Mini Project 1: Structured Data

IST652 - Scripting for Data Analysis

M002: Spring 2020

Due: Sunday, March 8th by midnight

For this mini project, you must work individually.

Structured Data Processing:

For the purposes of explanation only, we will use examples from Donors data (Donors_Data.csv) to help convey the requirements.

The main outline of your assignment is to write a program that will read in data from a file, such as a .csv, .tsv, .txt, or a file saved from excel. This will be in a format that is structured with lines of data representing one type of unit (e.g. one donor in the donors file). Your program must represent the data using learned Python data structures. You may choose the overall structure to be one of the following:

- Dictionaries, lists, or tuples
- NumPy Arrays (this topic will be covered in class on 2/17)
- pandas DataFrame (this topic will be covered in class on 2/24)
- Or some combination of the above

You will perform data cleaning and exploration on this data.

The programs you write will do some processing to convert the data to a form that will answer two questions, as described below, and write files with the data suitable for answering each question. Graphing is optional (e.g. Matplotlib).

Data:

You must first choose a dataset to work with. As a guideline, datasets should be chosen that have from about 500 to 4,000 lines of data with some number of columns between 5 and 50.

If the data comes in an Excel spread sheet with a lot of columns, it is OK to first edit the excel file to remove columns that you don't need for your processing. For example, in the Donors data, you might wish to create a separate excel spread sheet with only a few columns of data.

Questions:

For this assignment, at least one question that you choose should look at the data in a different unit of analysis than is present in the data file. For example, instead of looking at individual donors, you could look at the aggregated donors of each of the 9 income or wealth types.

Sample example question (NOTE: you should do a more complex problem than this):

For each wealth type, what is the average home value of all the donors of that type?

- Unit of analysis: wealth types
- Comparison: for each wealth type, compute the average home value of the neighborhoods of all the donors of that type
- Output: should be in a file with 9 rows of data (you may also produce header and label rows), where each row has an income type (1 – 9) and the average home values.

One way to have increased the complexity of this particular question would be to add more items to be compared to for income types (e.g. add columns to the output with average total gifts or values of the last gifts).

Another option would have been to introduce a more detailed unit of analysis, for example, suppose that for each income level, you reported by gender, giving the average home values for both men and women in each category.

Other ideas:

Compare donors in the various zip codes with various types or amounts of giving.

Compare donors by the number of promotions with the total amount of donations and the frequency of donations.

Compare the number of months since the last donation to the donation amounts.

Deliverable [total: 15 points]:

For this mini project, you must submit your data set, a program*, a report**, and output files. Your program must be submitted as a .ipynb file (same as the labs). You may submit the above either as separate files or as a single compressed file (.zip/.rar).

- • * A program (.ipynb) which does the following [subtotal: 10 points]:
  - Reads in data from a file [1 points]
  - Cleans and formats the data [2 points]
  - Analyzes/Summarizes the data in two or more different ways [6 points (3 x 2)]
  - Outputs the summaries in new file(s) (.cvs, .excel) with column headers [1 point]

- ** A report (.docx) which describes the following [subtotal: 5 points]:
- The data and its source [1 point]
- A description of your data exploration and data cleaning steps [1 point]
- Two clearly stated comparison questions with the unit of analysis, the comparison values and how they are computed. [1 point]
- A description of the program [1 point]

- A description of the output files [1 point]

For your program, you may use any of the code developed in class as a template, but it is absolutely essential that you use appropriate variable names and that you write original comments for what your program does. Recall that good comments demonstrate your understanding of the code that you write and the problem that you are trying to solve.