COLLEGE CODE:4212
REGISTER NUMBER:421221106005

INTERNET OF THINGS -GROUP:2:

ENVIRONMENTAL MONITORING - PHASE4

IBM NAAN MUDHULVAN

In the previous phases we have discussed about the step-by-step process, Design thinking and at the phase3 we have discussed about the data preprocessing techniques and many more in the last steps and in this step we have given some problem statements to solve in the IBM COGNOS ANALYTICS.

In this part we will continue building our project, Building the analysis by creating visualizations using IBM Cognos.

Problem:1 Continue building the analysis by creating visualizations using IBM Cognos and developing a predictive model.

Problem:2 Create interactive dashboards and reports in IBM Cognos to visualize churn patterns, retention rates, and key factors influencing churn. Use machine learning algorithms to build a predictive model that identifies potential churners based on historical data and relevant features.

According to the problems we come to know that we have to find the relation between the two variables so for that we need to visualize the relation between the two variables using the IBM Cognos

The visual insights that we have created are shown in the upcoming papers. so here we have prepared a necessary visualizationand also the dashboard using the IBM Cognos.

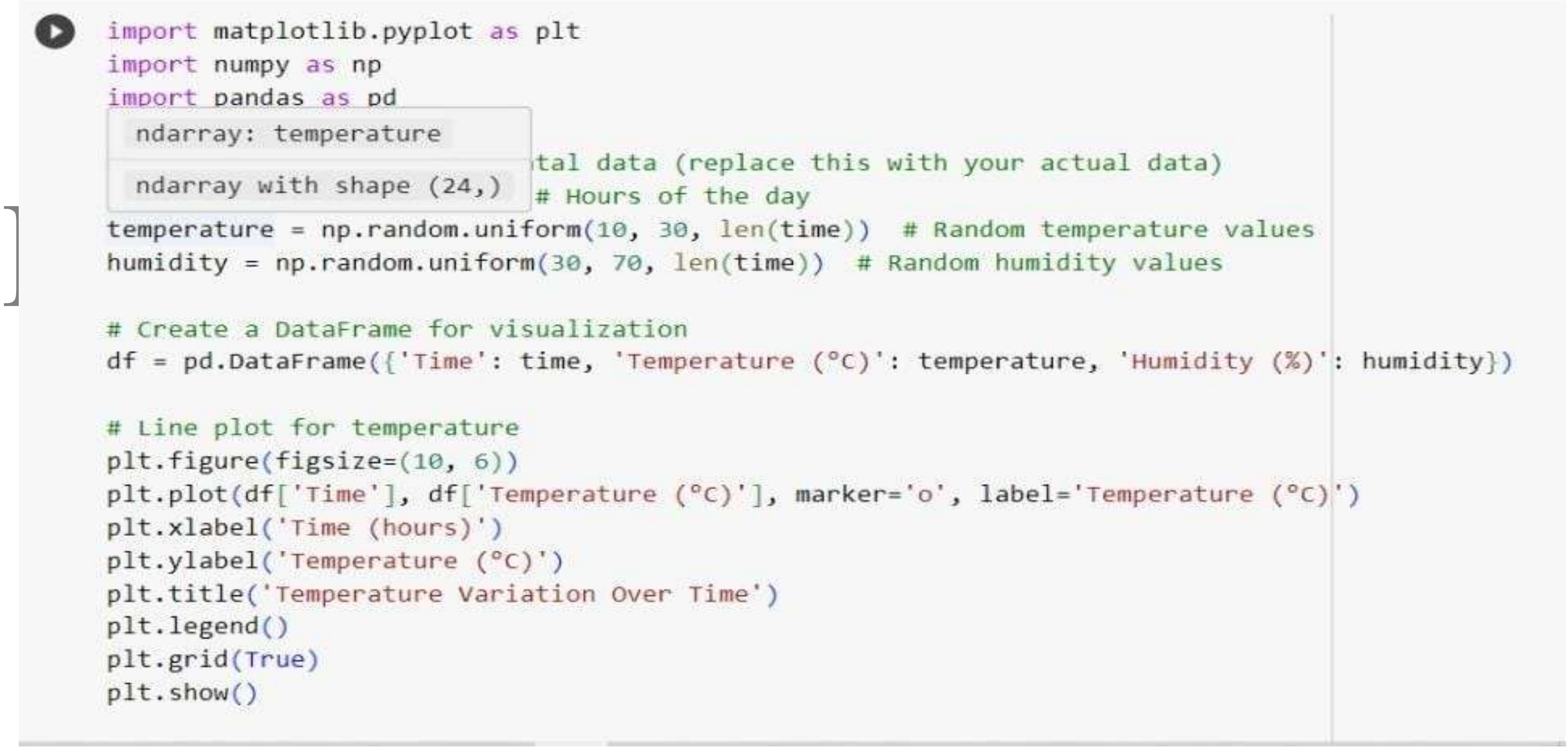
Note: The Narrative insights(i.e. The explanation of the visualization) can be at the right side of the picture.

HOW TO USE IBM COGNOS:

for this we need to login in the IBM Cognos the IBM Cognos is free for one month.

Then we have to go to main page and we need to upload our data in the in my content on the left side of the home page then we need to prepare the dashboard by dragging down option we will easily create the visual insights.

The visualization methods is as follows:

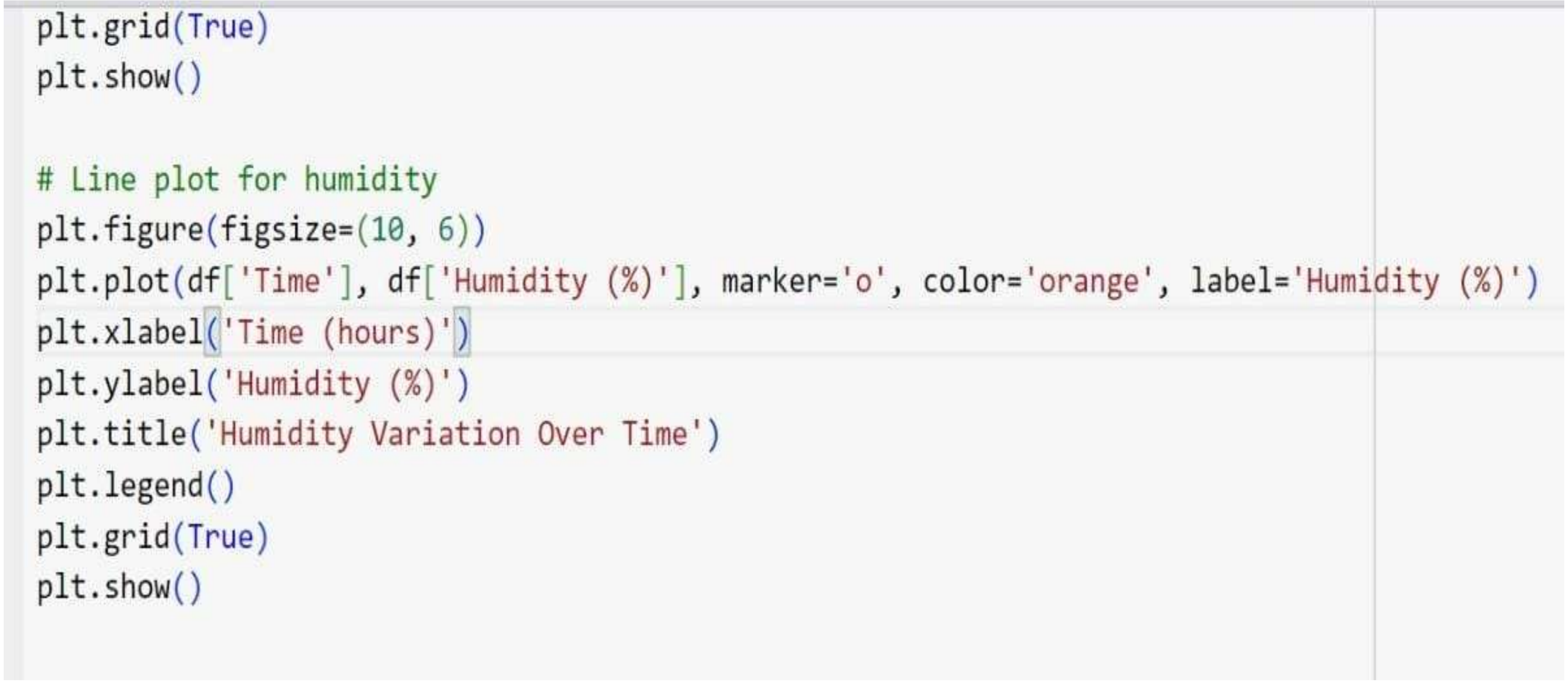


```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

                    tal data (replace this with your actual data)
                    # Hours of the day
temperature = np.random.uniform(10, 30, len(time))  # Random temperature values
humidity = np.random.uniform(30, 70, len(time))  # Random humidity values

# Create a DataFrame for visualization
df = pd.DataFrame({'Time': time, 'Temperature (°C)': temperature, 'Humidity (%)': humidity})

# Line plot for temperature
plt.figure(figsize=(10, 6))
plt.plot(df['Time'], df['Temperature (°C)'], marker='o', label='Temperature (°C)')
plt.xlabel('Time (hours)')
plt.ylabel('Temperature (°C)')
plt.title('Temperature Variation Over Time')
plt.legend()
plt.grid(True)
plt.show()
```

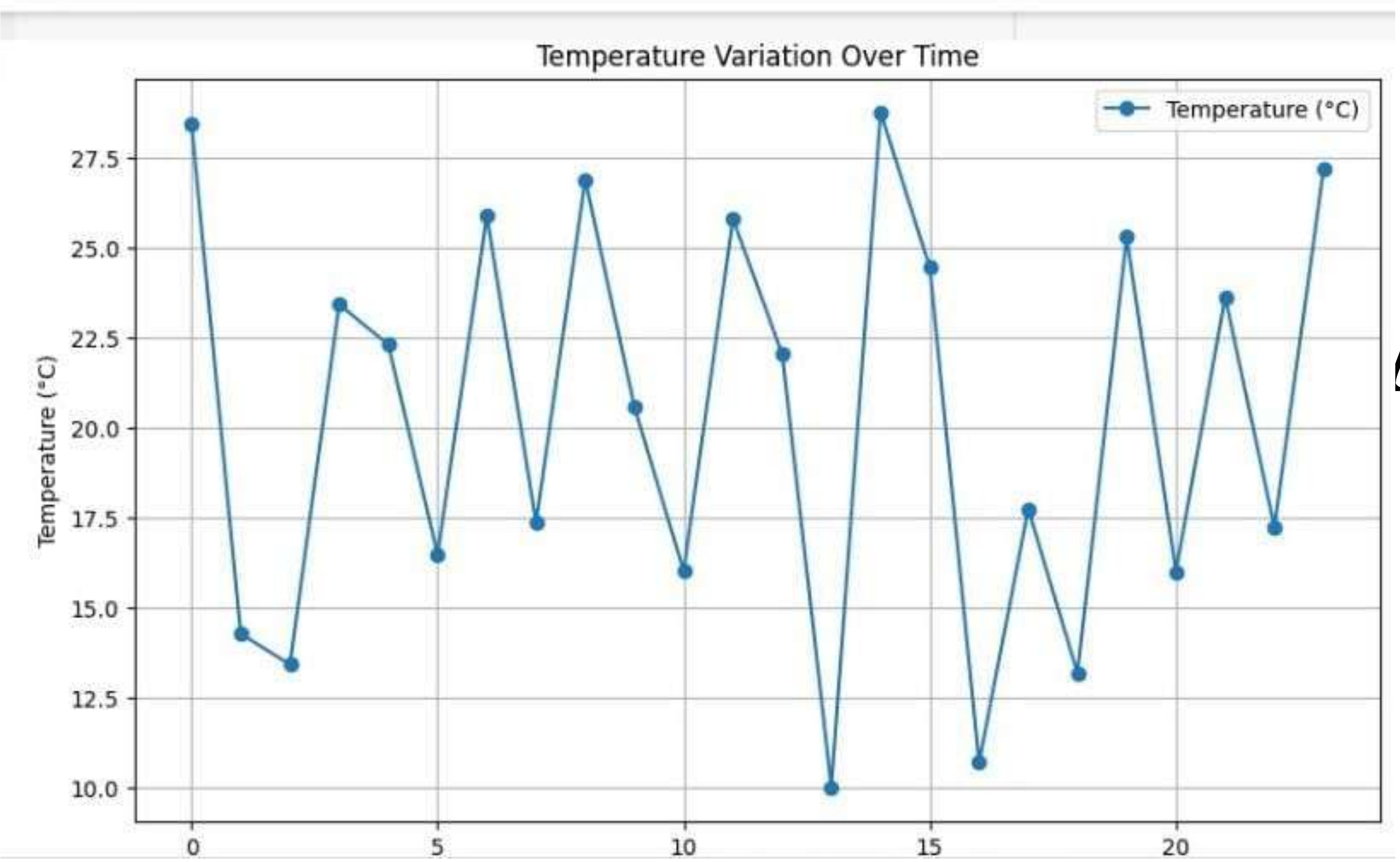


```
plt.grid(True)
plt.show()

# Line plot for humidity
plt.figure(figsize=(10, 6))
plt.plot(df['Time'], df['Humidity (%)'], marker='o', color='orange', label='Humidity (%)')
plt.xlabel('Time (hours)')
plt.ylabel('Humidity (%)')
plt.title('Humidity Variation Over Time')
plt.legend()
plt.grid(True)
plt.show()
```

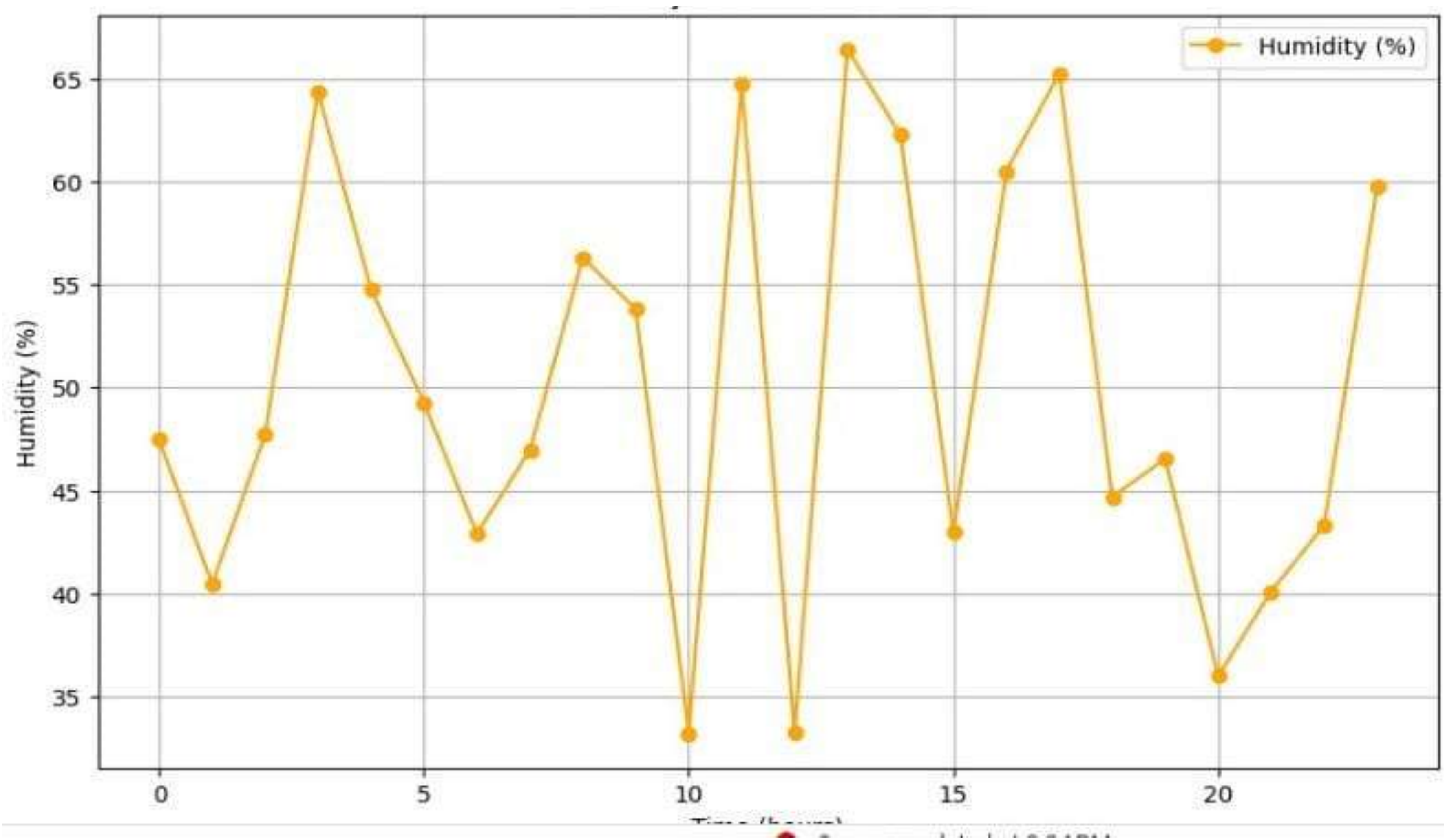
1q`2

LINEPLOT FOR TEMPERATURE



AN

LINEPLOT FOR HUMIDITY


Humidity (%)
Humidity (%)
Time (hours)