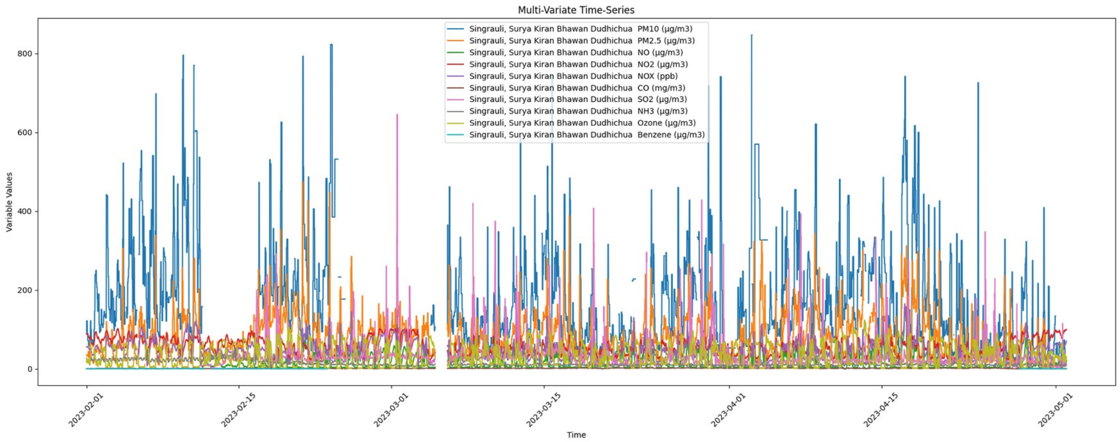
ASSIGNMENT

This is the code to plot the raw multivariate time series by importing the data through the csv file

```
import pandas as pd
import matplotlib.pyplot as plt
# Import data from CSV
df = pd.read_csv('/content/Open pit blasting 01-02-2023 000000 To 01-05-2023 235959.csv')

# Handle missing values
df.replace('NA', pd.NA, inplace=True)

# Exclude rows where "From" value is "Min"
df = df[df['From'] != 'Min']

# Delete the last 3 rows
df = df.iloc[:-3]

# Convert date or time columns to datetime format
df['From'] = pd.to_datetime(df['From'])
df['To (Interval: 15M)'] = pd.to_datetime(df['To (Interval: 15M)'])

# Plotting the time-series data
plt.figure(figsize=(20, 8))

# Plot each variable
for column in df.columns[3:]:
    plt.plot(df['From'], df[column], label=column)

# Set plot title and labels
plt.title('Multi-Variate Time-Series')
plt.xlabel('Time')
plt.ylabel('Variable Values')

# Show a legend for the variables
plt.legend()

# Rotate x-axis tick labels for better readability
plt.xticks(rotation=45)
plt.tight_layout()
# Display the plot
plt.show()
```



As there are multiple variable and I have plotted them all in single plot so I implemented different line styles or colors to differentiate between the lines.

CHECKING FOR MISSING VALUES

```
import pandas as pd

# Read the data from a CSV file
# Read the dataset
data = pd.read_csv('/content/Open pit blasting 01-02-2023 000000 To 01-05-2023 235959.csv')
data = data.iloc[:-3]

# Clean column names
data.columns = data.columns.str.replace('[^a-zA-Z0-9]', '')

# Convert date columns to datetime format
data['From'] = pd.to_datetime(data['From'])
data['ToInterval15M'] = pd.to_datetime(data['ToInterval15M'])



# Check for missing values
missing_values = data.isnull().sum()
print(missing_values)
```

```
                                                   0
From                                               0
ToInterval15M                                      0
SingrauliSuryaKiranBhawanDudhichuaPM10gm3       1681
SingrauliSuryaKiranBhawanDudhichuaPM25gm3        226
SingrauliSuryaKiranBhawanDudhichuaNOgm3         1369
SingrauliSuryaKiranBhawanDudhichuaNO2gm3         416
SingrauliSuryaKiranBhawanDudhichuaNOXppb         415
SingrauliSuryaKiranBhawanDudhichuaCOmgm3         496
SingrauliSuryaKiranBhawanDudhichuaSO2gm3        1451
```

```
SingrauliSuryaKiranBhawanDudhichuaNH3gm3         326
SingrauliSuryaKiranBhawanDudhichuaOzonegm3       453
SingrauliSuryaKiranBhawanDudhichuaBenzenegm3    6195
dtype: int64
```

```
<ipython-input-43-9b2f7fdde57a>:9: FutureWarning: The default value of
regex will change from True to False in a future version.
  data.columns = data.columns.str.replace('[^a-zA-Z0-9]', '')
```

As we can see that dataset has NA values in data which can also be seen in the plot. We will have to implement techniques to tackle it.

Replacing NA values with 0 is a valid strategy in certain scenarios, but it depends on the nature of your data and the analysis you are performing. Here are a few considerations:

Impact on data analysis: Replacing NA values with 0 may introduce bias or distort the patterns in the data, especially if the missing values have a specific meaning or are related to the variable's behavior. It's important to carefully consider the implications of replacing missing values with 0 and how it might affect your analysis results.

Type of missing data: Not all missing data is the same. There are different types of missing data, such as missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). The type of missing data can influence the appropriateness of replacing missing values with 0.

Data context: The decision to replace missing values with 0 should also consider the specific context and domain of your data. For example, if the missing values represent measurements that were not taken or data points that are genuinely zero, then replacing NA with 0 might be reasonable. However, if the missing values represent unknown or unrecorded data, replacing them with 0 could be misleading.

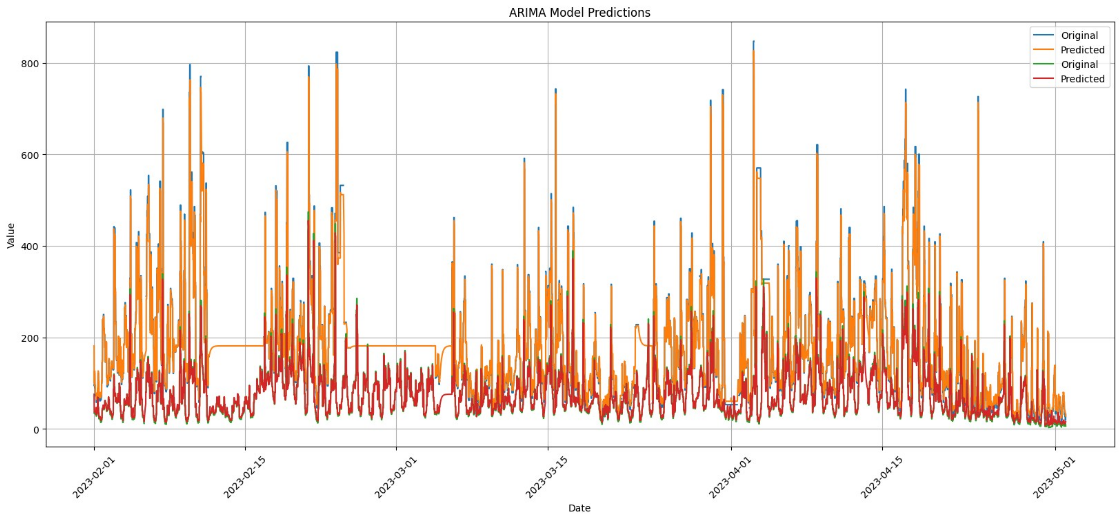
Impact on downstream analysis: Replacing missing values with 0 can affect various analyses, such as calculating averages, correlations, or performing time series forecasting. It's important to understand how the presence of 0 values may impact the interpretation of your results.

Yes! we can establish **ARMA/ARIMA process** for the dataset and it will be **Per-Column**

```
from statsmodels.tsa.arima.model import ARIMA

# Read the dataset
df = pd.read_csv('/content/Open pit blasting 01-02-2023 000000 To 01-
05-2023 235959.csv')
df = df.iloc[:-3]

# Clean column names
df.columns = df.columns.str.replace('[^a-zA-Z0-9]', '')
```

```python
# Convert date columns to datetime format
df['From'] = pd.to_datetime(df['From'])
df['ToInterval15M'] = pd.to_datetime(df['ToInterval15M'])


# Select the columns for analysis
columns_to_analyze = ['SingrauliSuryaKiranBhawanDudhichuaPM10gm3',
                      'SingrauliSuryaKiranBhawanDudhichuaPM25gm3'
                      ]




# Set the figure size
plt.figure(figsize=(20, 8))

# Iterate over the columns
for column in columns_to_analyze:
    # Fit the ARIMA model
    model = ARIMA(df[column], order=(1, 0, 1))
    model_fit = model.fit()

    # Make predictions
    predictions = model_fit.predict(start=0, end=len(df)-1)

    # Plot the original data and predictions
    plt.plot(df['ToInterval15M'], df[column], label='Original')
    plt.plot(df['ToInterval15M'], predictions, label='Predicted')

# Set the plot labels and title
plt.xlabel('Date')
plt.ylabel('Value')
plt.title('ARIMA Model Predictions')
plt.xticks(rotation=45)
plt.legend()
plt.grid(True)
plt.show()
```

```
<ipython-input-44-a2ed9e978fd9>:8: FutureWarning: The default value of
regex will change from True to False in a future version.
  df.columns = df.columns.str.replace('[^a-zA-Z0-9]', '')
```



this code implements ARIMA for first two columns and plot the orignial and the predicted plots. BLUE- PM10 original GREEN- PM2.5 orignial followed by their predicted graph

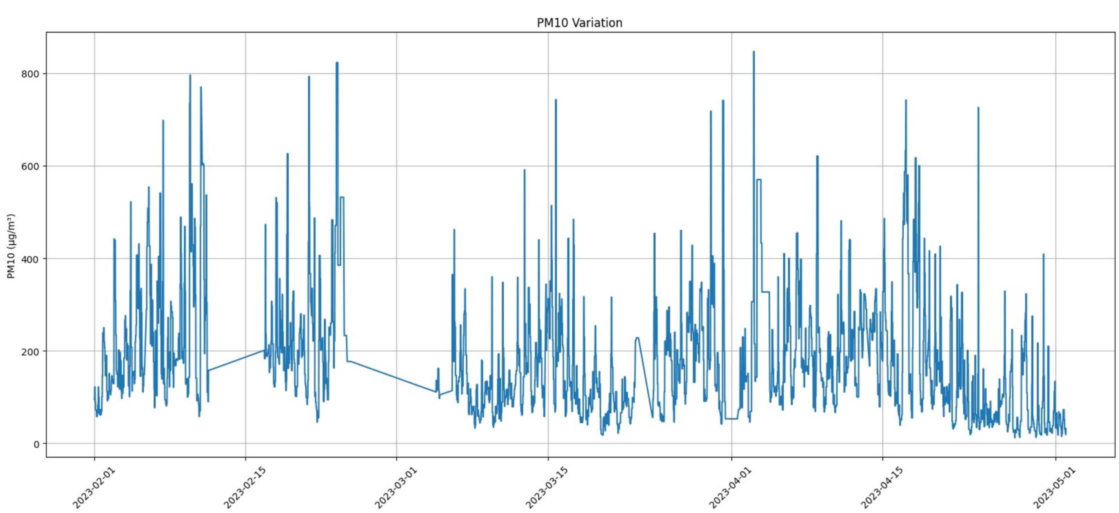
Regarding interpolation, it can be used to fill in missing values or smooth out irregularities in the time series data. Interpolation methods like linear interpolation, spline interpolation, or polynomial interpolation can help estimate the missing values based on the surrounding data points. However, it's important to note that interpolation is not directly related to ARMA/ARIMA modeling, which focuses on time series forecasting and modeling the temporal dependencies of the data. Interpolation can be used as a preprocessing step before applying ARMA/ARIMA modeling or as a separate technique to handle missing data.

NOW we tried to implement **interpolation** to fill values.

```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA

# Read the dataset
df = pd.read_csv('/content/Open pit blasting 01-02-2023 000000 To 01-
05-2023 235959.csv')
df = df.iloc[:-3]

# Clean column names
df.columns = df.columns.str.replace('[^a-zA-Z0-9]', '')

# Convert date columns to datetime format
df['From'] = pd.to_datetime(df['From'])
df['ToInterval15M'] = pd.to_datetime(df['ToInterval15M'])

# Interpolate missing values for specific columns
columns_to_interpolate = ['SingrauliSuryaKiranBhawanDudhichuaPM10gm3',
                          'SingrauliSuryaKiranBhawanDudhichuaPM25gm3',
                          'SingrauliSuryaKiranBhawanDudhichuaNOgm3',
                          'SingrauliSuryaKiranBhawanDudhichuaNO2gm3',
```

```
                                    'SingrauliSuryaKiranBhawanDudhichuaNOXppb',
                                    'SingrauliSuryaKiranBhawanDudhichuaCOmgm3',
                                    'SingrauliSuryaKiranBhawanDudhichuaSO2gm3',
                                    'SingrauliSuryaKiranBhawanDudhichuaNH3gm3',

'SingrauliSuryaKiranBhawanDudhichuaOzonegm3',

'SingrauliSuryaKiranBhawanDudhichuaBenzenegm3']

df[columns_to_interpolate] =
df[columns_to_interpolate].interpolate(method='linear')

# Select the column for analysis
pm10_column = 'SingrauliSuryaKiranBhawanDudhichuaPM10gm3'

# Plot the data
plt.figure(figsize=(20, 8))
plt.plot(df['ToInterval15M'], df[pm10_column])
plt.xlabel('Date')
plt.ylabel('PM10 (µg/m³)')
plt.title('PM10 Variation')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```

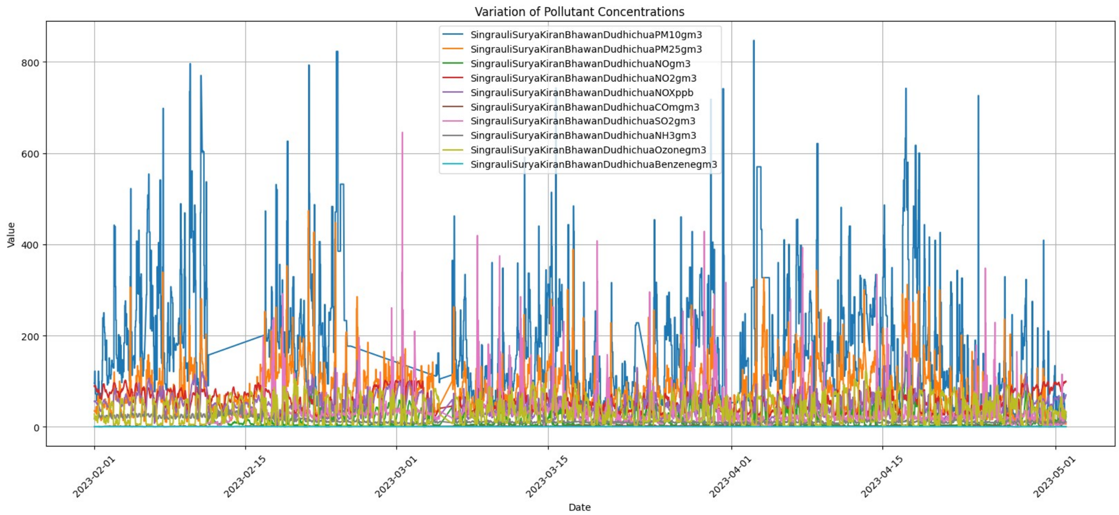
```
<ipython-input-45-8532b000dfe3>:10: FutureWarning: The default value
of regex will change from True to False in a future version.
  df.columns = df.columns.str.replace('[^a-zA-Z0-9]', '')
```



This is a plot of PM10 in which missing values are filled using **Linear Interpolation**

```
import pandas as pd
import matplotlib.pyplot as plt

# Read the dataset
df = pd.read_csv('/content/Open pit blasting 01-02-2023 000000 To 01-
05-2023 235959.csv')
df = df.iloc[:-3]

# Clean column names
df.columns = df.columns.str.replace('[^a-zA-Z0-9]', '')

# Convert date columns to datetime format
df['From'] = pd.to_datetime(df['From'])
df['ToInterval15M'] = pd.to_datetime(df['ToInterval15M'])



# Select the column for analysis
columns_for_analysis = ['SingrauliSuryaKiranBhawanDudhichuaPM10gm3',
                        'SingrauliSuryaKiranBhawanDudhichuaPM25gm3',
                        'SingrauliSuryaKiranBhawanDudhichuaNOgm3',
                        'SingrauliSuryaKiranBhawanDudhichuaNO2gm3',
                        'SingrauliSuryaKiranBhawanDudhichuaNOXppb',
                        'SingrauliSuryaKiranBhawanDudhichuaCOmgm3',
                        'SingrauliSuryaKiranBhawanDudhichuaSO2gm3',
                        'SingrauliSuryaKiranBhawanDudhichuaNH3gm3',
                        'SingrauliSuryaKiranBhawanDudhichuaOzonegm3',

'SingrauliSuryaKiranBhawanDudhichuaBenzenegm3']

df[columns_for_analysis] =
df[columns_for_analysis].interpolate(method='linear')

# Plot all columns together
plt.figure(figsize=(20, 8))
for column in columns_for_analysis:
    plt.plot(df['ToInterval15M'], df[column], label=column)

plt.xlabel('Date')
plt.ylabel('Value')
plt.title('Variation of Pollutant Concentrations')
plt.xticks(rotation=45)
plt.legend()
plt.grid(True)
plt.show()
```

```
<ipython-input-46-2277297f5945>:9: FutureWarning: The default value of regex will change from True to False in a future version.
  df.columns = df.columns.str.replace('[^a-zA-Z0-9]', '')
```



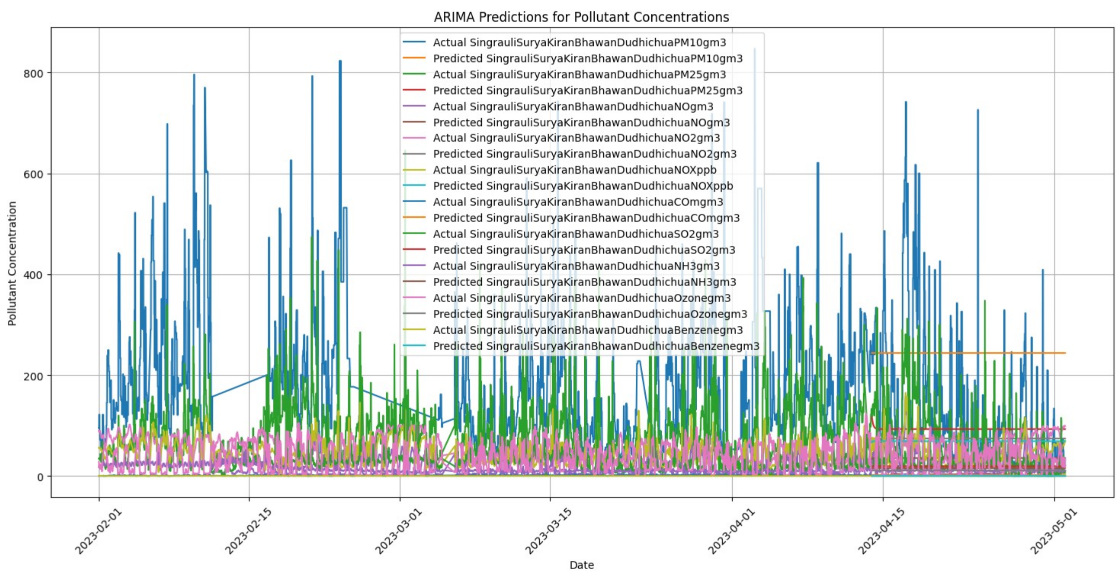
THIS is the plot in which is used **linear interpolation** in all the pollutants to fill missing values and then plotted its time series.

```python
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA

# Read the dataset
df = pd.read_csv('/content/Open pit blasting 01-02-2023 000000 To 01-05-2023 235959.csv')
df = df.iloc[:-3]

# Clean column names
df.columns = df.columns.str.replace('[^a-zA-Z0-9]', '')

# Convert date columns to datetime format
df['From'] = pd.to_datetime(df['From'])
df['ToInterval15M'] = pd.to_datetime(df['ToInterval15M'])

# Select the columns for analysis
columns_to_analyze = ['SingrauliSuryaKiranBhawanDudhichuaPM10gm3',
                      'SingrauliSuryaKiranBhawanDudhichuaPM25gm3',
                      'SingrauliSuryaKiranBhawanDudhichuaNOgm3',
                      'SingrauliSuryaKiranBhawanDudhichuaNO2gm3',
                      'SingrauliSuryaKiranBhawanDudhichuaNOXppb',
                      'SingrauliSuryaKiranBhawanDudhichuaCOmgm3',
                      'SingrauliSuryaKiranBhawanDudhichuaSO2gm3',
                      'SingrauliSuryaKiranBhawanDudhichuaNH3gm3',
                      'SingrauliSuryaKiranBhawanDudhichuaOzonegm3',
                      'SingrauliSuryaKiranBhawanDudhichuaBenzenegm3']
```

```
# Interpolate missing values for all columns
df[columns_to_analyze] =
df[columns_to_analyze].interpolate(method='linear')

# Define ARIMA order parameters for each column
arima_orders = {
    'SingrauliSuryaKiranBhawanDudhichuaPM10gm3': (1, 1, 1),
    'SingrauliSuryaKiranBhawanDudhichuaPM25gm3': (1, 1, 1),
    'SingrauliSuryaKiranBhawanDudhichuaNOgm3': (1, 1, 1),
    'SingrauliSuryaKiranBhawanDudhichuaNO2gm3': (1, 1, 1),
    'SingrauliSuryaKiranBhawanDudhichuaNOXppb': (1, 1, 1),
    'SingrauliSuryaKiranBhawanDudhichuaCOmgm3': (1, 1, 1),
    'SingrauliSuryaKiranBhawanDudhichuaSO2gm3': (1, 1, 1),
    'SingrauliSuryaKiranBhawanDudhichuaNH3gm3': (1, 1, 1),
    'SingrauliSuryaKiranBhawanDudhichuaOzonegm3': (1, 1, 1),
    'SingrauliSuryaKiranBhawanDudhichuaBenzenegm3': (1, 1, 1)
}

# Plot the pollutant concentrations and predicted values
plt.figure(figsize=(18, 8))
for column in columns_to_analyze:
    # Select the column data
    data = df[column]

    # Split the data into training and testing sets
    train_size = int(len(data) * 0.8)
    train_data, test_data = data[:train_size], data[train_size:]

    # Fit the ARIMA model
    arima = ARIMA(train_data, order=arima_orders[column])
    arima_model = arima.fit()

    # Predict the values for the testing set
    predicted_values = arima_model.predict(start=len(train_data),
end=len(train_data) + len(test_data) - 1)

    # Plot the actual values
    plt.plot(df['ToInterval15M'], data, label=f'Actual {column}')

    # Plot the predicted values
    plt.plot(df['ToInterval15M'][train_size:], predicted_values,
label=f'Predicted {column}')

plt.xlabel('Date')
plt.ylabel('Pollutant Concentration')
plt.title('ARIMA Predictions for Pollutant Concentrations')
plt.xticks(rotation=45)
plt.legend()
```

```
plt.grid(True)
plt.show()
```

```
<ipython-input-47-9521d4a24bea>:10: FutureWarning: The default value of regex will change from True to False in a future version.
  df.columns = df.columns.str.replace('[^a-zA-Z0-9]', '')
```

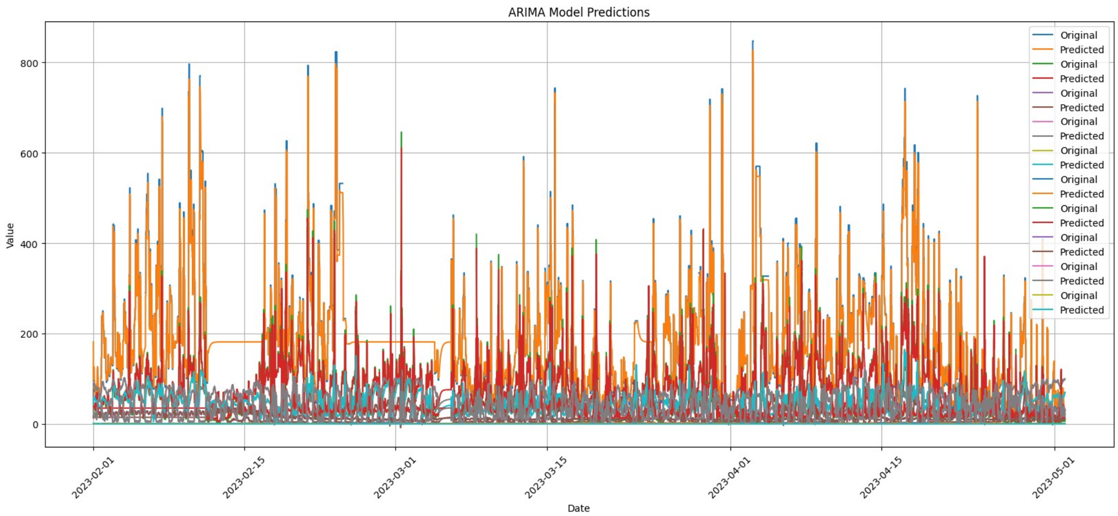


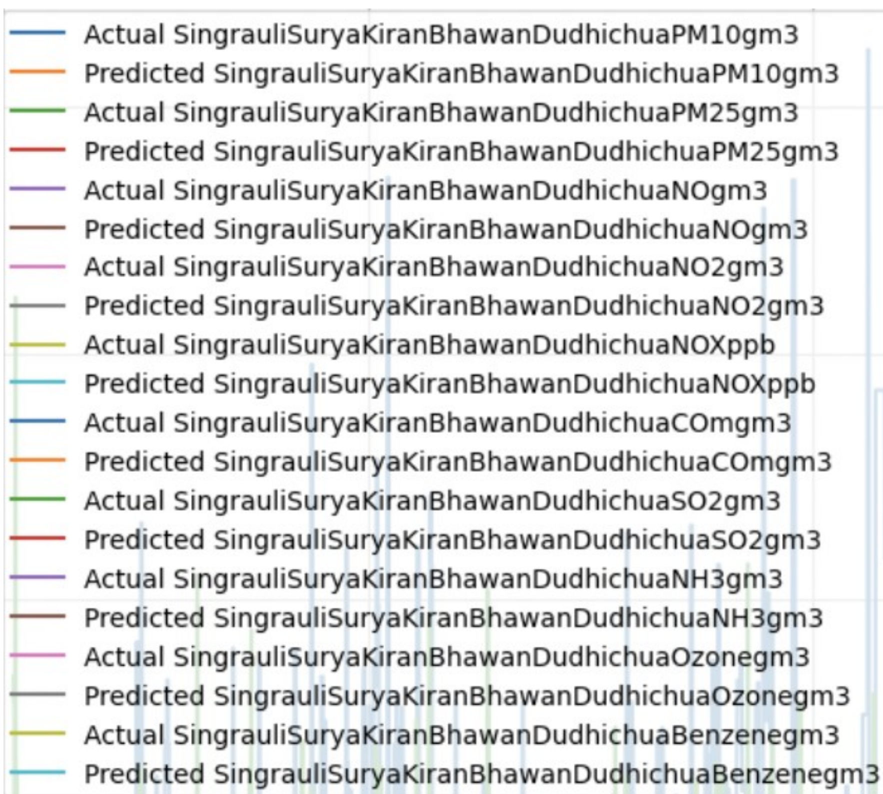
This code uses arima/arma model to handle missing values and plots both the orignial series and the predicted series by applying the model.

```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA

# Read the dataset
df = pd.read_csv('/content/Open pit blasting 01-02-2023 000000 To 01-05-2023 235959.csv')
df = df.iloc[:-3]

# Clean column names
df.columns = df.columns.str.replace('[^a-zA-Z0-9]', '')

# Convert date columns to datetime format
df['From'] = pd.to_datetime(df['From'])
df['ToInterval15M'] = pd.to_datetime(df['ToInterval15M'])


# Select the columns for analysis
columns_to_analyze = ['SingrauliSuryaKiranBhawanDudhichuaPM10gm3',
                      'SingrauliSuryaKiranBhawanDudhichuaPM25gm3',
                      'SingrauliSuryaKiranBhawanDudhichuaNOgm3',
                      'SingrauliSuryaKiranBhawanDudhichuaNO2gm3',
                      'SingrauliSuryaKiranBhawanDudhichuaNOXppb',
```

```
                      'SingrauliSuryaKiranBhawanDudhichuaCOmgm3',
                      'SingrauliSuryaKiranBhawanDudhichuaSO2gm3',
                      'SingrauliSuryaKiranBhawanDudhichuaNH3gm3',
                      'SingrauliSuryaKiranBhawanDudhichuaOzonegm3',
                      'SingrauliSuryaKiranBhawanDudhichuaBenzenegm3']



# Set the figure size
plt.figure(figsize=(20, 8))

# Iterate over the columns
for column in columns_to_analyze:
    # Fit the ARIMA model
    model = ARIMA(df[column], order=(1, 0, 1))
    model_fit = model.fit()

    # Make predictions
    predictions = model_fit.predict(start=0, end=len(df)-1)
    # Plot the original data and predictions
    plt.plot(df['ToInterval15M'], df[column], label='Original')
    plt.plot(df['ToInterval15M'], predictions, label='Predicted')



# Set the plot labels and title
plt.xlabel('Date')
plt.ylabel('Value')
plt.title('ARIMA Model Predictions')
plt.xticks(rotation=45)
plt.legend()
plt.grid(True)
plt.show()
```

```
<ipython-input-48-51f7c6f21d0c>:10: FutureWarning: The default value
of regex will change from True to False in a future version.
  df.columns = df.columns.str.replace('[^a-zA-Z0-9]', '')
```

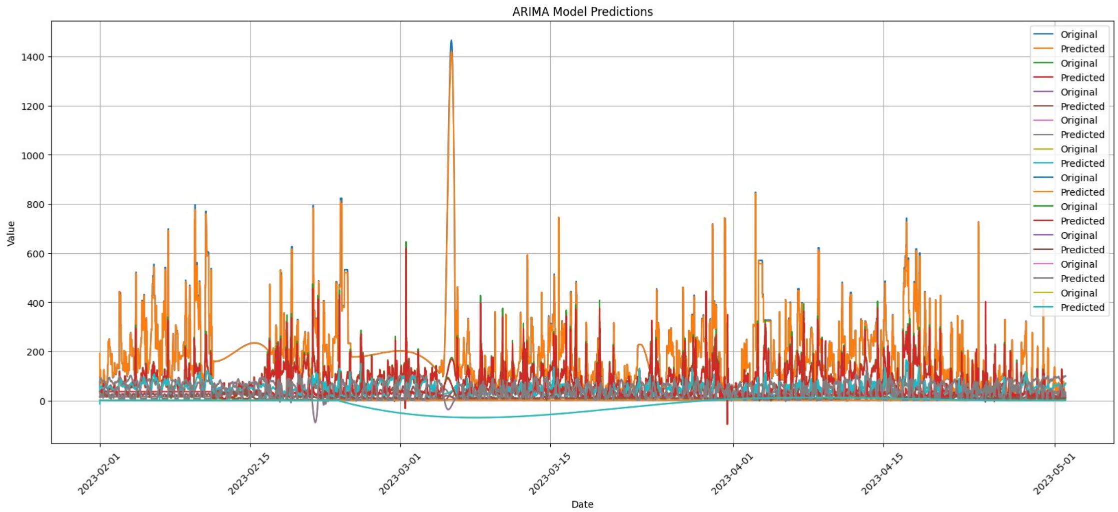





```python
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA

# Read the dataset
df = pd.read_csv('/content/Open pit blasting 01-02-2023 000000 To 01-
05-2023 235959.csv')
df = df.iloc[:-3]

# Clean column names
```

```python
df.columns = df.columns.str.replace('[^a-zA-Z0-9]', '')

# Convert date columns to datetime format
df['From'] = pd.to_datetime(df['From'])
df['ToInterval15M'] = pd.to_datetime(df['ToInterval15M'])


# Select the columns for analysis
columns_to_analyze = ['SingrauliSuryaKiranBhawanDudhichuaPM10gm3',
                      'SingrauliSuryaKiranBhawanDudhichuaPM25gm3',
                      'SingrauliSuryaKiranBhawanDudhichuaNOgm3',
                      'SingrauliSuryaKiranBhawanDudhichuaNO2gm3',
                      'SingrauliSuryaKiranBhawanDudhichuaNOXppb',
                      'SingrauliSuryaKiranBhawanDudhichuaCOmgm3',
                      'SingrauliSuryaKiranBhawanDudhichuaSO2gm3',
                      'SingrauliSuryaKiranBhawanDudhichuaNH3gm3',
                      'SingrauliSuryaKiranBhawanDudhichuaOzonegm3',
                      'SingrauliSuryaKiranBhawanDudhichuaBenzenegm3']

                      # Interpolate missing values for all columns
df[columns_to_analyze] =
df[columns_to_analyze].interpolate(method='cubic')

# Set the figure size
plt.figure(figsize=(20, 8))

# Iterate over the columns
for column in columns_to_analyze:
    # Fit the ARIMA model
    model = ARIMA(df[column], order=(1, 0, 1))
    model_fit = model.fit()

    # Make predictions
    predictions = model_fit.predict(start=0, end=len(df)-1)

    # Plot the original data and predictions
    plt.plot(df['ToInterval15M'], df[column], label='Original')
    plt.plot(df['ToInterval15M'], predictions, label='Predicted')

# Set the plot labels and title
plt.xlabel('Date')
plt.ylabel('Value')
plt.title('ARIMA Model Predictions')
plt.xticks(rotation=45)
plt.legend()
plt.grid(True)
plt.show()
```

```
<ipython-input-50-10f7fb072e06>:10: FutureWarning: The default value of regex will change from True to False in a future version.
  df.columns = df.columns.str.replace('[^a-zA-Z0-9]', '')
```

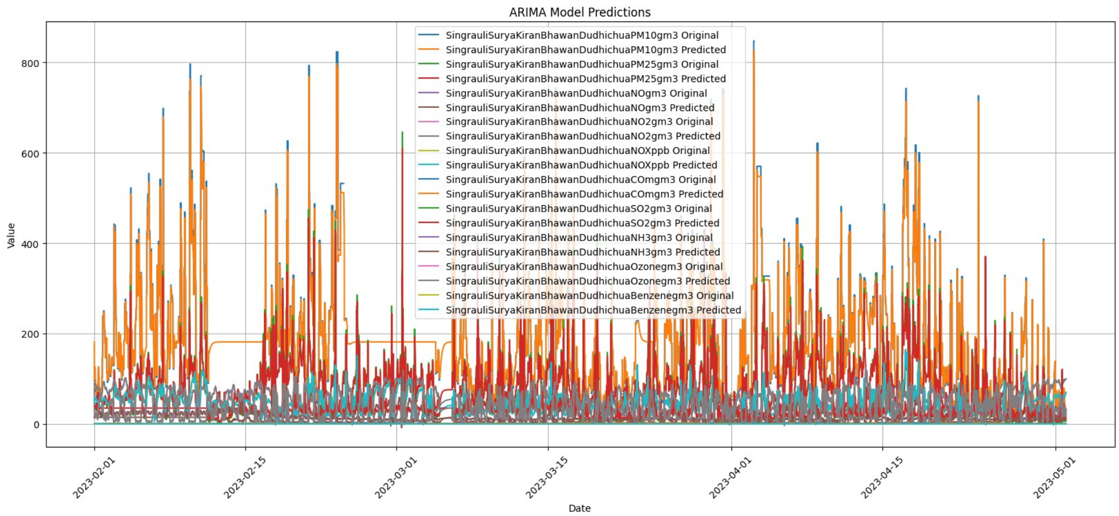


applied **cubic interpolation**

```python
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from statsmodels.tsa.arima.model import ARIMA
from scipy.interpolate import make_interp_spline

# Read the dataset
df = pd.read_csv('/content/Open pit blasting 01-02-2023 000000 To 01-05-2023 235959.csv')
df = df.iloc[:-3]

# Clean column names
df.columns = df.columns.str.replace('[^a-zA-Z0-9]', '')

# Convert date columns to datetime format
df['From'] = pd.to_datetime(df['From'])
df['ToInterval15M'] = pd.to_datetime(df['ToInterval15M'])

# Select the columns for analysis
columns_to_analyze = ['SingrauliSuryaKiranBhawanDudhichuaPM10gm3',
                      'SingrauliSuryaKiranBhawanDudhichuaPM25gm3',
                      'SingrauliSuryaKiranBhawanDudhichuaNOgm3',
                      'SingrauliSuryaKiranBhawanDudhichuaNO2gm3',
                      'SingrauliSuryaKiranBhawanDudhichuaNOXppb',
                      'SingrauliSuryaKiranBhawanDudhichuaCOmgm3',
                      'SingrauliSuryaKiranBhawanDudhichuaSO2gm3',
                      'SingrauliSuryaKiranBhawanDudhichuaNH3gm3',
                      'SingrauliSuryaKiranBhawanDudhichuaOzonegm3',
```

```
                      'SingrauliSuryaKiranBhawanDudhichuaBenzenegm3']

# Set the figure size
plt.figure(figsize=(20, 8))

# Iterate over the columns
for column in columns_to_analyze:
    # Fit the ARIMA model
    model = ARIMA(df[column], order=(1, 0, 1))
    model_fit = model.fit()

    # Make predictions
    predictions = model_fit.predict(start=0, end=len(df)-1)

    # Interpolate missing values for the current column using spline
interpolation
    data_column = df[column]
    x = np.arange(len(data_column))
    mask = ~pd.isnull(data_column)
    spline = make_interp_spline(x[mask], data_column[mask])
    interpolated_values = spline(np.linspace(x[0], x[-1], len(df)))
 # Plot the original data for the current column
    plt.plot(df['ToInterval15M'], df[column], label=f'{column}
Original')

    # Plot the predictions for the current column
    plt.plot(df['ToInterval15M'], predictions, label=f'{column}
Predicted')
# Set the plot labels and title
plt.xlabel('Date')
plt.ylabel('Value')
plt.title('ARIMA Model Predictions')
plt.xticks(rotation=45)
plt.legend()
plt.grid(True)
plt.show()
```

```
<ipython-input-51-c2fade66c030>:12: FutureWarning: The default value
of regex will change from True to False in a future version.
  df.columns = df.columns.str.replace('[^a-zA-Z0-9]', '')
```



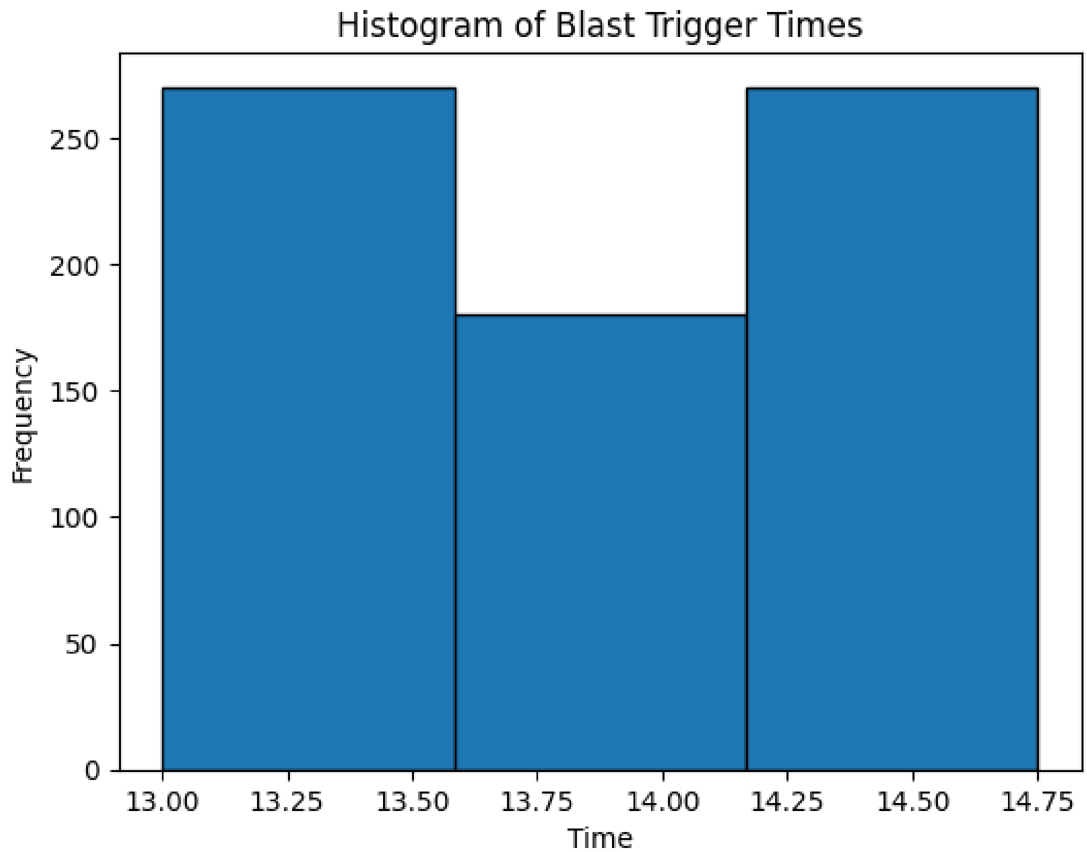
Applied **Spline interpolation**

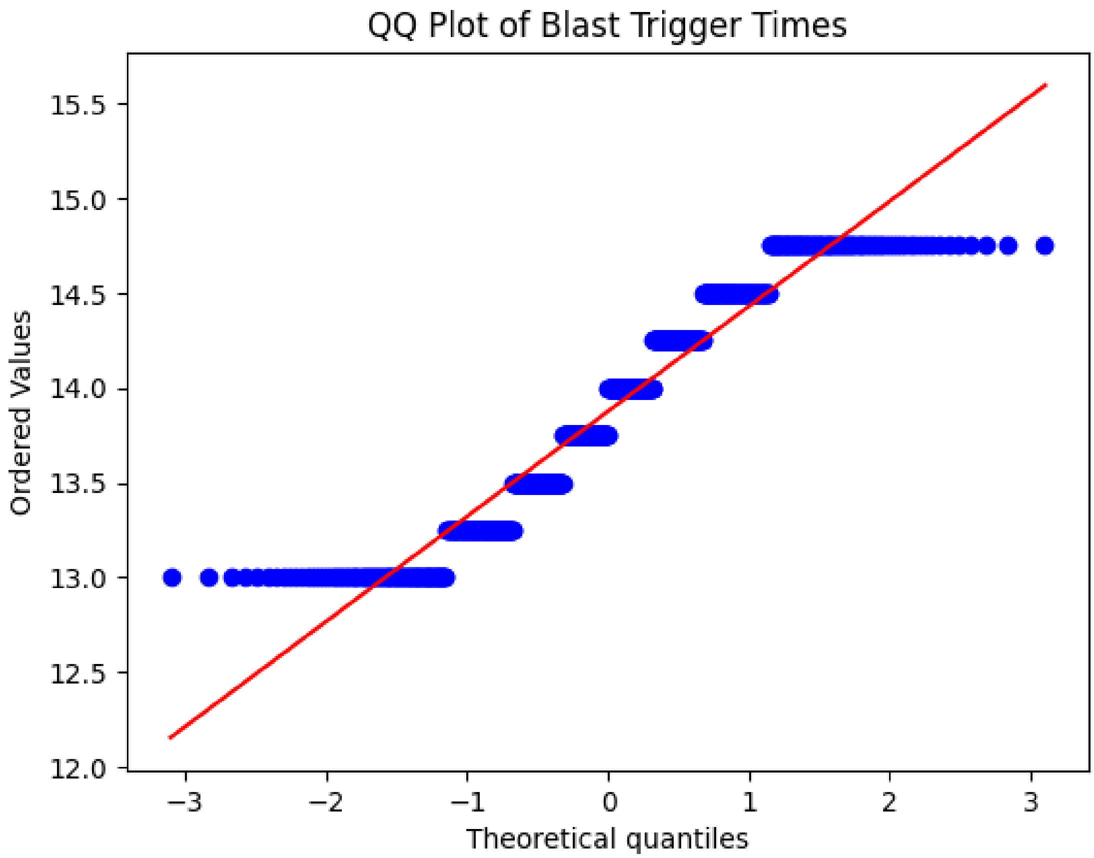
A particular method of analysing a set of air pollution data points gathered over a period of time is called a 'time series analysis.' Instead of just capturing the data points intermittently or arbitrarily, time series analyzers record the data points at regular intervals over a predetermined length of time. Blasting time in coal India is 13:45 pm to 14:45 pm major effect on air pollution.

```python
import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats as stats

# Read the dataset
df = pd.read_csv('/content/Filled_Data.csv')
df = df.iloc[:-3]

# Clean column names
df.columns = df.columns.str.replace('[^a-zA-Z0-9]', '')

# Convert date columns to datetime format
df['From'] = pd.to_datetime(df['From'])
df['ToInterval15M'] = pd.to_datetime(df['ToInterval15M'])

# Forward fill missing values in each column
df.fillna(method='ffill', inplace=True)

# Step 3: Create a weighted combination of air polluting factors
weights = {
    'SingrauliSuryaKiranBhawanDudhichuaPM10gm3': 0.3,
    'SingrauliSuryaKiranBhawanDudhichuaPM25gm3': 0.2,
    'SingrauliSuryaKiranBhawanDudhichuaNOgm3': 0.1,
    'SingrauliSuryaKiranBhawanDudhichuaNO2gm3': 0.1,
    'SingrauliSuryaKiranBhawanDudhichuaNOXppb': 0.1,
    'SingrauliSuryaKiranBhawanDudhichuaCOmgm3': 0.15,
```

```
    'SingrauliSuryaKiranBhawanDudhichuaSO2gm3': 0.05,
    'SingrauliSuryaKiranBhawanDudhichuaNH3gm3': 0.1,
    'SingrauliSuryaKiranBhawanDudhichuaOzonegm3': 0.01,
    'SingrauliSuryaKiranBhawanDudhichuaBenzenegm3': 0.5
}

df['Combined_Pollution'] = sum(df[fac] * weights[fac] for fac in
weights)

# Step 4: Extract blasting time and plot histogram
blast_trigger_times = df[(df['From'].dt.hour >= 13) &
(df['From'].dt.hour <= 14)]['From'].dt.time
blast_trigger_times_numeric = blast_trigger_times.apply(lambda x:
x.hour + x.minute / 60)  # Convert to numeric format

# Calculate the bin range
bin_range = max(blast_trigger_times_numeric) -
min(blast_trigger_times_numeric)

# Plot the histogram
plt.hist(blast_trigger_times_numeric, bins=int(bin_range * 2),
edgecolor='black')
plt.xlabel('Time')
plt.ylabel('Frequency')
plt.title('Histogram of Blast Trigger Times')
plt.show()

# Step 5: Analyze distribution using QQ plot
stats.probplot(blast_trigger_times_numeric, dist='norm', plot=plt)
plt.title('QQ Plot of Blast Trigger Times')
plt.show()

# Step 6: Calculate probability of blast happening during 14:15 to
14:30
blast_probability = ((blast_trigger_times >=
pd.to_datetime('14:15').time()) &
                     (blast_trigger_times <=
pd.to_datetime('14:30').time())).mean()

print('Probability of a blast happening during 14:15 to 14:30:',
blast_probability)
```

```
<ipython-input-63-9034344dfb62>:10: FutureWarning: The default value
of regex will change from True to False in a future version.
  df.columns = df.columns.str.replace('[^a-zA-Z0-9]', '')
```


Histogram of Blast Trigger Times
Frequency
0
50
100
150
200
250
13.00
13.25
13.50
13.75
14.00
14.25
14.50
14.75
Time




```
Probability of a blast happening during 14:15 to 14:30: 0.25
```

to save a new file with linear interpolated data

```
import pandas as pd

def fill_missing_values_linear_interpolation(file_path,
output_file_path):
    # Read the CSV file into a DataFrame
    df = pd.read_csv(file_path)

    # Iterate over columns starting from the 4th column
    for column_name in df.columns[3:]:
        # Convert "NA" to NaN
        df[column_name] = df[column_name].replace("NA", float("nan"))

        # Perform linear interpolation
        df[column_name] = df[column_name].interpolate(method='linear')

    # Save the updated DataFrame to a new CSV file
    df.to_csv(output_file_path, index=False)

# Usage example
input_file_path = '/content/Open pit blasting 01-02-2023 000000 To 01-
```

```
05-2023 235959.csv'
output_file_path = '/content/linear_filled.csv'

fill_missing_values_linear_interpolation(input_file_path,
output_file_path)
```

Problem Statement:

The objective of this analysis is to examine the air quality data collected from Singrauli, Surya Kiran Bhawan Dudhichua, and gain insights into the levels of various pollutants present in the air. The dataset provides information on PM10, PM2.5, NO, NO2, NOX, CO, SO2, NH3, Ozone, and Benzene concentrations at 15-minute intervals over a specific time period.

Key Questions to Address:

Are there any significant fluctuations or patterns observed in the pollutant levels? Are there any correlations between different pollutants? Are there any periods of time when certain pollutants exceeded the permissible limits? Are there any relationships between pollutant levels and time of day? Are there any pollutant concentration outliers that need further investigation?

1. What is the overall trend of each pollutant concentration during the recorded time period? plot time series
2. Are there any correlations between different pollutants?
3. Are there any periods of time when certain pollutants exceeded the permissible limits?
4. Are there any relationships between pollutant levels and time of day? DO a descriptive analysis
5. Predict the future Data of atleast 15 days
6. Are there any pollutant concentration outliers that need further investigation?

```
import pandas as pd

# Read the data from a CSV file
# Read the dataset
data = pd.read_csv('/content/Open pit blasting 01-02-2023 000000 To
01-05-2023 235959.csv')
data = data.iloc[:-3]

# Clean column names
data.columns = data.columns.str.replace('[^a-zA-Z0-9]', '')
```

```python
# Convert date columns to datetime format
data['From'] = pd.to_datetime(data['From'])
data['ToInterval15M'] = pd.to_datetime(data['ToInterval15M'])
```

```python
# Check for missing values
missing_values = data.isnull().sum()
print(missing_values)
```

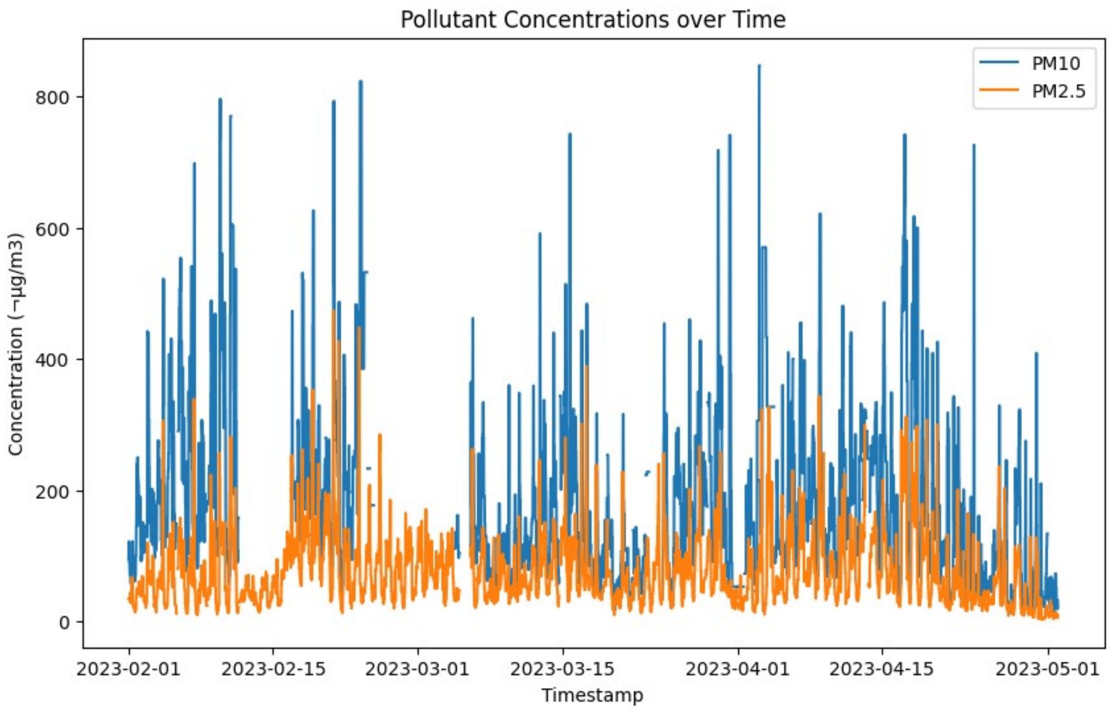
```
                                                    0
From                                                0
ToInterval15M                                       0
SingrauliSuryaKiranBhawanDudhichuaPM10gm3        1681
SingrauliSuryaKiranBhawanDudhichuaPM25gm3         226
SingrauliSuryaKiranBhawanDudhichuaNOgm3          1369
SingrauliSuryaKiranBhawanDudhichuaNO2gm3          416
SingrauliSuryaKiranBhawanDudhichuaNOXppb          415
SingrauliSuryaKiranBhawanDudhichuaCOmgm3          496
SingrauliSuryaKiranBhawanDudhichuaSO2gm3         1451
SingrauliSuryaKiranBhawanDudhichuaNH3gm3          326
SingrauliSuryaKiranBhawanDudhichuaOzonegm3        453
SingrauliSuryaKiranBhawanDudhichuaBenzenegm3     6195
dtype: int64
```

```
<ipython-input-54-9b2f7fdde57a>:9: FutureWarning: The default value of
regex will change from True to False in a future version.
  data.columns = data.columns.str.replace('[^a-zA-Z0-9]', '')
```

Visualize time series of pollutant concentrations

```python
import matplotlib.pyplot as plt

# Visualize time series of pollutant concentrations
plt.figure(figsize=(10, 6))
plt.plot(data['From'],
data['SingrauliSuryaKiranBhawanDudhichuaPM10gm3'], label='PM10')
plt.plot(data['From'],
data['SingrauliSuryaKiranBhawanDudhichuaPM25gm3'], label='PM2.5')
# Repeat for other pollutants
plt.xlabel('Timestamp')
plt.ylabel('Concentration (¬µg/m3)')
plt.title('Pollutant Concentrations over Time')
plt.legend()
plt.show()

# Calculate summary statistics
summary_stats = data.describe()
print(summary_stats)
```

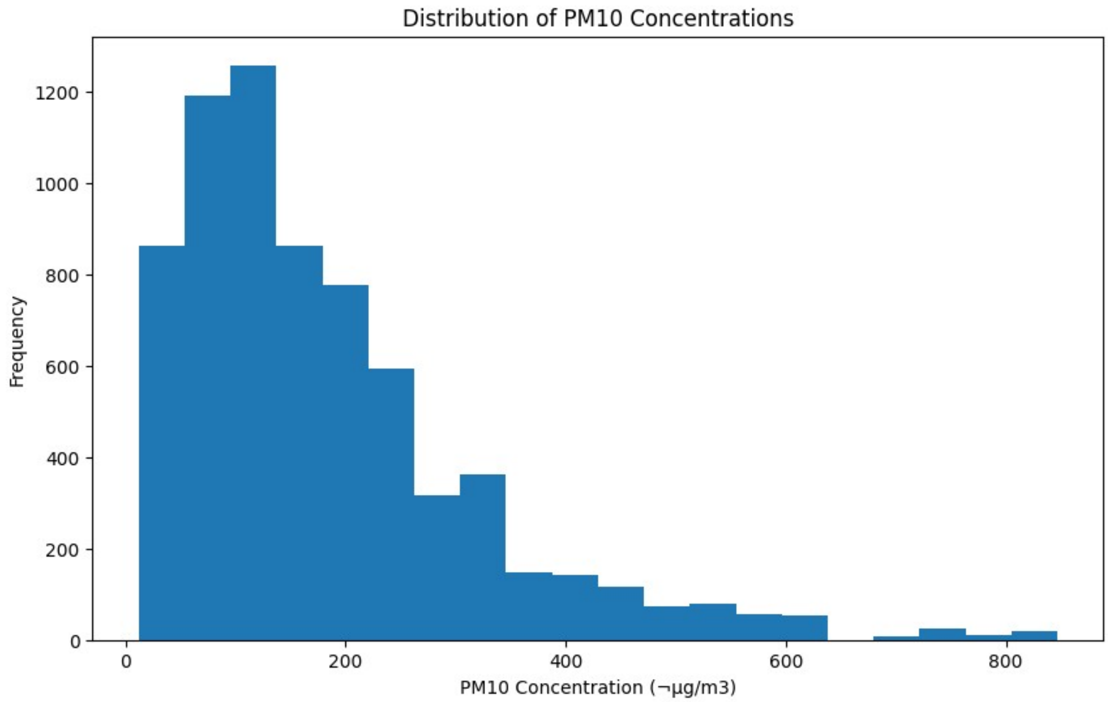
```
# Examine the distribution of pollutant concentrations
plt.figure(figsize=(10, 6))
plt.hist(data['SingrauliSuryaKiranBhawanDudhichuaPM10gm3'], bins=20)
plt.xlabel('PM10 Concentration (¬µg/m3)')
plt.ylabel('Frequency')
plt.title('Distribution of PM10 Concentrations')
plt.show()
```




```
                    SingrauliSuryaKiranBhawanDudhichuaPM10gm3  \
count  8640.000000                                  6959.000000
mean   4320.500000                                   181.408679
std    2494.297496                                   136.016142
min       1.000000                                    12.000000
25%    2160.750000                                    84.000000
50%    4320.500000                                   145.000000
75%    6480.250000                                   238.000000
max    8640.000000                                   847.000000

       SingrauliSuryaKiranBhawanDudhichuaPM25gm3  \
count                                8414.000000
mean                                   75.690397
std                                    55.245265
min                                     3.000000
25%                                    36.000000
50%                                    61.000000
75%                                   101.000000
max                                   474.000000
```

```
       SingrauliSuryaKiranBhawanDudhichuaNOgm3  \
count                                7271.000000
mean                                   14.649636
std                                    19.221385
min                                     0.100000
25%                                     3.900000
50%                                     6.100000
75%                                    16.500000
max                                   157.500000

       SingrauliSuryaKiranBhawanDudhichuaNO2gm3  \
count                                 8224.000000
mean                                    55.757028
std                                     20.231407
min                                      0.200000
25%                                     39.400000
50%                                     53.200000
75%                                     71.025000
max                                    106.900000

       SingrauliSuryaKiranBhawanDudhichuaNOXppb  \
count                                 8225.000000
mean                                    42.672219
std                                     22.435262
min                                      4.200000
25%                                     25.000000
50%                                     37.700000
75%                                     53.800000
max                                    165.200000

       SingrauliSuryaKiranBhawanDudhichuaCOmgm3  \
count                                 8144.000000
mean                                     1.408538
std                                      0.631056
min                                      0.100000
25%                                      0.950000
50%                                      1.420000
75%                                      1.850000
max                                      4.000000

       SingrauliSuryaKiranBhawanDudhichuaSO2gm3  \
count                                 7189.000000
mean                                    34.232731
std                                     39.452131
min                                      0.100000
25%                                     16.100000
50%                                     25.300000
75%                                     35.200000
max                                    645.600000
```

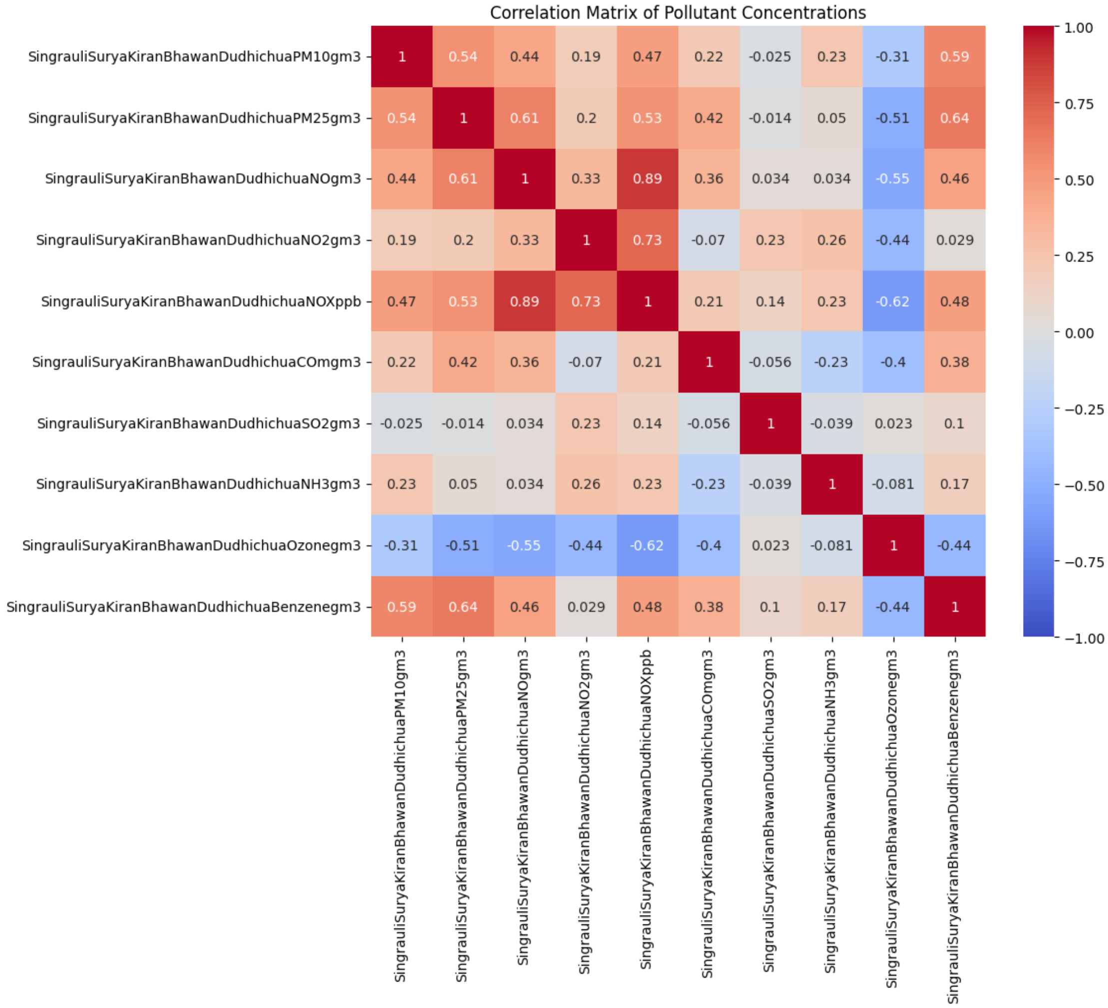
```
       SingrauliSuryaKiranBhawanDudhichuaNH3gm3  \
count                              8314.000000
mean                                 13.242663
std                                   6.151034
min                                   4.600000
25%                                   9.400000
50%                                  11.000000
75%                                  14.000000
max                                  62.400000

       SingrauliSuryaKiranBhawanDudhichuaOzonegm3  \
count                                8187.000000
mean                                   35.626530
std                                    27.018693
min                                     0.100000
25%                                    10.500000
50%                                    32.400000
75%                                    58.800000
max                                   123.800000

       SingrauliSuryaKiranBhawanDudhichuaBenzenegm3
count                                  2445.000000
mean                                      0.177505
std                                       0.098895
min                                       0.100000
25%                                       0.100000
50%                                       0.100000
75%                                       0.200000
max                                       0.600000
```



Correlation Analysis:

Correlation Matrix of Pollutant Concentrations

```
import seaborn as sns
# Select relevant pollutant columns for correlation analysis
pollutant_columns = ['SingrauliSuryaKiranBhawanDudhichuaPM10gm3',
    'SingrauliSuryaKiranBhawanDudhichuaPM25gm3',
    'SingrauliSuryaKiranBhawanDudhichuaNOgm3',
    'SingrauliSuryaKiranBhawanDudhichuaNO2gm3',
    'SingrauliSuryaKiranBhawanDudhichuaNOXppb',
    'SingrauliSuryaKiranBhawanDudhichuaCOmgm3',
    'SingrauliSuryaKiranBhawanDudhichuaSO2gm3',
    'SingrauliSuryaKiranBhawanDudhichuaNH3gm3',
    'SingrauliSuryaKiranBhawanDudhichuaOzonegm3',
    'SingrauliSuryaKiranBhawanDudhichuaBenzenegm3'
]
# Calculate correlation coefficients
corr_matrix = data[pollutant_columns].corr()

# Visualize correlations using a heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Correlation Matrix of Pollutant Concentrations')
plt.show()
```



check for threshold values for each pollutant

```
# Define threshold values for each pollutant
thresholds = {
   'SingrauliSuryaKiranBhawanDudhichuaPM25gm3':50,
    'SingrauliSuryaKiranBhawanDudhichuaNOgm3':45,
    'SingrauliSuryaKiranBhawanDudhichuaNO2gm3':20,
    # Add thresholds for other pollutants
}

# Identify exceedances for each pollutant
exceedances = {}
for pollutant, threshold in thresholds.items():
    try:
        exceedances[pollutant] = data[data[pollutant] > threshold]
    except KeyError:
        print(f"Column '{pollutant}' not found in the dataset.")

# Print the exceedances for each pollutant
for pollutant, exceedance_data in exceedances.items():
```

```
    print(f"Exceedances for {pollutant}:")
    print(exceedance_data)

# Count the number of exceedances for each pollutant
exceedance_counts = {pollutant: len(exceedance_data) for pollutant,
exceedance_data in exceedances.items()}
print("\nExceedance Counts:")
print(exceedance_counts)
```

```
Exceedances for SingrauliSuryaKiranBhawanDudhichuaPM25gm3:
                          From        ToInterval15M  \
31      32 2023-02-01 07:45:00 2023-02-01 08:00:00
32      33 2023-02-01 08:00:00 2023-02-01 08:15:00
33      34 2023-02-01 08:15:00 2023-02-01 08:30:00
34      35 2023-02-01 08:30:00 2023-02-01 08:45:00
95      96 2023-02-01 23:45:00 2023-02-02 00:00:00
...    ...                 ...                 ...
8450  8451 2023-04-30 00:30:00 2023-04-30 00:45:00
8451  8452 2023-04-30 00:45:00 2023-04-30 01:00:00
8452  8453 2023-04-30 01:00:00 2023-04-30 01:15:00
8453  8454 2023-04-30 01:15:00 2023-04-30 01:30:00
8454  8455 2023-04-30 01:30:00 2023-04-30 01:45:00

      SingrauliSuryaKiranBhawanDudhichuaPM10gm3  \
31                                         83.0
32                                         83.0
33                                         83.0
34                                         83.0
95                                        178.0
...                                         ...
8450                                       43.0
8451                                       23.0
8452                                       23.0
8453                                       23.0
8454                                       23.0

      SingrauliSuryaKiranBhawanDudhichuaPM25gm3  \
31                                         68.0
32                                         68.0
33                                         68.0
34                                         68.0
95                                         61.0
...                                         ...
8450                                       95.0
8451                                       87.0
8452                                       87.0
8453                                       87.0
8454                                       87.0

      SingrauliSuryaKiranBhawanDudhichuaNOgm3  \
```

```
31                                        NaN
32                                        NaN
33                                        NaN
34                                        NaN
95                                        NaN
...                                       ...
8450                                      4.3
8451                                      3.7
8452                                      3.3
8453                                      3.0
8454                                      3.1

      SingrauliSuryaKiranBhawanDudhichuaNO2gm3  \
31                                        76.6
32                                        79.1
33                                        78.9
34                                        77.8
95                                        79.0
...                                        ...
8450                                      92.3
8451                                      90.7
8452                                      89.5
8453                                      90.1
8454                                      90.7

      SingrauliSuryaKiranBhawanDudhichuaNOXppb  \
31                                        49.3
32                                        50.8
33                                        50.6
34                                        49.1
95                                        53.1
...                                        ...
8450                                      52.6
8451                                      51.3
8452                                      50.3
8453                                      50.4
8454                                      50.8

      SingrauliSuryaKiranBhawanDudhichuaCOmgm3  \
31                                        0.47
32                                        0.49
33                                        0.34
34                                        0.29
95                                         NaN
...                                        ...
8450                                      0.61
8451                                      0.63
8452                                      0.63
8453                                      0.63
8454                                      0.63
```

```
      SingrauliSuryaKiranBhawanDudhichuaSO2gm3  \
31                                         NaN   
32                                         NaN   
33                                         NaN   
34                                         NaN   
95                                         NaN   
...                                        ...   
8450                                       1.7   
8451                                       0.4   
8452                                       0.8   
8453                                       0.4   
8454                                       0.6   

      SingrauliSuryaKiranBhawanDudhichuaNH3gm3  \
31                                        18.1   
32                                        17.3   
33                                        18.2   
34                                        20.4   
95                                        19.7   
...                                        ...   
8450                                       9.4   
8451                                       9.3   
8452                                       9.1   
8453                                       8.8   
8454                                       8.6   

      SingrauliSuryaKiranBhawanDudhichuaOzonegm3  \
31                                          22.1   
32                                          30.1   
33                                          35.8   
34                                          36.9   
95                                          15.3   
...                                          ...   
8450                                        45.1   
8451                                        43.9   
8452                                        44.2   
8453                                        41.9   
8454                                        38.8   

      SingrauliSuryaKiranBhawanDudhichuaBenzenegm3  
31                                             0.4  
32                                             0.4  
33                                             0.4  
34                                             0.4  
95                                             0.1  
...                                            ...  
8450                                           0.1  
8451                                           0.1  
8452                                           0.1  
```

```
8453                                               0.1
8454                                               0.1

[4999 rows x 13 columns]
Exceedances for SingrauliSuryaKiranBhawanDudhichuaNOgm3:
                          From       ToInterval15M  \
1902  1903 2023-02-20 19:30:00 2023-02-20 19:45:00
1903  1904 2023-02-20 19:45:00 2023-02-20 20:00:00
1904  1905 2023-02-20 20:00:00 2023-02-20 20:15:00
1905  1906 2023-02-20 20:15:00 2023-02-20 20:30:00
1906  1907 2023-02-20 20:30:00 2023-02-20 20:45:00
...    ...                 ...                 ...
8286  8287 2023-04-28 07:30:00 2023-04-28 07:45:00
8287  8288 2023-04-28 07:45:00 2023-04-28 08:00:00
8288  8289 2023-04-28 08:00:00 2023-04-28 08:15:00
8289  8290 2023-04-28 08:15:00 2023-04-28 08:30:00
8290  8291 2023-04-28 08:30:00 2023-04-28 08:45:00

      SingrauliSuryaKiranBhawanDudhichuaPM10gm3  \
1902                                      136.0
1903                                      436.0
1904                                      436.0
1905                                      436.0
1906                                      436.0
...                                         ...
8286                                      323.0
8287                                      188.0
8288                                      188.0
8289                                      188.0
8290                                      188.0

      SingrauliSuryaKiranBhawanDudhichuaPM25gm3  \
1902                                      175.0
1903                                      474.0
1904                                      474.0
1905                                      474.0
1906                                      474.0
...                                         ...
8286                                       97.0
8287                                       57.0
8288                                       57.0
8289                                       57.0
8290                                       57.0

      SingrauliSuryaKiranBhawanDudhichuaNOgm3  \
1902                                     49.9
1903                                     70.2
1904                                     78.8
1905                                     68.4
1906                                     59.2
```

```
...                                                ...
8286                                              69.5
8287                                              64.6
8288                                              60.0
8289                                              54.0
8290                                              47.9

      SingrauliSuryaKiranBhawanDudhichuaNO2gm3  \
1902                                      70.4
1903                                      69.0
1904                                      72.9
1905                                      78.4
1906                                      72.5
...                                        ...
8286                                      95.1
8287                                      92.5
8288                                      93.0
8289                                      94.1
8290                                      94.0

      SingrauliSuryaKiranBhawanDudhichuaNOXppb  \
1902                                      78.0
1903                                      93.8
1904                                     102.8
1905                                      97.2
1906                                      86.8
...                                        ...
8286                                     107.1
8287                                     101.7
8288                                      98.2
8289                                      94.0
8290                                      88.9

      SingrauliSuryaKiranBhawanDudhichuaCOmgm3  \
1902                                      2.86
1903                                      3.15
1904                                      2.10
1905                                      1.02
1906                                      1.28
...                                        ...
8286                                      1.56
8287                                      1.50
8288                                      1.48
8289                                      1.44
8290                                      1.28

      SingrauliSuryaKiranBhawanDudhichuaSO2gm3  \
1902                                      49.7
1903                                      52.9
1904                                      49.4
```

```
1905                                          39.9
1906                                          36.1
...                                            ...
8286                                          23.7
8287                                          22.8
8288                                          23.2
8289                                          23.1
8290                                          22.2

      SingrauliSuryaKiranBhawanDudhichuaNH3gm3  \
1902                                          16.3
1903                                          18.5
1904                                          22.2
1905                                          20.2
1906                                          14.8
...                                            ...
8286                                          10.1
8287                                          10.7
8288                                          10.2
8289                                           9.7
8290                                           9.4

      SingrauliSuryaKiranBhawanDudhichuaOzonegm3  \
1902                                             NaN
1903                                             NaN
1904                                             NaN
1905                                             NaN
1906                                             NaN
...                                              ...
8286                                            22.9
8287                                            28.4
8288                                            34.7
8289                                            40.2
8290                                            48.2

      SingrauliSuryaKiranBhawanDudhichuaBenzenegm3
1902                                               0.4
1903                                               0.5
1904                                               0.5
1905                                               0.6
1906                                               0.5
...                                                ...
8286                                               0.1
8287                                               0.1
8288                                               0.1
8289                                               0.1
8290                                               0.1

[577 rows x 13 columns]
Exceedances for SingrauliSuryaKiranBhawanDudhichuaNO2gm3:
```

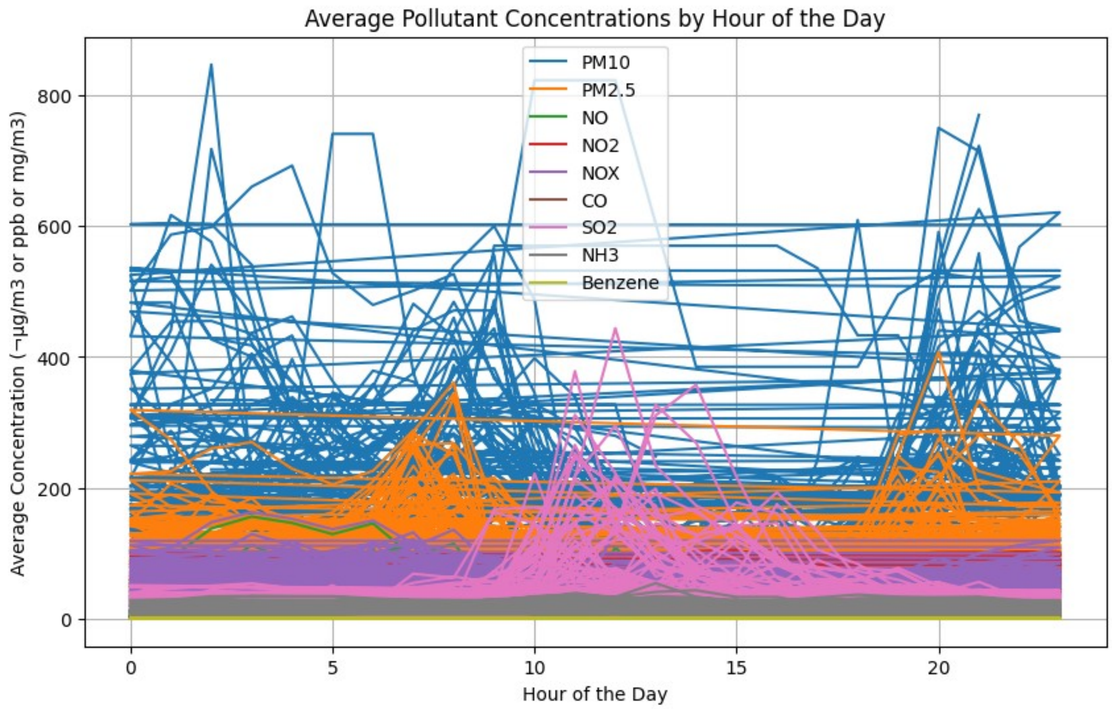
```
                           From       ToInterval15M  \
0        1 2023-02-01 00:00:00 2023-02-01 00:15:00
1        2 2023-02-01 00:15:00 2023-02-01 00:30:00
2        3 2023-02-01 00:30:00 2023-02-01 00:45:00
3        4 2023-02-01 00:45:00 2023-02-01 01:00:00
4        5 2023-02-01 01:00:00 2023-02-01 01:15:00
...    ...                 ...                 ...
8635  8636 2023-05-01 22:45:00 2023-05-01 23:00:00
8636  8637 2023-05-01 23:00:00 2023-05-01 23:15:00
8637  8638 2023-05-01 23:15:00 2023-05-01 23:30:00
8638  8639 2023-05-01 23:30:00 2023-05-01 23:45:00
8639  8640 2023-05-01 23:45:00 2023-05-02 00:00:00

      SingrauliSuryaKiranBhawanDudhichuaPM10gm3  \
0                                          95.0
1                                          95.0
2                                          95.0
3                                         122.0
4                                         122.0
...                                         ...
8635                                       19.0
8636                                       19.0
8637                                       19.0
8638                                       19.0
8639                                       32.0

      SingrauliSuryaKiranBhawanDudhichuaPM25gm3  \
0                                          35.0
1                                          35.0
2                                          35.0
3                                          34.0
4                                          34.0
...                                         ...
8635                                       11.0
8636                                       11.0
8637                                       11.0
8638                                       11.0
8639                                        6.0

      SingrauliSuryaKiranBhawanDudhichuaNOgm3  \
0                                         NaN
1                                         NaN
2                                         NaN
3                                         NaN
4                                         NaN
...                                       ...
8635                                     17.9
8636                                     17.9
8637                                     19.6
8638                                     20.8
```

```
8639                                       21.8

      SingrauliSuryaKiranBhawanDudhichuaNO2gm3  \
0                                         90.1
1                                         88.0
2                                         87.7
3                                         88.9
4                                         90.0
...                                        ...
8635                                     100.0
8636                                     100.0
8637                                     100.2
8638                                     100.2
8639                                      98.8

      SingrauliSuryaKiranBhawanDudhichuaNOXppb  \
0                                         56.2
1                                         55.1
2                                         55.2
3                                         55.7
4                                         55.8
...                                        ...
8635                                      67.8
8636                                      67.7
8637                                      69.2
8638                                      70.2
8639                                      70.3

      SingrauliSuryaKiranBhawanDudhichuaCOmgm3  \
0                                         0.31
1                                         0.33
2                                         0.38
3                                         0.38
4                                         0.38
...                                        ...
8635                                      0.63
8636                                      0.57
8637                                      0.58
8638                                      0.58
8639                                       NaN

      SingrauliSuryaKiranBhawanDudhichuaSO2gm3  \
0                                          NaN
1                                          NaN
2                                          NaN
3                                          NaN
4                                          NaN
...                                        ...
8635                                      10.0
8636                                      10.0
```

```
8637                                             9.9
8638                                             9.5
8639                                             NaN

      SingrauliSuryaKiranBhawanDudhichuaNH3gm3  \
0                                             17.7
1                                             18.3
2                                             19.7
3                                             21.3
4                                             22.3
...                                            ...
8635                                          10.7
8636                                          10.4
8637                                          10.5
8638                                          10.8
8639                                          11.0

      SingrauliSuryaKiranBhawanDudhichuaOzonegm3  \
0                                               28.1
1                                               27.1
2                                               24.9
3                                               21.9
4                                               16.7
...                                              ...
8635                                            26.1
8636                                            30.9
8637                                            29.6
8638                                            30.0
8639                                            33.5

      SingrauliSuryaKiranBhawanDudhichuaBenzenegm3
0                                                0.4
1                                                0.4
2                                                0.4
3                                                0.4
4                                                0.4
...                                              ...
8635                                             0.1
8636                                             0.1
8637                                             0.1
8638                                             0.1
8639                                             0.1

[8172 rows x 13 columns]

Exceedance Counts:
{'SingrauliSuryaKiranBhawanDudhichuaPM25gm3': 4999,
'SingrauliSuryaKiranBhawanDudhichuaNOgm3': 577,
'SingrauliSuryaKiranBhawanDudhichuaNO2gm3': 8172}
```

```
import pandas as pd
import matplotlib.pyplot as plt

# Convert the 'From' column to datetime format
data['Timestamp'] = data['From']

# Group data by hourly intervals and calculate average pollutant
concentrations
hourly_data = data.resample('H', on='Timestamp').mean()


# Plot average pollutant concentrations by hour of the day
plt.figure(figsize=(10, 6))
plt.plot(hourly_data.index.hour,
hourly_data['SingrauliSuryaKiranBhawanDudhichuaPM10gm3'],
label='PM10')
plt.plot(hourly_data.index.hour,
hourly_data['SingrauliSuryaKiranBhawanDudhichuaPM25gm3'],
label='PM2.5')
plt.plot(hourly_data.index.hour,
hourly_data['SingrauliSuryaKiranBhawanDudhichuaNOgm3'], label='NO')
plt.plot(hourly_data.index.hour,
hourly_data['SingrauliSuryaKiranBhawanDudhichuaNO2gm3'], label='NO2')
plt.plot(hourly_data.index.hour,
hourly_data['SingrauliSuryaKiranBhawanDudhichuaNOXppb'], label='NOX')
plt.plot(hourly_data.index.hour,
hourly_data['SingrauliSuryaKiranBhawanDudhichuaCOmgm3'], label='CO')
plt.plot(hourly_data.index.hour,
hourly_data['SingrauliSuryaKiranBhawanDudhichuaSO2gm3'], label='SO2')
plt.plot(hourly_data.index.hour,
hourly_data['SingrauliSuryaKiranBhawanDudhichuaNH3gm3'], label='NH3')
plt.plot(hourly_data.index.hour,
hourly_data['SingrauliSuryaKiranBhawanDudhichuaBenzenegm3'],
label='Benzene')

plt.xlabel('Hour of the Day')
plt.ylabel('Average Concentration (¬µg/m3 or ppb or mg/m3)')
plt.title('Average Pollutant Concentrations by Hour of the Day')
plt.legend()
plt.grid(True)
plt.show()
```

```
<ipython-input-58-df9fc27dfa0a>:8: FutureWarning: The default value of
numeric_only in DataFrameGroupBy.mean is deprecated. In a future
version, numeric_only will default to False. Either specify
numeric_only or select only columns which should be valid for the
function.
  hourly_data = data.resample('H', on='Timestamp').mean()
```



Forecasting

I used linear interpolated dataset to predict future by training the model for 2months (5760 rows as each row denoted 15mins) and tested it on to predict 15 days and stored the new file of future predicted 15 days. I have also plotted the future prediction of 15 days with the last 15 days of dataset provided

```
import pandas as pd
from statsmodels.tsa.arima.model import ARIMA

# Read the dataset
data = pd.read_csv('/content/linear_filled.csv')
data = data.iloc[:-3]
# Preprocess the dataset
# Clean column names
data.columns = data.columns.str.replace('[^a-zA-Z0-9]', '')

# Convert date columns to datetime format
data['From'] = pd.to_datetime(data['From'])
data['ToInterval15M'] = pd.to_datetime(data['ToInterval15M'])


# Fill missing values in each column using ARIMA
pollutants = [
    'SingrauliSuryaKiranBhawanDudhichuaPM10gm3',
    'SingrauliSuryaKiranBhawanDudhichuaPM25gm3',
```

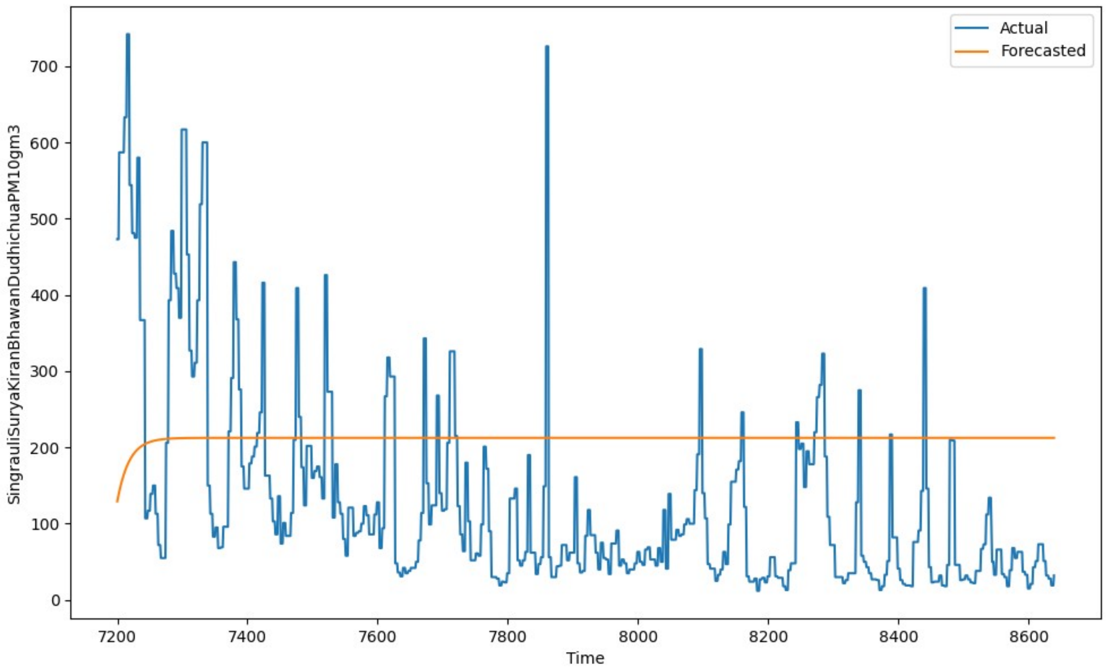
```
    'SingrauliSuryaKiranBhawanDudhichuaNOgm3',
    'SingrauliSuryaKiranBhawanDudhichuaNO2gm3',
    'SingrauliSuryaKiranBhawanDudhichuaNOXppb',
    'SingrauliSuryaKiranBhawanDudhichuaCOmgm3',
    'SingrauliSuryaKiranBhawanDudhichuaSO2gm3',
    'SingrauliSuryaKiranBhawanDudhichuaNH3gm3',
    'SingrauliSuryaKiranBhawanDudhichuaOzonegm3',
    'SingrauliSuryaKiranBhawanDudhichuaBenzenegm3'
]
# Split the data into training and test sets
train_data = data.iloc[:-5760]  # Use the first 3 months of data for
training
test_data = data.iloc[-1440:]  # Use the last month of data for
testing

# Perform forecasting for each pollutant
forecasted_data = pd.DataFrame()
for pollutant in pollutants:
    # Build and train the ARIMA model
    model = ARIMA(train_data[pollutant], order=(1, 0, 0))  # (p, d, q)
parameters for ARIMA model
    model_fit = model.fit()

    # Forecast the next month's data
    forecast = model_fit.get_forecast(steps=1440)  # Forecast for the
next month (30 days x 24 hours = 720 steps)
    forecasted_values = forecast.predicted_mean

    # Store the forecasted values for the pollutant in a DataFrame
    forecasted_data[pollutant] = forecasted_values

# Print the forecasted data for the 4th month
print(forecasted_data)

fig, ax = plt.subplots(len(pollutants), figsize=(10, 6 *
len(pollutants)))
for i, pollutant in enumerate(pollutants):
    ax[i].plot(test_data.index, test_data[pollutant], label='Actual')
    ax[i].plot(test_data.index, forecasted_data[pollutant],
label='Forecasted')
    ax[i].set_xlabel('Time')
    ax[i].set_ylabel(pollutant)
    ax[i].legend()

plt.tight_layout()
plt.show()

# Store the forecasted data in a new DataFrame or export it to a file
```

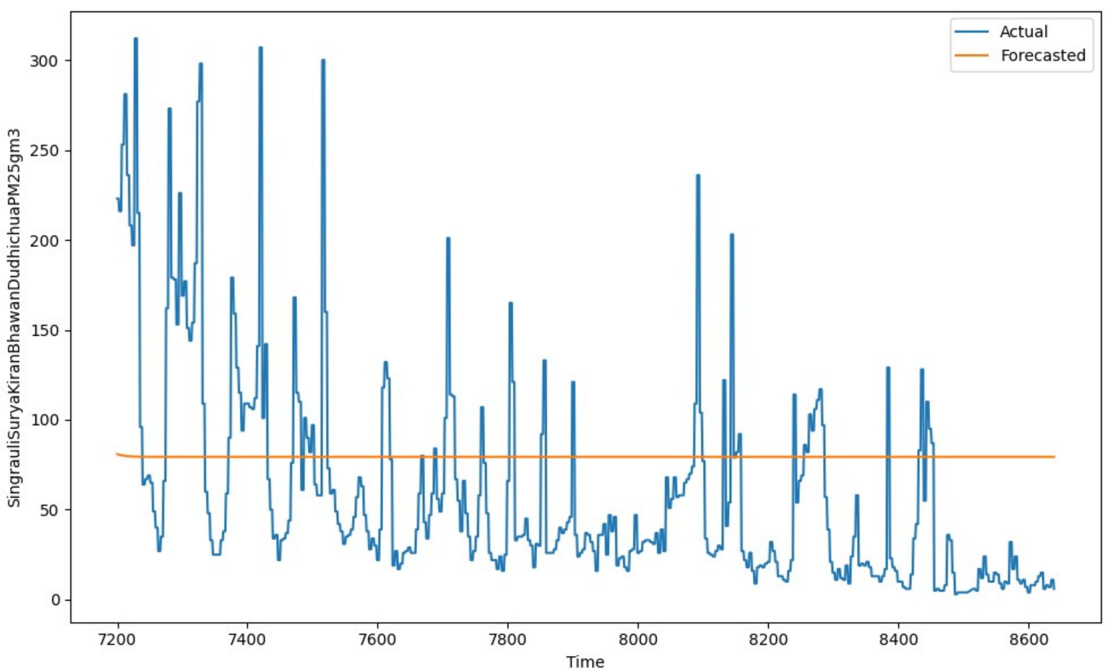
```
if needed
forecasted_data.to_csv('forecasted_data.csv')
```

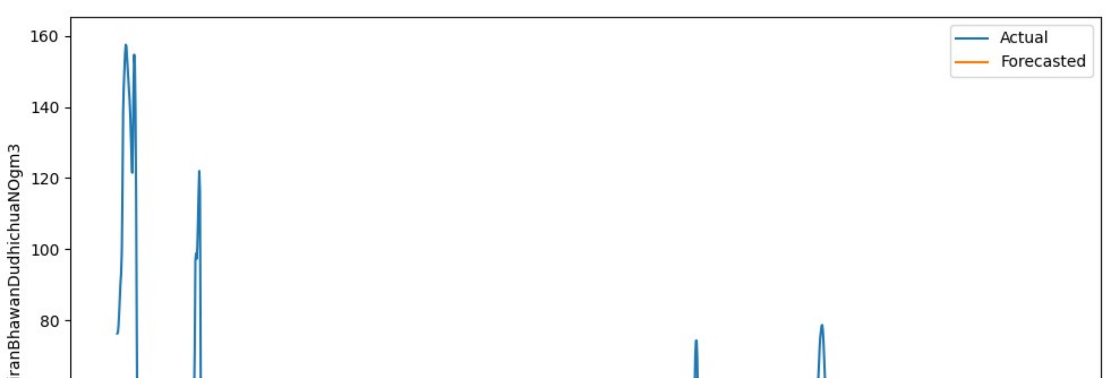
```
<ipython-input-59-9686256baeb9>:9: FutureWarning: The default value of
regex will change from True to False in a future version.
  data.columns = data.columns.str.replace('[^a-zA-Z0-9]', '')
/usr/local/lib/python3.10/dist-packages/statsmodels/base/model.py:604:
ConvergenceWarning: Maximum Likelihood optimization failed to
converge. Check mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "

      SingrauliSuryaKiranBhawanDudhichuaPM10gm3  \
2880                                 129.302180
2881                                 133.735914
2882                                 137.933530
2883                                 141.907604
2884                                 145.670039
...                                         ...
4315                                 212.557443
4316                                 212.557443
4317                                 212.557443
4318                                 212.557443
4319                                 212.557443

      SingrauliSuryaKiranBhawanDudhichuaPM25gm3  \
2880                                  80.868501
2881                                  80.747686
2882                                  80.636685
2883                                  80.534701
2884                                  80.441003
...                                         ...
4315                                  79.381338
4316                                  79.381338
4317                                  79.381338
4318                                  79.381338
4319                                  79.381338

      SingrauliSuryaKiranBhawanDudhichuaNOgm3  \
2880                                17.596747
2881                                17.593574
2882                                17.590479
2883                                17.587460
2884                                17.584515
...                                       ...
4315                                17.467484
4316                                17.467484
4317                                17.467484
4318                                17.467484
4319                                17.467484
```

```
      SingrauliSuryaKiranBhawanDudhichuaNO2gm3  \
2880                                  99.200580
2881                                  98.708492
2882                                  98.223629
2883                                  97.745884
2884                                  97.275153
...                                         ...
4315                                  65.682144
4316                                  65.682144
4317                                  65.682144
4318                                  65.682144
4319                                  65.682144

      SingrauliSuryaKiranBhawanDudhichuaNOXppb  \
2880                                  67.068505
2881                                  66.840238
2882                                  66.615155
2883                                  66.393210
2884                                  66.174361
...                                         ...
4315                                  50.700363
4316                                  50.700363
4317                                  50.700363
4318                                  50.700363
4319                                  50.700363

      SingrauliSuryaKiranBhawanDudhichuaCOmgm3  \
2880                                   1.340307
2881                                   1.326390
2882                                   1.313207
2883                                   1.300720
2884                                   1.288891
...                                         ...
4315                                   1.076536
4316                                   1.076536
4317                                   1.076536
4318                                   1.076536
4319                                   1.076536

      SingrauliSuryaKiranBhawanDudhichuaSO2gm3  \
2880                                  36.873230
2881                                  37.122304
2882                                  37.349359
2883                                  37.556341
2884                                  37.745024
...                                         ...
4315                                  39.690619
4316                                  39.690619
4317                                  39.690619
4318                                  39.690619
```

```
4319                                 39.690619

      SingrauliSuryaKiranBhawanDudhichuaNH3gm3  \
2880                                  7.583914
2881                                  7.764834
2882                                  7.942808
2883                                  8.117884
2884                                  8.290109
...                                        ...
4315                                 18.695524
4316                                 18.695524
4317                                 18.695524
4318                                 18.695524
4319                                 18.695524

      SingrauliSuryaKiranBhawanDudhichuaOzonegm3  \
2880                                   10.939213
2881                                   11.272631
2882                                   11.600354
2883                                   11.922478
2884                                   12.239100
...                                          ...
4315                                   30.456917
4316                                   30.456917
4317                                   30.456917
4318                                   30.456917
4319                                   30.456917

      SingrauliSuryaKiranBhawanDudhichuaBenzenegm3
2880                                      0.102030
2881                                      0.103997
2882                                      0.105904
2883                                      0.107752
2884                                      0.109543
...                                            ...
4315                                      0.166008
4316                                      0.166008
4317                                      0.166008
4318                                      0.166008
4319                                      0.166008

[1440 rows x 10 columns]
```


Actual
Forecasted
SingrauliSuryaKiranBhawanDudhichuaPM10gm3
Time


Actual
Forecasted
SingrauliSuryaKiranBhawanDudhichuaPM25gm3
Time


Actual
Forecasted
ranBhawanDudhichuaNOgm3

DESCRIPTIVE ANALYSIS

Descriptive analysis: Identifies patterns in time series data at the time of coal India open-pit blasting effect, in coal india blasting effect time is 13:45 pm to 14:45 pm.

```
import pandas as pd

# Read the dataset
df = pd.read_csv('/content/Open pit blasting 01-02-2023 000000 To 01-05-2023 235959.csv')
df = df.iloc[:-3]


# Convert date and time columns to datetime format
df['From'] = pd.to_datetime(df['From'], dayfirst=True)
df['To (Interval: 15M)'] = pd.to_datetime(df['To (Interval: 15M)'], dayfirst=True)

# Set the index of the DataFrame as the 'From' column
df.set_index('From', inplace=True)

# Filter the dataset to include only data during the blasting effect period
start_time = '13:45:00'
end_time = '14:45:00'
blast_period_data = df.between_time(start_time, end_time)

# Calculate measures of frequency
frequency = blast_period_data.shape[0]
print("Frequency of air pollution measurements during blasting effect period:", frequency)

# Calculate measures of central tendency
mean = blast_period_data.mean()
median = blast_period_data.median()
mode = blast_period_data.mode().iloc[0]
print("Mean pollution levels during blasting effect period:\n", mean)
print("\nMedian pollution levels during blasting effect period:\n", median)
print("\nMode of pollution levels during blasting effect period:\n", mode)

# Calculate measures of dispersion or variation
range_val = blast_period_data.max() - blast_period_data.min()
std_dev = blast_period_data.std()
variance = blast_period_data.var()
print("\nRange of pollution levels during blasting effect period:\n", range_val)
print("\nStandard deviation of pollution levels during blasting effect
```

```
period:\n", std_dev)
print("\nVariance of pollution levels during blasting effect period:\
n", variance)

# Calculate measures of position
percentile_25 = blast_period_data.quantile(0.25)
percentile_50 = blast_period_data.quantile(0.50)
percentile_75 = blast_period_data.quantile(0.75)
print("\n25th percentile of pollution levels during blasting effect
period:\n", percentile_25)
print("\n50th percentile (median) of pollution levels during blasting
effect period:\n", percentile_50)
print("\n75th percentile of pollution levels during blasting effect
period:\n", percentile_75)
```

```
Frequency of air pollution measurements during blasting effect period:
450
Mean pollution levels during blasting effect period:
 #
4330.000000
Singrauli, Surya Kiran Bhawan Dudhichua  PM10 (µg/m3)
109.045070
Singrauli, Surya Kiran Bhawan Dudhichua  PM2.5 (µg/m3)
34.732719
Singrauli, Surya Kiran Bhawan Dudhichua  NO (µg/m3)
5.507595
Singrauli, Surya Kiran Bhawan Dudhichua  NO2 (µg/m3)
47.611486
Singrauli, Surya Kiran Bhawan Dudhichua  NOX (ppb)
29.928378
Singrauli, Surya Kiran Bhawan Dudhichua  CO (mg/m3)
1.070167
Singrauli, Surya Kiran Bhawan Dudhichua  SO2 (µg/m3)
53.607634
Singrauli, Surya Kiran Bhawan Dudhichua  NH3 (µg/m3)
12.756306
Singrauli, Surya Kiran Bhawan Dudhichua  Ozone (µg/m3)
66.457175
Singrauli, Surya Kiran Bhawan Dudhichua  Benzene (µg/m3)
0.104762
dtype: float64

Median pollution levels during blasting effect period:
 #                                                              4330.00
Singrauli, Surya Kiran Bhawan Dudhichua  PM10 (µg/m3)             96.00
Singrauli, Surya Kiran Bhawan Dudhichua  PM2.5 (µg/m3)            31.00
Singrauli, Surya Kiran Bhawan Dudhichua  NO (µg/m3)                3.80
Singrauli, Surya Kiran Bhawan Dudhichua  NO2 (µg/m3)              41.80
Singrauli, Surya Kiran Bhawan Dudhichua  NOX (ppb)                25.85
Singrauli, Surya Kiran Bhawan Dudhichua  CO (mg/m3)                1.01
```

```
Singrauli, Surya Kiran Bhawan Dudhichua  SO2 (µg/m3)          28.10
Singrauli, Surya Kiran Bhawan Dudhichua  NH3 (µg/m3)          10.10
Singrauli, Surya Kiran Bhawan Dudhichua  Ozone (µg/m3)        68.00
Singrauli, Surya Kiran Bhawan Dudhichua  Benzene (µg/m3)       0.10
dtype: float64

Mode of pollution levels during blasting effect period:
 #
56
To (Interval: 15M)                                             2023-02-01
14:00:00
Singrauli, Surya Kiran Bhawan Dudhichua  PM10 (µg/m3)
85.0
Singrauli, Surya Kiran Bhawan Dudhichua  PM2.5 (µg/m3)
30.0
Singrauli, Surya Kiran Bhawan Dudhichua  NO (µg/m3)
3.4
Singrauli, Surya Kiran Bhawan Dudhichua  NO2 (µg/m3)
27.9
Singrauli, Surya Kiran Bhawan Dudhichua  NOX (ppb)
17.8
Singrauli, Surya Kiran Bhawan Dudhichua  CO (mg/m3)
1.06
Singrauli, Surya Kiran Bhawan Dudhichua  SO2 (µg/m3)
17.0
Singrauli, Surya Kiran Bhawan Dudhichua  NH3 (µg/m3)
10.8
Singrauli, Surya Kiran Bhawan Dudhichua  Ozone (µg/m3)
74.4
Singrauli, Surya Kiran Bhawan Dudhichua  Benzene (µg/m3)
0.1
Name: 0, dtype: object

Range of pollution levels during blasting effect period:
 #
8548
To (Interval: 15M)                                             89 days
01:00:00
Singrauli, Surya Kiran Bhawan Dudhichua  PM10 (µg/m3)
555.0
Singrauli, Surya Kiran Bhawan Dudhichua  PM2.5 (µg/m3)
98.0
Singrauli, Surya Kiran Bhawan Dudhichua  NO (µg/m3)
60.6
Singrauli, Surya Kiran Bhawan Dudhichua  NO2 (µg/m3)
93.1
Singrauli, Surya Kiran Bhawan Dudhichua  NOX (ppb)
91.2
Singrauli, Surya Kiran Bhawan Dudhichua  CO (mg/m3)
2.64
```

```
Singrauli, Surya Kiran Bhawan Dudhichua  SO2 (µg/m3)
392.7
Singrauli, Surya Kiran Bhawan Dudhichua  NH3 (µg/m3)
42.6
Singrauli, Surya Kiran Bhawan Dudhichua  Ozone (µg/m3)
97.1
Singrauli, Surya Kiran Bhawan Dudhichua  Benzene (µg/m3)
0.1
dtype: object

Standard deviation of pollution levels during blasting effect period:
 #
2496.775337
To (Interval: 15M)                                          26 days
00:11:37.803111455
Singrauli, Surya Kiran Bhawan Dudhichua  PM10 (µg/m3)
78.624677
Singrauli, Surya Kiran Bhawan Dudhichua  PM2.5 (µg/m3)
18.66138
Singrauli, Surya Kiran Bhawan Dudhichua  NO (µg/m3)
7.747044
Singrauli, Surya Kiran Bhawan Dudhichua  NO2 (µg/m3)
20.502687
Singrauli, Surya Kiran Bhawan Dudhichua  NOX (ppb)
15.041362
Singrauli, Surya Kiran Bhawan Dudhichua  CO (mg/m3)
0.557918
Singrauli, Surya Kiran Bhawan Dudhichua  SO2 (µg/m3)
63.239623
Singrauli, Surya Kiran Bhawan Dudhichua  NH3 (µg/m3)
7.115346
Singrauli, Surya Kiran Bhawan Dudhichua  Ozone (µg/m3)
16.537547
Singrauli, Surya Kiran Bhawan Dudhichua  Benzene (µg/m3)
0.021381
dtype: object

Variance of pollution levels during blasting effect period:
 #
6.233887e+06
Singrauli, Surya Kiran Bhawan Dudhichua  PM10 (µg/m3)
6.181840e+03
Singrauli, Surya Kiran Bhawan Dudhichua  PM2.5 (µg/m3)
3.482471e+02
Singrauli, Surya Kiran Bhawan Dudhichua  NO (µg/m3)
6.001669e+01
Singrauli, Surya Kiran Bhawan Dudhichua  NO2 (µg/m3)
4.203602e+02
Singrauli, Surya Kiran Bhawan Dudhichua  NOX (ppb)
2.262426e+02
```

```
Singrauli, Surya Kiran Bhawan Dudhichua  CO (mg/m3)         3.112729e-
01
Singrauli, Surya Kiran Bhawan Dudhichua  SO2 (µg/m3)
3.999250e+03
Singrauli, Surya Kiran Bhawan Dudhichua  NH3 (µg/m3)
5.062815e+01
Singrauli, Surya Kiran Bhawan Dudhichua  Ozone (µg/m3)
2.734904e+02
Singrauli, Surya Kiran Bhawan Dudhichua  Benzene (µg/m3)    4.571429e-
04
dtype: float64

25th percentile of pollution levels during blasting effect period:
 #                                                           2170.250
Singrauli, Surya Kiran Bhawan Dudhichua  PM10 (µg/m3)          65.000
Singrauli, Surya Kiran Bhawan Dudhichua  PM2.5 (µg/m3)         22.000
Singrauli, Surya Kiran Bhawan Dudhichua  NO (µg/m3)             3.200
Singrauli, Surya Kiran Bhawan Dudhichua  NO2 (µg/m3)           29.500
Singrauli, Surya Kiran Bhawan Dudhichua  NOX (ppb)             18.200
Singrauli, Surya Kiran Bhawan Dudhichua  CO (mg/m3)             0.640
Singrauli, Surya Kiran Bhawan Dudhichua  SO2 (µg/m3)           16.800
Singrauli, Surya Kiran Bhawan Dudhichua  NH3 (µg/m3)            8.675
Singrauli, Surya Kiran Bhawan Dudhichua  Ozone (µg/m3)         60.050
Singrauli, Surya Kiran Bhawan Dudhichua  Benzene (µg/m3)        0.100
Name: 0.25, dtype: float64

50th percentile (median) of pollution levels during blasting effect
period:
 #                                                           4330.00
Singrauli, Surya Kiran Bhawan Dudhichua  PM10 (µg/m3)          96.00
Singrauli, Surya Kiran Bhawan Dudhichua  PM2.5 (µg/m3)         31.00
Singrauli, Surya Kiran Bhawan Dudhichua  NO (µg/m3)             3.80
Singrauli, Surya Kiran Bhawan Dudhichua  NO2 (µg/m3)           41.80
Singrauli, Surya Kiran Bhawan Dudhichua  NOX (ppb)             25.85
Singrauli, Surya Kiran Bhawan Dudhichua  CO (mg/m3)             1.01
Singrauli, Surya Kiran Bhawan Dudhichua  SO2 (µg/m3)           28.10
Singrauli, Surya Kiran Bhawan Dudhichua  NH3 (µg/m3)           10.10
Singrauli, Surya Kiran Bhawan Dudhichua  Ozone (µg/m3)         68.00
Singrauli, Surya Kiran Bhawan Dudhichua  Benzene (µg/m3)        0.10
Name: 0.5, dtype: float64

75th percentile of pollution levels during blasting effect period:
 #                                                           6489.750
Singrauli, Surya Kiran Bhawan Dudhichua  PM10 (µg/m3)         132.000
Singrauli, Surya Kiran Bhawan Dudhichua  PM2.5 (µg/m3)         44.000
Singrauli, Surya Kiran Bhawan Dudhichua  NO (µg/m3)             4.600
Singrauli, Surya Kiran Bhawan Dudhichua  NO2 (µg/m3)           60.825
Singrauli, Surya Kiran Bhawan Dudhichua  NOX (ppb)             36.125
Singrauli, Surya Kiran Bhawan Dudhichua  CO (mg/m3)             1.475
Singrauli, Surya Kiran Bhawan Dudhichua  SO2 (µg/m3)           69.200
```

```
Singrauli, Surya Kiran Bhawan Dudhichua  NH3 (µg/m3)         12.325
Singrauli, Surya Kiran Bhawan Dudhichua  Ozone (µg/m3)       76.500
Singrauli, Surya Kiran Bhawan Dudhichua  Benzene (µg/m3)      0.100
Name: 0.75, dtype: float64

<ipython-input-60-6d96b353d197>:25: FutureWarning: DataFrame.mean and
DataFrame.median with numeric_only=None will include datetime64 and
datetime64tz columns in a future version.
  mean = blast_period_data.mean()
<ipython-input-60-6d96b353d197>:26: FutureWarning: DataFrame.mean and
DataFrame.median with numeric_only=None will include datetime64 and
datetime64tz columns in a future version.
  median = blast_period_data.median()
<ipython-input-60-6d96b353d197>:35: FutureWarning: The default value
of numeric_only in DataFrame.var is deprecated. In a future version,
it will default to False. In addition, specifying 'numeric_only=None'
is deprecated. Select only valid columns or specify the value of
numeric_only to silence this warning.
  variance = blast_period_data.var()
<ipython-input-60-6d96b353d197>:41: FutureWarning: The default value
of numeric_only in DataFrame.quantile is deprecated. In a future
version, it will default to False. Select only valid columns or
specify the value of numeric_only to silence this warning.
  percentile_25 = blast_period_data.quantile(0.25)
<ipython-input-60-6d96b353d197>:42: FutureWarning: The default value
of numeric_only in DataFrame.quantile is deprecated. In a future
version, it will default to False. Select only valid columns or
specify the value of numeric_only to silence this warning.
  percentile_50 = blast_period_data.quantile(0.50)
<ipython-input-60-6d96b353d197>:43: FutureWarning: The default value
of numeric_only in DataFrame.quantile is deprecated. In a future
version, it will default to False. Select only valid columns or
specify the value of numeric_only to silence this warning.
  percentile_75 = blast_period_data.quantile(0.75)
```