Final Project Rubric

Due: Dec. 13, 11:59pm

The final project is here...Time to take over the world!

All projects should be submitted as a zip-file entitled `FinalProject_<your name>.zip`. This zip file should contain all of your python code, spreadsheets, html, etc. If you have a website for your project, please include a file entitled `WEBSITE.txt` with the url of your site. All projects must include a `README.txt` in the zipfile. This text document should contain a description of each of the items in the folder as well as any concerns or important information not addressed elsewhere in your project presentation.

Rubric

Proposal (10%)

- You should do this write-up after you meet with a course staff, and before you start to write code or manipulate Excel spreadsheets. This is your plan.
- Explain your overall goal in a few sentences. Think of this as your project's thesis statement.
- Tell us where you are pulling your data from and how you are going about this.
- Break down your project into steps and explain how they relate to the goal. For example, simply stating that you're going to find voting blocks isn't enough. Talk about the specifics, such as the actual data, formulas, and skills you will use. For example, if you were looking for voting blocs, you should then describe what you're defining quantifiably as a "voting bloc". Then you might describe how you determine whether two senators are similar. For example, "I will look at each vote for the two senators of interest, and if they match, I will record a one and if they don't, a negative one. I will then add up these votes and divide by the number of votes that both senators were present for to get a number representing their similarity in voting patterns.

Execution (25%)

- This portion of your grade will reflect how well you were able to execute your plan of action.
- Make sure to look at your data at intermediate steps of your computation (intermediate spreadsheets, specific functions) and verify that it looks correct. This is where you will lose big points for bugs if you don't notice them.
- Besides bugs, your data will sometimes have errors in it; make sure you deal with this or at least mention it in your analysis.

Interactive Grading (15%)

- Ownership: do you understand every line of your code? Every formula you used in your spreadsheet? The nature of the data?
-

Analysis of Results and Presentation (15%)

- You must present your results in Google Sites web page, which may contain screenshots of maps, spreadsheet results, graphs, text, etc. Make sure that you set the permission of your website to be "anyone with the link".
- This is where you should address any problem you had in converting your initial design into a result. If you do not address and reflect on issues in your code, you will likely lose points in both this section and in Execution.
- Don't forget to mention any reflections on how you might have been able to improve your process if you were to do it again.

Code Quality(10%)

- Everything listed under Coding Style in the grading rubrics for project 2.

- "Production Values" refers to the cleanliness and professionalism of your over-all project. Basically, this means spell-check and remove the print statements in your python code you used for testing. Don't have meaningless files lying around in your zip folder.
- For Excel:
  - Give a name to each "sheet" (instead of the default "sheet1", "sheet2", etc).
  - Use "comments" on cells to explain their contents
  - Every cell should generally either be the result of reading in some data (a CSV file, or an XML file, ...) or should have its value produced by a formula. Exceptions are:
    * You might want to make a table for converting strings to numerical values, like

YEA	1
NAY	2
MISSING	3
NO VOTE	4

    * If your computation is really huge, recomputing things via formulas may become very slow. In such a case, with TA approval, you can copy-and-paste computed values into a new Sheet, from which you then derive further things. This copy-paste operation should be well-documented in your writeup so that its reproducible by someone trying to replicate your results

Degree of Difficulty (25%)

If you propose a really simple project, youll get no DD points, or your mentor may suggest things you could add to get more DD points. If you propose a difficult project but dont complete it, you might get 25 DD points, but lose a few points for execution. And since the execution of complex project is bigger than that for a simple one, the same minor error in those two projects might count differently: one error in 70 lines of program is a bigger deal than the same error in 250 lines of program.

Good Luck!