Michael Amodeo
W18 Python
Water Balance - MidTerm Project

Project Reflection

Project instructions are included in the file readme.txt

Over the course of my midterm project, I made many adjustments to the overall structure of the program. My initial setup of the classes mostly worked pretty well, as I knew what specific objects I was going to work with (parcels) and the attributes I wanted them to contain. In the end, I removed many attributes that I deemed unnecessary to be contained in every class instance because I only needed the aggregate value, which I could find later through a function call. For instance, the irrigation volume calculation was something that I moved outside of the classes because I ultimately did not need to know the irrigation demand each month for each parcel. I was only using this calculation on some total (project total or land use total), and it was not necessary to create 12 additional attributes for each instance. Instead, I passed the irrigated area value from the class instance to the function with the climate data, and performed calculations there.

Initially, I found that I was putting most of the functionality of the tool into the main run file and outside of the classes. However, as I built the full tool, I realized that certain elements were being repeated, and that those could be moved either into the classes as methods, or as separate functions that I could call repeatedly. The difference between these two approaches ultimately depended on where the function would be called within the program and whether it aligned with each class instance. The separate functions stayed separate in many cases because it was not needed for every class instance. I tried to minimize the number of objects I created, so some things that would not change stayed outside of the classes (climate data, land use assumptions). Again, this influenced the decision to put the irrigation calculation as a separate function and not a class method. Also, because I was working with so many instances, the need for iteration really dictated what I could do inside the classes and what I would do outside.

One change I made during the course of the project was to create new classes out of some data that had been lists. When I had multiple lists that had a common order because they were linked somehow, I found it efficient to create a list of class instances so I could call values by attribute instead of just their offset. I did this specifically for the months and the water use assumptions by land use type. I had gone back and forth on whether to make these dictionaries, lists, or named tuples, but ultimately went with classes because it seemed the best way to contain and call the data. I like being able to link across different lists based on the common object (a month has a number of days as well as an average rainfall, etc).

Time constraints and project requirements also dictated what I could include in the project overall. There were many scenarios or analyses that I had wanted to add, but it became clear that a lot of that functionality was outside of the scope of the project and I felt the basic functionality demonstrated the content that we were supposed to show. Secondary functionality like the webscraping and exports were included where possible, though not all were used.