Project 1: Problem Solving with Python

Due : March 10th

1 Project Overview

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, as we discussed in Lectures #1 and #2:

(a) develop a hypothesis which can be computationally tested;

(b) find an appropriate dataset (as discussed in Lecture #3);

(c) design and implement a method for computing a solution via Python code (as discussed in Lecture #4 onwards);

(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 performing 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 that we describe below and in the rubric. Regardless, the TAs and I are here to help along the way, and since we want it to be a manageable project for you, we require that you first draft a *proposal*, (which includes outlining all parts of your project, from data, potential issues, etc) and meet with a TA in person so that we can ensure your project is appropriate and not too ambitious (or too short).

The **proposal portion is due first, on February 27th (Thursday).** Please talk to us as you draft your proposal so that we can help ensure that your project is appropriate and that you can start on writing the Python solution as soon as possible – as opposed to submitting a proposal right before the deadline, then hearing from us a few days later with large suggestions, leaving you with just a week to do the rest of the project.

The **completed project is due by March 10th (Tuesday) at 11:59pm,** which allows for roughly three weeks of work. We ask that in addition to submitting your Python code (`.py` file) which performs your calculations and results, you provide a write-up of your project (details below). The intent of this project at large is to help you hone your skills in formulating problems while keeping in mind the individual steps and eventual need to communicate your findings.

2 Task 1: Proposal (due February 27th by 11:59pm)

Write a concise description of the project you would like to execute before you start working with Python. Proposals should be 1-2 pages in length, but the more specific details you provide, the better feedback we can give. This description should cover everything in the rubric for the proposal, including the following parts:

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 plan to test – something that is quantifiably true or false.
3. **Data:** a clear description of the data you plan to use, including:
   - What the data represents.
   - The format of the data and how you are going to import it into your spreadsheet.
   - Where is it located (include the URL)
   - Who/How was it curated? For example, was the data originally collected from the United States Census Bureau, but then gathered and cleaned (removes the noisy, bad, or missing elements) by a different organization, and the dataset is now available on Kaggle.com?
   - What it currently looks like (what the data includes). If you want to include a picture of it, you can capture a screenshot by pressing `PrtScr` (for "Print Screen") and pasting it into a document. If you have a Mac, press `Cmd + Shift + 4` and click and drag the selection box around the area of your screen you want to capture (this will save your image to the Desktop).
   - What are possible biases with the data.
   - What are *other* potential issues with the data. For example, having biases is a huge cause for issues, but maybe your data is also small, or intimidatingly large, or certain attributes seem to be messy (not uniform, such as if a US State is sometimes abbreviated, sometimes spelled out, and sometimes referred to in an even different way), etc.
4. **Analysis Steps:** a list of steps to carry out your analysis. Be concrete about what you plan to do.
   - The first step(s) should be about importing and formatting your data.
   - Describe your main computation. This should be on the par with what you did for Homework #2. That is, describe in English or Pseudocode the main, high-level overview of what you'll calculate.
   - Describe how you plan to present your results. Will it be a single # (such as the case when calculating an alignment score between -1 and 1), or a % of something, a count of something, or any other fancy statistic that you may want Excel or any other 3rd-party program to calculate for you? NOTE: We don't expect any complicated math,

but if you're interested in calculating, for example, a correlation, and you're familiar w/ getting Excel to calculate a correlation value, then by all means please use it. Otherwise, a fraction of something, a count, etc is perfectly sufficient!

- Describe how the end product(s) of your analysis (e.g., in Lecture #1 where the average number of times Senators are mentioned within a political label) might support your claim, or show it is false.

5. **Potential Roadblocks:** a list of potential hangups and needed resources. There are all kinds of roadblocks you might encounter; try hard to imagine what might trip you up. The following examples are to get you started thinking:

   - The data might have to be formatted in a special way, or you plan to import LOTS of data (i.e. from many years/sessions).
   - There's a calculation that might be tricky.
   - You think there might be a more interesting claim to test but you don't know how to test it.
   - You need to use 2 or more datasets, and there may not be enough data that is common amongst all datasets (e.g., data that corresponds to zip codes, but the other useful dataset may not have data for all zip codes).

2.1 Finding Data

In lecture, we have learned about reading input from files, which is what we expect for this project. In particular, datasets are commonly separated into columns by delimiters such as commas. So far, we have mentioned Kaggle.com as a great resource, which typically has files of such format, too. However, also feel free to Google for "large free datasets," "great public datasets", etc. Regardless of where you get your dataset from, it's your responsibility to judge if the data on that site is actually credible and appropriate for your task. After all, it's just another human who curated each dataset, so a lone dataset might be faulty and non-ideal.

2.2 Submitting

Check the rubric for the proposal – did you cover everything?

The earlier you submit the proposal, the sooner we can provide feedback and thus give you the green light to start writing Python Code to solve your problem.

Submit your proposal in a pdf via Gradescope before February 27th (Thursday) 11:59pm. At the top of your document, answer the following:

1. How many hours did this assignment take you?
2. Did you seek help from TAs or Enrique? (We hope so)
3. Is there any material which you feel particularly confused about?
4. Is there anything we can do to help you further understand said material / do you have a learning preference such as wanting more visuals, more examples, in-class activities, etc.

3 Task 2: Python Program + Write-up (due March 10th by 11:59pm)

3.1 Python

Carry out the project you proposed. It's OK if the project changes — that's why it was a proposal, but be realistic with respect to the time remaining before it is due. Check the rubric for the project for details of everything we will expect.

In short, we expect you to write 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). We won't be harsh on grading style, since this is the first project, but you should have good variable names and please just have appropriate spacing and white space to 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 have a possible, elegant solution that is only a few lines of code. We only ask and expect you to demonstrate knowledge of Python which we cover in class: e.g., reading files, using functions, and when appropriate, using basic data structures like lists and dictionaries, and constructs like if-statements and for-loops. Every project and every solution is unique, so we have no requirements as to which data structures and how many data structures you use. Related, there's no such thing as perfect code, so don't stress, as we will not harshly grade you on which exact data structures you use. The most important thing is that your code works and is readable.

3.2 Write-up

Communicating results is an important part of any project, as it allows your work to reach others and potentially affect change and extend further. Consequentyly, for this project we ask for you to provide a short write-up (2-3 pages will likely be sufficient.) If, for example, you have several graphs that you want to show, then it's okay if you exceed 3 pages. **Much** of your proposal's content will be used in the write-up. The main difference being that you'll remove future-tense actions and concerns and replace them with past-tense methods, issued that you encountered, results you calculated, etc.

Your writing should be for an audience of (1) TAs and Enrique to understand at a high-level everything you did; (2) the general population. Imagine this content being accessible from the web, and that it's something you'd want to show the world. That is, it's important to convey what your project was about, while getting into specifics of what you cared about, what your data was, and what your results and thoughts are, but **do not go into in-depth details about your computations.** We will read your code to glean such, and the general population should be

shielded from knowing your exact calculations. That is, describe your general method, but not the exact process of using loops, functions, etc.

Specifically, your write-up should 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 you used, including:
   - What the data represents.
   - The format of the data.
   - Where is it located (include the URL)?
   - Who/How was it curated? For example, was the data originally collected from the United States Census Bureau, but then gathered and cleaned (removes the noisy, bad, or missing elements) by a different organization, and the dataset is now available on Kaggle.com?
   - What it currently looks like (what the data includes). If you want to include a picture of it, you can capture a screenshot by pressing `PrtScr` (for "Print Screen") and pasting it into a document. If you have a Mac, press `Cmd + Shift + 4` and click and drag the selection box around the area of your screen you want to capture (this will save your image to the Desktop).
   - What are possible biases with the data?
   - Did you encounter any issues with the data, such as missing information, bad formatting, etc.?
4. **Analysis Steps:**
   - The first step(s) should be about importing and formatting your data if you had to.
   - Describe your main computation in high-level English; please don't describe your computations in in-depth detail (we will read your code for such).
   - Describe what your computation's output is (e.g., a percentage increase/decrease in something, a single-valued # that will be above or below 0, etc) and what expected values would support or refute your claim
5. **Results/Discussion:**
   - What were your results – hypothesis supported or refuted? We encourage you to make a graph or plot if it's appropriate for your results, which could lead to receiving extra credit. Although, in some cases, a plot is completely unnecessary and can be more distracting. Use your judgment (or talk with us) on deciding if a graph is needed, while keeping in mind that 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.
- What were the most challenging parts?
- Reflection: Did you have to change anything from the project you originally proposed? Any particular challenges or problems you ran into? If you could do it again, is there anything that you'd change to make your experiment easy to code or more sound.

4 Extra Credit

This project has plenty of opportunity for extra credit. Be creative. At the top of your submitted Python (`.py`) file, feel free to explicitly mention things that you did which you feel were above and beyond and **not mentioned in your proposal.** That is, when we approve your proposal, that is our accepting that the work you outlined is warranted of a project worth full-credit. If you add graphs or do multiple analysis, or happen to do an impressive job with loading multiple datasets, etc, these things will likely warrant extra credit, so please bring to our attention anything you did which was beyond your proposed work. The TAs will do their best to pay attention to all the nuanced elements of your code, but you can help.

4.1 Submitting

Check the rubric for the proposal – did you cover everything?

Submit your final project via Gradescope before March 10 (Tuesday) 11:59pm. We expect two files: (1) your Python (`.py`) file(compressed to zip file like previous assignments); (2) your write-up (in PDF format). At the top of your Python (`.py`) file, list:

1. **Any work you did which you feel warrants extra credit**
2. How many hours did this assignment take you?
3. Did you seek help from TAs or Enrique? (We hope so)