590PR Assignment A5 - Pandas
"Intro to Aircraft Tracking: Analysis on a Continuous Data Stream"

As discussed in class, many real-world data processing and analytics situations require "real-time" (nearly instant) and continuous calculations of a stream that doesn't stop.

For logistical convenience reasons, at this point you'll work with some fixed data files that were captured from a stream over a span of time, but still design the program to process them as if they were coming in a continuous stream of unbounded length. You'd want to develop and test on static data first anyway, to verify that your program works correctly before using it in production on live data. It also allows the graders to compare your program's outputs with known inputs. (Later, we'll discuss how to feed the actual live stream into your program to run it that way too.)

Use a similar technique to the "incremental processing" we discussed in Week 2 and used for Assignment 2, and continually update your calculations (and relevant outputs) while you read in more data.

The files I'm supplying you include several hours' continuous ADS-B transponder messages broadcast by aircraft within range of my ADS-B receiver. They were recorded on different dates and from two locations here in Champaign.

The "Jan15_3_hours.csv" file was recorded with my receiver antenna at lat/long (40.0893, -88.266) at approx 741 feet altitude.

The "Jan30_5_hours.csv" file was recorded from my office, with antenna at lat/long (40.10762, -88.23145) at approx 741 feet altitude

This format of the messages is just CSV, and the descriptive metadata you need is here:

The column headings:

```
['MessageType', 'TransmissionType', 'SessionID', 'AircraftID', 'AircraftHex', 'FlightRecorderNum', 'DateGenerated',
'TimeGenerated', 'DateLogged', 'TimeLogged', 'Callsign/FlightNum', 'Altitude', 'GroundSpeed', 'Track/Heading',
'Latitude', 'Longitude', 'VerticalRate', 'SquawkCode', 'Alert (Squawk Change)', 'Emergency','SPI', 'OnGround']
```

The units for those columns containing measurements are:

Date*	UTC date
Time*	UTC time
Altitude	feet
Latitude	degrees (should all be positive numbers, indicating Northern hemisphere)
Longitude	degrees (should all be negative numbers, indicating Western hemisphere)
GroundSpeed	knots
Track/Heading	degrees (compass heading, 0-360)
VerticalRate	feet/minute

Make your program display on the console or to a text file as it runs, and create a parameter that defines the "update_interval". The interval should work properly with a range anywhere from 1-60 minutes.

Use the "TimeLogged" data field as a way to determine how much time has elapsed as you read through the file. That way if you later run this on true real-time stream it will still work right. Update the display by adding a new line every "update_interval", formatted in fixed-width aligned columns similar to the example, to show all these things:

Example output format (with fake data) with an interval of 5 minutes. Aircraft that haven't been heard from within the interval will disappear from CURRENT RESULTS.

```
****************** CURRENT RESULTS ******************** | ******* CUMULATIVE RESULTS *******
TIME     #Craft Fastest (kts)  Highest (ft)  Msgs/Sec | #Craft  LongestTrack (nm)
22:45:00    15   AABBC0  552   ACF08F 42000     45.1  |    15     A78842   27.2
22:50:00    12   AABBC0  560   ACF08F 42000     32.8  |    16     A78842   52.3
22:55:00    18   ACF08F  590   ACF08F 40500     60.3  |    21     A78842   77.2
23:00:00    22   A9978A  525   ACF08F 37300     72.7  |    30     A78842  101.2
23:05:00    21   A9978A  520   A9978A 35000     70.7  |    30     A9978A  114.0
```

Use the string class format() method to force all those outputs to fixed-width, rounded properly, and with numbers decimal-justified.

The aircraft IDs all come from the `AircraftHex` field to uniquely designate them. (That's also their ICAO-assigned transponder's broadcast ID).

"#Craft" means number of DISTINCT AircraftHex values from which we received messages within the current update_interval, or for cumulative, that means since the program began running.

"Msgs/Sec" means how many ADS-B Messages (rows of data) were received within the update_interval, divided by number of seconds.

For "LongestTrack (nm)" you'll have to continually calculate distance traveled per Aircraft, and at the point of output, you only display the one aircraft with the longest traveled distance you've seen so far. Since it's under the "CUMULATIVE RESULTS" section, it doesn't matter if you haven't seen that plane for hours, you keep showing it *until some other plane's track distance exceeds it*. This is similar to the tropical storm propagation distances, where each new GPS position for a craft allows you to add on a new straight line segment from its previous position.