Mini Project 1: Structured Data
IST652 - Scripting for Data Analysis **M002: Spring 2023**
Due: Monday, March 6th by 5:00PM

For this mini project, you must work individually.

Structured Data Processing:
For the purposes of this write-up, we will use examples from Donors data (*Donors_Data.csv*) to help convey the requirements.

- **You must find your own data set for this project**

The main outline of your assignment is to write a program that will read in the data from a file, such as a *.csv, .tsv, .txt,* or .xlsx. This will be in a format that is structured with lines (aka rows) 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, and tuples
- NumPy Arrays (*this topic will be covered in class on **2/20***)
- pandas DataFrame (*this topic will be covered in class on **2/27***)
    - Accessible also from your readings (Python for Data Analysis)
- 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 but encouraged (try matplotlib, seaborn)***.

Data:
You must first choose a dataset to work with. As a guideline, datasets should be chosen that have between 500 to 5,000 rows and between 5 and 25 columns.

If the data comes in an Excel spread sheet with a lot of columns, it is **encouraged** that you manipulate the file in python. 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 donors of each of the 9 income/wealth types.

Simple 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

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 months since the last donation to the donation amounts.

Deliverable [total: 15 points]:
For this mini project, you must submit 3 files: your (1) data set, (2) a program*****, and (3) a report******. Your program must be submitted as a jupyter notebook (**.ipynb**). You may submit the above files to blackboard either as separate files (3) or as a single file (*.zip*).

- ***** A program (.ipynv) which does the following **[subtotal: 10 points]**:
  - Reads in data from a file [1 points]
  - Cleans and formats the data [3 points]
  - Analyzes/Summarizes the data in **two** different ways [6 points (3 x 2)]

- ****** A report (.docx or .pdf) 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 result of the analysis [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.

Using AI for Problem Solving:
The use of A.I. systems for this assignment **is not allowed**. This is an assessment of your problem-solving capabilities and requires mastery of the Python materials learned in this course to-date. Any usage of A.I. on this assignment will be considered a violation of academic integrity and will be handled accordingly.