Data Science Project
Owen Shaffer and Owen Himmel
10/20/2024

This project helped us learn a lot about data conversion and processing, but it also came with its fair share of hardships as well. Firstly, we will start by describing what was easier than we thought it was going to be. The main thing that we thought was going to take more time than it did was the actual conversion from one data type to another. Pandas has a few functions that make converting data types fairly easy, and examples of these are df.to_csv or df.to_json. These made the steps of the coding pipeline pretty simple. Additionally, it was relatively simple to import the data. By calling the repo straight into the google colab notebook, we are able to seamlessly sync the data that we had uploaded into the repo for use in the code. Also, there was a lot of open source data and we happened to get ours from data.gov which has an abundance of environmental-related datasets.

While some parts of the project were easier than anticipated, some were much harder than we first thought they would be. For example, we had expected to store the files much easier than we were able to. For example, when we convert from csv to json, the table only outputs one row with thousands of variables instead of several rows and columns. One general strategy that we applied to the data to correct for this problem was to convert the data as strings before we converted the file type. Another issue that we encountered was how to best export the new converted files once converted. However, we learned that by adding the file type to the end of the saving process (.csv, .json, .sql) it automatically populates in the google colab session and is available for download onto the local disk. One final area that was harder than we initially realized was error coding and identifying the dimensions of the new datasets. Neither of us were that familiar with how to code for errors, so we needed to look up some tips from various help tools online. Also, we learned how to identify the number of rows and columns for all the newly converted datasets by looking back at only class material (but this took a while).

I think that this pipeline could be very useful for all sorts of data science projects that we may work on. Specifically, it is easily adaptable due to the fact that it uses user input as a key to complete functions. For example, if instead of the data that we currently have we wanted to substitute it with another csv or json file, all that we would need to do is to provide the path name and use the same pipeline to investigate the data dimensions and convert it to another type of data format. We are also very interested in implementing an API call into this pipeline framework, so it would constantly be updated with new json fields and perhaps graphed as an output of a csv file. This pipeline also allows the user to easily drop certain unwanted columns which may occur when doing API calls on new internet pages. All in all, this pipeline makes the data analysis and processing much more efficient by providing error messages, data descriptions (like dimensions), and makes editing the data fairly simple all with user input.