

Computer Project #08

(Modified 3/20 to clarify prompting in the plot function.)

Assignment Overview

This assignment focuses on the design, implementation and testing of Python programs to process data files and extract meaningful information from them.

It is worth 50 points (5% of course grade) and must be completed no later than 11:59 PM on Monday, April 3, 2017.

Assignment Deliverable

The deliverable for this assignment is the following file:

proj08.py – the source code for your Python program

Be sure to use the specified file name and to submit it for grading via the **handin system** before the project deadline.

Assignment Background

The United States, home to approximately 320 million citizens, is a large country made of 50 states and the nation's capital, the District of Columbia (also called Washington D.C.). In this project, you will explore some of the economic statistics of each state and create some visualization of the data. No data on U.S. territories are included in the file.

Data will be read from a CSV (comma-separated values) file called *State_Data.csv*. Each row in the file contains the following information on each state in the United States from 2010. Values are separated by commas.:

- 1st Value: State
- 2nd Value: Region (defined by the Bureau of Economic Analysis)
- 3rd Value: Population (in millions)
 - o The total number of people living in the state.
- 4th Value: GDP (in billions)
 - o Measure of the state's economic activity, a higher GDP means higher monetary value for goods and services within the state's boarder.
- 5th Value: Personal Income (in billions)
 - o All incomes received by individuals and households.
- 6th Value: Subsidies (in millions)
 - o Money granted by the state's government to help an industry or business.
- 7th Value: Compensation of Employees (in billions)
 - o Pre-taxed wages paid by employers to employees.

- 8th Value: Taxes on Production and Imports (in billions)
 - o Taxes chargeable to business expenses of producing and importing

Note that Python uses Zero-based indexing, meaning that when data is put in a list the first value (State) will be found by taking the 0th index of the list.

Also recognize that some values are in millions while others are in billions, so when using operations on two values (which you will) make sure to adjust them accordingly.

The Bureau of Economic Analysis contains the following regions. This information is already in the provided data file.

- **Far_West:** Alaska, California, Hawaii, Nevada, Oregon, Washington
- **Great_Lakes:** Illinois, Indiana, Michigan, Ohio, Wisconsin
- **Mideast:** Delaware, District of Columbia (Washington D.C.), Maryland, New_Jersey, New_York, Pennsylvania
- **New_England:** Connecticut, Maine, Massachusetts, New_Hampshire, Rhode_Island, Vermont
- **Plains:** Iowa, Kansas, Minnesota, Missouri, Nebraska, North_Dakota, South_Dakota
- **Rocky_Mountain:** Colorado, Idaho, Montana, Utah, Wyoming
- **Southeast:** Alabama, Arkansas, Florida, Georgia, Kentucky, Louisiana, Mississippi, North_Carolina, South_Carolina, Tennessee, Virginia, West_Virginia
- **Southwest:** Arizona, New_Mexico, Oklahoma, Texas

This data is provided by the U.S. Bureau of Economic Analysis in the following link, [GDP and Personal Income of the U.S. \(annual\)](#)

Assignment Specifications

Provide the following functions – you get to choose the name and the parameters. Also, you will likely want more functions. I have a total of ten functions.

1. **Function that opens the file and returns a file object**
 - Your program should prompt the user to enter a valid file name. If the file is not found print an error message and keep prompting until a valid file name is entered. This should be done using try and except. You likely already have this function from a previous project.
2. **Function that reads data from a file and returns it in a data structure of your choice (e.g. a list or dictionary).**
 - Prompt the user to select a region to gather data from. You can then read the contents of the file into your data structure. If you use a dictionary, you likely want the state name as the key, and a list of data about the state as the value. If you use a list, you

likely want the state name as the first item in the list followed by the remaining data. You will only need states from the selected region in your data structure.

- Append the following information to each state value (note that “per capita” means “per person” so you must divide by the population):
 - GDP per capita
 - Per capita personal income
 That is, you have two new pieces of information in addition to those values read from the file.
- Your code should check for invalid region names and continue to prompt until the user provides a valid region name.

3. Function that displays state information for the region

- Print the states with the highest and lowest GDP per capita and Per capita income respectively, along with the values of each one. Include the dollar sign and commas when displaying these max and min values (see note on formatting below).
- Then print out all of the states in the region and their data. Format this information nicely in columns with column headers. Print by state in alphabetical order. Include commas in values, but in this table you do not need dollar signs.

4. Function that plots data on the selected region:

- The user will be prompted to provide an x and a y list value to graph for each state. The x and y value lists are selected from the following, **exactly** as they are listed below (where GDPp is GDP per capita and Plp is personal income per capita):

Pop, GDP, Pl, Sub, CE, TPI, GDPp, Pip

prompt for both values separated by spaces on the same line with x first and y second (see the test cases below).

- You will then create a graph for each state in the region based on the chosen x and y. The x and y values can be the same. Use `pylab.scatter(x, y)` as noted below.
- Your code should print an error message if the x and y values are not in the list above. and keep prompting until both the x and y values are valid.
- The following can be used to label your scatter plot with state names:

```
for i,txt in enumerate(State):
    pylab.annotate(txt, (x[i],y[i]))
```

Where `State` is a list of state names in the region and `x` and `y` are lists of the corresponding x and y values for each state.

5. Drawing the regression line:

- We provide the function `plot_regression` to plot a regression line. What follows is an explanation of the function—you do not need to change the function, simply use it.
- The parameters `x` and `y` are lists of x and y coordinates (so the lists must be the same length). This function should be called after `pylab.scatter` and before `pylab.show()`. This function draws a best fit (regression) line to the (x,y) data. This line will show the relationship between the x and y values that are graphed.

- Linear regression uses the equation $Y = mX + b$ so the function calls `pylab.polyfit` to calculate values for `m` and `b`. With those two values in hand you can plot `(x,y)` where `x` is `xarr` and `y` is `m*xarr+b` as `pylab.plot(xarr,m*xarr + b, '-')`

6. Main

Call functions from here.

Prompt the user whether they want to create a plot: “yes” means to create the plot; all other input will skip the plot.

Assignment Notes

- We provide a file `proj08.py` that has the function `plot_regression` as well as some optional constants that you are free to use or modify. There are also some suggested lines of code for plotting—you need to provide arguments.
- To plot you will need to import `pylab`. Appendix D of the text describes plotting.
- Use `pylab.scatter(x, y)` to graph a scatter plot of the data, where `x` and `y` are both lists of data (of the same length). (`pylab` uses the functionality of the graphing package that you imported)
- Your code should be able to recognize any combination of capital and lowercase letters.
 - ex: `grEAT_Lakes`, `SOUTHWEST`, `souTHEast`, `aLI`, etc.
 - If the user enters `all`, then information on every state should be printed
- Every value printed by your code should be rounded to the hundredth decimal place (.00).
- State and region names that are composed of two or more words will have underscores between each word
 - `Rocky_Mountains`, `North_Carolina`, `New_Hampshire`
- The following methods may prove useful in completing this project:
 - `lower()`, `upper()`, `split()`, `strip()`, `title()`, `readline()`
- Use of expression “in list” should prove helpful as well:
 - `mylist = [1,2,3]`
 - if 2 in `mylist`:
 `print("2 in my list")`
 - Because the integer, 2, is in the list, `mylist`, the statement will be printed
- In order to format a number into a dollar amount (\$ and ,) use the following:
 - `${:,2f}`
 - `f`: denotes it will be a float value
 - `.2`: exclusive to float values, determines the number of decimal places

- ,: is for putting commas inside the number
- \$: place a money sign while printing
- {:}: formatting syntax
-
- We provide some constants – you may use them or create your own.
- List comprehension may be useful, especially for gathering data for plotting.
 - http://www.secnexix.de/olli/Python/list_comprehensions.hawk
- Items 1-9 of the Coding Standard will be enforced for this project.
<https://www.cse.msu.edu/~cse231/General/coding.standard.html>

Suggested Procedure

- Solve the problem using pencil and paper first. You cannot write a program until you have figured out how to solve the problem. This first step may be done collaboratively with another student. However, once the discussion turns to Python specifics and the subsequent writing of Python statements, you must work on your own.
- First focus on error checking for opening a file using try and except. Once you have this working, then you can set it automatically open a file to ease testing other areas, but don't forget to change it back in the final stages of the project.
- Read through the file, line by line, changing every line to a list, separated by commas. Add states that are in the selected region to your dictionary. Find extreme values for GDP per capita and Per capita income.
- Create a list with the following values as strings: Pop, GDP, PI, Sub, CE, TPI, GDPp, Plp. Check if the user's choices for x and y are contained in these values.
- Use the corresponding lists of the selected strings for graphing.
- Use the main() function if your TA requires it. Don't forget the end comments.
- Use the **handin system** to turn in the first version of your solution. Cycle through the steps to incrementally develop your program:
 - Edit your program to add new capabilities.
 - Run the program and fix and errors.
 - Use the **handin system** to submit the current version of your solution.
- Be sure to log out when you leave the room, if you're working in a public lab.

Tests and Output

Test 1

Input a file: State_Data.csv

Specify a region from this list -- far_west,great_lakes,mideast,new_england,plains,rocky_mountain,southeast,southwest,all: southeast

Data for the Southeast region:

Virginia has the highest GDP per capita at \$47,036.17

Mississippi has the lowest GDP per capita at \$28,749.45

Virginia has the highest income per capita at \$44,853.78

Mississippi has the lowest income per capita at \$30,847.16

Data for all states in the Southeast region:

State	Population(m)	GDP(b)	Income(b)	Subsidies(m)	Compensation(b)	Taxes(b)	GDP per capita	Income per capita
Alabama	4.78	153.84	162.23	485.00	98.28	11.02	32,151.81	33,904.93
Arkansas	2.92	92.08	93.68	405.00	56.79	7.51	31,504.04	32,052.52
Florida	18.85	650.29	725.44	2,522.00	399.29	69.98	34,505.47	38,492.85
Georgia	9.71	358.84	333.63	1,215.00	227.03	25.88	36,937.84	34,343.08
Kentucky	4.35	141.98	143.21	484.00	92.35	12.33	32,663.86	32,947.06
Louisiana	4.54	200.94	169.12	759.00	105.14	14.31	44,219.98	37,216.76
Mississippi	2.97	85.36	91.59	387.00	52.85	7.37	28,749.45	30,847.16
North_Carolina	9.56	380.69	338.99	1,300.00	218.62	30.47	39,825.30	35,462.65
South_Carolina	4.64	143.41	151.54	472.00	93.03	11.88	30,935.33	32,688.38
Tennessee	6.36	227.36	225.22	730.00	138.68	19.04	35,766.99	35,431.07
Virginia	8.03	377.47	359.96	1,113.00	248.95	28.02	47,036.17	44,853.78
West_Virginia	1.85	53.58	58.95	121.00	35.19	5.15	28,899.68	31,796.17

Do you want to create a plot? no

Test 2

Input a file: State_Data.csv

Specify a region from this list -- far_west,great_lakes,mideast,new_england,plains,rocky_mountain,southeast,southwest,all: all

Data for the All region:

D.O.C has the highest GDP per capita at \$148,710.74

Mississippi has the lowest GDP per capita at \$28,749.45

D.O.C has the highest income per capita at \$69,767.44

Mississippi has the lowest income per capita at \$30,847.16

Data for all states in the All region:

State	Population(m)	GDP(b)	Income(b)	Subsidies(m)	Compensation(b)	Taxes(b)	GDP per capita	Income per capita
Alabama	4.78	153.84	162.23	485.00	98.28	11.02	32,151.81	33,904.93
Alaska	0.71	43.47	32.65	68.00	22.96	6.33	60,873.83	45,721.33
Arizona	6.41	221.02	217.76	763.00	135.60	16.94	34,476.20	33,967.48
Arkansas	2.92	92.08	93.68	405.00	56.79	7.51	31,504.04	32,052.52
California	37.33	1,672.50	1,579.10	9,235.00	1,009.60	133.61	44,797.83	42,296.11
Colorado	5.05	230.98	210.61	868.00	141.00	16.79	45,752.20	41,716.89
Connecticut	3.58	197.61	197.84	717.00	121.06	15.10	55,250.80	55,314.91
D.O.C	0.60	89.97	42.21	833.00	75.89	3.67	148,710.74	69,767.44
Delaware	0.90	55.50	36.96	249.00	25.36	3.07	61,680.37	41,073.13
Florida	18.85	650.29	725.44	2,522.00	399.29	69.98	34,505.47	38,492.85
Georgia	9.71	358.84	333.63	1,215.00	227.03	25.88	36,937.84	34,343.08
Hawaii	1.36	59.67	56.83	276.00	37.88	5.45	43,736.71	41,653.01
Idaho	1.57	50.73	50.39	337.00	29.36	3.31	32,295.65	32,076.14
Illinois	12.84	571.23	540.22	2,746.00	362.94	50.56	44,486.59	42,071.83
Indiana	6.49	241.93	223.16	898.00	144.02	17.48	37,277.92	34,385.45
Iowa	3.05	124.01	119.08	1,039.00	72.04	9.10	40,655.02	39,038.68
Kansas	2.86	113.32	110.88	777.00	71.43	9.03	39,639.01	38,787.15
Kentucky	4.35	141.98	143.21	484.00	92.35	12.33	32,663.86	32,947.06
Louisiana	4.54	200.94	169.12	759.00	105.14	14.31	44,219.98	37,216.76
Maine	1.33	45.56	49.36	175.00	29.50	4.54	34,317.57	37,180.02
Maryland	5.79	264.32	289.65	1,128.00	175.61	18.69	45,666.90	50,043.73
Massachusetts	6.56	340.16	337.93	1,564.00	229.30	21.00	51,827.59	51,488.06
Michigan	9.88	329.81	346.82	1,159.00	215.58	29.91	33,389.35	35,111.23
Minnesota	5.31	240.42	226.32	1,327.00	154.01	18.85	45,270.87	42,615.83
Mississippi	2.97	85.36	91.59	387.00	52.85	7.37	28,749.45	30,847.16
Missouri	6.00	216.68	219.48	911.00	142.84	14.96	36,136.82	36,604.46
Montana	0.99	31.92	34.27	238.00	20.06	2.47	32,219.64	34,590.49
Nebraska	1.83	80.64	73.07	313.00	47.02	5.65	44,072.80	39,935.02
Nevada	2.70	109.61	99.21	313.00	62.41	10.27	40,539.24	36,691.40
New_Hampshire	1.32	55.24	59.19	162.00	35.26	4.49	41,950.18	44,953.45
New_Jersey	8.80	431.41	449.06	1,733.00	267.58	42.41	49,004.93	51,009.83
New_Mexico	2.06	70.79	68.49	302.00	42.64	5.55	34,284.19	33,169.85
New_York	19.40	1,013.30	960.83	6,156.00	638.67	90.88	52,234.11	49,529.18
North_Carolina	9.56	380.69	338.99	1,300.00	218.62	30.47	39,825.30	35,462.65
North_Dakota	0.67	31.62	29.15	603.00	18.64	2.37	46,893.07	43,235.65
Ohio	11.54	413.99	418.54	1,608.00	272.72	33.55	35,879.64	36,273.55
Oklahoma	3.76	132.92	135.06	448.00	80.06	9.23	35,355.77	35,925.68
Oregon	3.84	174.17	137.67	703.00	88.90	8.03	45,378.04	35,868.82
Pennsylvania	12.71	493.53	529.81	2,125.00	323.80	39.32	38,826.08	41,680.06
Rhode_Island	1.05	43.15	45.27	200.00	27.11	3.97	40,988.79	42,997.34
South_Carolina	4.64	143.41	151.54	472.00	93.03	11.88	30,935.33	32,688.38
South_Dakota	0.82	34.37	33.14	617.00	18.24	2.73	42,109.78	40,597.53
Tennessee	6.36	227.36	225.22	730.00	138.68	19.04	35,766.99	35,431.07
Texas	25.24	1,116.27	961.83	2,887.00	621.10	93.06	44,221.50	38,103.22
Utah	2.78	105.20	90.11	326.00	62.60	6.56	37,908.54	32,471.87
Vermont	0.63	23.34	25.12	105.00	15.12	2.52	37,284.35	40,120.93
Virginia	8.03	377.47	359.96	1,113.00	248.95	28.02	47,036.17	44,853.78
Washington	6.74	307.69	286.74	1,526.00	187.42	28.48	45,626.96	42,520.88
West_Virginia	1.85	53.58	58.95	121.00	35.19	5.15	28,899.68	31,796.17
Wisconsin	5.69	219.08	220.50	973.00	141.45	18.43	38,505.34	38,755.33
Wyoming	0.56	32.00	25.43	82.00	15.68	3.66	56,697.38	45,063.08

Do you want to create a plot? No

Test 3

Input a file: State_Data.csv

Specify a region from this list -- far_west,great_lakes,mideast,new_england,plains,rocky_mountain,southeast,southwest,all: plains

Data for the Plains region:

North_Dakota has the highest GDP per capita at \$46,893.07

Missouri has the lowest GDP per capita at \$36,136.82

North_Dakota has the highest income per capita at \$43,235.65

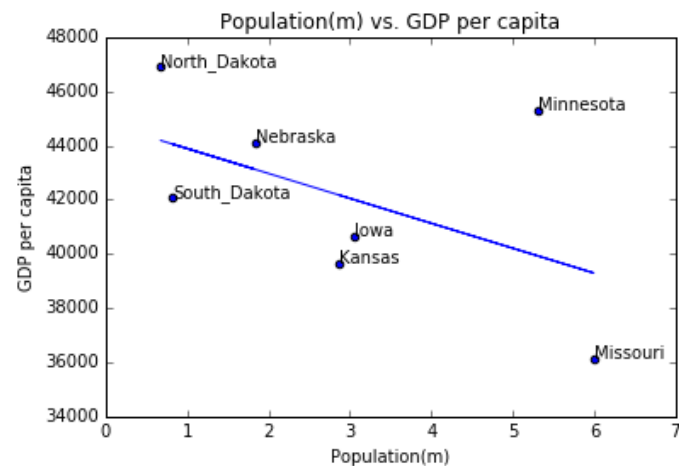
Missouri has the lowest income per capita at \$36,604.46

Data for all states in the Plains region:

State	Population(m)	GDP(b)	Income(b)	Subsidies(m)	Compensation(b)	Taxes(b)	GDP per capita	Income per capita
Iowa	3.05	124.01	119.08	1,039.00	72.04	9.10	40,655.02	39,038.68
Kansas	2.86	113.32	110.88	777.00	71.43	9.03	39,639.01	38,787.15
Minnesota	5.31	240.42	226.32	1,327.00	154.01	18.85	45,270.87	42,615.83
Missouri	6.00	216.68	219.48	911.00	142.84	14.96	36,136.82	36,604.46
Nebraska	1.83	80.64	73.07	313.00	47.02	5.65	44,072.80	39,935.02
North_Dakota	0.67	31.62	29.15	603.00	18.64	2.37	46,893.07	43,235.65
South_Dakota	0.82	34.37	33.14	617.00	18.24	2.73	42,109.78	40,597.53

Do you want to create a plot? yes

Specify x and y values, space separated from Pop, GDP, PI, Sub, CE, TPI, GDPp, PIp: Pop GDPp



Test 4

Input a file: abcd
Error opening file. Please try again.

Input a file: State_Data.csv

Specify a region from this list -- far_west,great_lakes,mideast,new_england,plains,rocky_mountain,southeast,southwest,all: xxxyyy
Error in region name. Please try again

Specify a region from this list -- far_west,great_lakes,mideast,new_england,plains,rocky_mountain,southeast,southwest,all: Great_lakes

Data for the Great_Lakes region:

Illinois has the highest GDP per capita at \$44,486.59
Michigan has the lowest GDP per capita at \$33,389.35

Illinois has the highest income per capita at \$42,071.83
Indiana has the lowest income per capita at \$34,385.45

Data for all states in the Great_Lakes region:

State	Population(m)	GDP(b)	Income(b)	Subsidies(m)	Compensation(b)	Taxes(b)	GDP per capita	Income per capita
Illinois	12.84	571.23	540.22	2,746.00	362.94	50.56	44,486.59	42,071.83
Indiana	6.49	241.93	223.16	898.00	144.02	17.48	37,277.92	34,385.45
Michigan	9.88	329.81	346.82	1,159.00	215.58	29.91	33,389.35	35,111.23
Ohio	11.54	413.99	418.54	1,608.00	272.72	33.55	35,879.64	36,273.55
Wisconsin	5.69	219.08	220.50	973.00	141.45	18.43	38,505.34	38,755.33

Do you want to create a plot? no

Grading RubricComputer Project #08
Scoring Summary

General Requirements

__0__ (5 pts) Coding Standard 1-9
(descriptive comments, function header,
etc...)

Implementation:

__0__ (10 pts) Pass test1: Display one region

__0__ (5 pts) Pass test2: Display all

__0__ (15 pts) Pass test3: Plotting

__0__ (10 pts) Pass test4: Error Checks

__0__ (5 pts) Further Testing
-- Error Checks not tested in test4
-- Displays other regions

TA Comments:

Optional Testing

This test suite has a test program for each of the three functions and two for the entire program. They are handled slightly differently.

1. Test the entire program using `run_file.py`.

You need to have the files `test1.txt` to `test3.txt` from the project directory.

IMPORTANT: the plot doesn't draw in `test3`, but you can have the plot output to a file named `plot.png` by replacing `pylab.show()` with `pylab.savefig("plot.png")`.

Make sure that you have the following lines at the top of your program (only for testing):

```
import sys
def input( prompt=None ):
    if prompt != None:
        print( prompt, end="" )
    aaa_str = sys.stdin.readline()
    aaa_str = aaa_str.rstrip( "\n" )
    print( aaa_str )
    return aaa_str
```

Educational Research

When you have completed the project insert the 5-line comment specified below.

For each of the following statements, please respond with how much they apply to your experience completing the programming project, on the following scale:

1 = Strongly disagree / Not true of me at all

2

3

4 = Neither agree nor disagree / Somewhat true of me

5

6

7 = Strongly agree / Extremely true of me

****Please note that your responses to these questions will not affect your project grade, so please answer as honestly as possible.****

Q1: Upon completing the project, I felt proud/accomplished

Q2: While working on the project, I often felt frustrated/annoyed

Q3: While working on the project, I felt inadequate/stupid

Q4: Considering the difficulty of this course, the teacher, and my skills, I think I will do well in this course.

Q5: I ran the optional test cases (choose 7=Yes, 1=No)

Please insert your answers into the bottom of your project program as a comment, formatted exactly as follows (so we can write a program to extract them).

Questions

Q1: 5

Q2: 3

Q3: 4

Q4: 6

Q5: 7