Lab Guide: Weather Data Visualization with Python in Power BI

# Objective

In this lab, you will use Power BI to load time-series weather data from Excel and visualize it using Python. You will create multiple Python-based visuals directly within Power BI Desktop to analyze temperature, humidity, and precipitation trends.

# Prerequisites

- Power BI Desktop installed

- Python installed (Anaconda or Python 3.8+ recommended)

- Required libraries installed via terminal:  
 pip install pandas matplotlib seaborn

# Files Provided

- generated\_weather\_data.xlsx: The source dataset with 300 rows of hourly weather data

- This lab guide

# Step 1: Launch Power BI and Load Excel File

1. Open Power BI Desktop.  
2. Click on 'Get Data' > 'Excel'.  
3. Browse to and select 'generated\_weather\_data.xlsx'.  
4. Select the worksheet and click 'Load'.

# Step 2: Configure Python in Power BI (One-time)

1. Go to File > Options and Settings > Options.  
2. In the left panel, choose 'Python scripting'.  
3. Set the path to your Python executable (e.g., C:\Users\YourName\anaconda3\python.exe).  
4. Click OK.

# Step 3: Create Your First Python Visual – Temperature Over Time

1. Click the Python Visual icon from the Visualizations pane.  
2. Drag 'Timestamp' and 'Temperature\_C' into the Values section.  
3. Paste the following code:  
  
 import pandas as pd  
 import matplotlib.pyplot as plt  
 import seaborn as sns  
  
 dataset['Timestamp'] = pd.to\_datetime(dataset['Timestamp'])  
 plt.figure(figsize=(10, 5))  
 sns.lineplot(data=dataset, x='Timestamp', y='Temperature\_C', color='orange', label='Temperature')  
 plt.title('Temperature Over Time')  
 plt.xlabel('Timestamp')  
 plt.ylabel('Temperature (°C)')  
 plt.grid(True)  
 plt.tight\_layout()  
 plt.show()

# Step 4: Add More Python Visuals

A. Scatter Plot - Temperature vs. Humidity:  
 - Drag 'Temperature\_C' and 'Humidity\_%'.  
 - Use:  
 sns.scatterplot(data=dataset, x='Temperature\_C', y='Humidity\_%', alpha=0.6)  
 plt.title('Temperature vs Humidity')  
 plt.grid(True); plt.tight\_layout(); plt.show()  
  
B. Hourly Average Temperature:  
 - Drag 'Timestamp', 'Temperature\_C'  
 - Use:  
 dataset['Timestamp'] = pd.to\_datetime(dataset['Timestamp'])  
 dataset['Hour'] = dataset['Timestamp'].dt.hour  
 avg\_by\_hour = dataset.groupby('Hour')['Temperature\_C'].mean().reset\_index()  
 sns.lineplot(data=avg\_by\_hour, x='Hour', y='Temperature\_C')  
 plt.title('Hourly Avg Temp'); plt.grid(True); plt.tight\_layout(); plt.show()  
  
C. Histogram of Precipitation:  
 - Drag 'Precipitation\_mm'  
 - Use:  
 sns.histplot(data=dataset, x='Precipitation\_mm', bins=10, kde=True)  
 plt.title('Precipitation Distribution'); plt.tight\_layout(); plt.show()  
  
D. Heatmap of Correlation:  
 - Drag 'Temperature\_C', 'Humidity\_%', 'Precipitation\_mm'  
 - Use:  
 sns.heatmap(dataset.corr(), annot=True, cmap='coolwarm')  
 plt.title('Correlation Heatmap'); plt.tight\_layout(); plt.show()

# Step 5: Publish Your Report

1. Go to Home > Publish.  
2. Sign in to Power BI Service.  
3. Choose a workspace and publish.  
4. From Power BI Web, open the report and pin visuals to a dashboard.

# Lab Summary

You have successfully created multiple Python visuals in Power BI using time-series weather data. You learned to configure Python, use different chart types like line plots, scatter plots, histograms, and heatmaps. You also published your report for dashboard sharing.