Coding Exercises

These exercises are here to allow you to do what you do best - write great code. It's your opportunity to shine and show us what you can do. There are two problems to pick from for this exercise. Please pick one challenge to show case your coding skill, analytics skill, and your familiarity with data wrangling.

Happy coding!

The Synpulse team

Date Difference Challenge

Date processing is very essential in many data projects. In this challenge, you need to calculate the number of full days elapsed between two events.

The first and the last day are considered partial days and never counted. Following this logic, the distance between two related events on 03/08/2021 and 04/08/2021 is 0, since there are no fully elapsed days contained in between those, and 01/01/2021 to 03/01/2021 should return 1. The solution needs to cater for all valid dates between 01/01/1901 and 31/12/2999.

Test cases

1) 02/06/1983 - 22/06/1983 = 19 days

2) 04/07/1984 - 25/12/1984 = 173 days

3) 03/01/1989 - 03/08/1983 = 1979 days

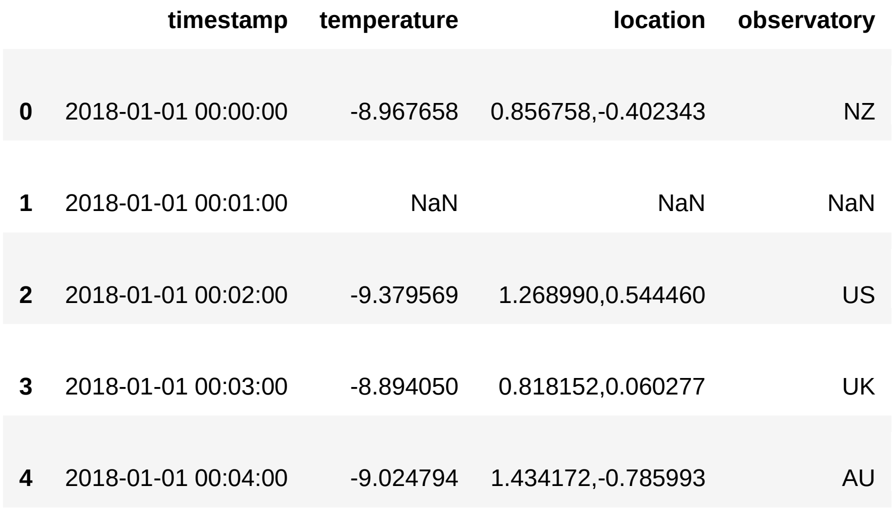
(Please note these dates are formatted as DD/MM/YYYY)

Instructions

- Write a program that accepts date input from the console.
- You can choose any language for your program.
- Please make sure your code is clear and robust.
- Please attach your test cases.
- You should not use any existing date libraries for your implementation.
- You may however use date libraries to test your solution. We encourage it!
- Consider other potential input sources & how your app might fit into a bigger system.

Weather Balloon Challenge

We are part of a global weather monitoring program. A lot of hydrogen weather balloons were released to flow around the globe to collect important weather data. The data was collected by on-ground observatory stations. The collected data includes timestamp, temperature, latitude, longitude, observatory site code, etc. The data looks like:



	timestamp	temperature	location	observatory
0	2018-01-01 00:00:00	-8.967658	0.856758,-0.402343	NZ
1	2018-01-01 00:01:00	NaN	NaN	NaN
2	2018-01-01 00:02:00	-9.379569	1.268990,0.544460	US
3	2018-01-01 00:03:00	-8.894050	0.818152,0.060277	UK
4	2018-01-01 00:04:00	-9.024794	1.434172,-0.785993	AU

Timestamp is the time the data been collected; temperature is in Celsius degrees; location is the longitude and latitude of the balloon; observatory is the country code of the observatory site. As depicted in the sample data, the real-world data is full of imperfection. The sensor could be faulty, the data points could be missed completely, and the datapoints were not guaranteed to come in order.

At the first stage, we need a program (or set of programs) that can perform the following tasks:

- Given that it is difficult to obtain real data from the weather balloon we would first like to be able to generate a test file of representative (at least in form) data for use in simulation and testing. This tool should be able to generate at least 500 million lines of data for testing your tool. Remember that the data is not reliable, so consider including invalid and out of order lines.
- Use a Jupyter notebook to showcase how you do EDA on the simulated data.
- Showcase how you wrangle the data and generate insight from the simulated data.