SI 601 Winter 2016 Homework 1 (100 points)

Due at 5:30pm on Wednesday, Jan. 13, 2015

The purpose of this assignment is to (a) to help you become more familiar with Python's basic programming constructs, data-handling methods and libraries, and (b) to get you started doing some basic manipulation of real data.

This assignment uses a summary dataset from the World Bank, containing 18 'indicators' (statistics) from over 200 countries, collected for each year in the period 2000-2010. (This is a subset of a larger dataset with 331 indicators spanning 1960-present: more information about each indicator and those not included here can be found at: http://data.worldbank.org/indicator)

The main dataset (world_bank_indicators.tsv) is stored in "Tab Separated Value" format, which is a standard exchange format for simple tabular data. TSV files store one table row per line, with the first row being a header row with the names of the columns; the other rows consist of a list of tab-separated values, corresponding to the columns of the header row.

This homework involves multiple steps: you will get credit for each step you complete successfully. You should complete the steps one by one and verify that you've completed each correctly before moving on to the next one.

Getting started:

Download the zip file si601_w16_hw1.zip and unzip it. You'll see the following files:

world_bank_indicators.tsv
world_bank_regions.tsv
country_data_desired_output.csv
region_data_desired_output.csv
sample chart.png

The first two files are in tab-separated format and they are your input files. For each file, the first row is a header row that contains the column names, which should be read, but is not actually part of the data to be used/analyzed.

The world_bank_indicators.tsv file contains country data, formatted into 20 columns per line, which are:

Column Number	*Column Name*
0	Country Name
1	Date
2	Transit: Railways, (million passenger-km)
3	Transit: Passenger cars (per 1,000 people)
4	Business: Mobile phone subscribers
5	Business: Internet users (per 100 people)
6	Health: Mortality, under-5 (per 1,000 live births)
7	Health: Health expenditure per capita (current US$)
8	Health: Health expenditure, total (% GDP)
9	Population: Total (count)
10	Population: Urban (count)

11	Population:: Birth rate, crude (per 1,000)
12	Health: Life expectancy at birth, female (years)
13	Health: Life expectancy at birth, male (years)
14	Health: Life expectancy at birth, total (years)
15	Population: Ages 0-14 (% of total)
16	Population: Ages 15-64 (% of total)
17	Population: Ages 65+ (% of total)
18	Finance: GDP (current US$)
19	Finance: GDP per capita (current US$)

The world_bank_regions.tsv file has 3 columns per line:

Column Number	*Column Name*
0	Region
1	Subregion
2	Country Name

Your code will load these two files and do some manipulation on them to get the desired outputs.

You should review all steps before starting to write code, to think about what kinds of data structures you're going to need that can be used for both the earlier and later steps.

Step 1 (55 points) Finding out correlations between urban population ratio and life expectancy

In this step, you need to write Python code that reads the input file world_bank_indicators.tsv, and generates an output file that looks the same as country_data_desired_output.csv

To do this, first read the file world_bank_indicators.tsv line by line, and parse each line into a Python data structure of your choice, where the individual values for each column can be used by later code.

HINT: For platform independence, using 'rU' to turn on Python universal newline support in the open function. For example:

```
input_file = open('inputfile.txt', 'rU')
```

Read about file methods at http://docs.python.org/2/library/stdtypes.html#file-objects

We are only interested in the following columns in world_bank_indicators.tsv:

- Country Name
- Total population
- Urban population
- Health: Life expectancy at birth, total (years)

(20 points) Store the total population, urban population, and life expectancy data from all years for each country in a suitable data structure.

(10 points) For each country, you should calculate the average urban population ratio. To do this, first sum up the total populations in all years, and sum up the urban populations in all years, then calculate the average urban population ratio as (sum of urban population/sum of total population). Note that the urban population data is missing for some countries, so for these countries, urban population ratio is not calculated.

(10 points) For each country, you should also calculate the average life expectancy across all years.

(10 points) Once you have all data you need, output it to a file in "comma-separated values" (CSV) format. Note that the provided country_data_desired_output.csv is sorted by the "country name" column in alphabetical order. Your step 1 output file should be sorted in the same way.

CSV format is a standard data exchange format for simple tabular data, typically used by spreadsheet programs like Excel. CSV format uses one row per line, with the first row being a header row with the names of the columns; the other rows consist of a list of comma-separated values (hence the name), corresponding to the columns of the header row.

You should use the existing csv package for this task as described in lecture slides. Also see http://docs.python.org/2/library/csv.html

Your step 1 CSV output file should be named si601_w16_hw1_step1_ ***youruniquename***.csv

(5 points) Load you step 1 output CSV file into either Excel or Google Docs Spreadsheet, and generate a scatter plot of the relationship between urban population ratio and life expectancy. Your plot should look like the sample given in sample chart.png

Save your plot file as si601_w16_hw1_chart_***youruniquename***.pdf

Step 2 (45 points) Adding region data

(15 points) Add code to read the region file (world_bank_regions.tsv) into an appropriate data structure to facilitate lookup operations.

(20 points) Use the region data structure you created to look up, for each country, a country's corresponding region, and then calculate the average urban population ratio and the average life expectancy for each region based on the country data you calculated in step 1. The average urban population ratio for a region should be calculated as the sum of sum of urban population in all years of all countries in that region/ the sum of sum of total population in all years of all countries in that region. The average life expectancy for a region is calculated as a simple average of the average life expectancy of all countries in that region (Technically, this average should be calculated as a weighted average with population in each country as weight. But in the homework, we just cheat a bit by calculating the simple average.)

(10 points) Save your step 2 results in a CSV file named si601_w16_hw1_step2_ ***youruniquename***.csv. It should contain the same data as the provided region_data _desired_output.csv file.

Submission instructions

The name of your python file should be si601_w16_hw1_***youruniquename***.py. When you run it, it should write out two output files named si601_w16_hw1_step1_ ***youruniquename.***csv and si601_w16_hw1_step2_ ***youruniquename.***csv

What to submit: A zip file, **named si601_w16_hw1_youruniquename.zip**, that contains your python source code file, the two output CSV files, and your plot file.