VISUALIZATION - USING FOLIUM LIBRARY FOR GEOGRAPHIC OVERLAYS

Objective

The objective of this project is to visualize CO2 emissions per capita for various countries using the Folium library. This involves creating an interactive geographic map to overlay CO2 emissions data, allowing for a visual comparison of emissions across different countries. The steps include:

1. **Reading** the World Development Indicators dataset.
2. **Filtering** the dataset for CO2 emissions data for the year 2011.
3. **Creating** a geographic map with Folium to visualize CO2 emissions per capita.
4. **Saving and displaying** the interactive map.

Summary

This project uses the Folium library to create an interactive choropleth map, which overlays CO2 emissions data onto a geographic map of the world. The data is sourced from the World Development Indicators dataset provided by the World Bank. The visualization helps in understanding the distribution of CO2 emissions across different countries in 2011.

Results

1. **Data Preparation**:
    - The dataset was filtered to obtain CO2 emissions per capita for the year 2011.
    - A new DataFrame was created containing only the country codes and corresponding CO2 emissions values.
2. **Visualization**:
    - An interactive map was generated using Folium, centered globally with a zoom level that provides a broad overview.
    - A choropleth map was created to visualize CO2 emissions per capita. The map uses a color gradient to represent different levels of emissions, making it easy to identify countries with high and low emissions.
3. **Map Output**:
    - The interactive map was saved as an HTML file and successfully opened in a web browser.
    - The map visually represents CO2 emissions per capita, with countries shaded according to their emissions levels. The legend provides context for interpreting the color scale.

Conclusion

The Folium-based interactive map effectively visualizes CO2 emissions per capita for various countries, highlighting differences in emissions levels across the globe. The map offers a clear and intuitive way to explore and compare emissions data, revealing geographical patterns and outliers.

Code :

```
'''
Project 3 : Using Folium Library for Geographic Overlays
Further exploring CO2 Emissions per capita in the World Development Indicators Dataset
'''
import webbrowser

import folium
import pandas as pd
import warnings
import os

warnings.filterwarnings("ignore")
country_geo = ("D:\\Aditya's Notes\\Aditya's Data Science Notes\\Projects and Other Datasets\\ML PROJECTS\\data\\world-countries.json")
# Read in the World Development Indicators Database
data = pd.read_csv("D:\\Aditya's Notes\\Aditya's Data Science Notes\\Projects and Other Datasets\\ML PROJECTS\\data\\Indicators.bz2", compression='bz2')
data.shape
print(data.head())

#Select CO2 emissions for all countries in 2011
hist_indicator = 'CO2 emissions \\(metric'
hist_year = 2011
mask1 = data['IndicatorName'].str.contains(hist_indicator)
mask2 = data['Year'].isin([hist_year])
# Apply our mask
stage = data[mask1 & mask2]
stage.head()

#Setup the data for plotting
#Create a data frame with just the country codes and the values we want to be plotted.
```

```
plot_data = stage[['CountryCode','Value']]
plot_data.head()
# Label for the legend
hist_indicator = stage.iloc[0]['IndicatorName']
print(hist_indicator)

#Visualize CO2 emissions per capita using Folium
#Folium provides interactive maps with the ability to create sophisticated
# overlays for data visualization
#Setup a folium map at a high-level zoom.
map = folium.Map(location=[100, 0], zoom_start=1.5)
#Choropleth maps bind Pandas Data Frames and json geometries.
# This allows us to quickly visualize data combinations
folium.Choropleth(geo_data=country_geo, data=plot_data,columns=['CountryCode','Value'],
key_on='feature.id',fill_color='YlGnBu', fill_opacity=0.7,
line_opacity=0.2,
legend_name=hist_indicator)
# Create Folium plot
map.save("D:\\Aditya's Notes\\Aditya's Data Science Notes\\Projects and Other Datasets\\ML PROJECTS\\data\\saved_info\\plot_data.html")
# Import the Folium interactive html file
from IPython.display import HTML
map_file_path = ("D:\\Aditya's Notes\\Aditya's Data Science Notes\\Projects and Other Datasets\\ML PROJECTS\\data\\saved_info\\plot_data.html")

# Check if the HTML file was created
if os.path.exists(map_file_path):
    print(f"File saved successfully at {map_file_path}")
    # Open the HTML file in the default web browser
    webbrowser.open('file://' + map_file_path)
else:
    print("Error: HTML file was not created!")
```

#HTML(f'<iframe src="{map_file_path}" width=700 height=450></iframe>') used above method instead of this line