Unit 9 Assignment - W200 Introduction to Data Science Programming, UC Berkeley MIDS

Write code in this Jupyter Notebook to solve the following problems. Please upload this **Notebook** with your solutions to your GitHub repository in your SUBMISSIONS/week_10 folder by 11:59PM PST the night before class. Do not upload the data files or the answer .csv (we want your notebook to make the answers when we run it)

This homework assignment is assigned during Week 10 but corresponds to the Unit #9 async.

Objectives

- Demonstrate how to import different data files
- Get a small glimpse on how messy data can be
- Design and implement an algorithm to standardize the information and fix the messiness
- Work with Python data structures to sort and output the correct information
- Demonstrate how to export required information to a .csv file

Reading and Writing Data (25 Points)

In this assignment, you will be reading and writing data. Yes, finally some data science (or at least some exploratory data analysis)! In the week_10 assignment folder, there are three data files named:

- data.csv
- data.json
- data.pkl

These are three common file formats. You can run the following **on the bash command line** to see what is in each file (this will not work from a Windows prompt but will work in git bash):

```
head data.csv
head data.pkl
head data.json
```

You'll see that there is some method to the madness but that each file format has its peculiarities. Each file contains a portion of the total dataset that altogether comprises 100 records, so you need to **read in all of the files and combine them into some standard format** with which you are comfortable. Aim for something standard where each "row" is represented in the same format. **Name this object that contains the data for all three files combined `full_data`**

Questions to answer (75 points: each question is worth 15 points):

After you've standardized all of the data, report the following information:

1. What are the unique countries in the dataset, sorted alphabetically? Write to a new file called question_1.csv.

2. What are the unique complete email domains in the dataset, sorted alphabetically? Write to a new file called question_2.csv.
3. What are the first names of everyone (including duplicates) that do not have a P.O. Box address, sorted alphabetically? Write to a new file called question_3.csv.
4. What are the full names of the first 5 people when you sort the data alphabetically by country? Write to a new file called question_4.csv.
5. What are the full names of the first 5 people when you sort the data numerically ascending by phone number? Write to a new file called question_5.csv.

We will be using a script to examine and grade your .csv files so please make sure:

- The answers are all in one **column** with one list item per cell, sorted as stated in the question. I.e., looking at the .csv in a spreadsheet editor like Google Sheets, all answers would be in the 'A' column, with the first entry in A1, the second in A2, etc.
- Please do not include a header; just the answers to the questions.
- It is strongly recommended that you open each .csv file to ensure the answers are there and displayed correctly!
- Don't include quotes around the list items. I.e., strip the leading and trailing quotes, if necessary, from items when you write to the .csv files. For example, a list entry should look like `Spain` rather than `"Spain"`. One exception: Some country names do contain commas and it is ok to have quotes: `""` around just those country names so that they will be in one cell in the .csv.

In addition, show all of your work in this **Jupyter notebook**.

Assumptions

- You might have to make decisions about the data. For example, what to do with ties or how to sort the phone numbers numerically.
- Write your assumptions in this Jupyter notebook at the top of your code under the heading below that says ASSUMPTIONS
- Please do some research before making an assumption (e.g. what is a domain name?); put your notes inside that assumption so we can understand your thought process.
  - NOTE: If you don't know what an email domain is - do some research and write what you found in your assumptions; there is a correct answer to this question!
- This is a good habit to do as you analyze data so that you can remember why you made the decisions you did and other people can follow your analysis later!

Restrictions

You should use these standard library imports:

```
import json
import csv
import pickle
```

Some of you may be familiar with a Python package called `pandas` which would greatly speed up this sort of file processing. The point of this homework is to do the work manually. You can use `pandas` to independently check your work if you are so inclined but do not use `pandas` as the sole solution

method. Don't worry if you are not familiar with `pandas`. We will do this homework as a class exercise using `pandas` in the near future.

Hints (optional)

- You may use regular expressions if you wish to extract data from each row. You do not need to use them if you do not want to or see a need to. The Python regular expression module is called `re`.
- You may want to use the operator library or the sorted function to help in sorting.
- There are many data structures and formats that you might use to solve this problem. You will have to decide if you want to keep the information for each person together as one record or all the information for each of the fields together.
- You can put these files into sensible structures such as lists or or dictionaries. The async covers how to do this for csv and json. For pickle this might help https://wiki.python.org/moin/UsingPickle (https://wiki.python.org/moin/UsingPickle)
- .items() or .key() can be useful for dictionaries
- Once again, it is strongly recommended that you open each .csv file to ensure the answers