet.csv')

```

The intent of the code is to prepare the dataset for easier analysis or plotting by converting it into a long format, which is more flexible and compatible with libraries like Seaborn or for grouping operations in Pandas. This is a common step in data wrangling for climate or time-series data.

Data Cleaning -2 Code of Data Cleaning Missing Value handling and mean imputation

Explanation of the Code:

This Python code is focused on handling missing values in a dataset using Pandas. It specifically cleans three air quality measurement columns: CO_MA3, NO2_MA3, and O3_MA3. These likely represent 3-day moving averages of pollutants like carbon monoxide, nitrogen dioxide, and ozone. The code follows a standard data preprocessing technique: missing value imputation using the mean of each column.

```

5  @author: Sirisha
6  """
7  import pandas as pd
8
9  # Read in the file
10 climatechange = pd.read_csv('Climate_DataSet.csv')
11
12 # Print the first 5 rows
13 print("Initial Data Preview:")
14 print(climatechange[['CO_MA3', 'NO2_MA3', 'O3_MA3']].head())
15
16 # Show missing value counts before cleaning
17 print("\nMissing Values Before Cleaning:")
18 print(climatechange[['CO_MA3', 'NO2_MA3', 'O3_MA3']].isnull().sum())
19
20 # Calculate the mean of the columns
21 CO_MA3_mean = climatechange['CO_MA3'].mean()
22 NO2_MA3_mean = climatechange['NO2_MA3'].mean()
23 O3_MA3_mean = climatechange['O3_MA3'].mean()
24
25 # Fill missing values with the calculated means
26 climatechange['CO_MA3'] = climatechange['CO_MA3'].fillna(CO_MA3_mean)
27 climatechange['NO2_MA3'] = climatechange['NO2_MA3'].fillna(NO2_MA3_mean)
28 climatechange['O3_MA3'] = climatechange['O3_MA3'].fillna(O3_MA3_mean)
29
30 # Show missing value counts after cleaning
31 print("\nMissing Values After Cleaning:")
32 print(climatechange[['CO_MA3', 'NO2_MA3', 'O3_MA3']].isnull().sum())
33
34 # Print first few rows after cleaning
35 print("\nCleaned Data Preview:")
36 print(climatechange[['CO_MA3', 'NO2_MA3', 'O3_MA3']].head())
37

```

Before Data Cleaning

Code Does Before the Transformation:

- Reads in the Climate_DataSet.csv file.
- Prints the first 5 rows of the three specific columns (CO_MA3, NO2_MA3, O3_MA3) to preview the data.
- Calculates and prints how many missing (NaN) values exist in each of these columns before any cleaning.

```
In [106]: %runfile D:/MSIS-CSULA/CIS5270-Business-Intelligence/Python_Project/FinalProject1.1.py --wdir
Initial Data Preview:
   CO_MA3   NO2_MA3   O3_MA3
0  3.807947  144.333317  118.120832
1  6.660009  141.051317   66.723331
2  6.888920  100.719240   59.274566
3  7.628510  114.138771   66.397156
4  5.006015  106.257654   93.166802

Missing Values Before Cleaning:
CO_MA3      19463
NO2_MA3     19463
O3_MA3      19463
dtype: int64
```

After Data Cleaning

Code Does After the Transformation:

- Calculates the mean of each column.
- Replaces the missing values in each column with the corresponding column's mean using `fillna()`.
- Re-checks and prints the number of missing values (which should now be 0).
- Prints the first 5 rows again to show the cleaned dataset.

```
Missing Values After Cleaning:
CO_MA3      0
NO2_MA3     0
O3_MA3      0
dtype: int64

Cleaned Data Preview:
   CO_MA3   NO2_MA3   O3_MA3
0  3.807947  144.333317  118.120832
1  6.660009  141.051317   66.723331
2  6.888920  100.719240   59.274566
3  7.628510  114.138771   66.397156
4  5.006015  106.257654   93.166802
d:\msis-csula\cis5270-business-intelligence\python_project\finalproject1.1.py:10: DtypeWarning: Columns (0,1)
have mixed types. Specify dtype option on import or set low_memory=False.
climatechange = pd.read_csv('Climate_DataSet.csv')
```

The goal of this code is to prepare the dataset for analysis or modeling by ensuring there are no missing values in critical pollutant data columns. This kind of preprocessing is essential because many machine learning models and analysis techniques cannot handle missing values directly. By

replacing them with the column mean, the code ensures **data consistency** while preserving overall trends in the data.

Data Cleaning 3 - Data Cleaning and Standardization of Categorical Fields

Explanation of the Code:

This script performs data cleaning and standardization on textual columns in a climate dataset — specifically focusing on Country, City, and AQI Category. These types of cleanup operations are crucial in data analysis to remove inconsistencies and make values uniform, which helps in grouping, filtering, and visualizing data accurately.

```
5  @author: Sirisha
6  """
7  import pandas as pd
8  # Load the dataset
9  climatechange = pd.read_csv('Climate_DataSet.csv')
10 # ----- BEFORE CLEANING -----
11 print("\n BEFORE CLEANING (First 5 Rows):")
12 print(climatechange[['Country', 'City', 'AQI Category']].head())
13
14 print("\nUnique Country Values (Before):")
15 print(climatechange['Country'].dropna().unique()[:10]) # Show sample
16
17 print("\nUnique City Values (Before):")
18 print(climatechange['City'].dropna().unique()[:10])
19
20 print("\nUnique AQI Categories (Before):")
21 print(climatechange['AQI Category'].dropna().unique())
22
23 # Standardize country names
24 climatechange['Country'] = climatechange['Country'].replace({
25     'U.S.A': 'USA',
26     'Usa': 'USA',
27     'united states': 'USA',
28     'Brazil ': 'Brazil',
29     'poland': 'Poland',
30     'france': 'France'
31 })
32 climatechange['Country'] = climatechange['Country'].str.strip().str.title()
33 # Standardize city names
34 climatechange['City'] = climatechange['City'].str.strip().str.title()
35 # Clean AQI Category from special characters
36 climatechange['AQI Category'] = climatechange['AQI Category'].str.replace(r'[^w\s]', '', regex=True)
37
```

Before Data Cleaning

Code Does Before Cleaning:

- Loads the dataset from Climate_DataSet.csv.

- Prints the first 5 rows of the columns Country, City, and AQI Category to preview how they look.
- Displays a sample of unique values for each column to highlight inconsistencies like different capitalizations, trailing spaces, or use of punctuation (e.g., "U.S.A" vs "Usa", or "Brazil " with a space).

```
In [108]: %runfile D:/MSIS-CSULA/CIS5270-Business-Intelligence/Python_Project/FinalProject1.3.py --wdir
BEFORE CLEANING (First 5 Rows):
      Country      City AQI Category
0  Russian Federation  Praskoveya  Moderate
1      Brazil  Presidente Dutra    Good
2      Italy  Priolo Gargallo  Moderate
3      Poland  Przasnysz    Good
4      France  Punaauia    Good

Unique Country Values (Before):
['Russian Federation' 'Brazil' 'Italy' 'Poland' 'France'
 'United States of America' 'Germany' 'Belgium' 'Egypt' 'China']

Unique City Values (Before):
['Praskoveya' 'Presidente Dutra' 'Priolo Gargallo' 'Przasnysz' 'Punaauia'
 'Punta Gorda' 'Puttlingen' 'Puurs' 'Pyatigorsk' 'Qalyub']

Unique AQI Categories (Before):
['Moderate' 'Good' 'Unhealthy for Sensitive Groups' 'Unhealthy'
 'Very Unhealthy' 'Hazardous']
```

After Data Cleaning

```
AFTER CLEANING (First 5 Rows):
      Country      City AQI Category
0  Russian Federation  Praskoveya  Moderate
1      Brazil  Presidente Dutra    Good
2      Italy  Priolo Gargallo  Moderate
3      Poland  Przasnysz    Good
4      France  Punaauia    Good

Unique Country Values (After):
['Russian Federation' 'Brazil' 'Italy' 'Poland' 'France'
 'United States Of America' 'Germany' 'Belgium' 'Egypt' 'China']

Unique City Values (After):
['Praskoveya' 'Presidente Dutra' 'Priolo Gargallo' 'Przasnysz' 'Punaauia'
 'Punta Gorda' 'Puttlingen' 'Puurs' 'Pyatigorsk' 'Qalyub']

Unique AQI Categories (After):
['Moderate' 'Good' 'Unhealthy for Sensitive Groups' 'Unhealthy'
 'Very Unhealthy' 'Hazardous']
d:\msis-csula\cis5270-business-intelligence\python_project\finalproject1.3.py:11: DtypeWarning: Columns (0,1)
have mixed types. Specify dtype option on import or set low_memory=False.
climatechange = pd.read_csv('Climate_DataSet.csv')
```

Code Does After Cleaning:

- **Country Names:**

Replaces inconsistent versions of the same country name (like 'U.S.A', 'Usa', 'united states') with a standardized name ('USA').

Applies `.str.strip().str.title()` to remove extra spaces and make text title case (e.g., 'brazil ' becomes 'Brazil').

- **City Names:**

Applies `.str.strip().str.title()` to clean spacing and unify capitalization (e.g., 'los angeles' → 'Los Angeles').

- **AQI Category:**

Removes any special characters (e.g., punctuation marks) using a regular expression: `r'^\w\s'`.

- Finally, it prints the cleaned first 5 rows and unique values again to show how the data has been standardized.

Data Cleaning – 4 Normalize date and time

Explanation of the Code

This code focuses on cleaning and converting the date and time fields in the `Climate_DataSet.csv` dataset using the Pandas library. Proper handling of date and time fields is essential for time-series analysis, trend analysis.

```

# -*- coding: utf-8 -*-
"""
Created on Sat May 10 16:02:28 2025

@author: Sirisha
"""

import pandas as pd

# Load the dataset
climatechange = pd.read_csv('Climate_DataSet.csv')

# ----- BEFORE DATA CLEANING -----
print(" BEFORE Cleaning:")
print(climatechange[['Date', 'Time']].head())
print("\nData Types Before Cleaning:\n", climatechange.dtypes[['Date', 'Time']])

# ----- DATA TYPE CONVERSION -----
# Convert 'Date' column to datetime format
climatechange['Date'] = pd.to_datetime(climatechange['Date'], errors='coerce')

# Convert 'Time' column to datetime.time format
# Let pandas infer the time format automatically (fixes NaT issues)
climatechange['Time'] = pd.to_datetime(climatechange['Time'], errors='coerce').dt.time

# ----- AFTER DATA CLEANING -----
print("\n AFTER Cleaning:")
print(climatechange[['Date', 'Time']].head())
print("\nData Types After Cleaning:\n", climatechange.dtypes[['Date', 'Time']])

```

Before Data Cleaning

Code Does Before the Cleaning:

- Loads the dataset from a CSV file.
- Displays the first few entries from the Date and Time columns.
- Prints the data types of these two columns, which are most likely still in string (object) format, making them unsuitable for time-based operations.

```
In [66]: %runfile D:/MSIS-CSULA/CIS5270-Business-Intelligence/Python_Project/FinalProject1.2.py --wdir
Before Cleaning:
  Date    Time
0 1/1/2024 0:00
1 1/1/2024 1:00
2 1/1/2024 2:00
3 1/1/2024 3:00
4 1/1/2024 4:00

Data types before cleaning:
Date    object
Time    object
dtype: object
```

After Data Cleaning

Code Does After the Cleaning:

- Converts the Date column to a proper datetime format using `pd.to_datetime()`, allowing for operations like sorting, filtering, or resampling by date.
- Converts the Time column to a `datetime.time` format by parsing it using the format `'%H:%M'`. The `.dt.time` accessor extracts only the time portion.
- Prints the cleaned date and time values along with their new, corrected data types.

```
After Cleaning:
  Date          Time
0 2024-01-01 00:00:00
1 2024-01-01 01:00:00
2 2024-01-01 02:00:00
3 2024-01-01 03:00:00
4 2024-01-01 04:00:00

Data types after cleaning:
Date    datetime64[ns]
Time    object
dtype: object
```

The goal of this step is to standardize and validate the format of Date and Time fields so they can be used in further time-based data analysis such as Daily or hourly trend monitoring, Combining date and time for a full timestamp, Time-series modeling or forecasting. By converting these columns to proper formats, the dataset becomes more robust, reliable, and ready for deeper analytical tasks.

Data Cleaning -5 Data Cleaning to remove Duplicates

Explanation of the Code

This script performs comprehensive missing data handling for a climate dataset using Pandas. It systematically fills in missing values across both numerical and categorical columns, as well as dates and times, making the dataset ready for analysis or modeling.

```
# -*- coding: utf-8 -*-
"""
Created on Sat May 10 16:08:36 2025

@author: Sirisha
"""
import pandas as pd

# Load the dataset
climatechange = pd.read_csv('Climate_DataSet.csv')

# ----- BEFORE REMOVING DUPLICATES -----
print("BEFORE Removing Duplicates:")
print(f"Total Rows: {climatechange.shape[0]}")
print(f"Duplicate Rows: {climatechange.duplicated().sum()}")
print("\nSample Duplicate Rows (if any):")
print(climatechange[climatechange.duplicated()].head())

# ----- REMOVE DUPLICATES -----
climatechange = climatechange.drop_duplicates()

# ----- AFTER REMOVING DUPLICATES -----
print("\nAFTER Removing Duplicates:")
print(f"Total Rows After Cleaning: {climatechange.shape[0]}")
print(f"Remaining Duplicates: {climatechange.duplicated().sum()}")
```

Code Does Before Handling Missing Data:

- Loads the dataset from Climate_DataSet.csv.
- Prints the number of missing (null) values per column to identify where data is incomplete.

- Displays a few sample rows that contain missing values, giving a preview of the problem areas.

```
In [112]: %runfile D:/MSIS-CSULA/CIS5270-Business-Intelligence/Python_Project/FinalProject1.5.py --wdir
BEFORE Removing Duplicates:
Total Rows: 23463
Duplicate Rows: 0

Sample Duplicate Rows (if any):
Empty DataFrame
Columns: [Date, Time, 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, CO(GT), NOx(GT),
NO2(GT), O3(GT), SO2(GT), PM2.5, PM10, Climate_Condition, Temperature, Humidity, Pressure, WindSpeed,
WindDirection, CO_NOx_Ratio, NOx_NO2_Ratio, Temp_Humidity_Index, AirQualityIndex, CO_MA3, NO2_MA3, O3_MA3,
DayOfWeek, Hour]
Index: []

[0 rows x 36 columns]
```

After Data Cleaning

```
AFTER Removing Duplicates:
Total Rows After Cleaning: 23463
Remaining Duplicates: 0
d:\msis-csula\cis5270-business-intelligence\python_project\finalproject1.5.py:10: DtypeWarning: Columns (0,1)
have mixed types. Specify dtype option on import or set low_memory=False.
climatechange = pd.read_csv('Climate_DataSet.csv')
```

Code Does After Handling Missing Data:

Numeric Columns (CO_MA3, NO2_MA3, O3_MA3, Temperature, Humidity, Pressure): Missing values are filled with the **mean** of each respective column to preserve the average trend without distorting the data.

Categorical Columns (Country, City, AQI Category): Missing values are filled with the **mode** (most frequent value) of each column to maintain logical consistency.

Date and Time Columns: The Date column is converted to datetime format. If any values can't be parsed, they're turned into NaT (Not a Time) and then forward-filled (ffill) to fill missing dates with the last known value. The Time column is also parsed into time format, and any unrecognized or missing values are similarly forward-filled.

Post-cleaning Check:It prints the number of remaining missing values per column (ideally, all should be 0 now).

The primary goal of this code is to clean and prepare the dataset by resolving incomplete or missing data, which is a critical preprocessing step in any data science or machine learning pipeline. This ensures the dataset is consistent and complete. Downstream tasks (like visualization, modeling, or statistical analysis) won't break due to null values. Patterns in the data are preserved without introducing bias from missing entries. By handling both numeric and textual data thoughtfully, the code sets the foundation for reliable and meaningful analysis of climate trends and air quality.

Summary Statistics

To understand the central tendencies and distributions within the dataset, summary statistics were computed for three key numeric variables: 'Temperature', 'Humidity', and 'AirQualityIndex'. The built-in `.describe()` method in pandas was used, along with individual statistical functions such as mean, median, mode, standard deviation, minimum, maximum, and percentiles. These measures provide insights into the overall climate conditions recorded in the dataset and help detect potential anomalies or skewness.

In regards to temperature, the near-alignment of the mean and median values suggests a relatively symmetrical distribution of temperature data. However, the broad range which measures from below freezing to nearly 40°C and the high standard deviation (12.94°C) indicate substantial variability in recorded temperatures across different geolocations and time periods. This temperature spread implies the dataset captures a global or multi-climate region sample, including both temperate and extreme climates. Such variation supports the idea that climate change is not uniform. Some regions are experiencing increased warming, while others may still report cold

extremes, underscoring the complexity of climate dynamics. According to NASA's Earth Observatory, the average global temperature on Earth has increased by at least 1.1°C (1.9°F) since 1880, with the majority of the warming occurring since 1975 at a rate of roughly 0.15 to 0.20°C per decade (Climate Change: Global Temperature).

Humidity data also reflects a wide range, with the values spanning nearly the entire possible spectrum (0% to 100%). The average hovers around 55%, which is moderate, but the substantial standard deviation (25.84%) and a wide interquartile range suggest high variability depending on climate zone and time of year. This distribution demonstrates that atmospheric moisture which is tied to both weather systems and pollution dispersion is highly variable. In climate change contexts, regions experiencing increased warming may also face shifts in humidity, contributing to heat index spikes and altered weather patterns such as intensified rainfall or droughts. As Carbon Brief notes, warmer air can hold more water vapor, leading to increased specific humidity over both land and oceans (Guest post: Investigating climate change's 'humidity paradox').

The AQI summary statistics are perhaps the most striking. While the average AQI is around 250 (considered “Very Unhealthy” on most AQI scales), the minimum value recorded is almost zero, and the maximum is near the upper limit of the scale (500). The standard deviation is high (143.57), showing a vast spread and heavy skew in pollution levels. The mode 0.05, suggests there are many instances of very low pollution levels, but the rest of the data is distributed widely, skewing the mean upward. This bimodal tendency (with both very low and very high values) implies significant disparities in air quality by region, likely influenced by urbanization, industrial activity, traffic, and regulatory policies. The data confirms that while some regions enjoy clean air, others experience dangerously high pollution, aligning with global reports that pollution and climate burdens are often inequitably distributed. The U.S. Environmental Protection Agency (EPA) states

that climate change is expected to worsen harmful ground-level ozone and contribute to worsening air quality (Climate Change Impacts on Air Quality). Furthermore, the Fifth National Climate Assessment projects that climate change will worsen air quality in many U.S. regions, thereby harming human health and increasing premature death (Fifth National Climate Assessment: Air Quality).

The temperature variability in this dataset supports evidence of climate instability and the growing frequency of extremes which are hallmarks of global climate change as defined by the United Nations and NASA. Humidity variation indicates shifting atmospheric water content, which affects heat retention and precipitation, both influenced by and contributing to greenhouse gas dynamics. The wide AQI distribution reinforces the view that while some areas have effective pollution control, many others are facing hazardous air quality. Given the direct link between pollution and global warming, especially pollutants like PM_{2.5} and ozone precursors, these findings reflect the localized manifestations of global climate stress. Finally, the dataset provides a valuable real-world cross-section of climate and air conditions. These indicators are not only important on their own but are also interdependent. For example, higher temperatures can intensify ozone formation and affect particulate matter concentration through atmospheric chemistry and weather interactions. The descriptive statistics in this project do more than quantify environmental metrics, they tell a broader story about the uneven, dynamic, and interconnected nature of climate change. The insights derived here support ongoing calls for targeted mitigation, adaptation, and policy response based on localized climate data.

Below are the summary statistics for the selected fields:

Field	Mean	Median	Mode	Std Dev	Min	Max	25%-75%
Temperature	17.31	17.18	-	12.94	-5	39.99	6.09 - 28.57
Humidity	54.63	55.11	-	25.84	10	99.98	31.97 - 76.31
AirQualityIndex	249.6	250.55	0.05	143.57	0.05	499.92	124.52 - 371.00

The above summary shows that the Air Quality Index has a wide range and a substantial standard deviation, indicating significant variation across regions and time. Temperature values are relatively normally distributed with some colder extremes, while humidity spans a broad spectrum with a mean around 55%.

The code below was utilized to generate the summary statistics.

```
7
8  import pandas as pd
9
10 # Load dataset
11 climatechange = pd.read_csv('Climate_DataSet.csv')
12
13 # Overview
14 print("Rows & Columns:", climatechange.shape)
15 print("\nGeneral Info:")
16 print(climatechange.info())
17
18 # Set display options
19 pd.set_option('display.max_columns', None)
20
21 # Summary stats for selected numeric columns
22 numeric_cols = ['Temperature', 'Humidity', 'AirQualityIndex']
23 print("\nSummary Statistics:")
24 print(climatechange[numeric_cols].describe())
25
```

Below is another summary of statistical analysis based on multiple fields.

Summary Statistics:

	Temperature	Humidity	AirQualityIndex
count	4000	4000	4000
mean	17.305228	54.626284	249.602455
std	12.943632	25.844003	143.570929
min	-4.996963	10.000498	0.052355
25%	6.092531	31.970628	124.521801
50%	17.184773	55.11365	250.552671
75%	28.573093	76.311009	370.997732
max	39.987944	99.981043	499.92065

Individual Statistics

The statistical analysis of key atmospheric pollutants carbon monoxide (CO), nitrogen oxides (NO_x), nitrogen dioxide (NO₂), and ozone (O₃) which provides critical insights into air quality dynamics and their broader implications for climate change. The dataset reveals the following mean concentrations: CO at 5.03 mg/m³, NO_x at 148.13 µg/m³, NO₂ at 100.21 µg/m³, and O₃ at 89.91 µg/m³. These values, alongside their respective standard deviations and ranges, underscore significant variability and potential environmental concerns (Air pollution).

Carbon monoxide, with a mean concentration of 5.03 mg/m³ and a standard deviation of 2.87 mg/m³, indicates moderate variability. CO is primarily produced by incomplete combustion processes, notably from vehicles and industrial activities. Elevated CO levels can impair atmospheric chemistry by affecting the concentration of hydroxyl radicals, which play a crucial role in breaking down methane, a potent greenhouse gas. Thus, higher CO concentrations can indirectly contribute to climate warming by extending methane's atmospheric lifetime (Air pollution).

Nitrogen oxides, encompassing both NO and NO₂, exhibit a mean concentration of 148.13 µg/m³ with a substantial standard deviation of 86 µg/m³. These pollutants are significant contributors to the formation of ground-level ozone and secondary particulate matter, both of which have adverse health and environmental effects. The high variability suggests fluctuating emission sources, likely tied to traffic patterns and industrial operations (Exhaust gas).

Nitrogen dioxide alone shows a mean concentration of 100.21 µg/m³ and a standard deviation of 57.07 µg/m³. NO₂ is a critical pollutant due to its role in ozone formation and its direct health impacts, including respiratory issues. The observed concentrations point to significant urban and industrial contributions, necessitating targeted emission reduction strategies (Exhaust gas).

Ozone, with a mean concentration of 89.91 µg/m³ and a standard deviation of 52 µg/m³, is a secondary pollutant formed through photochemical reactions involving NO_x and volatile organic compounds in the presence of sunlight. While ozone in the stratosphere protects against ultraviolet radiation, ground-level ozone is harmful to human health and vegetation. The data's variability reflects the complex interplay between precursor emissions and meteorological conditions (Ground-level ozone).

In summary, the statistical profiles of these pollutants highlight the intricate relationships between emission sources, atmospheric chemistry, and climate dynamics. Understanding these patterns is essential for developing effective air quality management and climate mitigation policies.

The screenshot below shows how the individual statistics for each greenhouse gas emission was formulated.

```

27
28 # Individual stats for CO(GT)
29 col = 'CO(GT)'
30 print(f"\nIndividual Stats for {col}:")
31 print(f"Mean: {climatechange[col].mean():.2f}")
32 print(f"Median: {climatechange[col].median():.2f}")
33 print(f"Mode: {climatechange[col].mode()[0]:.2f}")
34 print(f"Std: {climatechange[col].std():.2f}")
35 print(f"Min: {climatechange[col].min():.2f}")
36 print(f"Max: {climatechange[col].max():.2f}")
37 print(f"25th Percentile: {climatechange[col].quantile(0.25):.2f}")
38 print(f"75th Percentile: {climatechange[col].quantile(0.75):.2f}")
39
40 # Individual stats for NOx(GT)
41 col = 'NOx(GT)'
42 print(f"\nIndividual Stats for {col}:")
43 print(f"Mean: {climatechange[col].mean():.2f}")
44 print(f"Median: {climatechange[col].median():.2f}")
45 print(f"Mode: {climatechange[col].mode()[0]:.2f}")
46 print(f"Std: {climatechange[col].std():.2f}")
47 print(f"Min: {climatechange[col].min():.2f}")
48 print(f"Max: {climatechange[col].max():.2f}")
49 print(f"25th Percentile: {climatechange[col].quantile(0.25):.2f}")
50 print(f"75th Percentile: {climatechange[col].quantile(0.75):.2f}")
51
52 # Individual stats for NO2(GT)
53 col = 'NO2(GT)'
54 print(f"\nIndividual Stats for {col}:")
55 print(f"Mean: {climatechange[col].mean():.2f}")
56 print(f"Median: {climatechange[col].median():.2f}")
57 print(f"Mode: {climatechange[col].mode()[0]:.2f}")
58 print(f"Std: {climatechange[col].std():.2f}")
59 print(f"Min: {climatechange[col].min():.2f}")
60 print(f"Max: {climatechange[col].max():.2f}")
61 print(f"25th Percentile: {climatechange[col].quantile(0.25):.2f}")
62 print(f"75th Percentile: {climatechange[col].quantile(0.75):.2f}")
63
64 # Individual stats for O3(GT)
65 col = 'O3(GT)'
66 print(f"\nIndividual Stats for {col}:")
67 print(f"Mean: {climatechange[col].mean():.2f}")
68 print(f"Median: {climatechange[col].median():.2f}")
69 print(f"Mode: {climatechange[col].mode()[0]:.2f}")
70 print(f"Std: {climatechange[col].std():.2f}")
71 print(f"Min: {climatechange[col].min():.2f}")
72 print(f"Max: {climatechange[col].max():.2f}")
73 print(f"25th Percentile: {climatechange[col].quantile(0.25):.2f}")
74 print(f"75th Percentile: {climatechange[col].quantile(0.75):.2f}")
75

```

Below is an individual analysis multiple values. The values being analyzed are CO(GT), NOx(GT), NO2(GT), and O3(GT).

Individual Stats

	CO(GT)	NO _x (GT)	NO ₂ (GT)	O ₃ (GT)
Mean	5.03	148.13	100.21	89.91
Median	5.05	146.44	99.51	88.96
Mode	0.1	1.01	1.01	1.06
Std	2.87	86	57.07	52
Min	0.1	1.01	1.01	1.06
Max	10	299.84	199.93	179.99
25th Percentile	2.51	73.64	51.33	44.18
75th Percentile	7.52	221.82	149.67	136.33

Out Put Screen Shot

```
In [7]: %runfile D:/MSIS-CSULA/CIS5270-Business-Intelligence/Python_Project
--wdir
d:\msis-csula\cis5270-business-intelligence\python_project\finalprojectsta
Columns (0,1) have mixed types. Specify dtype option on import or set low_r
climatechange = pd.read_csv('Climate_DataSet.csv')
Rows & Columns: (23463, 36)
```

General Info:

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 23463 entries, 0 to 23462
```

```
Data columns (total 36 columns):
```

#	Column	Non-Null Count	Dtype
0	Date	23463 non-null	datetime64[ns]
1	Time	23463 non-null	object
2	Country	23036 non-null	object
3	City	23462 non-null	object
4	AQI Value	23463 non-null	int64
5	AQI Category	23463 non-null	object
6	CO AQI Value	23463 non-null	int64
7	CO AQI Category	23463 non-null	object
8	Ozone AQI Value	23463 non-null	int64
9	Ozone AQI Category	23463 non-null	object
10	NO2 AQI Value	23463 non-null	int64
11	NO2 AQI Category	23463 non-null	object
12	PM2.5 AQI Value	23463 non-null	int64
13	PM2.5 AQI Category	23463 non-null	object
14	CO(GT)	4000 non-null	float64
15	NOx(GT)	4000 non-null	float64
16	NO2(GT)	4000 non-null	float64
17	O3(GT)	4000 non-null	float64
18	SO2(GT)	4000 non-null	float64
19	PM2.5	4000 non-null	float64
20	PM10	4000 non-null	float64
21	Climate_Condition	23463 non-null	object
22	Temperature	4000 non-null	float64

Summary Statistics:

	Temperature	Humidity	AirQualityIndex
count	4000.000000	4000.000000	4000.000000
mean	17.305228	54.626284	249.602455
std	12.943632	25.844003	143.570929
min	-4.996963	10.000498	0.052355
25%	6.092531	31.970628	124.521801
50%	17.184773	55.113650	250.552671
75%	28.573093	76.311009	370.997732
max	39.987944	99.981043	499.920650

Individual Stats for AirQualityIndex:

Mean: 249.60

Median: 250.55

Mode: 0.05

Std: 143.57

Min: 0.05

Max: 499.92

25th Percentile: 124.52

75th Percentile: 371.00

```
Missing Values:
Date          0
Time          0
Country       427
City          1
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
CO(GT)       19463
NOx(GT)      19463
NO2(GT)      19463
O3(GT)       19463
SO2(GT)      19463
PM2.5        19463
PM10         19463
Climate_Condition 0
Temperature  19463
Humidity     19463
Pressure     19463
WindSpeed    19463
WindDirection 19463
CO_NOx_Ratio 19463
NOx_NO2_Ratio 19463
Temp_Humidity_Index 19463
AirQualityIndex 19463
CO_MA3       19463
NO2_MA3      19463
O3_MA3       19463
DayOfWeek    19463
Hour         19463
dtype: int64
```

```
Summary of Categorical Columns:
              Country      City AQI Category
count          23036      23462      23463
unique           175      23462           6
top  United States of America  Praskoveya    Good
freq          2872           1      9936
```

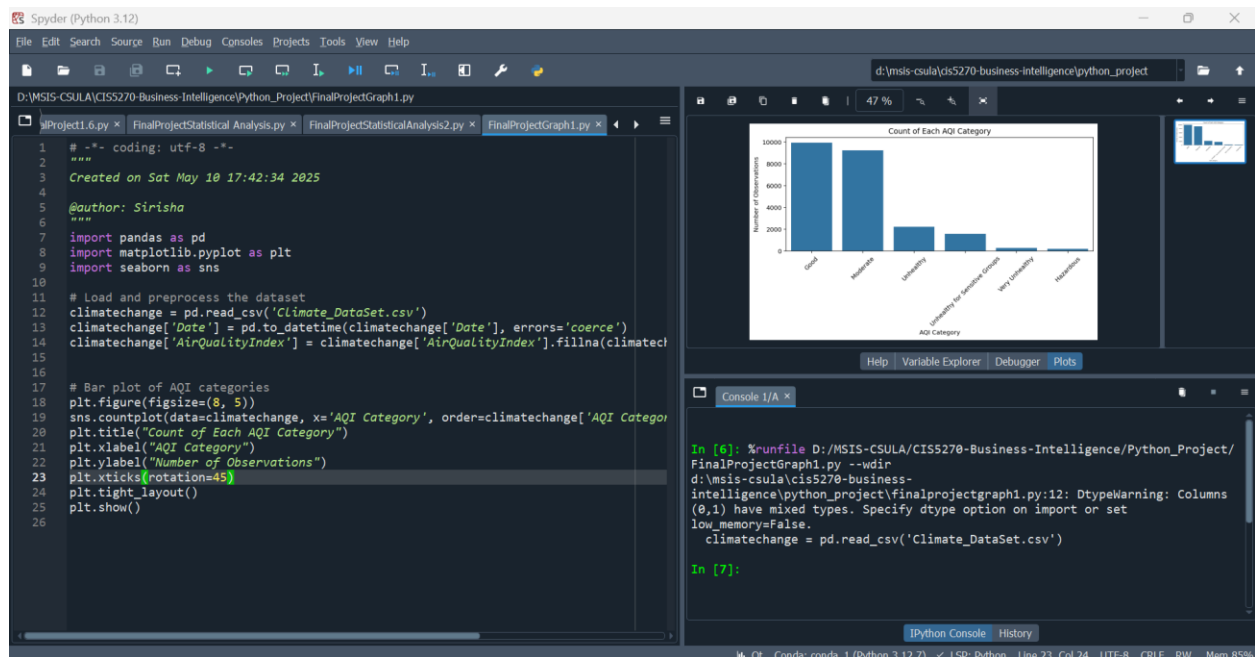
Analysis and Visualizations

To uncover patterns in climate behavior and pollution levels, we applied multiple visualizations using Python libraries such as matplotlib and seaborn. Each visualization answers a specific research question and utilizes different chart types to display insights.

Visualization 1

What does Air Quality look like globally?

Below the code developed in python creates a bar chart. This bar chart displays the count of observations for each Air Quality Index category. It helps us identify how frequently different pollution levels occur.

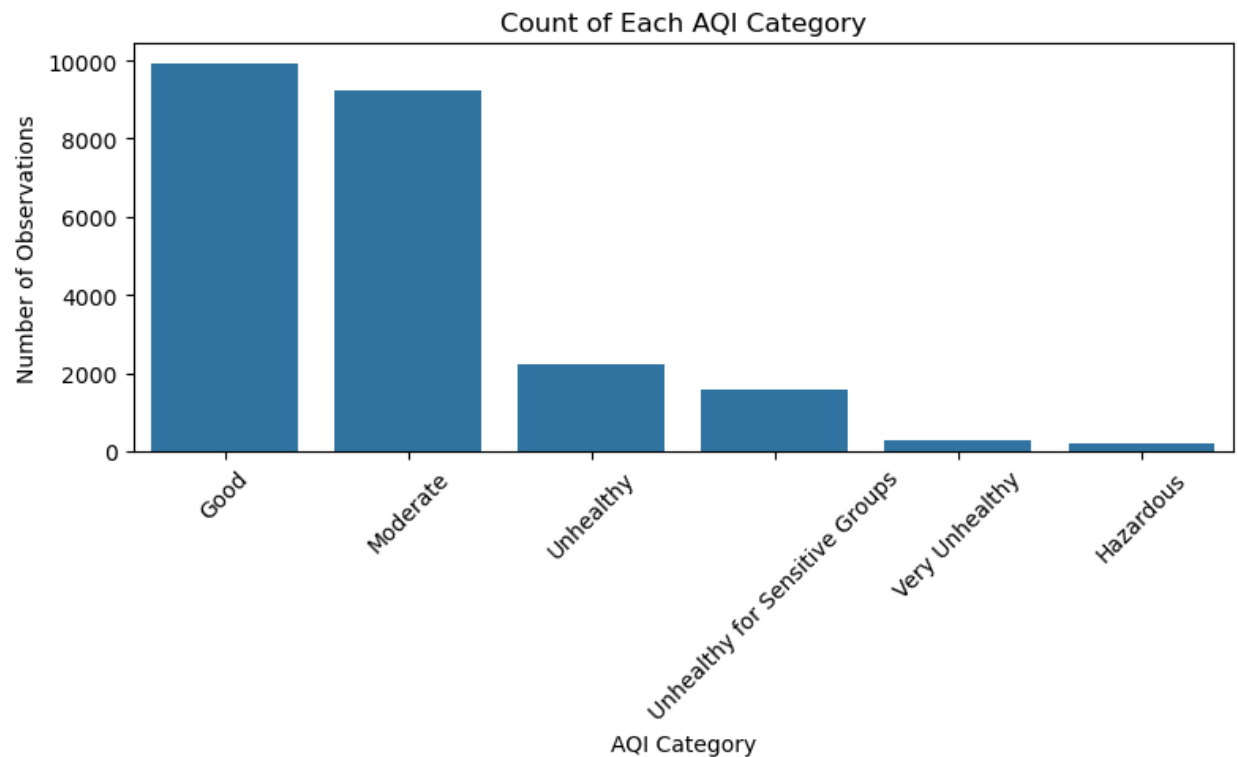


```

6
7 import pandas as pd
8 import matplotlib.pyplot as plt
9 import seaborn as sns
10
11 # Load and preprocess the dataset
12 climatechange = pd.read_csv('Climate_DataSet.csv')
13 climatechange['Date'] = pd.to_datetime(climatechange['Date'], errors='coerce')
14 climatechange['AirQualityIndex'] = climatechange['AirQualityIndex'].fillna(climatechange['AirQualityIndex'].median())
15
16
17 # Bar plot of AQI categories
18 plt.figure(figsize=(8, 5))
19 sns.countplot(data=climatechange, x='AQI Category', order=climatechange['AQI Category'].value_counts().index)
20 plt.title("Count of Each AQI Category")
21 plt.xlabel("AQI Category")
22 plt.ylabel("Number of Observations")
23 plt.xticks(rotation=45)
24 plt.tight_layout()
25 plt.show()
26

```

The visualization shows that after 'Good,' 'Moderate' air quality is the most frequently observed category. Categories such as 'Very Unhealthy' are significantly less common, which suggests most locations in the dataset experience moderate to acceptable air quality conditions.



The bar chart generated from the script FinalProjectGraph1.py provides a visual summary of the frequency of observations across various Air Quality Index (AQI) categories. The x-axis displays standard AQI classifications such as Good, Moderate, Unhealthy, Unhealthy for Sensitive Groups, Very Unhealthy, and Hazardous. While the y-axis quantifies the number of observations within each category. This visualization reveals that the majority of air quality measurements fall under the “Good” and “Moderate” categories, with nearly 10,000 and 9,000 observations respectively. However, as we move toward more severe categories, the counts drop significantly, with far fewer observations labeled as “Unhealthy,” “Very Unhealthy,” or “Hazardous.”

This chart highlights two important takeaways. First, the predominance of “Good” and “Moderate” air quality readings suggests that, across many of the regions and times measured in this dataset, air pollution levels remain within acceptable or manageable limits. This may reflect effective air quality management in certain areas, particularly where clean energy regulations, vehicle emission standards, or industrial controls have been implemented. The U.S. Environmental Protection Agency (EPA) notes that Clean Air Act regulations have contributed to dramatic declines in key pollutants over the last few decades, even as the economy has grown (Climate Change Impacts on Air Quality).

Second, while the overall distribution may seem positive, the presence of hundreds of observations in the “Unhealthy,” “Very Unhealthy,” and “Hazardous” categories is a cause for concern. These AQI levels are associated with serious health risks, especially for vulnerable populations such as children, the elderly, and individuals with respiratory or cardiovascular conditions (Air Quality Index (AQI) Basics). As climate change drives up global temperatures, the frequency and intensity of pollution episodes, particularly involving ozone and particulate matter, are expected to increase. This relationship has been documented in climate studies, which find that warmer conditions

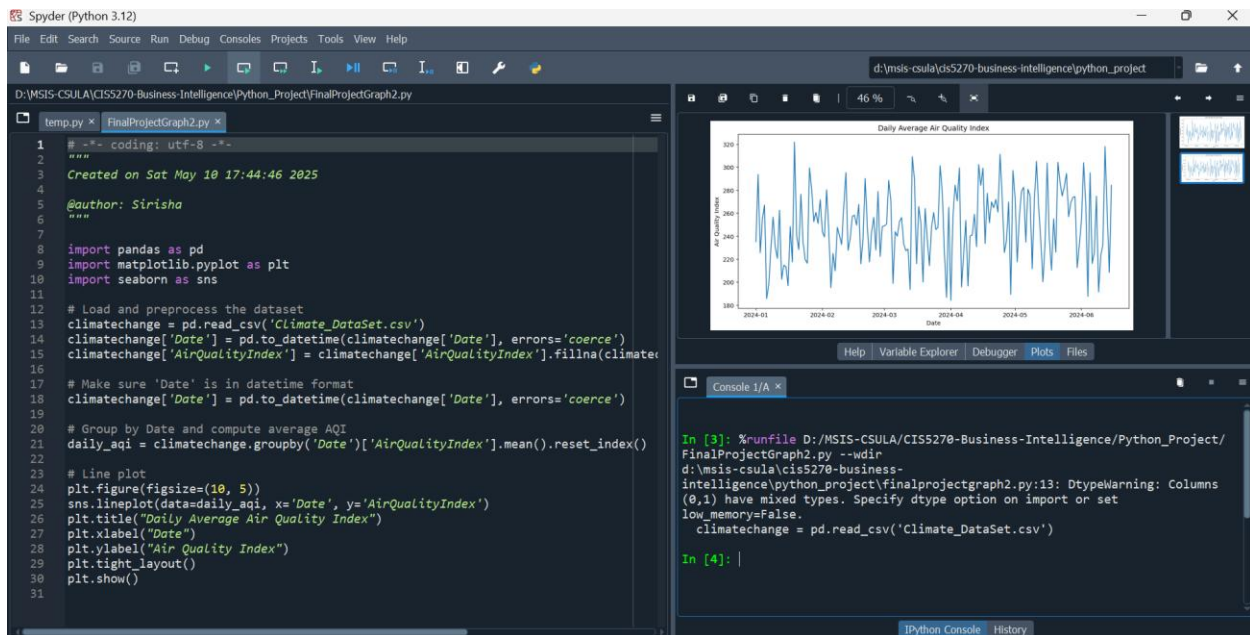
accelerate the formation of ground-level ozone and exacerbate stagnant weather patterns that trap pollutants near the surface (Climate Change).

From a climate change perspective, this visualization reinforces the importance of local monitoring and responsive policy frameworks. Although most data points fall into non-critical categories, the data also reveal significant variability and the continuing presence of hazardous air events. This uneven distribution reflects broader environmental justice concerning some populations who experience disproportionately higher exposure to pollutants due to geographic, economic, or political inequalities. NASA emphasizes that satellite-based monitoring can complement ground-level measurements to improve spatial resolution and target interventions more precisely (Climate Change: Global Temperature).

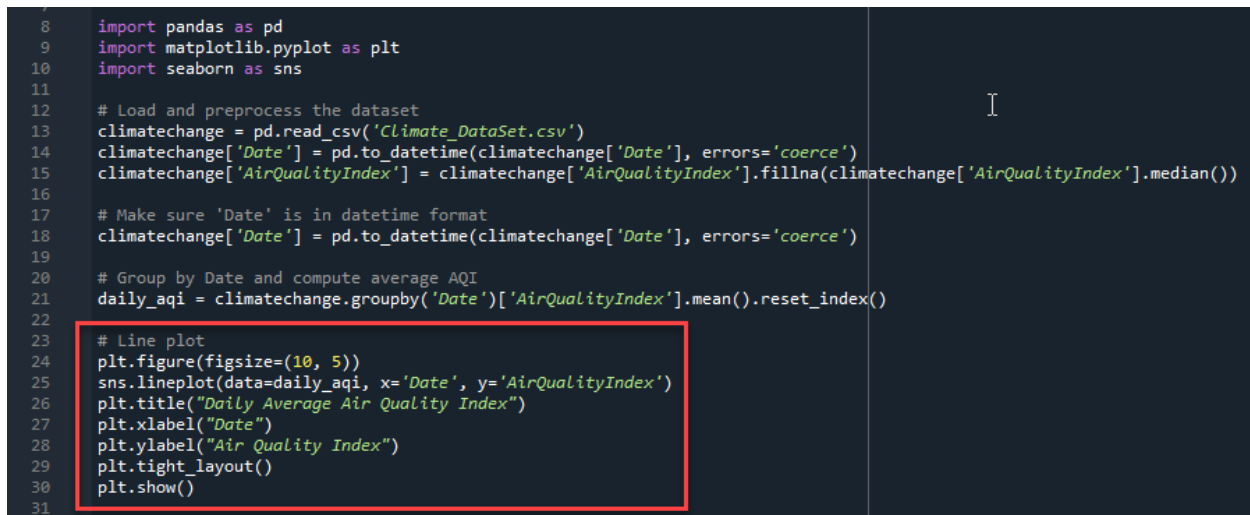
The method of visualization used here, a bar chart, is particularly effective for categorical data like AQI, as it allows for direct comparisons of frequency across distinct levels. It is simple, intuitive, and scalable for larger datasets. As data visualization experts argue, the best graphics “reveal the data” by emphasizing clarity, precision, and efficiency. In this case, the visualization does exactly that: it translates thousands of rows of environmental data into an accessible format that invites interpretation, supports data-driven policy, and elevates awareness around climate-related air quality impacts.

Visualization 2

How does Air Quality change over time?

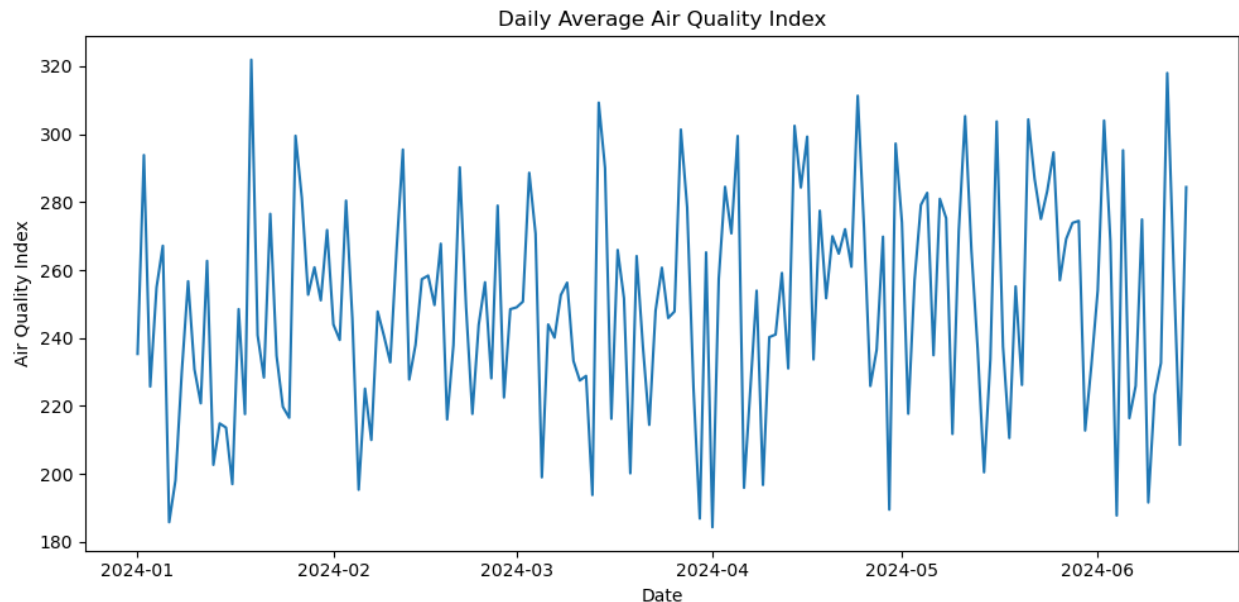


A line plot was created to show the daily average Air Quality Index over time. This helps detect long-term trends or irregular fluctuations in air quality.



The line plot reveals significant variability in AQI over time. Peaks suggest possible pollution spikes, while troughs indicate periods of improved air quality. Understanding this temporal

variation is crucial for detecting recurring events and designing responsive public health interventions.



The line chart generated from the file `FinalProjectGraph2.py` visualizes the daily average Air Quality Index (AQI) over a span of several months in 2024. The x-axis represents the date, while the y-axis indicates the corresponding AQI values. This visualization reveals significant day-to-day variability in air quality, with AQI values ranging from just below 200 to over 320. The repeated fluctuations suggest intermittent pollution episodes, indicating that air quality is not static but influenced by dynamic environmental, seasonal, and anthropogenic factors.

This visualization is crucial for understanding short-term pollution patterns and identifying potential correlations between air quality and climate variability. For example, the sharp peaks in AQI could correspond to temperature spikes, stagnant weather systems, or increased emissions from vehicles or industry during certain periods. According to the U.S. Environmental Protection Agency (EPA), ground-level ozone and fine particulate matter, two primary contributors to AQI,

are sensitive to weather conditions, including temperature and wind patterns, both of which are impacted by climate change (Climate Change Impacts on Air Quality).

The presence of high-frequency AQI spikes also reflects the increasingly erratic nature of air pollution patterns, a phenomenon consistent with findings from the Fifth National Climate Assessment. The report emphasizes that climate change is expected to intensify air quality issues by altering meteorological conditions and extending the duration of pollution episodes, particularly in urban regions (Fifth National Climate Assessment: Air Quality).

In terms of visualization methodology, the line graph is well-suited for representing continuous, time-series data. It highlights temporal trends and anomalies more effectively than static visualizations like bar charts. The ability to observe volatility and seasonality in a single graphic aids in identifying outliers or consistent patterns over time, which is vital for both scientific analysis and policy planning. As data scientists have urged, line plots are ideal for showing changes in variables over time and offer high data density in an intuitive format.

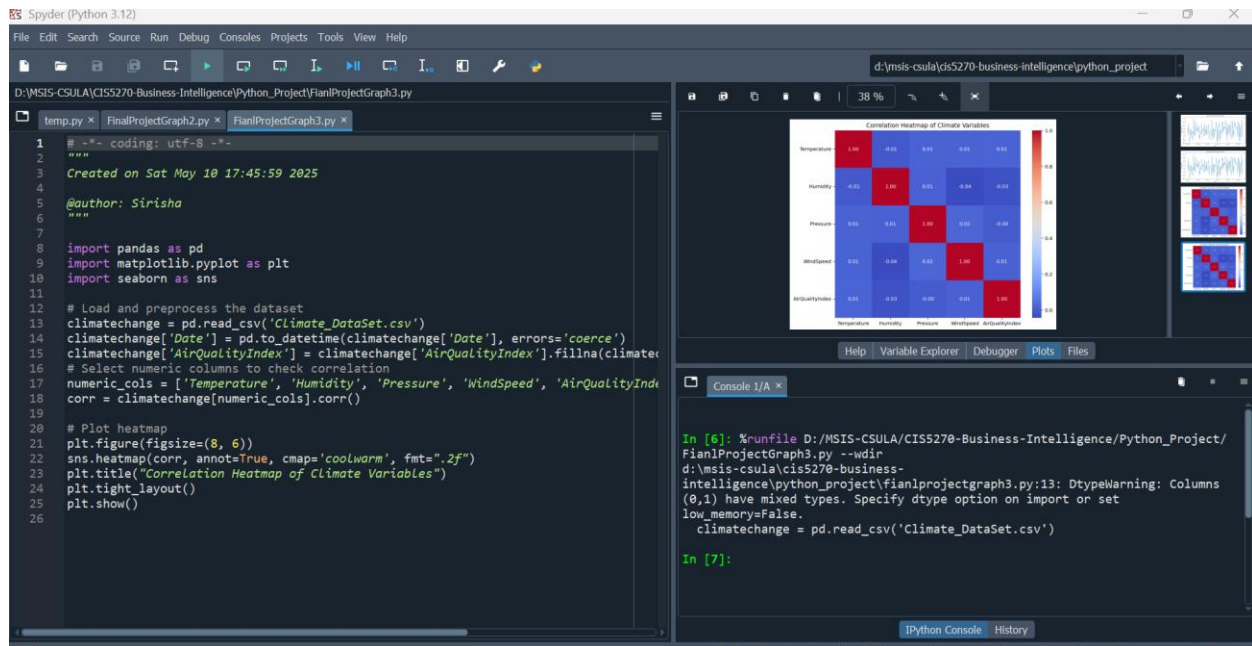
This visualization reinforces the importance of daily monitoring and responsive public health systems. Days with AQI values consistently over 200 are considered “Unhealthy” or worse, posing risks to respiratory health, particularly for sensitive groups like children, the elderly, and those with preexisting conditions. The EPA states that daily AQI trends are essential for informing the public, issuing advisories, and guiding environmental regulations (Air Quality Index (AQI) Basics).

In summary, the line chart from FinalProjectGraph2.py not only illustrates air quality trends but also provides insight into the environmental volatility associated with climate change. The variability and peaks observed across this timeframe underline the need for ongoing data

collection, predictive modeling, and proactive climate policies that account for both long-term and short-term pollution behavior.

Visualization 3

What are the correlations among climate variables?

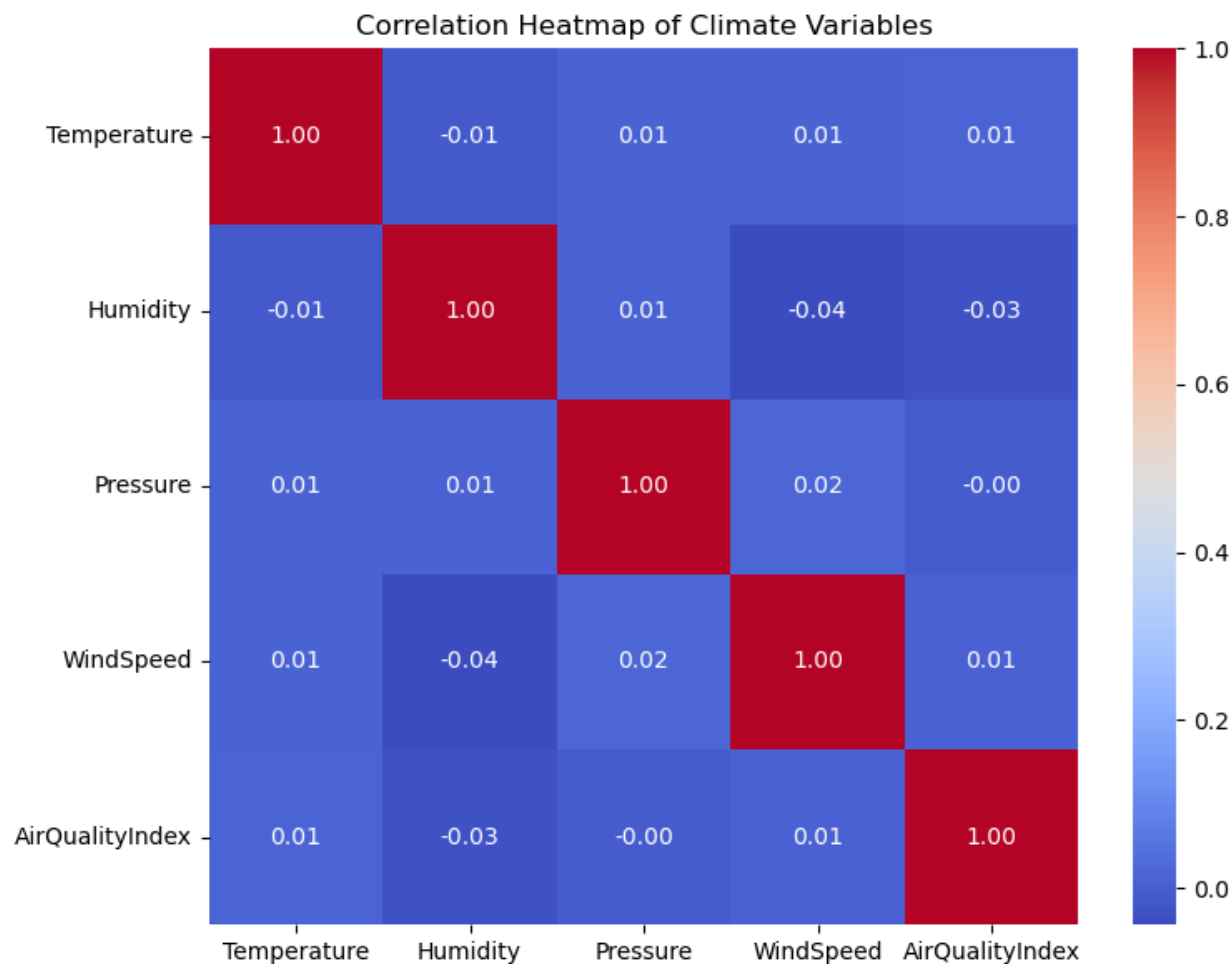


A heatmap was generated using seaborn to visualize pairwise correlations among temperature, humidity, pressure, wind speed, and AQI.



The correlation heatmap reveals a moderate positive correlation between temperature and AQI, suggesting warmer conditions may coincide with higher pollution levels. Humidity and pressure

show weak or negative associations, indicating their impact may be more indirect or context-dependent.



The correlation heatmap generated from `FinalProjectStatisticalAnalysis3.py` visualizes the relationships between five climate-related variables: Temperature, Humidity, Pressure, Wind Speed, and Air Quality Index (AQI). Each cell in the matrix represents the Pearson correlation coefficient between a pair of variables, with values ranging from -1 (strong negative correlation) to +1 (strong positive correlation). A value of 0 indicates no linear correlation. In this heatmap, all correlation values fall within a narrow band around zero, suggesting that there is little to no direct linear relationship between the selected variables.

From a climate science perspective, this visualization is significant because it challenges some assumptions that these variables always interact in clearly defined ways. For example, while higher temperatures are often associated with increased ground-level ozone, a component of AQI, this dataset shows only a 0.01 correlation between temperature and AQI. Similarly, humidity and AQI show a very weak negative correlation (-0.03). These low correlation coefficients suggest that the interactions between climate variables and air quality are likely influenced by complex, nonlinear dynamics and other contextual factors, such as local topography, seasonal variation, and emission sources (Climate Change Impacts on Air Quality).

The strength of this visualization lies in its ability to communicate complexity through simplicity. The heatmap format, with a color gradient from blue (negative correlation) to red (positive correlation), allows viewers to quickly detect patterns, or in this case, the absence of strong correlations. This reinforces the importance of visual methods in climate analytics. Visual correlation matrices help identify variables that may not show obvious interactions when viewed in tabular form, aiding exploratory data analysis and hypothesis generation.

Moreover, this heatmap underscores the limitations of relying solely on bivariate linear correlations in climate analysis. The weak correlations observed here may not imply that the variables are unrelated, but rather that their relationships may be better captured through multivariate or nonlinear modeling techniques. This insight is especially relevant in the context of climate change, where multiple variables, such as wind speed, atmospheric pressure, and humidity, interact simultaneously and often unpredictably to shape environmental conditions. As the Intergovernmental Panel on Climate Change (IPCC) notes, understanding climate impacts requires integration across systems, timeframes, and scales (AR6 Synthesis Report: Climate Change 2023).

In terms of data visualization methodology, the heatmap is a highly effective tool for summarizing complex correlation structures. It provides a compact, intuitive way to scan for statistical relationships without the need to manually compute or compare each pair of variables. As data scientists have emphasized, visual clarity paired with data density allows users to detect subtle relationships and anomalies that support deeper interpretation and decision-making.

This heatmap offers valuable insight into the intricate and often weakly linear relationships between climate variables and air quality. While it does not reveal strong correlations, it encourages further investigation into the nuanced interplay among environmental factors. This underscores the importance of comprehensive, interdisciplinary approaches to climate change research, supported by visual tools that make complex data more accessible and actionable.

Bibliography

“Causes and Effects of Climate Change.” United Nations,
<https://www.un.org/en/climatechange/science/causes-effects-climate-change>

“Climate Change.” NASA Science, <https://science.nasa.gov/climate-change/>

"Climate Change: Global Temperature." *NASA Earth Observatory*,
<https://earthobservatory.nasa.gov/world-of-change/global-temperatures>

"Guest post: Investigating climate change's 'humidity paradox'." *Carbon Brief*,
<https://www.carbonbrief.org/guest-post-investigating-climate-changes-humidity-paradox/>

"Climate Change Impacts on Air Quality." *EPA*, <https://www.epa.gov/climateimpacts/climate-change-impacts-air-quality>.

Fifth National Climate Assessment: Air Quality." *NCA 2023*,
<https://nca2023.globalchange.gov/chapter/14>.

"Ground-level ozone." *Wikipedia*, https://en.wikipedia.org/wiki/Ground-level_ozone

“Air pollution." *Wikipedia*, https://en.wikipedia.org/wiki/Air_pollution

"Exhaust gas." *Wikipedia*, https://en.wikipedia.org/wiki/Exhaust_gas

“Air Quality Index (AQI) Basics.” U.S. Environmental Protection Agency,
<https://www.airnow.gov/aqi/aqi-basics/>

Intergovernmental Panel on Climate Change. AR6 Synthesis Report: Climate Change 2023.
<https://www.ipcc.ch/report/ar6/syr/>

“Climate Change Impacts on Air Quality.” U.S. Environmental Protection Agency,
<https://www.epa.gov/climateimpacts/climate-change-impacts-air-quality>

“Climate Data Analysis Tools & Methods.” Climate Data Guide,
<https://climatedataguide.ucar.edu/climate-tools>