# Assignment 2: Northern Hemisphere Cyclones

IS597PR. Copyright John Weible. Modified 1/28/25

The NOAA National Hurricane Center has several databases available. We’re going to work with the historical “HURDAT2”. From <https://www.nhc.noaa.gov/data/#hurdat> we can download two data files (N. Pacific and Atlantic. The accompanying PDFs on that same webpage describe the data format in fine detail. Read through the PDF and view the data files to get familiar with the custom format.

## Submission Requirements:

1. Your final submitted solution to this assignment must be a Python version 3.11 or higher .py file (A module, not a Jupyter Notebook.)
2. If you are a team of two students, please have only one student from your group submit the completed work.
3. You must include comments at the top of your program file that list each student’s full name & NetID, and a summary of how you worked together (who wrote which parts, etc.).
4. INCLUDE a text file showing the complete output from your program having processed both files together! If the program prints to the console, then you may copy-and-paste from the console window into an editor to create that file OR have your program print directly to a single text file.

## CORE GOALS and Specifications:

### Package Restrictions:

Because there are some relative novice Python programmers in this class and they’ll need to review and understand your code, we don’t want you to jump ahead by using complex features and libraries we haven’t discussed yet or that aren't in the assigned readings from class sessions 1-3. Just use the fundamentals for now, only *standard library* modules. This means even if you have used Pandas or NumPy modules before, **do not** use them on this assignment, it would be very confusing to some of your peers. Those libraries are not as much help here as you may expect anyway, but we will study and use them soon.

TIPS! You’ll need to use strings, lists, and dictionaries (or the similar collections.Counter class instead), the datetime library and text file I/O for this program.

1. Download both current HURDAT2 data files (Atlantic & N. Pacific). Do not modify or rename the data files, make your program process them without modification.
2. Remember the DRY principle (Don't Repeat Yourself). Try to use loops or custom functions instead of duplicating lines of code to do identical or similar work again.
3. [Skill: Memory-efficient programming]  
   Write your code so that it does not read in or accumulate the entire data set/file into memory. *[We discuss a similar example in class Week 2.]* The data files are already sorted chronologically, so you can take advantage of that. An incremental algorithm here is far more memory efficient and nearly unlimited in how much data it can process. Many students want to work on "Big Data" analytics – one of the important techniques for that is incremental processing. This means your program will need to release all the detail rows for each storm from memory before it proceeds to load the next storm.
4. As it processes each HURDAT file (which should be unmodified including keeping the original filenames), it should compute and print out the following data for each storm:
   1. Storm system ID (e.g. ‘AL011851’). Also show its name if not ‘UNNAMED’
   2. The duration of time the storm was tracked, as days + hours. (datetime library is helpful).
   3. The highest ***Maximum sustained wind (in knots)***.
   4. The lowest ***Minimum in central pressure (in millibars)***.
   5. How many landfalls, if any, did it have *while* at Category 1 or higher? (see below for Categories)

1. After all the per-storm output above (for BOTH files), your program should output these aggregate annual counts (for both files combined). Use either the string .format() method or an f-string to arrange these as right-justified, fixed-width columns, sorted chronologically, by year. Do not omit years that have no storms at all -- show the zeroes. The columns to show are: “Year”; "Storms” - *Total* number of storms tracked per year; “Cat1” – the number of storms (not rows) that *peaked* in Category 1; “Cat2” = the number that *peaked* in Category 2; “Cat3”; “Cat4”; “Cat5”. These yearly totals are combined from both oceans/files. See <https://www.nhc.noaa.gov/aboutsshws.php> for the ranges.

Example: The output for this part should begin with:

Year Storms Cat1 Cat2 Cat3 Cat4 Cat5

1851 6 2 0 1 0 0

1. Go back through your program to improve anything you have unnecessarily hard-coded. This is part of high-quality coding practices for flexibility, reliability, and re-usability. As examples:
   1. If years 1851, 1949, or 2023 are in your program source code as a literal value in loops, ranges, or indices, they shouldn't be. Make the loops work no matter what dates are in the data files!
   2. If your program has a full filesystem path to a datafile, it will not work as is on anyone else's computer, so only use *relative* path names. (Python's file I/O functions default to looking in the same directory where the executing program module is located.)

## STRETCH GOALS:

Each of these are individually **optional**, to encourage you to try and learn more beyond the *Core Goals*. However, we strongly encourage everyone to try implementing as many as you can to learn more and earn “Mastery” points. See the Syllabus on grading for more about how this relates to “Evidence of Mastery”.

Please always mention at the top of your program which stretch goals you attempted! This helps our TA grade more efficiently, and keeping your graders happy is wise! ☺

1. [Skill: Writing Functions *Properly*] In all later assignments, writing good functions will be a Core expectation. Define your functions *following proper rules for scope access, use of parameters, and return values* and have *complete Docstrings*.
2. [Skill: Debugging] Use the PyCharm Debugger to help you verify that your program successfully does memory-efficient "incremental file processing". To do this, make it pause the execution by setting a breakpoint at a line after the program has *finished* reading the input files. Then carefully inspect all the list or dictionary type variables you created to make sure that they don't contain things like all the Storm IDs, or the storm Names, and not individual detail rows of data. How confident are you that it's working right? Take screenshot(s) to show evidence of using the debugger in this way. Let us see where in the code execution is paused and the relevant variables expanded if necessary.
3. [Skill: Basic Analytical Calculations] The Tropical Zone is the belt of Earth between the Tropic of Cancer (23.436° North) and the Tropic of Capricorn (23.436° South). Calculate these statistics considering all storms in both HURDAT2 files together:
   1. What percentage of all storms ORIGINATE within the Tropics?
   2. What percentage of all storms STAYED ENTIRELY within the Tropics?