BSc (Hons) Artificial Intelligence and Data Science

Academic Year	2024
Semester	2
Module Number	CM2606
Module Title	Data Engineering
Assessment Method	Coursework
Deadline	21st April 2024
Module Co-Ordinator	Mr. Mohamed Ayoob

Name	Seth Rajarathne
IIT Student ID	20211344
RGU ID	2237948

Table of Contents

1. Introduction

Formaldehyde (HCHO) has an impact on both public health and air quality, hence its levels in the atmosphere must be monitored. HCHO is a harmful air pollutant that has been connected to a variety of health concerns. To anticipate future HCHO levels, we examined HCHO data from seven major cities in Sri Lanka over a five-year period.

Our purpose is to,

- Analyze HCHO data in Sri Lanka using data engineering approaches.
- Recognizing the importance of HCHO monitoring in climate change studies and air quality.
- Understanding the probable sources of HCHO, as well as geographical and temporal trends.

This project includes the cities of Kurunegala, Monaragala, Jaffna, Nuwara Eliya, Kandy, and Matara. The data for each of these cities is cleansed, preprocessed, and shown to identify potential patterns. I then investigate the influence of exogenous influences on HCHO emissions. Time series techniques are employed to anticipate the HCHO's future values, and the findings are shown as a dashboard using Microsoft Power BI.

2. Data Preprocessing

Merge Dataset

```
# Define column names
cols = ["HCHO reading", "Location", "Current Date", "Next Date"]

# Load data from each CSV file
col_mat_nuw_data = pd.read_csv("col_mat_nuw_output.csv", names=cols)
mon_kur_jaf_data = pd.read_csv("mon_kur_jaf_output.csv", names=cols)
kan_data = pd.read_csv("kan_output.csv", names=cols)

# Merge data from all files
merged_data = pd.concat([col_mat_nuw_data,mon_kur_jaf_data,kan_data])
```

✓ 0.1s

```
merged_data
```

✓ 0.0s

	HCHO reading	Location	Current Date	Next Date
0	0.000197	Colombo Proper	2019-01-01	2019-01-02
1	0.000263	Colombo Proper	2019-01-02	2019-01-03
2	0.000099	Colombo Proper	2019-01-03	2019-01-04
3	0.000210	Colombo Proper	2019-01-04	2019-01-05
4	0.000179	Colombo Proper	2019-01-05	2019-01-06
...	...	...	...	...
1821	NaN	Kandy Proper	2023-12-27	2023-12-28
1822	NaN	Kandy Proper	2023-12-28	2023-12-29
1823	NaN	Kandy Proper	2023-12-29	2023-12-30
1824	0.000056	Kandy Proper	2023-12-30	2023-12-31
1825	NaN	Kandy Proper	2023-12-31	2024-01-01

Column names are defined as "HCHO reading", "Location", "Current Date", and "Next Date". It then imports data from three CSV files, each providing information on formaldehyde (HCHO) measurements at various places, using the provided column names. The files are "col_mat_nuw_output.csv", "mon_kur_jaf_output.csv", and "kan_output.csv". The code then combines the data from all three files into a single DataFrame named "merged_data" using the 'pd.concat()' method. This concatenation merges the rows from each file into a single dataset for further analysis or processing.

Handling Null Values

```
# Convert 'Current Date' column to datetime
merged_data['Current Date'] = pd.to_datetime(merged_data['Current Date'])

# Extract year and month
merged_data['Year'] = merged_data['Current Date'].dt.year
merged_data['Month'] = merged_data['Current Date'].dt.month

# Group by Location, Year, and Month and calculate mean
mean_by_location_year_month = merged_data.groupby(['Location', 'Year', 'Month'])['HCHO reading'].mean()

# Replace null values with corresponding means
merged_data['HCHO reading'] = merged_data.apply(
    lambda row: mean_by_location_year_month[row['Location'], row['Year'], row['Month']]
                if pd.isnull(row['HCHO reading']) else row['HCHO reading'],
    axis=1
)

# Drop the temporary columns 'Year' and 'Month'
merged_data.drop(['Year', 'Month'], axis=1, inplace=True)

# Print the updated DataFrame
print(merged_data)
```

[5] ✓ 0.5s

This code begins by converting the 'Current Date' column in the'merged_data' DataFrame to datetime format using the pd.to_datetime() method. It then pulls the year and month from the 'Current Date' column, generating new columns 'Year' and 'Month'. The data is then categorized by 'Location', 'Year', and 'Month', and the average 'HCHO reading' for each group is computed. Then it substitutes any null values in the 'HCHO reading' column with the mean derived for the location, year, and month group. After the null values are replaced, the temporary columns 'Year' and 'Month' are removed from the DataFrame. Finally, an updated DataFrame is printed. This code effectively preprocesses the data by managing missing values and guaranteeing consistency in the 'HCHO reading' column based on location, year, and month.

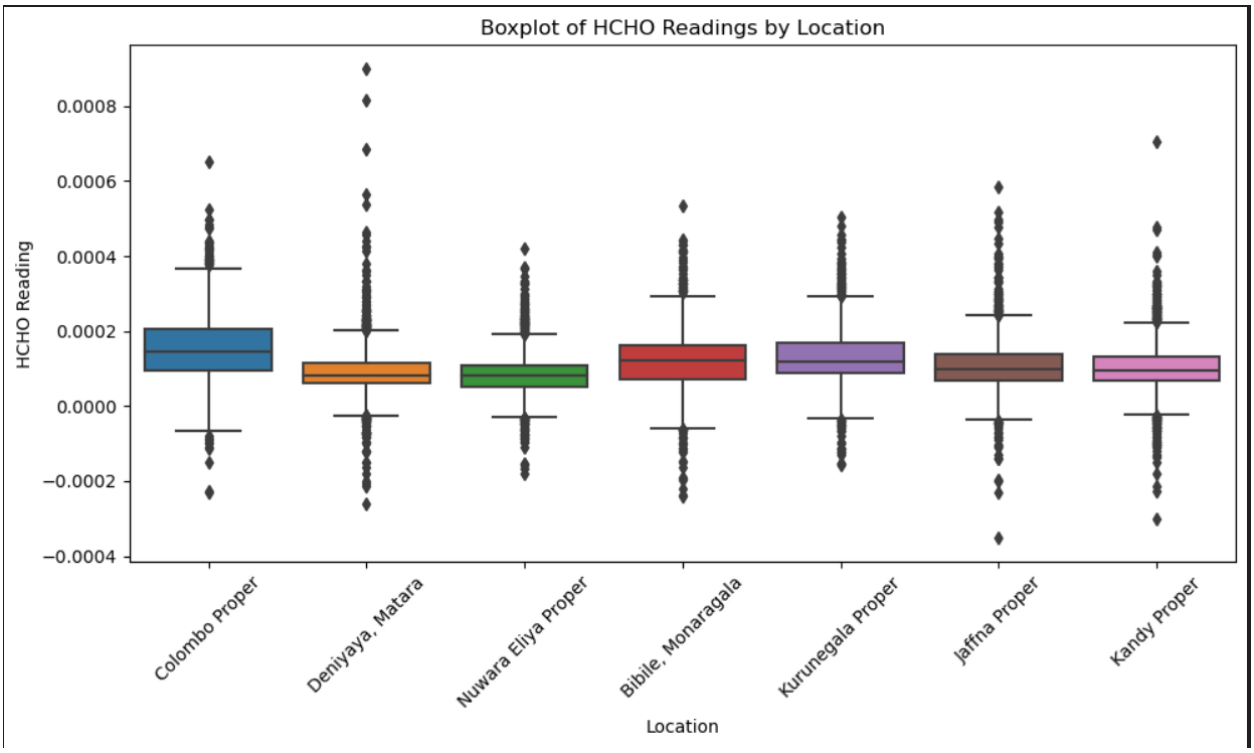
```
      HCHO reading        Location Current Date   Next Date
0         0.000197  Colombo Proper   2019-01-01  2019-01-02
1         0.000263  Colombo Proper   2019-01-02  2019-01-03
2         0.000099  Colombo Proper   2019-01-03  2019-01-04
3         0.000210  Colombo Proper   2019-01-04  2019-01-05
4         0.000179  Colombo Proper   2019-01-05  2019-01-06
...            ...             ...          ...         ...
1821      0.000057    Kandy Proper   2023-12-27  2023-12-28
1822      0.000057    Kandy Proper   2023-12-28  2023-12-29
1823      0.000057    Kandy Proper   2023-12-29  2023-12-30
1824      0.000056    Kandy Proper   2023-12-30  2023-12-31
1825      0.000057    Kandy Proper   2023-12-31  2024-01-01

[12782 rows x 4 columns]
```

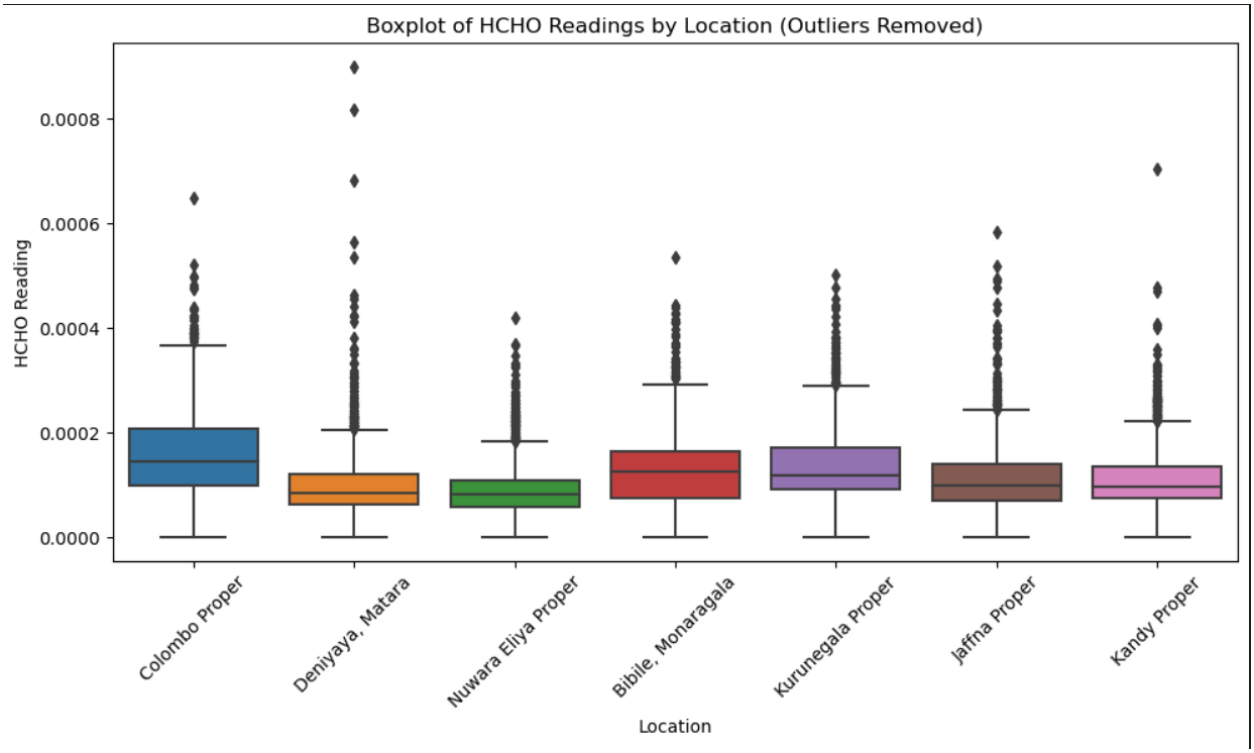
Handling Outliers

```
# Create boxplots to visualize outliers
plt.figure(figsize=(10, 6))
sns.boxplot(x='Location', y='HCHO reading', data=merged_data)
plt.title('Boxplot of HCHO Readings by Location')
plt.xlabel('Location')
plt.ylabel('HCHO Reading')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# Define a function to remove outliers (values below zero)
def remove_outliers_below_zero(df, column):
    df_filtered = df[df[column] >= 0]  # Remove values below zero
    return df_filtered

# Remove outliers (values below zero) from 'HCHO reading' column
merged_data = remove_outliers_below_zero(merged_data, 'HCHO reading')

# Plot the boxplot again after removing outliers
plt.figure(figsize=(10, 6))
sns.boxplot(x='Location', y='HCHO reading', data=merged_data)
plt.title('Boxplot of HCHO Readings by Location (Outliers Removed)')
plt.xlabel('Location')
plt.ylabel('HCHO Reading')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

✓ 0.9s

This code initially generates a boxplot to show outliers in the 'HCHO reading' column of the'merged_data' DataFrame, organized by location. It creates the plot with Matplotlib and Seaborn, labeling the x-axis 'Location' and the y-axis 'HCHO reading'. After presenting the initial boxplot, a function named'remove_outliers_below_zero' is defined to filter out outliers by deleting values less than zero from the chosen column. This function is then used to eliminate outliers from the 'HCHO reading' column of the'merged_data' dataframe. Finally, another boxplot is created to view the new data with no outliers, using the same plot parameters as previously. This method effectively displays a visual depiction of outliers in the formaldehyde measurements by location, and then proceeds to delete outliers below zero.


Boxplot of HCHO Readings by Location
HCHO Reading
0.0008
0.0006
0.0004
0.0002
0.0000
−0.0002
−0.0004
Colombo Proper
Deniyaya, Matara
Nuwara Eliya Proper
Bibile, Monaragala
Kurunegala Proper
Jaffna Proper
Kandy Proper
Location


Boxplot of HCHO Readings by Location (Outliers Removed)
HCHO Reading
0.0008
0.0006
0.0004
0.0002
0.0000
Colombo Proper
Deniyaya, Matara
Nuwara Eliya Proper
Bibile, Monaragala
Kurunegala Proper
Jaffna Proper
Kandy Proper
Location

Statistical Summery

```
# Group by Location and calculate statistics
summary_stats_by_city = merged_data.groupby('Location')['HCHO reading'].agg(['mean', 'median', 'std'])

# Calculate statistics across the entire dataset
overall_summary_stats = merged_data['HCHO reading'].agg(['mean', 'median', 'std'])

# Print summary statistics for each city
print("Summary statistics for each city:")
print(summary_stats_by_city)

# Print summary statistics across the entire dataset
print("\nOverall summary statistics:")
print(overall_summary_stats)
```

✓ 0.0s

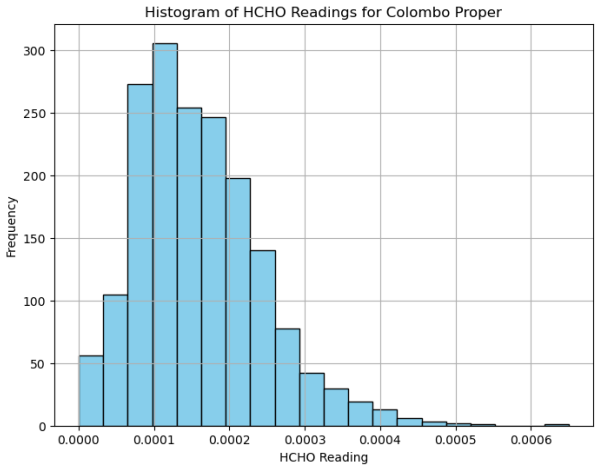
```
Summary statistics for each city:
                        mean    median       std
Location
Bibile, Monaragala   0.000130  0.000126  0.000069
Colombo Proper       0.000160  0.000146  0.000083
Deniyaya, Matara     0.000099  0.000084  0.000066
Jaffna Proper        0.000112  0.000101  0.000065
Kandy Proper         0.000111  0.000098  0.000062
Kurunegala Proper    0.000135  0.000119  0.000068
Nuwara Eliya Proper  0.000090  0.000083  0.000050

Overall summary statistics:
mean      0.000120
median    0.000105
std       0.000071
Name: HCHO reading, dtype: float64
```

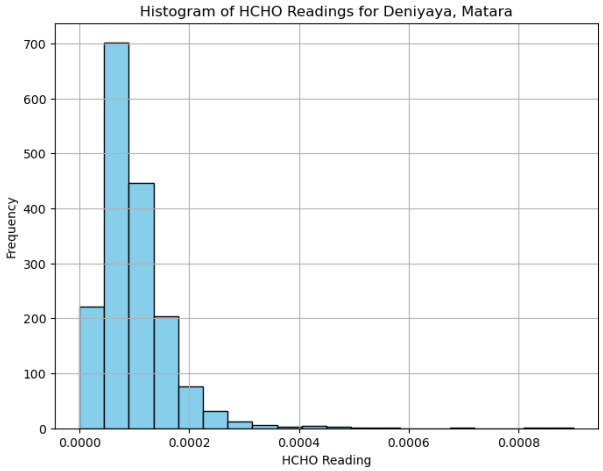
This code applies statistical analysis on the 'HCHO reading' column of the'merged_data' DataFrame. Initially, it organizes the data by location using the 'groupby()' function and produces three statistics (mean, median, and standard deviation) for each place using the 'agg()' method. The findings are saved in a new DataFrame called'summary_stats_by_city'. It then calculates the same statistics (mean, median, and standard deviation) for the whole dataset using the 'agg()' function on the 'HCHO reading' column, and saves the results in a new DataFrame named 'overall_summary_stats'. Finally, the code outputs the summary statistics for each city, as well as the overall summary statistics for the dataset. This code gives insight into the distribution of formaldehyde readings across different regions as well as an overview of the dataset's statistical properties.

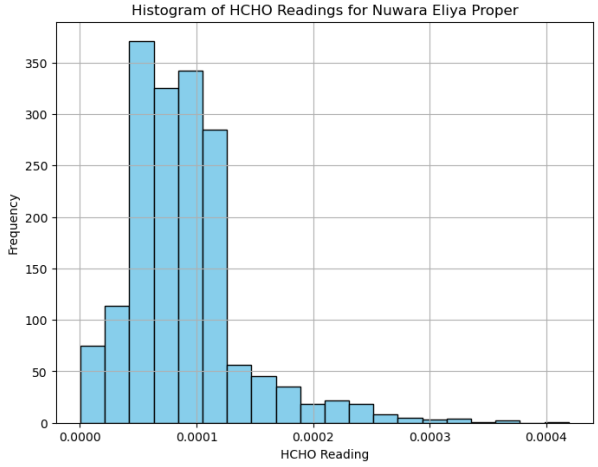
Visualization

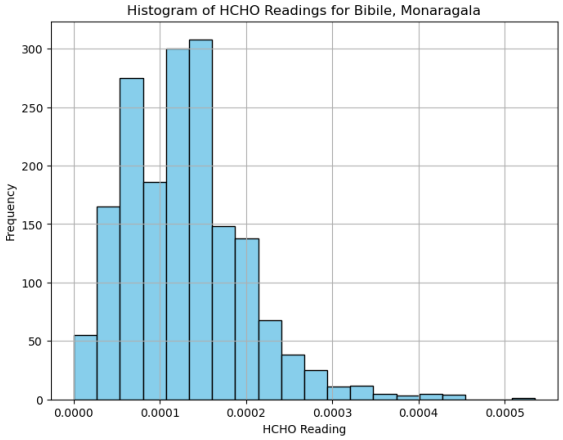
```
# Create histograms for each location
for location in unique_locations:
    # Filter DataFrame for the current location
    location_df = merged_data[merged_data['Location'] == location]

    # Create histogram
    plt.figure(figsize=(8, 6))
    plt.hist(location_df['HCHO reading'], bins=20, color='skyblue', edgecolor='black')
    plt.title(f'Histogram of HCHO Readings for {location}')
    plt.xlabel('HCHO Reading')
    plt.ylabel('Frequency')
    plt.grid(True)
    plt.show()

# Create boxplots for each location
for location in unique_locations:
    # Filter DataFrame for the current location
    location_df = merged_data[merged_data['Location'] == location]

    # Create boxplot
    plt.figure(figsize=(8, 6))
    sns.boxplot(x='Location', y='HCHO reading', data=location_df)
    plt.title(f'Boxplot of HCHO Readings for {location}')
    plt.xlabel('Location')
    plt.ylabel('HCHO Reading')
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()
```

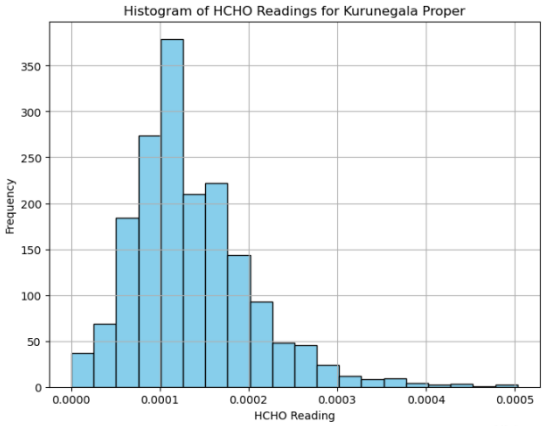
This code segment creates histograms and boxplots to show the distribution of formaldehyde (HCHO) measurements at each location in the dataset. It iterates over each distinct location, filtering the DataFrame to only contain data from the current location. For each location, it first generates a histogram using Matplotlib's hist() function, with the number of bins set to 20 to divide the data into intervals, and then shows the histogram with titles according to the individual location. It then uses Seaborn's boxplot() method to generate boxplots that display the distribution of HCHO measurements for each location. The boxplots depict the median, quartiles, and likely outliers, giving a picture of the data's distribution and central tendency for each location. Overall, this code segment visualizes the variance in HCHO values across multiple locations using histograms and boxplots.

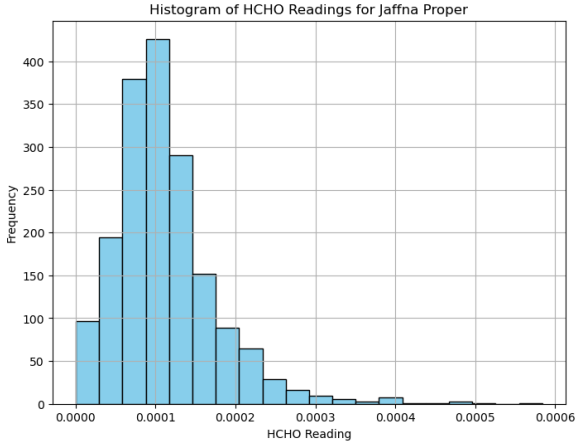

Histogram of HCHO Readings for Colombo Proper
Frequency
HCHO Reading

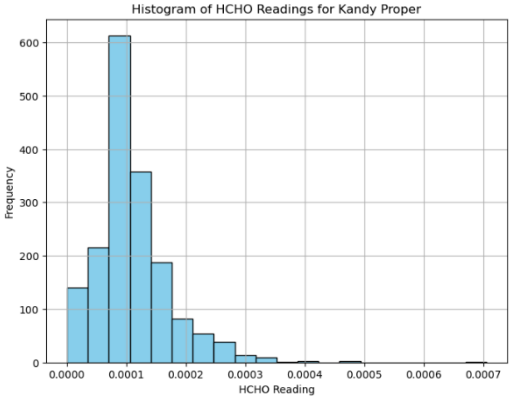

Histogram of HCHO Readings for Deniyaya, Matara
Frequency
HCHO Reading

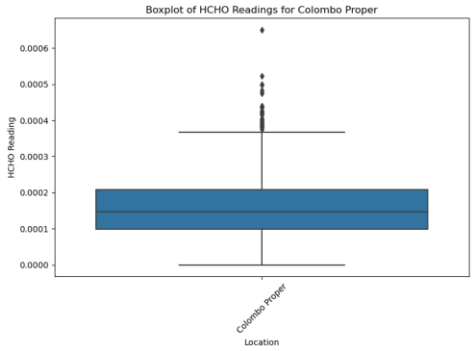

Histogram of HCHO Readings for Nuwara Eliya Proper
Frequency
HCHO Reading

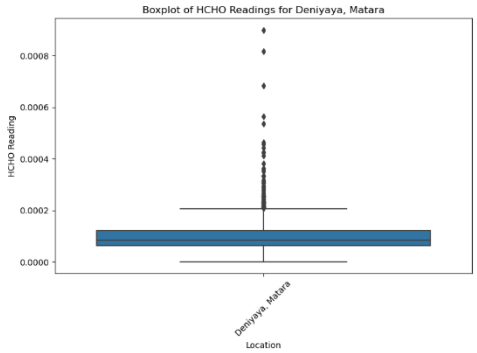

Histogram of HCHO Readings for Bibile, Monaragala
Frequency
HCHO Reading

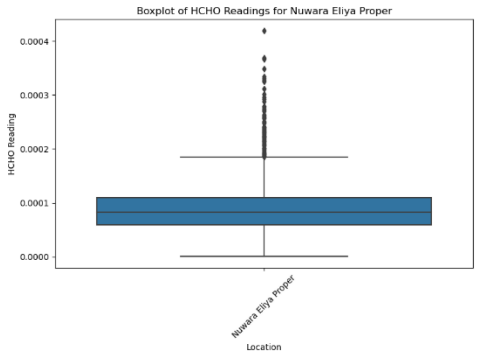

Histogram of HCHO Readings for Kurunegala Proper
Frequency
HCHO Reading

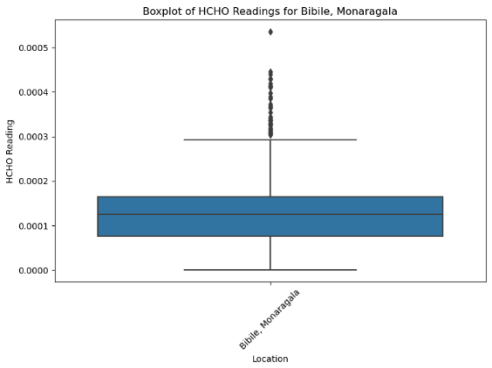

Histogram of HCHO Readings for Jaffna Proper
Frequency
HCHO Reading

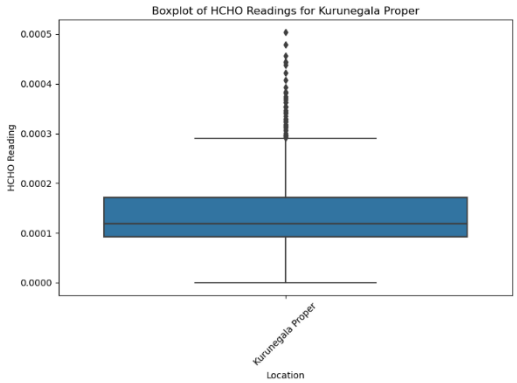

Histogram of HCHO Readings for Kandy Proper
Frequency
HCHO Reading

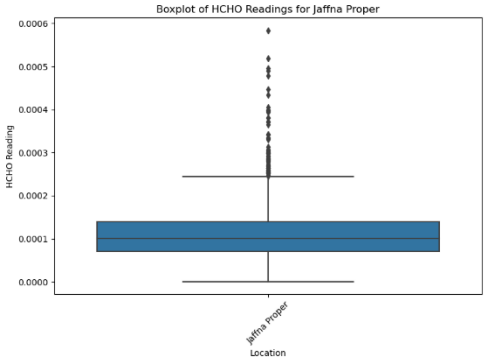

Boxplot of HCHO Readings for Colombo Proper
HCHO Reading
0.0006
0.0005
0.0004
0.0003
0.0002
0.0001
0.0000
Colombo Proper
Location

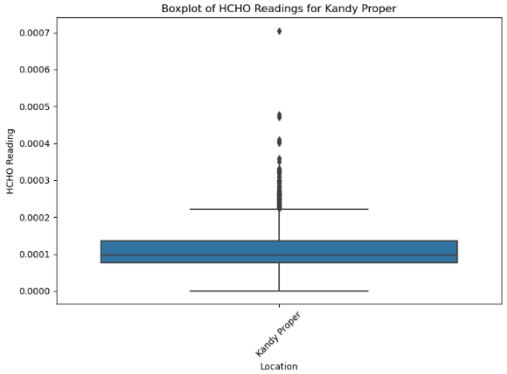

Boxplot of HCHO Readings for Deniyaya, Matara
HCHO Reading
0.0008
0.0006
0.0004
0.0002
0.0000
Deniyaya, Matara
Location


Boxplot of HCHO Readings for Nuwara Eliya Proper
HCHO Reading
0.0004
0.0003
0.0002
0.0001
0.0000
Nuwara Eliya Proper
Location


Boxplot of HCHO Readings for Bibile, Monaragala
HCHO Reading
0.0005
0.0004
0.0003
0.0002
0.0001
0.0000
Bibile, Monaragala
Location


Boxplot of HCHO Readings for Kurunegala Proper
HCHO Reading
0.0005
0.0004
0.0003
0.0002
0.0001
0.0000
Kurunegala Proper
Location

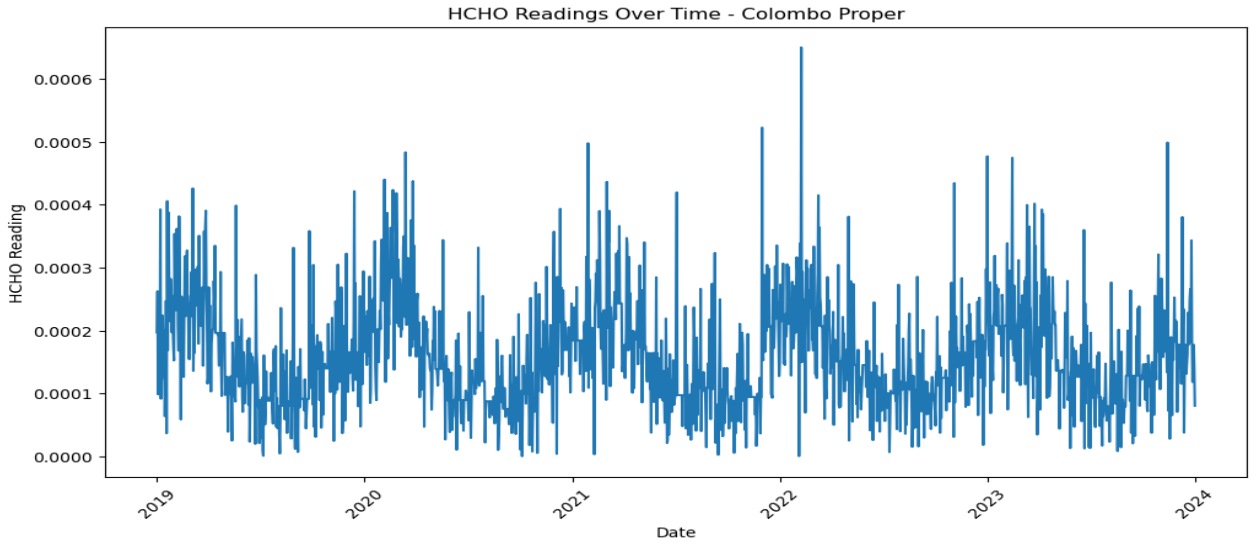

Boxplot of HCHO Readings for Jaffna Proper
HCHO Reading
0.0006
0.0005
0.0004
0.0003
0.0002
0.0001
0.0000
Jaffna Proper
Location

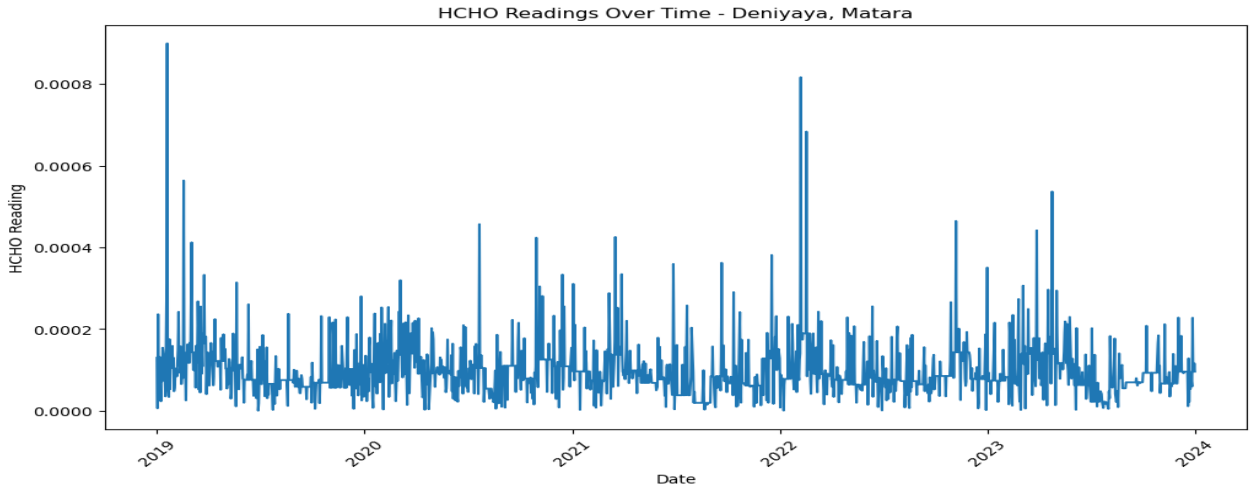

Boxplot of HCHO Readings for Kandy Proper
HCHO Reading
0.0007
0.0006
0.0005
0.0004
0.0003
0.0002
0.0001
0.0000
Kandy Proper
Location

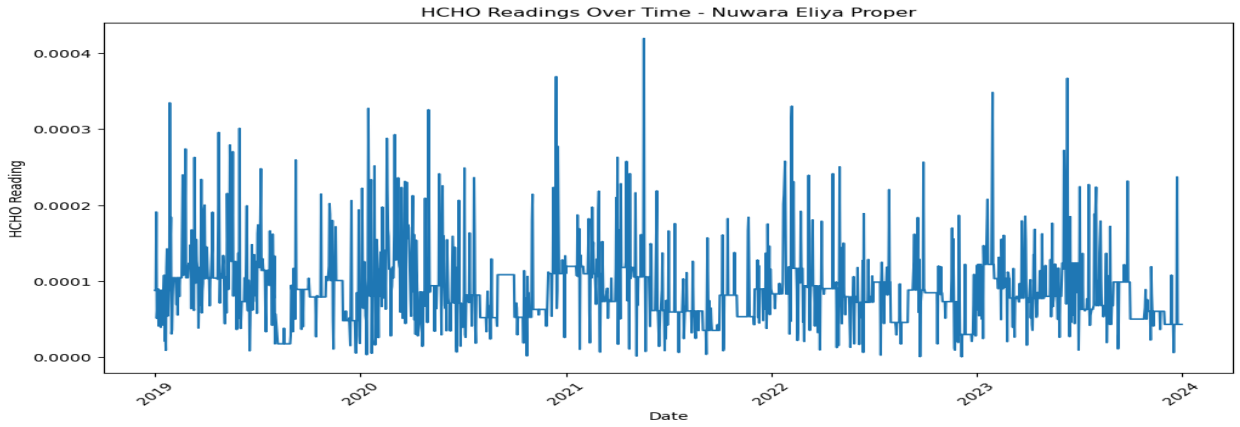
3. Spatio-Temporal Analysis

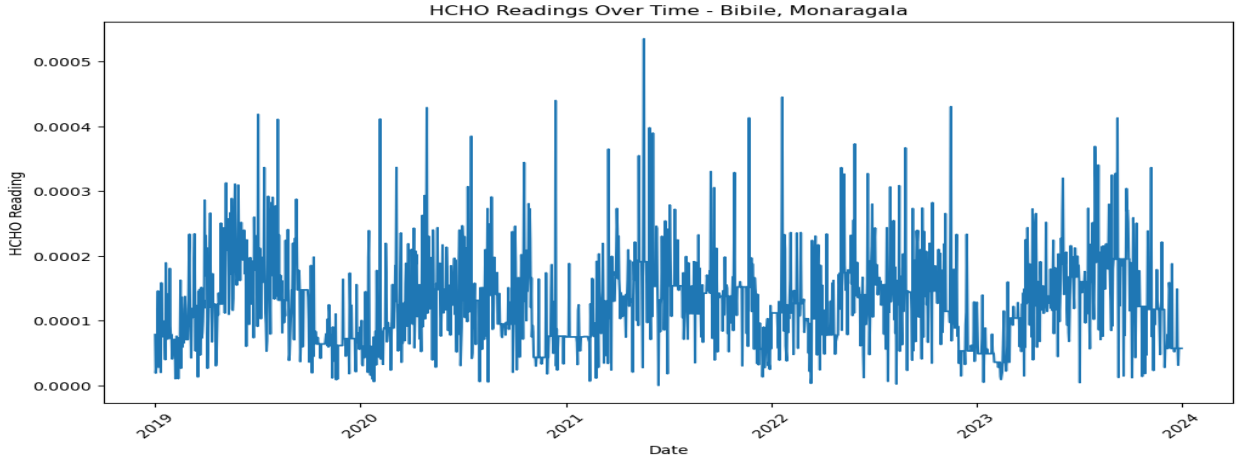
Analyzing Trends over time

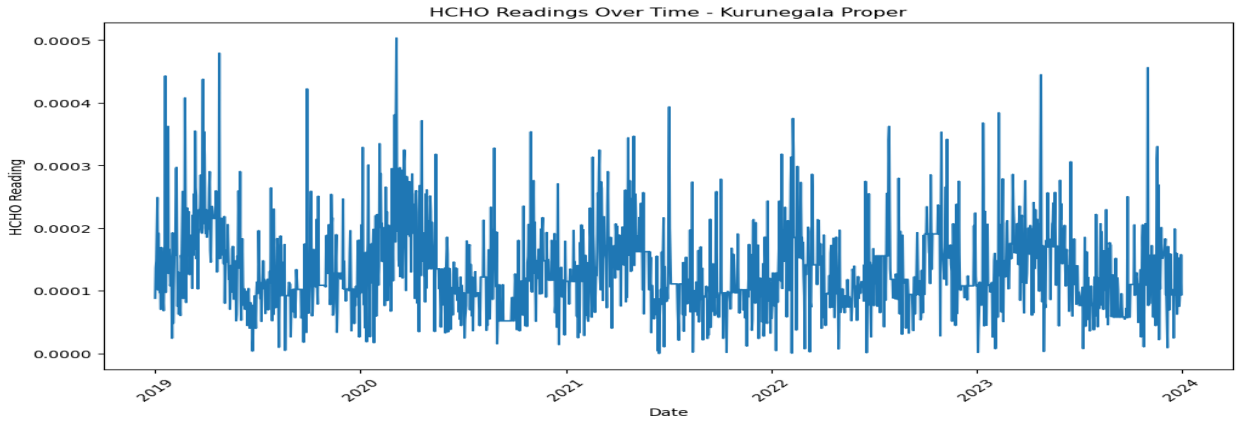
This generates line graphs for each individual site, exhibiting the trend in formaldehyde (HCHO) values over time. For each site, the line plot depicts the variance in HCHO levels over time, revealing any temporal trends or fluctuations in air quality. The plots are customized for each place, with titles denoting the unique location being studied. By displaying HCHO values over time, these plots allow for the detection of trends, seasonal patterns, or anomalies in air quality at distinct sites, assisting in the study of probable environmental variables or sources contributing to formaldehyde levels in various locales.

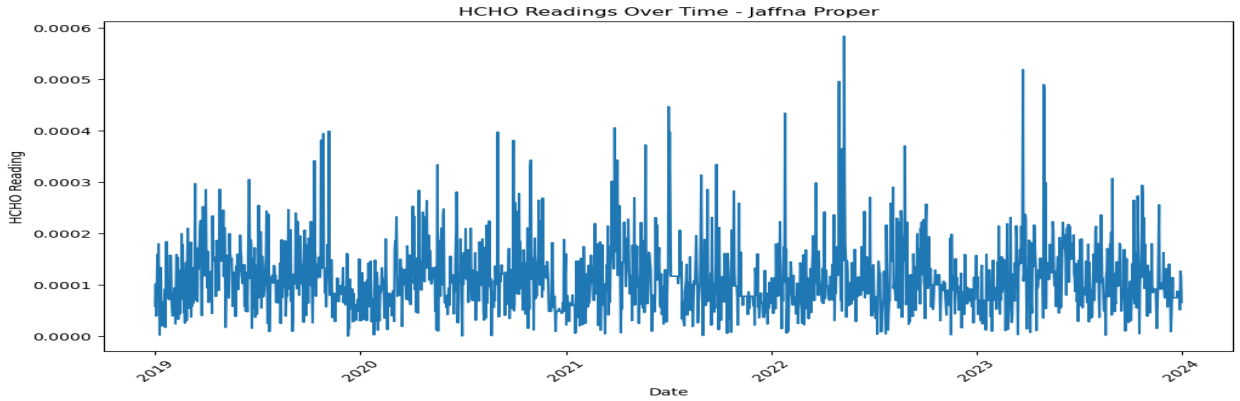


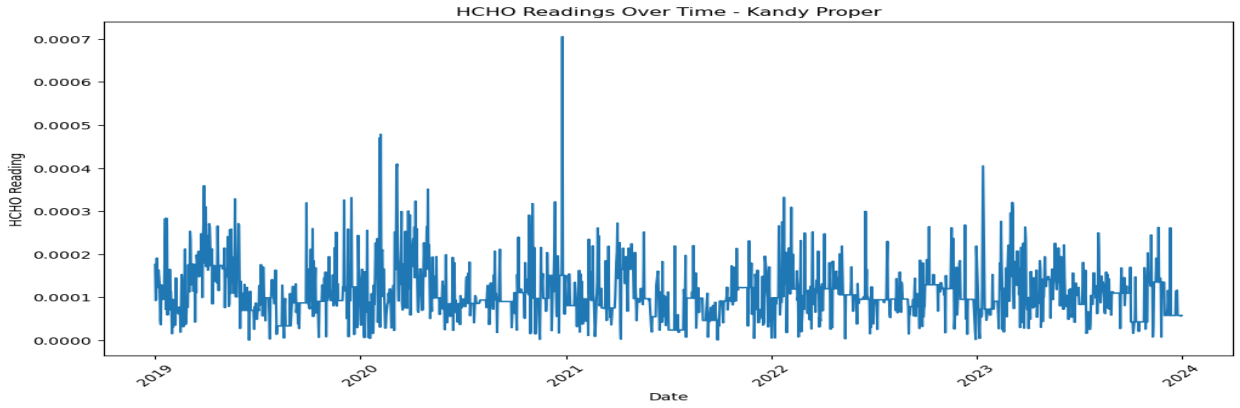



HCHO Readings Over Time - Nuwara Eliya Proper
HCHO Reading
0.0000
0.0001
0.0002
0.0003
0.0004
2019
2020
2021
2022
2023
2024
Date

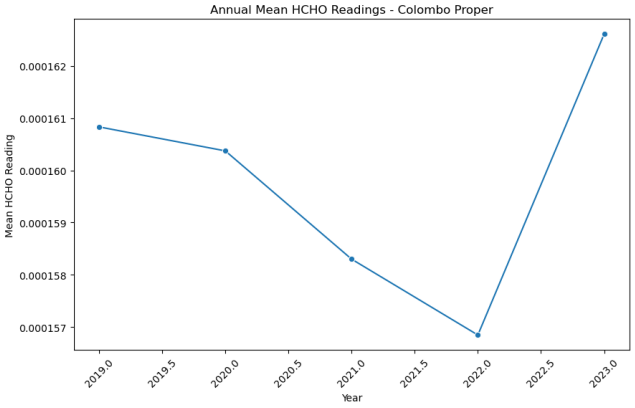

HCHO Readings Over Time - Bibile, Monaragala
HCHO Reading
0.0000
0.0001
0.0002
0.0003
0.0004
0.0005
2019
2020
2021
2022
2023
2024
Date

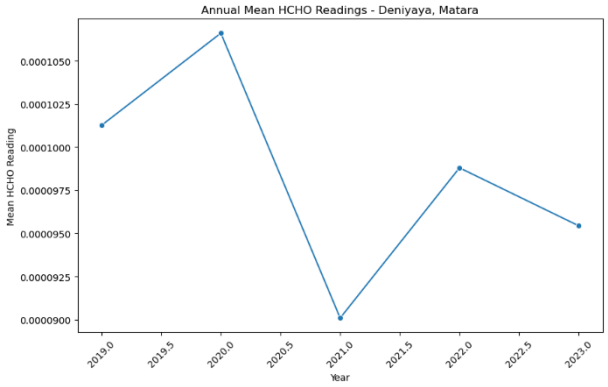

HCHO Readings Over Time - Kurunegala Proper
HCHO Reading
0.0000
0.0001
0.0002
0.0003
0.0004
0.0005
2019
2020
2021
2022
2023
2024
Date

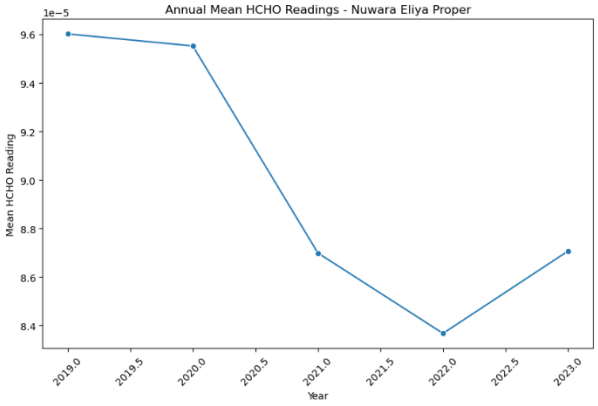


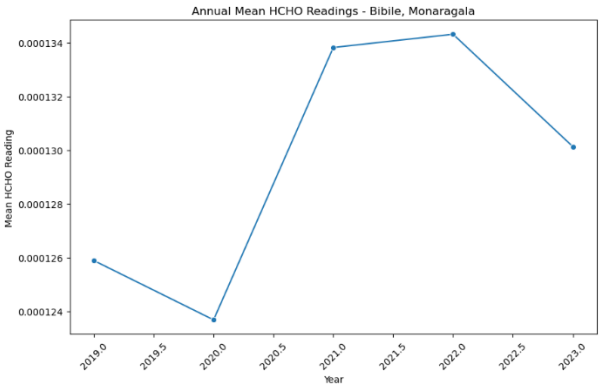


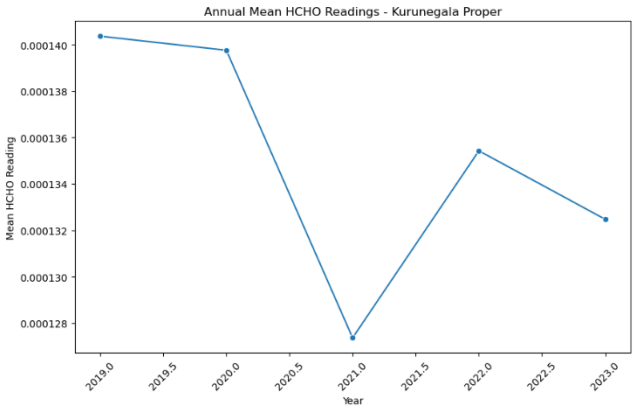
Below section creates line graphs for each individual site, displaying the yearly mean formaldehyde (HCHO) values over time. The figure shows the average HCHO levels for each site over time, allowing for a clear representation of long-term trends or changes in air quality. These charts, which group the data by year and calculate the mean HCHO measurements for each year, provide insights into the general yearly fluctuations in HCHO concentrations at different sites. The plots are tailored to each place, with titles denoting the exact location under study. By showing yearly mean HCHO measurements across time, these plots assist the detection of any noteworthy shifts or patterns in air quality trends at distinct places, aiding in understanding the larger environmental dynamics influencing HCHO levels.

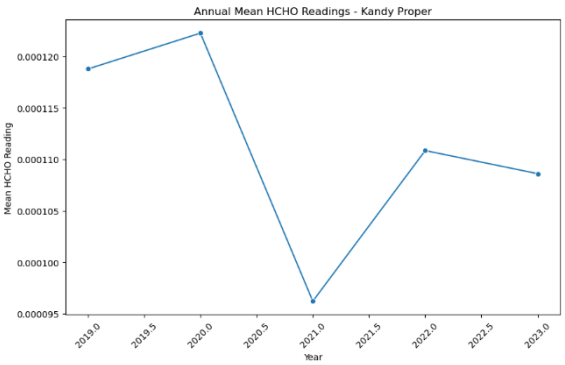

Annual Mean HCHO Readings - Colombo Proper
Mean HCHO Reading
Year

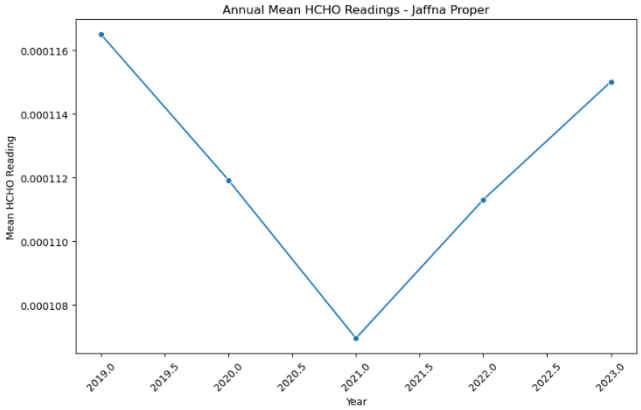

Annual Mean HCHO Readings - Deniyaya, Matara
Mean HCHO Reading
Year


Annual Mean HCHO Readings - Nuwara Eliya Proper
1e−5
Mean HCHO Reading
Year


Annual Mean HCHO Readings - Bibile, Monaragala
Mean HCHO Reading
Year


Annual Mean HCHO Readings - Kurunegala Proper
Mean HCHO Reading
Year

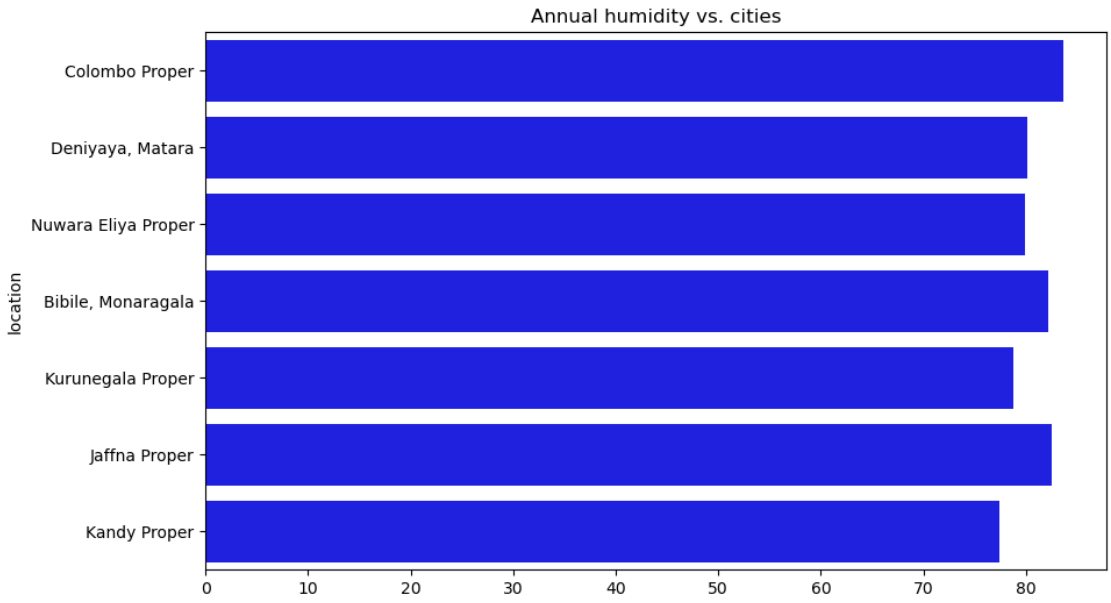

Annual Mean HCHO Readings - Kandy Proper
Mean HCHO Reading
Year


Annual Mean HCHO Readings - Jaffna Proper
Mean HCHO Reading
Year

External Factors

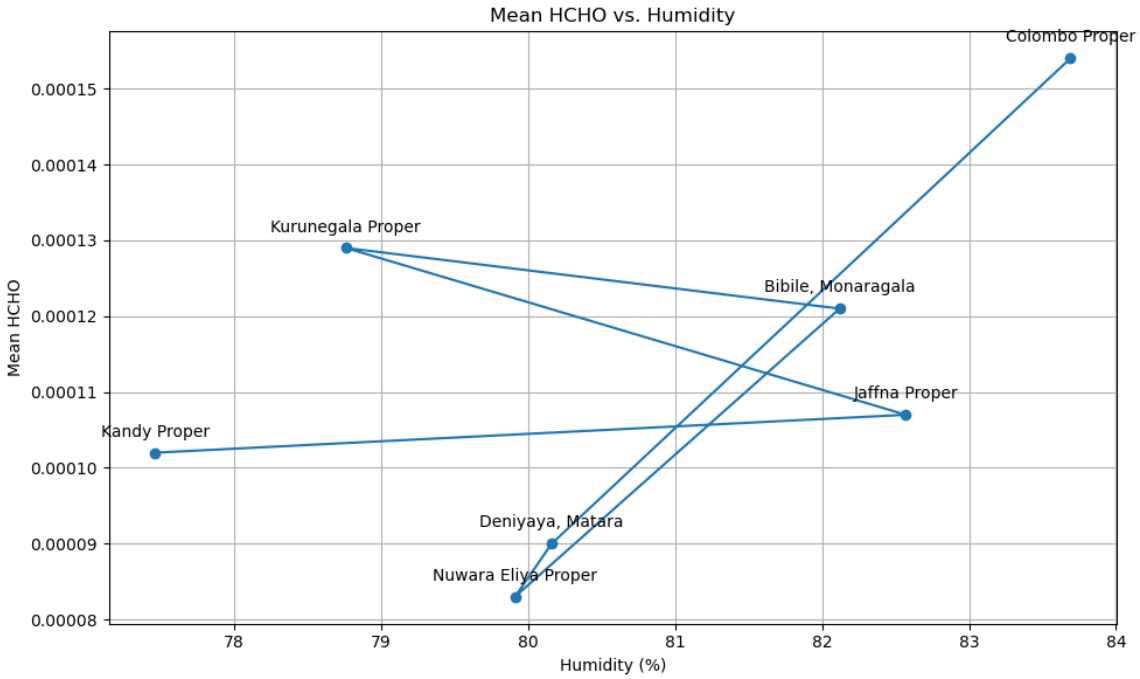
Humidity

This generates a bar plot comparing yearly humidity levels in various cities. The x-axis depicts humidity levels, and each bar corresponds to a specific region. The length of each bar represents the humidity level in the individual city. The map visualizes this information, allowing for a simple comparison of humidity levels across different places. The title "Annual humidity vs. cities" summarizes the plot's content, suggesting that it depicts the relationship between humidity and numerous cities across time. Overall, this visualization allows for the assessment of humidity fluctuations across different geographic regions, which aids in understanding regional disparities in atmospheric moisture levels.



The graphic depicts the association between mean formaldehyde (HCHO) levels and humidity in several cities. Each point on the figure indicates the city's average HCHO concentration and humidity level. The markers ('o') represent the data points for each city, while the lines linking the markers ('-') assist to visualize trends or patterns in the data. Additionally, the names of the cities are marked on the plot to make it clear which data points relate to which place. The "Mean HCHO vs. Humidity" label highlights the plot's content, emphasizing the link between these two factors. Overall, this visualization allows for the examination of potential connections or links between

HCHO levels and humidity across many cities, which aids in understanding the environmental factors impacting formaldehyde concentrations.



A correlation value of 0.50 suggests that humidity and HCHO concentrations tend to grow together, but not completely linearly. While this connection does not imply causality, it does indicate that there may be a link between atmospheric moisture levels and formaldehyde levels in the air. This research sheds light on the possible impact of humidity on formaldehyde concentrations, which may be useful for understanding and managing air quality concerns in various geographic locations.

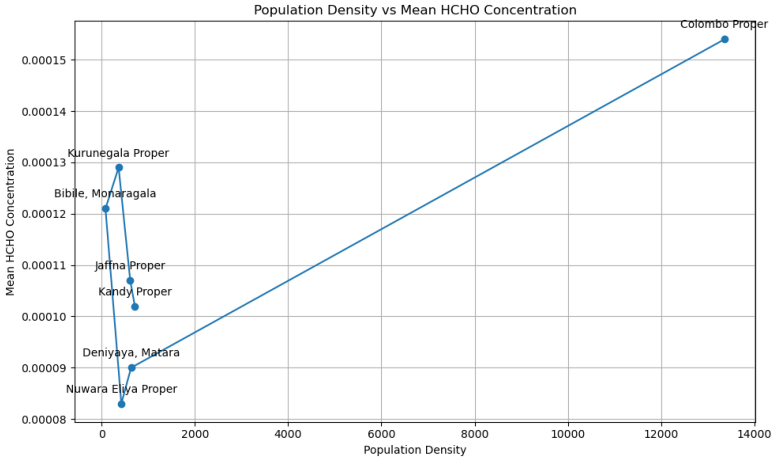
```
Humidity_corr = np.corrcoef(Humidity,HCHO)
print('Correlation  between humidity  and  HCHO :', Humidity_corr[0,1])
```

✓ 0.0s

```
Correlation  between humidity  and  HCHO : 0.5011712852490696
```

Population Density

The graphic shows the association between population density and mean formaldehyde (HCHO) concentrations in several cities. Each point on the figure indicates a city's mean HCHO concentration and accompanying population density. The markers ('o') represent data points for each city, while the lines linking the markers ('-') aid in seeing trends or patterns in data. Additionally, the names of the cities are marked on the plot to make it clear which data points relate to which place. The "Population Density vs Mean HCHO Concentration" graph highlights the plot's content, emphasizing the link between these two variables. Overall, this visualization allows for the examination of potential correlations between population density and formaldehyde levels in various places, shedding light on the potential influence of urbanization on air quality.



A correlation value of 0.74 suggests a strong propensity for population density and HCHO concentrations to rise together, but not exactly linearly. This association emphasizes the possible influence of urbanization and population density on formaldehyde levels in the atmosphere. While correlation does not imply causality, this evidence indicates that places with higher people density may have higher formaldehyde concentrations. Understanding this association can help with urban planning and environmental management initiatives to address air quality concerns related with formaldehyde exposure in densely populated areas.

```
population_corr = np.corrcoef(population, HCHO)
print('Correlation coefficient between population  and mean HCHO :', population_corr[0,1])
```

✓ 0.0s

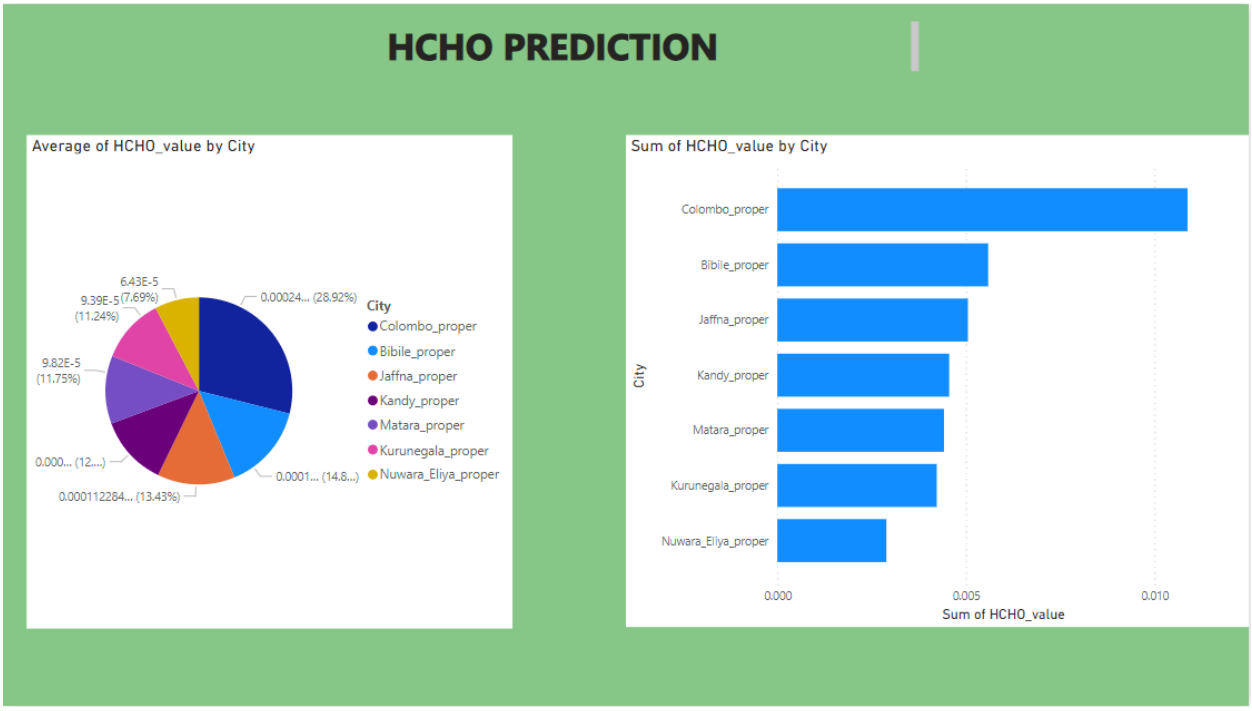
```
Correlation coefficient between population  and mean HCHO : 0.7371942511753211
```

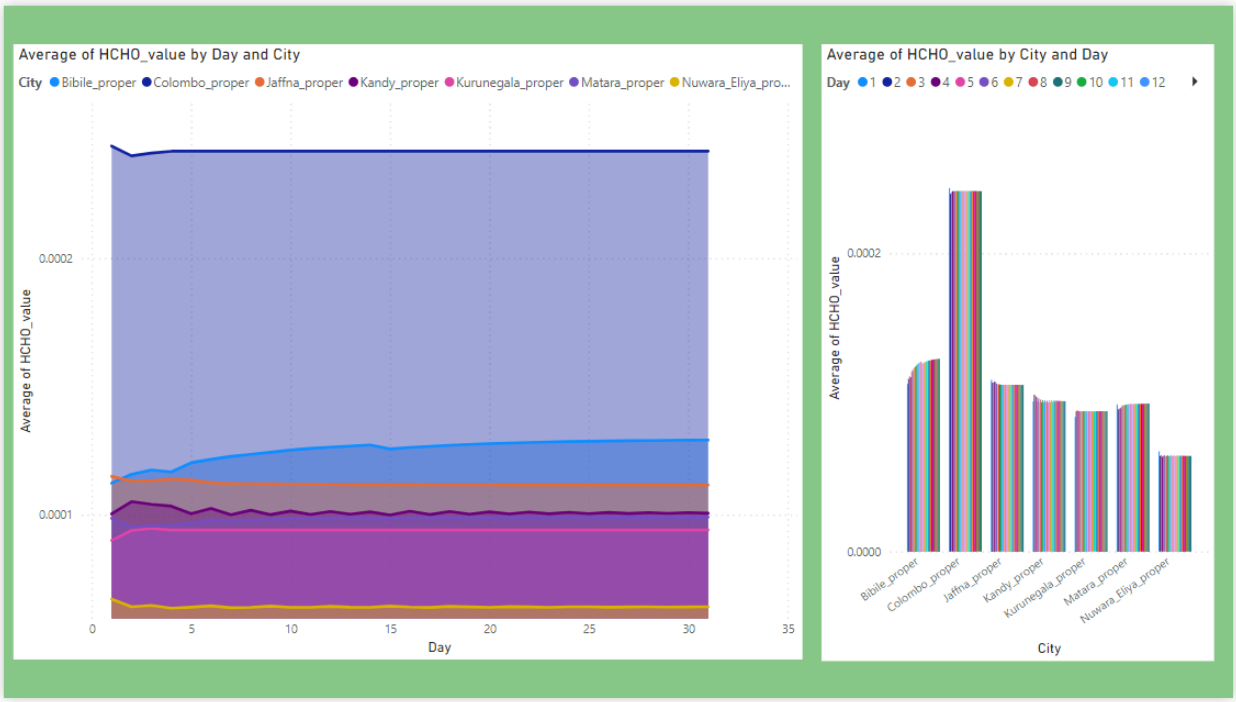
4. ARIMA Model

The Autoregressive Integrated Moving Average (ARIMA) model is a popular time series forecasting technique that uses both autoregressive (AR) and moving average (MA) components, as well as differencing to handle non-stationary data. In an ARIMA model, the time series data is turned into a stationary series via differencing, making it appropriate for modeling. The ARIMA model has three basic parameters: p, d, and q, which indicate the autoregressive, differencing, and moving average orders, respectively. The AR component captures the relationship between an observation and several lagged observations, the MA component models the relationship between an observation and a moving average residual error, and the differencing order aids in series stabilization by removing trends or seasonal patterns. By estimating these parameters and fitting the model to historical data, ARIMA models can predict future values in time series, making them useful tools in fields such as finance, economics, and environmental science for forecasting trends and making informed decisions based on time-dependent data.

City	Best Model	ARIMA Model
Colombo	ARIMA(0,1,4)(0,0,0)[12]	1.2449005624439747e-08
Deniyaya, Matara	ARIMA(5,0,0)(0,0,0)[12]	3.7965284355452705e-09
Nuwara Eliya	ARIMA(2,1,4)(0,0,0)[12]	2.6901717101307053e-09
Bibile, Monaragala	ARIMA(5,0,0)(0,0,0)[12]	4.991953730429999e-09
Kurunegala	ARIMA(0,1,4)(0,0,0)[12]	5.984465610186335e-09
Jaffna	ARIMA(5,0,0)(0,0,2)[12]	4.45852648357985e-09
Kandy	ARIMA(4,1,2)(0,0,0)[12]	3.2993472163395473e-09

5. Power BI


HCHO PREDICTION
Average of HCHO_value by City
6.43E-5
9.39E-5 (7.69%)
(11.24%)
9.82E-5
(11.75%)
0.000... (12....)
0.000112284... (13.43%)
0.00024... (28.92%)
0.0001... (14.8...)
City
Colombo_proper
Bibile_proper
Jaffna_proper
Kandy_proper
Matara_proper
Kurunegala_proper
Nuwara_Eliya_proper
Sum of HCHO_value by City
Colombo_proper
Bibile_proper
Jaffna_proper
Kandy_proper
Matara_proper
Kurunegala_proper
Nuwara_Eliya_proper
City
0.000
0.005
0.010
Sum of HCHO_value


Average of HCHO_value by Day and City
City Bibile_proper Colombo_proper Jaffna_proper Kandy_proper Kurunegala_proper Matara_proper Nuwara_Eliya_pro...
0.0002
0.0001
Average of HCHO_value
0
5
10
15
20
25
30
35
Day
Average of HCHO_value by City and Day
Day 1 2 3 4 5 6 7 8 9 10 11 12
0.0002
0.0000
Average of HCHO_value
Bibile_proper
Colombo_proper
Jaffna_proper
Kandy_proper
Kurunegala_proper
Matara_proper
Nuwara_Eliya_proper
City

6. Limitations

- Data Quality: Poor quality or inadequate training data might impair model performance. Inaccurate, incomplete, or biased data might result in models that generalize poorly to new data.

- Imbalanced Data: When the distribution of classes or outcomes in training data is unbalanced, models may be biased toward the majority class while doing badly on minority classes.

- Generalization: To be useful, the trained model must be able to generalize successfully to previously unknown data, including data from various distributions or domains.

- Interpretability: Some models, particularly deep learning models, may lack interpretability, making it difficult to comprehend the rationale underlying their predictions. This might be a drawback in sectors where interpretability is critical.

7. Recommendation

• Air Quality Policies: Use analysis results to design targeted limitations and emission reduction programs. Set aside areas with consistently high HCHO levels as priorities for enforcement and monitoring.

• Conduct more research to investigate how factors like population density and proximity to the sea may impact HCHO concentrations. Conduct cross-regional or cross-national comparative study to better understand regional variances and the effectiveness of air quality management.

• Identify emission sources by analyzing HCHO levels in relation to climatic circumstances, fire occurrences, and human activities. Implement techniques that target specific sources, such as industrial pollutants and forest fires.

8. References

Alex The Analyst (2022). *How to Install Power BI | Building First Visualization | Microsoft Power BI for Beginners*. *YouTube*. Available at: https://www.youtube.com/watch?v=g0m5sEHPU-s [Accessed 19 April 2024].