Lab Exercise #7

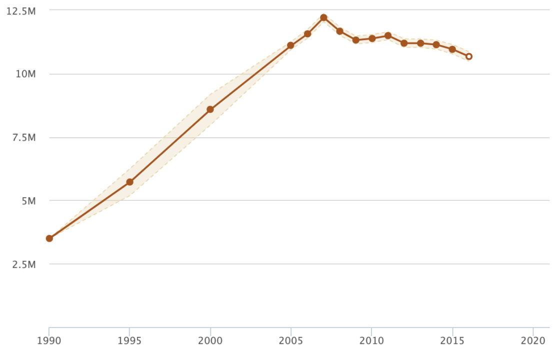
Assignment Overview

This lab exercise provides practice with lists, functions and namespaces in Python.

You will work with a partner on this exercise during your lab session. Two people should work at one computer. Occasionally switch the person who is typing. Talk to each other about what you are doing and why so that both of you understand each step.

Immigration

Immigration, especially illegal immigration, is a perennial, hot topic so in this lab we will take a look at some data. We use data from the Pew Research Center—data on illegal immigration is an estimate so its source is important.[1] Pew has published trends covering decades such as the following figure, but we will examine one data point: 2016 (the latest year of good data).



We downloaded a spreadsheet from Pew Research for 2016 data on each state as well as a summary for the US. Each state is a row in the spreadsheet. We provide the spreadsheet as a CSV (comma separated value) file named `immigration.csv` Open the file in Excel (or a similar program) to see what we are providing. Pew uses the term "unauthorized" for what most people call "illegal."

We are interested in answering three questions:

1. If we add up the individual state unauthorized immigration population (at index 1), do we get the same value as in the summative row labeled "U.S."? (Food for thought: why aren't they the same?)
2. Which states have a larger percentage unauthorized immigrant population (at index 2) than the value in the summative row labeled "U.S."?
3. Consider the column on the industry with the largest number of unauthorized immigrant workers (at index 9). Which industry is listed as the largest in the most states and how many states?

We provide a skeleton program with a `main()` and four function headers, one function to read the file plus one function for each question.

1. `def read_file(fp):` this function takes a file pointer as an argument and returns a list of lists. The list of lists where each list is a list of the contents of each data row of the file—

[1] http://www.pewhispanic.org/interactives/u-s-unauthorized-immigrants-by-state/

ignoring header rows, footnote rows, and empty rows. The list will contain 52 lists, one for each of the fifty states plus the District of Columbia and one summative row (the row labeled “U.S.”). The order of the list will be the same as the order in the file. (Hint: use `csv.reader(),` see notes below.)

2. `def get_totals(L)` **:** this function takes the list of lists returned by the `read_file()` function and returns two values. The data of interest for this function is the column at index 1: unauthorized immigrant population. Return the value in the summative row (labeled “U.S.”) and the sum of the other 51 data rows. The purpose of this function is to gather data to answer question #1 above.
3. `def get_largest_states(L)` **:** this function takes the list of lists returned by the `read_file()` function and returns a list. The data of interest for this function is the column at index 2: unauthorized immigration % of population. The returned list is a list of states whose value is greater than the summative value (the value in the row labeled “U.S.”). Since “District of Columbia” is in the file we will include it as a “state.” The returned list will be in alphabetical order (the order of the original file). The purpose of this function is to gather data to answer question #2 above.
4. `def get_industry_counts(L)` **:** this function takes the list of lists returned by the `read_file()` function and returns a list of lists. The data of interest for this function is the column at index 9: industry with largest number of unauthorized immigrant workers. The list of lists returned is a list of industries and occurrences in the column (excluding the summative data from the row labeled “U.S.”). The returned list will be sorted by occurrences (largest first) and will look like this:
   `[[industry1,count1],[industry2,count2],…]`
   The purpose of this function is to gather data to answer question #3 above. (Hint: use `key=itemgetter(1)` to sort on index 1 of the lists.)

Notes and Suggestions

1. Using the CSV package: Remember `import csv`

```
reader = csv.reader(fp)    # attaches a reader to the file fp
next(reader,None)          # skips a line, such as a header line
for line in reader:        # line is a list
```

★ **Demonstrate your completed program to your TA. On-line students should submit the completed program (named “lab07.py”) for grading via the Mirmir system.**