DS-GA-1007 Programming for Data Science

Assignment 9

Submission Instructions

You are free to use whichever development environment you wish to create and submit the assignment answers.

1. Create a directory using your *Net ID* as the directory name.
2. Place your Python code in this directory.
3. There should be at least one file called `assignment9.py` and one file called `results.txt` at the top level of this directory.
4. Fork the `assignment9` repository from the ds-gs-1007 user on GitHub.
5. Clone this repository onto your local system.
6. Place your new directory (the Net ID) into the working directory of this repository either using PyDev or manually.
7. Add your directory to the staging area, commit, and push to the remote repository.
8. Submit a pull request to the repository owner (ds-ga-1007).

Questions

1. Write a program to load the `countries.csv` file (located in the `assignment9` repository) into a pandas `DataFrame` and name this data set `countries.`
2. Load the `indicator gapminder gdp_per_capita_ppp.xlsx` data set (located in the `assignment9` repository) into a `DataFrame` called `income.` Hint: You can convert the data set to CSV, or consider using the pandas function `pandas.read_excel()` to read in the .xlsx file directly
3. The program should transform the data set to have years as the rows and countries as the columns, then show the head of this data set when it is loaded.
4. Provide a function to graphically display the distribution of income per person across all countries in the world for the given year. Choose the best kind of plot to display this information.

5. Provide a function called `merge_by_year(year)` to merge the countries and income data sets for any given year. The result should be a `DataFrame` with three columns titled `Country`, `Region`, and `Income`.

6. Provide a class that uses exploratory data analysis tools (histograms and boxplots) to graphically explore the distribution of the income per person by region data set from question 5 for a given year. Save these graphs to individual files.

7. The program should ask the user to enter a year, then display the graph using the function in question 4. Continue asking the user for a year and displaying the graph until the string 'finish' is entered.

8. The program should then use the class from question 6 to generate graphs for the years 2007-2012.

9. Visually inspect the graphs generated in question 8 and write a short description of the changes you observe over the period. Place your description in a file called `results.txt` and add this to the repository.

Grading

This assignment will be graded according to the criteria listed in the following 5 sections.

Correctness

- The program produces the correct output when run using the command

```
python assignment9.py
```

Exception/Error Handling

- All possible exceptions are handled correctly
- The code catches specific exceptions (e.g. `KeyboardInterrupt`) rather than using a catchall statement
- Invalid user input is handled correctly (when input is required by the assignment)
- User defined exception(s) are employed for indicating error conditions rather than raising generic exceptions

Comments

- The main program contains a comment that lists the authors, and describes the overall program behavior
- Comments are used to explain intent and/or warn of consequences where appropriate
- Doc strings are used to describe each function
- Comments are used to document public methods in the class
- There is no commented-out code

Structure

- At least one class is used
- The class is in separate module from main program
- Modules are used to structure the program
- The program is correctly structured as a Python package
- The code is easily understandable (i.e. divided into logical sections, well structured, etc.)
- The code uses meaningful names for variables, functions, and methods, and avoids “Hungarian” notation
- Function/method bodies are kept small

Testing

- Unit tests are provided with the solution code
- The unit tests pass correctly