## **Practical Python for News Investigations**

**Fall 2024**

**Gabriella Bass**

**Final Memo**

For my final project, I took on three possible projects that landed at various stages of completeness (incompleteness would be more accurate). That being said, I think the work, structure, and trajectory of the projects are clear.

1. I looked at Ready-Mix cement companies in the New York area. I worked off of three search criteria, codes 9999 (misc.), 3273 (Ready-Mix Concrete), and 3271 (Concrete Block and Brick), which are all associated with cement and concrete companies across the country. I originally took a nationwide database and whittled it down to New York companies. Identified companies that were located within the five boroughs and pulled them into a separate dataset. I then compared the column categories of each set to preserve the data from each group and contacted them into one larger set tagged with their original coding structure. The goal here is multifaceted.

* To create a master list of cement companies in the New York area
* To identify violations and oversite
* To identify companies that have no violations and have not been scrutinized by inspections.

I was successful in compiling and concatenating the data using Python. Structurally, this was a success. However, this is a classic case of ‘garbage in, garbage out.’ Firstly, the original data requires some refinement and scrutiny- especially dataset 9999 which includes many businesses that have nothing to do with the concrete industry but fall under the classification of ‘Misc’. Also, the data, which was acquired from the EPA’s APIs may not completely represent the target data I’m hoping to analyze. I will have to compare this to the data set I’ve been constructing manually to get a clearer picture. Although the original data and compiled data set are inaccurate, the code will be similar moving forward.

2. I attempted was to take a set of court documents (375 pages) that I copied by photographing each page with my phone. I organized the images and turned them into PDFs.

* I globbed the files to read them
* Used MangCR to convert them to python friendly text
* Used spaCy to identify keywords throughout the documents

My goal here is to parse through court documents to extract information and organize searches. I was able to pull terms from the individual documents and export them to CSVs.

3. Flattening JSON

I cURLed a cement company on the EPA ECHO site. The cURL produced a data set of the pollutant concentration for that company. It came in the form of a list of dictionaries. I turned that into a JSON file and flattened it into a data frame. I had hoped to be able to cURL the whole site, but I was unable to find the necessary files, which is what led me to work with the APIs above.

* cURLed website
* flattened list
* exported as csv

Original data:

https://npdes-ereporting.epa.gov/net-biosolids/api/public/v1/reference/pollutantConcentrationLimits