Assignment 4

Goals

The goal of this assignment is to work with files, iterators, strings, and string formatting in Python.

Instructions

You will be doing your work in a Jupyter notebook for this assignment. You may choose to work on this assignment on a hosted environment (e.g. [tiger](https://tiger.cs.niu.edu/jupyter/)) or on your own local installation of Jupyter and Python. You should use Python 3.8 or higher for your work. To use tiger, use the credentials you received. If you work remotely, make sure to download the .ipynb file to turn in. If you choose to work locally, [Anaconda](https://www.anaconda.com/download) is the easiest way to install and manage Python. If you work locally, you may launch Jupyter Lab either from the Navigator application or via the command-line as jupyter-lab.

In this assignment, we will be working with data from U.S. Department of Agriculture’s [Economic Research Service](https://www.ers.usda.gov/) about [weekly food sales](https://www.ers.usda.gov/data-products/weekly-retail-food-sales/) in 39 states (others are not available due to how the data is collected). I have downloaded and pre-processed the data [here](http://faculty.cs.niu.edu/~dakoop/cs503-2021fa/a4/food-prices-weekly.txt), converting it to a fixed-width file format. There are five fields, whose locations in each line are as follows:

* Date: (0-10) the end of the week of the data collection in yyyy-mm-dd format
* State: (12-26) the name of the state (**contains spaces**)
* Category: (28-55) the food category (**contains spaces**)
* Dollars: (57-66) the total value of sales
* LastYear: (68-77) the total value of sales **last year**

A [template notebook](http://faculty.cs.niu.edu/~dakoop/cs503-2021fa/a4/a4.ipynb) is provided with one cell to download the data. You will read this data, calculate monthly averages, and write a new output file in a similar format.

Due Date

The assignment is due at 11:59pm on Thursday, October 7.

Submission

You should submit the completed notebook file required for this assignment on [Blackboard](https://webcourses.niu.edu/). The filename of the notebook should be a4.ipynb. You **should not** turn in the food-prices-monthly.txt file as your notebook should contain the code to create it.

Details

Please make sure to follow instructions to receive full credit. Use a markdown cell to **Label** each part of the assignment with the number of the section you are completing. You may put the code for each part into one or more cells. In Parts 3 and 4, CSCI 503 students must compute and output **two** averages and a percent change while CSCI 490 students need only compute and output **one** average. Do not use external libraries for reading, parsing, or writing the data files.

0. Name & Z-ID (5 pts)

The first cell of your notebook should be a markdown cell with a line for your name and a line for your Z-ID. If you wish to add other information (the assignment name, a description of the assignment), you may do so after these two lines.

1. Read Data (15 pts)

Either download the [file](http://faculty.cs.niu.edu/~dakoop/cs503-2021fa/a4/food-prices-weekly.txt) and upload it to Jupyter, or use the provided [template notebook](http://faculty.cs.niu.edu/~dakoop/cs503-2021fa/a4/a4.ipynb) which contains a cell to download the file. Then, use an **iterator** to read the file food-prices-weekly.txt into a list of dictionaries named data (a format similar to what we used in [Assignment 3](http://faculty.cs.niu.edu/~dakoop/cs503-2021fa/assignment3.html)). Remember that a file object will provide an iterator if you pass it to the iter function. Read the header first (the first line), and then the rest of the file. The header will serve as the keys for each dictionary while the other lines are values. To split each line into its column values, use **slicing** and remove leading and trailing whitespace. You do not need to convert values (e.g. to integers) in this step (see Step 3). When you finish, your data should look like:

[

...,

{'Date': '2021-08-22',

'State': 'Wyoming',

'Category': 'Vegetables',

'Dollars': '621511',

'LastYear': '620174'}

]

Hints

* You **cannot** use split here because some states and categories have spaces in them. Use slicing.
* Remember that the iter() functions obtains an iterator for an object, and next() retrieves an item. For example, next(iter([1,2,3])) is 1.
* If you pass an iterator to a for loop, it will loop through the **remaining** items.
* strip will be useful to remove whitespace.
* Use zip to create pairs of tuples that can be used to create a dictionary. dict(zip(['a','b'], [1,2])) produces the dictionary {'a': 1, 'b': 2}.

2. Add Month Column (5 pts)

For each data item, create a new key-value pair, Month, from the Date column. Given a date in *yyyy-mm-dd* format, the Month value should be *yyyy-mm*. Add these pairs to the existing dictionaries in the list.

Hints

* You should be able to use slicing to extract the parts of the string for each data item.

3. Compute Monthly Values (15/25 pts)

CSCI 490 students should complete (a) and CSCI 503 students should complete (b)

a. [CSCI 490 Only] (15 pts)

Create a new list of dictionaries named monthly\_data that has **one** entry per month that and contains four key-value pairs (Month, State, Category, DollarsAvg). DollarsAvg should be the **average** of the Dollars values for the given month, state, and category. You will need to convert the individual values to integers before computing averages. monthly\_data should look like:

[

...,

{'Month': '2021-08',

'State': 'Wyoming',

'Category': 'Vegetables',

'Dollars': 619689.75}

]

Hints

* To compute an average, you need both the sum and the number of items being averaged. There are several ways to compute this ranging from storing these values as a tuple to storing the individual values in a list.

b. [CSCI 503 Only] (25 pts)

Create a new list of dictionaries named monthly\_data that has **one** entry per month that and contains four key-value pairs (Month, State, Category, DollarsAvg, LastYearAvg, PctChange). DollarsAvg and LastYearAvg should be the **average** of the Dollars and LastYear values, respectively, for the given month, state, and category. Then, PctChange is computed by100⋅DollarsAvg−LastYearAvgDollarsAvgYou will need to convert the individual values to integers before computing averages. Do this efficiently. You shouldn’t loop through the data more than twice (once to read, once to write). At the end, monthly\_data should look like:

[

...,

{'Month': '2021-08',

'State': 'Wyoming',

'Category': 'Vegetables',

'DollarsAvg': 619689.75,

'LastYearAvg': 640338.5,

'PctChange': -3.332110947453948}

]

Hints

* To compute an average, you need both the sum and the number of items being averaged. There are several ways to compute this ranging from storing these values as a tuple to storing the individual values in a list.
* To be efficient, compute DollarsAvg and LastYearAvg at the same time, and create the dictionary for each monthly\_data entry by accessing all data related to the *(Month, State, Category)* tuple at once.

4. Write Output (20/25 pts)

Write the new monthly\_data list of dictionaries to a file named food-prices-monthly.txt in a similar format to the original file. This means writing the header values as the first line. Second, write the floating point values for DollarsAvg (and for CSCI 503, LastYearAvg) with only **1** digit after the decimal point. CSCI 503 students should also write PctChange with **2** digits after the decimal point. All the numbers should be right-aligned with at least two spaces between each column. Write a positive sign (+) for positive values of PctChange and a negative sign for negative values. The columns should be:

* Month: (0-7) the month of the data collection in yyyy-mm format
* State: (9-23) the name of the state (**contains spaces**)
* Category: (25-52) the food category (**contains spaces**)
* DollarsAvg: (54-65) the total value of sales
* [CSCI 503] LastYearAvg: (67-78) the total value of sales **last year**
* [CSCI 503] PctChange: (80-89) the percent change from the last year

The output file should look like (CSCI 490 students will not have the last two columns):

Month State Category DollarsAvg LastYearAvg PctChange

2019-10 Alabama Alcohol 22064525.5 21262833.8 +3.63

2019-11 Alabama Alcohol 21828656.0 20955901.0 +4.00

2019-12 Alabama Alcohol 22924922.8 19847999.6 +13.42

... ... ... ... ... ...

2021-06 Wyoming Vegetables 619151.5 661495.0 -6.84

2021-07 Wyoming Vegetables 643430.0 693040.8 -7.71

2021-08 Wyoming Vegetables 619689.8 640338.5 -3.33

Hints

* Use print function calls to write a few data items to stdout first, then when you are satisfied the format is correct, write all of the data items to a file.
* Remember the file keyword argument for the print function.
* Use a with statement to make sure all data is written to the file (or make sure to call close).
* Remember to pass the w flag to open to be able to write to a file.
* Consult the [Format Specification Mini-Language](https://docs.python.org/3/library/string.html#formatspec) for the various flags
* You can see the contents of the file you wrote in the notebook using the !cat command: !cat food-prices-monthly.txt