Elizabeth Wickes
Data Cleaning
Final Project proposal

I am trying to solve the problem of overcoming the workload barrier of writing documentation for datasets. Simple and small datasets can be relatively easy to skim over and pick out columns for description. However, datasets can easily increase in complexity and quickly move out of reach for the average researcher to handle manually. The barrier between a researcher and robust documentation is one of time and detail.

Documentation will always need a human eye and attention, but a tool to create a template representing information that can be programmatically aggregated would allow the researcher to focus much needed efforts on the elements that truly require their personal knowledge. Tools like Yes Workflow tap into this philosophy, allowing the authors of scripts to add low-level markup within the code to programmatically produce a high-level description and visualization of the data process.

This project has a potentially unlimited scope when it comes to features. My goal with this project will be to produce a framework that could easily add in custom analysis for specific data types and file formats. The initial pass for this class would be something that could look over a set of CSV files, produce high level descriptions and aggregate statistics on that data set, and export out a SQL db file that could have queries run on it. This database could then also be run through a report system that automatically creates something like a rudimentary codebook for the researcher to review and fill in the contextual information that would be impossible to infer from the looking solely at the numbers.

A good example of this would be attempting to detect Likert scale questions. A detection algorithm would look for columns of data containing only integer values between 1 and 5, 6, or 7. Only these numbers would be seen, and we would expect to see all of them. However, a given survey, particularly with a low number of respondents, may only have values within limited clusters. This means that ranges on a given 1 to 7 Likert scale may only be observed between 2 and 4. Additionally, the test of 1 as an observed minimum could be true while the observed maximum in the dataset is incorrectly lower than what was offered on the instrument. Accompanying text matching these values may or may not be present within the data file. Certainly it would have been stripped and replaced by the respective integer value if the data file had been used for analysis. The tool could produce the core observed descriptive statistics of minimum, maximum, mean, standard deviation, and median. However, the researcher would need to use their knowledge of the survey instrument to complete the context of the actual responses and question text. Checking that Likert scale questions have correct balance is vital for appropriate analysis and reporting.

I will be using Python tools to create this tool, with a preference of standard library modules. However, as I've been exploring techniques, it appears that I cannot move away from using the Pandas module to arrange and output the data into a transportable format, such as a SQL database. I have also created a tool that will produce a set of CSV files with random values and lengths. This simulated data will be the files that I develop my script with. Again, the scope of the project is quite large and handling of special cases or messy data are features that could be added in at a later point.