Final Project Grading Rubric

DCDS510 / CSE314A

The primary deliverables for this project will be your code base, documentation for your project, and a presentation at the end of the semester describing your project and each team member's contributions.

For students working in teams: please make personal commits that accurately reflect your contributions and push/merge to your team's central repository/branch.

	0% credit	50% credit	100% credit
Speed **10 pts**	Steps take long enough (hours) to render the code unusable.	Certain steps take a while to run, but overall data manipulation runs smoothly.	All analysis steps can run near-instantaneously with either calculated or cached values. Caching is reserved for important network calls or expensive computation, and result in minimal storage footprint
Accuracy **10 pts**	Code clearly analyzes incorrect metrics, performs calculations incorrectly, etc., or throws errors for common use cases.	Minor calculation mistakes, some massaging of data is performed/is needed	Performs complete and accurate data manipulation and calculation, works with complete data set given scope of the problem.
Code cleanliness **10 pts**	Poor variable naming, extremely long, nested functions, code functionality cannot be deduced.	Code at least avoids long messy for loops and if/else branching. Code takes effort to trace but is overall	Code is highly modular and functions mostly follow the single responsibility principle. It is clear on first glance what each module does without excessive inline comments or complex documentation.

		manageable	
Project Architecture **10 pts**	Files are not organized in any way that makes sense, significant problems running the project on a new machine, significant clutter and bloated folders/files.	Some rearrangement of code is warranted but overall the project components are organized into appropriate folders and architecture is evident.	Each component of the project lives in a designated place that is easy to discover. No extra unnecessary files or directories are included in the repo. Data files delivered separately, only if needed (small lookup tables OK). All dependencies are included in .toml and/or documentation.
Deliverables (individual grade) **20 pts**	Not much progress was made towards executing deliverables.	Contributions show a clear attempt towards deliverables but there are large gaps between the state of the project and the completed deliverables.	Deliverables were executed with the appropriate scope from progress reports, or provided appropriate discussion around significant hurdles.
Relevance to course material **20 pts**	Work is entirely out of scope.	Code borrows concepts from the class but focuses on out of scope concepts or does not fully apply lessons learned	Code clearly uses techniques and approaches demonstrated or mentioned in class.
Presentation (Individual grade) **10 pts**	No-show, clear misunderstanding of project goals, does not demonstrate participation.	Participates in presentation but it is unclear how the work presented achieves project goals. Or, others need to present your work on your behalf.	Clearly demonstrates project contributions in the presentation
Documentation **10 pts**	No documentation.	README/documentation is confusing or missing project	A thorough README is provided to help with developer setup and orientation. Documentation for a wider audience (people seeking to use your data)

		components	is included and uses a static site generator, i.e. Sphinx or MkDocs. All components of the project are documented including API docs for your module code.