Scientific Programming with Python
Assignment: The Interaction Between Two Atoms

Karl N. Kirschner

Department of Computer Science, University of Applied Sciences Bonn-Rhein-Sieg,
Sankt Augustin, Germany

January 10, 2024

Goal The goal of this assignment is to make use of the scientific computing knowledge learned so far. Specifically, you will use your knowledge of matplotlib [1], NumPy [2], and Pandas [3, 4] libraries.

Problem and Input Data In Toronto Canada, the Environment and Climate Change Canada's National Air Pollution Surveillance (NAPS) program recorded the hourly concentrations (parts-per-billion; ppb) of ozone (O_3) and nitrogen dioxide (NO_2) at five different stations [5]. These two molecules are considered atmospheric pollutants when near the Earth's surface. Their relationship to each other can be seen the following reactions:

$$NO_2 \xrightarrow{h\nu} NO + O \tag{1}$$
$$O + O_2 \longrightarrow O_3 \tag{2}$$
$$O_3 + NO \longrightarrow O_2 + NO_2 \tag{3}$$

where $h\nu$ represents light. During the daytime, Eq. 1 dominates due to the sunlight that is present, resulting in an increase in O_3 concentration. However at night, Eq. 3 dominates, resulting in an increase of NO_2 concentration. These reactions result in a correlation to occur between the concentrations of O_3 and NO_2 [6].

The data was collected in five CSV-formatted files, entitled "Toronto2020_Station_n.csv", where $n = 1 - 5$. The content of these files include columns for the "Time" (i.e., date and time), O_3 concentration (in ppb), and NO_2 concentration in (ppb).

Assignment Tasks

Task 1 Air Quality Data (Pandas)

- Create a single dataframe that contains the data from all five stations.
- Convert the "Time" data to the datetime format using Pandas' 'to_datetime'.

Task 2 Calculating Means and Visualize the Data (Pandas and matplotlib)

- Using as much hourly data as available across all stations, compute an *hourly mean* concentrations for O_3 and NO_2.
- Smooth the O_3 and NO_2 *hourly mean* concentrations by computing a *rolling* (i.e., moving) *average*, using a window of 24 hours.
- Using matplotlib, create a plot that shows the following:
  - O_3 and NO_2 *rolling average* concentrations as a function of the recording time.

    (Hint: It will look similar to Figure 2A of Reference [6], but with a longer x-axis timescale.)

Task 3 Creating O_3 Prediction Models (NumPy and matplotlib)

- Using NumPy, create and optimize a 2-degree polynomial using the NO_2 and O_3 *hourly mean* concentrations. Print the resulting polynomial equation to the screen. The resulting polynomial should use NO_2 concentrations as the input data to predict O_3 concentrations.
- Using matplotlib, create a plot that shows the following:
  - NO_2 and O_3 *hourly mean* concentrations are correlated (i.e., a scatter plot), and
  - the resulting polynomial curve (i.e., a line) that fits the data.

  (Hint: It will look similar to Figure 2B of Reference [6].)

Task 4 Compute Root-Mean-Square Errors (NumPy)

- Write a function to compute the root-mean-square error (RMSE) between theoretical (i.e., predicted) and experimental (i.e., actual) values for a given observable (e.g., O_3 concentrations). This equation has the following form:

$$RMSE = \sqrt{\frac{1}{n} * \sum_{i=1}^{n} (X_i \text{ (Theoretical)} - Y_i \text{ (Experiment)})^2} \tag{4}$$

- For each of the five stations, compute an RMSE value when using its experimental NO_2 concentrations as input into your polynomial model to predict O_3 concentrations.

  (Hint 1: Drop rows that contain empty and 'NaN' cells from entire dataframe.)

Allowed Python3 [7, 8] functions & libraries/modules

- All built-in functions
- matplotlib [1], NumPy [2], Pandas [3, 4] library
- typing and unittest library (if needed)

Assignment Due Turn in your solution as a **Jupyter-notebook** [9] to **LEA** by **Monday, January 22nd, 2024 at 09:00**.

Note 1 : Please include your **SciPro_ID** at the **top** of your notebook.

Note 2 : **Do not** consider **significant figures** in this solution.

References

[1] Hunter, J.D., 2007. Matplotlib: A 2D graphics environment. IEEE Annals of the History of Computing, 9(03), pp.90-95.

[2] Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature, 585 (2020) 357–362 (DOI: 10.1038/s41586-020-2649-2)

[3] The Pandas Development Team pandas-dev/pandas: Pandas Zenodo, 2020 (https://pandas.pydata.org)

[4] Pandas user guide. Available at https://pandas.pydata.org/docs/user_guide/index.html. Accessed on November 7, 2023.

[5] Environment and Climate Change Canada, "National Air Pollution Surveillance (NAPS) Program". Available at https://data-donnees.az.ec.gc.ca/data/air/monitor/national-air-pollution-surveillance-naps-program. Accessed on December 15, 2023.

[6] David Ross Hall and Jessica C. D'eon (2023) How's the Air Out There? Using a National Air Quality Database to Introduce First Year Students to the Fundamentals of Data Analysis. J. Chem. Ed., 100, 3410-3418. (https://doi.org/10.1021/acs.jchemed.3c00333)

[7] Python Software Foundation. Python Language Reference, version 3.11. Available at http://www.python.org. Accessed on October 22, 2023.

[8] van Rossum, G. Python tutorial, Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, 1995.

[9] Kluyver, T. et al., (2016) Jupyter Notebooks – a publishing format for reproducible computational workflows. In F. Loizides & B. Schmidt, eds. Positioning and Power in Academic Publishing: Players, Agents and Agendas. pp. 87–90.