# INNOVATION

### **Consider incorporating data visualization techniques to showcase historical temperature and humidity trends.**

*Certainly! Visualizing historical temperature and humidity trends can help convey information effectively. Here are some suggestions for data visualization techniques:*

* *Line Charts*: Use line charts to display temperature and humidity trends over time. Plot temperature and humidity on the y-axis and time on the x-axis. This allows viewers to see how these variables change over different periods.
* *Area Charts*: Area charts can be used to show the relationship between temperature and humidity. You can stack temperature and humidity data to illustrate their combined impact over time.
* *Heatmaps*: Heatmaps are excellent for displaying variations in temperature and humidity across different time periods or locations. They use color gradients to represent different values, making it easy to spot patterns and anomalies.
* *Scatter Plots :*Scatter plots can be used to explore the correlation between temperature and humidity. Each data point represents a specific time, and you can color-code them to represent different conditions or seasons.
* *Box Plots***:** Box plots are useful for showing the distribution of temperature and humidity data over time. They provide insights into the median, quartiles, and potential outliers.
* *Animated Visualizations*: If you have data over a long time span, consider creating an animated visualization that shows how temperature and humidity change over the years. This can be done with tools like D3.js or Python libraries like Plotly.

## Coding:

import pandas as pd

import matplotlib.pyplot as plt

# Sample data (replace this with your historical temperature and humidity data)

data = {

'Date': ['2023-01-01', '2023-01-02', '2023-01-03', '2023-01-04'],

'Temperature (Celsius)': [20, 22, 19, 18],

'Humidity (%)': [50, 45, 60, 55]

}

# Convert data to a DataFrame

df = pd.DataFrame(data)

# Convert the 'Date' column to datetime

df['Date'] = pd.to\_datetime(df['Date'])

# Create subplots for temperature and humidity

fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 6))

# Plot temperature data

ax1.plot(df['Date'], df['Temperature (Celsius)'], marker='o', color='b', label='Temperature (°C)')

ax1.set\_xlabel('Date')

ax1.set\_ylabel('Temperature (°C)')

ax1.set\_title('Historical Temperature Trends')

ax1.grid(True)

ax1.legend()

# Plot humidity data

ax2.plot(df['Date'], df['Humidity (%)'], marker='o', color='g', label='Humidity (%)')

ax2.set\_xlabel('Date')

ax2.set\_ylabel('Humidity (%)')

ax2.set\_title('Historical Humidity Trends')

ax2.grid(True)

ax2.legend()

# Adjust spacing between subplots

plt.tight\_layout()

# Show the plots

plt.show()

**Program explanation:**

1. We start by importing the necessary libraries, Pandas and Matplotlib.
2. Load Data: You should replace 'your\_data.csv' with the path to your historical data file. This data is loaded into a Pandas DataFrame called data.
3. Data Preparation: We ensure that the 'Date' column is in datetime format, which is important for time-series data.
4. Create Visualizations: Two line charts are created. One for temperature trends and another for humidity trends. Each chart is customized with titles, labels, legends, and grids for clarity.
5. Display Visualizations: Finally, we use plt.show() to display the visualizations.
6. Make sure to customize the program with your actual data file and adjust plot settings as needed for your specific dataset and visualization preferences.