CSCI 1100: Humanities Computer Science 1

Homework 5: Trendlines & Algorithms

Reading: “Weapons of Math Destruction,” by Cathy O’Neil, Introduction and Chapter 1

This homework is worth 100 points towards your overall homework grade and is due Thursday, March 21st at 11:59:59 PM. There are two parts of the homework that are worth 45 points per part, and 10 points for the README.

Each part will be coded and submitted in separated files. These are the files that will be turned in for this homework:

```
hw5part1.py
hw5part2.py
README.txt
```

Respectively. All parts should be submitted by the deadline or your program will be considered late.

As a final note, in each part, you have to use loops. We will leave the choice of the loop type up to you. Please feel free to use while loops or for loops depending on the task and your own personal preference for completing the task efficiently.

PART ONE: Mapping Trends

While working on the following parts of the homework, think about the implications of the visualization and broad statements about safety. What assumptions are made? Do you have and problems with the way the homework questions are asked? We will talk about this further in the discussion portion of our lab.

As part of an open government initiative, New York State releases a large number of health-related statistics for researchers and developers to use. Among these are death statistics by region of the state at https://health.data.ny.gov. For this assignment, we have simplified the statistics to provide just the total number of deaths per 100,000 people. Our goal is to come up with a simple visualization comparing death trends between two different counties over the last few years.

Write a program to read in the death statistics for two different areas of NYS for the 11 years from 2003 to 2013. Remember that these are deaths per 100,000 people. We will take the naive view that areas with lower deaths per 100,000 are automatically healthier places to live, ignoring all other population demographics such as age.

Compute the difference between the death rates of the areas for each year, and encode this as a string visualization using the symbols =, +, and -, where = implies the two areas are within +/- 50 deaths per 100,000 for that year, + implies the first area has more than 50 fewer deaths for that year, and - implies that the first area has more than 50 more deaths for that year. Compare the number of + and - symbols in your string to determine the healthier area with + favoring the first area and - favoring the second. Print out the trend line including a header for the data labeling the start and end years – this should be in reverse order from 2013 to 2003 – along with your determination of the healthier area.

The utility module, **`hw5_util.py`**, provided for this homework will give you some help. Given a string containing a county name, it provides a function read_deaths(county) that returns you a list of 11 numbers.

Hint*: Separate as much of your code into functions as you can. You should have one function dedicated entirely to creating trendlines, one that determines the safety of an area, etc. The basis of these functions (with a few minor tweaks) can then be used for the next part of the homework. This is strongly encouraged. If you're having difficulty with this, please ask questions about it or come to office hours.*

Try the following:

```python
import hw5_util
cdata1 = hw5_util.read_deaths('Erie')
cdata2 = hw5_util.read_deaths('Cattaraugus')
print('Erie:', cdata1)
print('Cattaraugus:', cdata2)
```

would give:

```
Erie: [1061.0, 1032.0, 1047.0, 1020.0, 1040.0, 1037.0, 1029.4, 1010.0,
1043.0, 1014.0, 1046.0]
Cattaraugus: [1005.0, 1089.0, 1061.0, 978.7, 972.7, 978.8, 1010.2, 1083.0,
1002.0, 977.9, 990.0]
```

The difference (rounded to one decimal) is:

```
[56.0, -57.0, -14.0, 41.3, 67.3, 58.2, 19.2, -73.0, 41.0, 36.1, 56.0]
```

Running the program with these values would give:

```
Enter the first area to check => Erie
Erie
Enter the second area to check => Cattaraugus
Cattaraugus

       2013   2003
Trend: -==+=--==+-

I would rather live in Cattaraugus than Erie
```

Note that 2013 and 2003 are printed to line up with the margins of the trend data.
Here is another example where the number of ”+” and ”-” symbols are the same:

```
Enter the first area to check => Bronx
Bronx
Enter the second area to check => Queens
Queens

       2013   2003
Trend: ===========
Bronx and Queens are the same
```

The function **hw5_util.read_deaths** will return an empty list whenever the county cannot be found. Your program must give an error and exit when it encounters an invalid name as shown below.

```
Enter the first area to check => Errie
Errie
Errie is an invalid name
```

```
Enter the first area to check => Erie
Erie
Enter the second area to check => Wesstchaester
Wesstchaester
Wesstchaester is an invalid name
```

You can ignore any warnings you get from using **sys.exit()** that may show up after your error message. When you have tested your code, please submit it as hw5Part1.py.

For this part of the homework, you'll be graded on:

- How your program runs
  - Checking input
  - Calculating differences correctly
  - Printing out trendlines
- Style
  - Code structure
  - Using functions
  - Variable names
  - Commenting
- Using loops

PART TWO: Ranking and Budgeting

For this part of the homework, you will be writing code that ranks the safety of each county based on an algorithm you've created, and then divide funds from NY state for health insurance, social services, and school aid/special education, amongst them. *You must have completed Chapter 1 of "Weapons of Math Destruction" before completing this part of the assignment.*

To start off, you should first plan out on paper how you'll be deciding which counties are safer than others. Once finished, this algorithm should be a function named **sorting** in your program that takes in whatever inputs needed, but ultimately returns a list of strings of sorted counties. Some potential starting points for creating your algorithm are looking at the average change in deaths or basing your algorithm off of the trendlines created in the last part. You will be required to justify your algorithm, so consider this step carefully and thoughtfully.

After you've gotten your ranking of counties, you should move onto how you'll be dividing up each budget. Here are the numbers you'll be working with:

Health Insurance	$71 million
School Aid/Special Education	$42 million
Social Services	$33 million

In three separate functions, you should write code for each of these budgets that divides up the amount provided amongst the NY state counties. You must:

1. Use your ranking of counties in some way
2. Divide the money up in a way that doesn't just have the same amount go to all counties
3. Have reasoning behind why you allocate the funds in the way that you chose (you will write this in the README)

Your program should ask the user what budget they want to look at the statistics for, and then provide the statistics for what county gets what amount. For example:

```
NY State Budgets

You can see the broken up budgets by county for NY State. You can choose
from:
        1. Health Insurance
        2. School Aid/Special Education
        3. Social Services
```

```
Which budget would you like to see? => 1

Health Insurance Statistics
Albany : $1.5 million
Allegany : $3 million
Bronx : $8 million

[and so on....]
```

The user should be able to enter either the number that corresponds with a choice or the name of the budget they want to see. You should handle for error in the case that they number they enter is out of range, or the text they enter isn't a valid budget.

```
NY State Budgets

You can see the broken up budgets by county for NY State. You can choose
from:
         1. Health Insurance
         2. School Aid/Special Education
         3. Social Services

Which budget would you like to see? => 5
The budget entered could not be found.
Which budget would you like to see? =? Social Services
The budget entered could not be found.
```

Your program should continuously loop until the user enters an input that matches a choice.

For this part of the homework, you'll be graded on:

- How your program runs
  - Checking input
  - Outputting budget breakdowns
- Content
  - Present and working **`sorting`** function

  - Your three budget functions
  - Variable names
  - Commenting
  - Using loops

When you have tested your code and it is properly commented, please submit it as hw5Part1.py.

In a separate README file, detail your decision-making process for distributing funds. Remember to do this as it is worth 10 points.

When you have completed your decision-making justifications, please submit it as README.txt.