CPS 3320: Python Programming

Project Assignment #1

This independent project is designed to provide you an opportunity to focus on a problem that interests you — one realistic to something you may encounter in your field of study, or just any issue you are intellectually curious about. The project is very open-ended; the world is your oyster. Specifically, you are expected to illustrate each of the steps of the Problem Solving Workflow:

(a) develop a hypothesis which can be computationally tested;
(b) find an appropriate dataset;
(c) design and implement a method for computing a solution via Python code;
(d) analyze your results; and
(e) communicate your findings.

Since this is the first project, it is intended to be relatively small, but the intent is to provide sound practice for using Python to perform all of the main steps of computation. It may be the case that your entire program is just a couple dozen lines of code, and that is perfectly fine, as long as it exhibits each of the areas described below and in the rubric.

This project has three REQUIRED DELIVERABLES:

Deliverable #1) Project Documentation

You are expected to submit **a concise description of the project (1-2 pages in length)** that MUST include the following:

1. **Background:** a few sentences to put your project idea in context and the overall goal of your project.
2. **Hypothesis:** the specific hypothesis you tested – something that is quantifiably true or false.
3. **Data:** a clear description of the data that you used, including:
    - What the data represents
    - What is the source of the data? How was it curated?
    - What are the possible biases with the data?
    - What are other potential issues with the data?
4. **Analysis/Methodology:** a list of steps that were executed to carry out your analysis.
    - The first step(s) should be about importing and formatting your data.
    - Describe your main computation. That is, describe in English or Pseudocode the main, high-level overview of what you'll calculate.
    - Describe your results.
    - Describe how the end product(s) of your analysis might support your claim, or show it is false.
5. **Conclusion:** a list of steps that were executed to carry out your analysis.
    - What were your results – hypothesis supported or refuted? The goal is to convey your findings effectively.
    - If your results were unexpected, what do you think could explain such? Data? Methodology? Assumption was off?

- If your results were expected, how could you see expanding this (a short 1-sentence idea is sufficient) to test the limits of your hypothesis

Deliverable #2) Code

You are expected to submit **one Python (.py) file**, which contains:

- Bug-free code (shouldn't crash when we run it; it should run and produce the results you showcase).
- *# Python* comments to help make your code readable.
- Decent coding style (e.g., variable names, white space, etc.). Again, make your code readable.
- There is no minimum limit on the number of lines of code you must write to solve your problem. Your specific problem may even have a possible, elegant solution that is only a few lines of code. The main expectation is that you demonstrate knowledge of Python which we covered in class. Every project and every solution is unique, and there is no such thing as perfect code. The most important thing is that your code works, is readable, and the results are clearly communicated.

Deliverable #3) Project Reflection

Communicating a narrative is an important part of any project, as it allows your work to reach others and potentially affect change and extend further. Consequently, for this project **you are to provide a short write-up (1-2 pages) that "tells your CS story".**

- Provide context for what led you to choose this particular project idea. Why was this a problem that you wanted to solve?
- What were the most challenging parts of your development process? What was easier than expected and what was more difficult than expected?
- Did you have to change anything, or pivot, from your original project idea, or your programming approach? If so, how?
- If you had additional time to work on this project, in what ways would you want to build it out further and improve upon it?
- What were your main takeaways from this experience? If you could do it again, what would you do differently?