Exercise - 04

In today's exercise we will practice swimming in the waters of scientific computing and use python for it. There are two sets of exercises (set_01 and set_02). You are required to submit two jupyter notebooks with the following naming conventions.

1. <your_roll_no>_DA213_class_04_set_01.ipynb
2. <your_roll_no>_DA213_class_04_set_02.ipynb

The set_01 is a warm-up exercise.
You are recommended to start with this notebook. After completing this notebook we expect you to be familiar with usage of numpy and how this data type is different from the built in datatypes, from a computational standpoint.

To start, open the file: DA213_class_04_set_01.ipynb, create a copy of it and rename it to <your_roll_no>_DA213_class_04_set_01.ipynb and begin executing and filling the cells.

The set_02 aims at the following.
Introduce you to the process of breaking down a problem into parts, and use your knowledge of python to help come up with a solution. The end goal is to help you come up with an reasoning behind a physical phenomenon, evaluated using a computational approach.

Problem Description:
Makar Sankranti (MS) is a Hindu festival, and it is celebrated every year on 14th or 15th January. Every year 14th or 15th January witnesses the celebration of multiple festivals around India. One of them is Makar Sankranti. Most of the Hindu festivals follow the moon position, and this makes their date vary from year to year. However, Makar Sankranti follows the position of the Sun (not moon). But does the sun move? Of course, it does. I mean everything in this universe is moving, but - not everything is moving by the same amount. Have you experienced the movement of the Sun? No, at least I have not. Then what does it mean when we say Makar Sankranti follows the position of Sun.

Imagine you are inside a train, traveling alone. The train is moving and you are sitting on the lower berth. You took a deep nap, and something fell on your head - you woke up and now you have forgotten that the train is moving. Instead, you look out of the window, and start to think the world outside the train is moving and not the train. This is the scenario we will imagine for a while - you are sitting in the train, and the world outside is moving.
You noticed 12 stations on the way, but the train didn't stop at any of them. It is moving, moving and moving, and at the same speed. But wait - what happened on the 13th station! It looks similar to the 1st station. You are surprised. And then comes the 14th station, it looks similar to the 2nd station! Oh my goodness, the stations are repeating. After a while you conclude that the 12 stations keep repeating, and that too in the same order. You do not have a watch, and the train hasn't stopped for a long time. What do you do? You start keeping track of time via the cycle of the 12 stations!

Now let's switch to the world we live in. The train is our Earth. It is moving around the Sun on a fixed orbit. Depending on where we are on this orbit, when we look up at the sky we see different constellations. Sometimes we see the Sun entering some constellations, and sometimes leaving some. Humans have been smart. In our course of years of journey around the Sun, humans have documented that the Sun goes through a cycle of 12 constellations, and this cycle repeats! That's how we decided to have 12 solar months (my guess). The transition of the Sun from one constellation (also called rashi) to another is called Sankranti.

What is "Makar" in Makar Sankranti?
Now, what is "Makar '' in Makar Sankranti? Makar is a legendary sea-creature in hindu mythology. Likely the shape of one of the constellations appears like a "Makar". According to Hindu astrology, on the day of Makar Sankranti,

the Sun enters the Capricornus constellation. **This marks the end of short days, and from this day onwards the daylight duration begins to increase.**
Can we verify this? Let's try it out through this exercise.
==

Step 1:
How do we compute the daylight duration (that is, (sunset time) - (sunrise time))?
Is there a python package for it? Yes! Check this: https://pypi.org/project/suntime/
Install this package in your python environment. How? Use pip install suntime on your linux terminal.
How do you use this package? Check this documentation and see if you can figure out how to use the below function:
sun.get_local_sunrise_time(this_date)

Step 2:
To use the above function you will need to define a sun object. This can be done as follows.
sun = Sun(latitude, longitude)
For location we will use Guwahati. A CSV file is provided to you which list the latitude and longitudes of several Indian cities. Load it into your program and get the latitude and longitude for Guwahati.

You will also need the this_date variable which stores date in standard date format. For this, first install the datetime package. Then define following date object,
this_date = datetime.datetime(2022, 1, 1, 0, 0, 0)

Step 3:
Now you can use: sun.get_local_sunrise_time(this_date)
Analogously, check if there is a function to get sunset_time

Step 4:
Make a plot of daylight duration (in hrs) versus days of the year (for 2022).
The plot dimension should be 10 in x 5 in (width x height). Check the use of:
figure_name = plt.figure(figsize=(width, height)) for this. Do not miss adding grid lines, xlabel, ylabel, etc to make the plot easily readable.

Step 5:
Print the date corresponding to longest and shortest daylight duration. Write a paragraph (2-3 lines, in a markdown cell) about what happens around 14-15th Jan.

Step 6:
Following up, let's obtain the temperature variation in Guwahati for the same year. How do we get this data? Check this https://pypi.org/project/meteostat/
Install this package.

Step 7:
Fetch the weather data frame for Guwahati city for 2022, every day. Plot the resulting average temperature data.
The plot dimension should be 10 in x 5 in (width x height). Check the use of:
figure_name = plt.figure(figsize=(width, height)) for this. Do not miss adding grid lines, xlabel, ylabel, etc to make the plot easily readable.

Step 8:
Can we establish a relationship between temperature (in degree celsius) and daylight duration (in hrs)? Make a scatter plot of temperature versus daylight duration. Do you see a correlation? Check by computing the correlation coefficient and printing the same.

Step 9:

Use a second degree polynomial to fit the relationship between temperature and daylight duration. Obtain the coefficients of this polynomial from the data for the year 2022. How? Solve a linear algebra problem using numpy (example, np.linalg etc.). Print the coefficients, and the mean square error in the fit. Overlay the estimated temperature data in the original temperature versus day of the year plot.

Step 10:
Interpret any observation in a markdown cell. Did the prediction work? How can the error be reduced? Are there any additional variables which can improve the prediction?

Thank you!
