# EVR-5086 Assignment 2

# Reading On-line Data and Visualizing Hurricane Tracks using Python

## Objective:

The objective of this assignment is to practice data parsing, manipulation, and visualization using Python. Students will produce Python code to read hurricane track data from the NOAA HURDAT2 dataset and create a map of a hurricane track. Interactive maps are possible using the Folium library.

The code should allow users to select the hurricane or hurricanes of interest for visualization.

## Instructions:

**Part 1: Data Retrieval and Parsing**

1. Download the HURDAT2 data, ideally as part of your code. The statement

!curl -o junk.txt -X 'GET' 'https://www.nhc.noaa.gov/data/hurdat/hurdat2-1851-2024-040425.txt' worked well for me. A file that explains the data format is available at <https://www.nhc.noaa.gov/data/hurdat/hurdat2-format-atl-1851-2021.pdf>

2. Write Python code to implement a simple user input interface that allows the user to specify the hurricane ID code (like AL092021 for Atlantic storm 09 in 2021) or alternatively the name or names they want to visualize and the year or years of interest, because the storm names are not necessarily unique.

3. Use Python code to parse the downloaded HURDAT2 data to extract relevant information such as hurricane date, latitude, longitude, maximum wind speed, and minimum pressure. Organize the data into a suitable data structure for further processing.

**Part 2: Data Visualization**

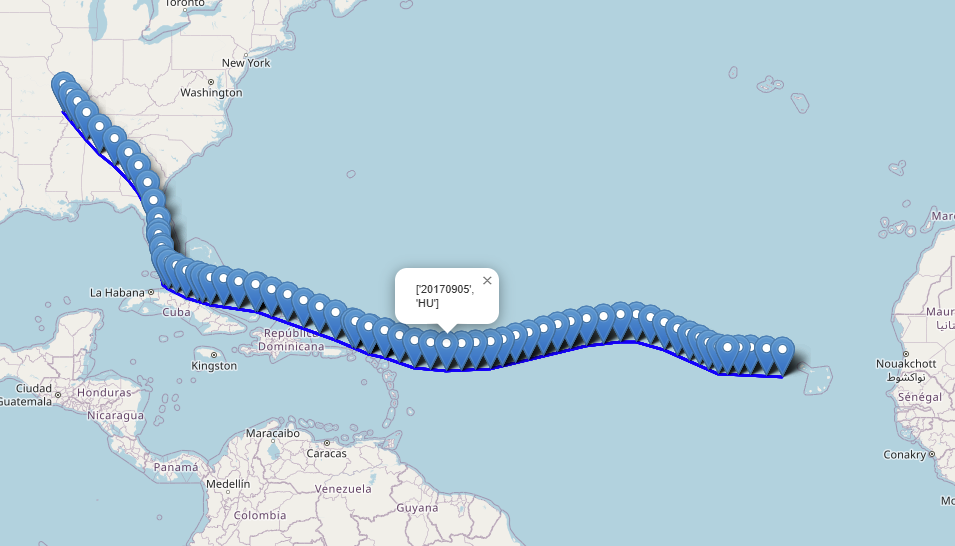
4. Find a way to properly plot the latitude and longitude data that constitute the hurricane’s track. Optionally, use the Folium library to create an interactive map, and ideally your map will include at least some reference to land outlines. A Folium command that might be helpful could look like marker = folium.Marker(location=[float(lat[i]), -float(lon[i])], popup=str(date[i])).add\_to(m). Inside a loop over previously stored latitude, longitude, and date lists, this adds a marker to each time point on the track and displays the date when you click on the marker.

Similarly, this piece of code is all that is needed to web-/browser-enable your result as an html file:

# Save the map to an HTML file

m.save('hurricane\_tracks\_map.html')

My map is below and also available as a great interactive and portable html file here: <https://faculty.fiu.edu/~sukopm/EVR5086/hurricane_tracks_map.html> Give it a try!



**Part 3: Submission**

5. Organize your Python code for readability and include explanatory comments.

6. Write a clear and concise report that includes the following:

- A brief explanation of the code's purpose and functionality.

- Instructions on how to run the code

- Sample screenshots or maps generated using your code.

- Any challenges encountered and how they were overcome.

**Grading Criteria:**

Your assignment will be evaluated based on the following criteria:

1. Correctness of code: The code should accurately parse the HURDAT2 data, handle user input, and create an accurate map.

2. Data visualization: The map should effectively visualize the selected hurricane tracks with appropriate markers and optional popups. You are encouraged to explore additional features and customization options for the map to enhance the assignment.

3. Report: The report should be well-written and provide clear documentation of the code's functionality and any challenges faced. As a reminder, English, spelling, units, significant figures, quality of graphics, accuracy of analysis, and quality of evaluation will all be considered in grading the homework. You must write a coherent report that explains what is presented.