# DATA 31500 Autumn 2024

## Project

*Due December 2, 2024*

In this project, students use visualization and/or web technology in the production of original data science research. The purpose of the project is to put the skills and knowledge gained in this course into action to advance the student's research interests. It is best if the project is agreed upon and planned in coordination with the student's advisor, such that the work produced can become publishable scholarship in the student's intended area of research.

Possible high-level project ideas include *but are not limited to*:

1. Build a custom data collection interface using web technology
2. Create public-facing data story, explorably explanation, or dashboard
3. Prototype a data-driven system or a design probe for user testing
4. Use visualization to advance scientific understanding of a complex dataset

Regardless of which option they take, students need to be clear about how visualization and/or interactive web technology was instrumental to advancing their research. All projects must be approved by the instuctor through the proposal process.

**Deliverables** include: a codebase with documentation; a presentation on the final week of class; and a 4-page technical write-up detailing the project's motivations, method or approach, provisional findings, and importance. These are in addition to the proposal due earlier in the quarter, which has a separate specification.

Students will *work alone, or in groups of 2-3*. Groups must have been formed at the time of the proposal deadline. Changes are not allowed at this time. *No free riders:* If the instructor finds out that you did not contribute to your project team, you will not receive credit for the project.

**All components should be submitted on Gradescope:** The codebase should be in a .zip file with a README. The presentation should be uploaded as a .zip file containing compressed slides. The write-up should be a PDF.

---

## Technical Specification

Given that the project is meant to advance the student's research interests, the specification is relatively open-ended. Here we outline the expectations for each component.

**Codebase:** Students must submit a codebase substantiating the efforts behind their project. *It should contain any code used to render visualizations or web pages created for the project. Work should be documented.* For visualizations created in a notebook, this documentation should be provided in a literate programming style with explanatory text blocks interleaved with code blocks. All notebook cells should be pre-executed, so the instructor can grade

them easily. For web infrastructure, documentation should be in the form of a README providing intructions on how to run the application locally. Any web application should be deployed on the internet and linked in the README, so the instructor can access it easily for grading.

**Presentation:** During the last week of classes, students will give a 3-5 minute lightning talk giving an exposition of their project. A short talk is more challenging than a long one in some ways, so this will be good practice in preparing a tight research presentation, in addition to practice with public speaking. Presentations must include a slide deck with relevant images from the project. *Critically, the presentation must motivate the research problem the student chose to work on, demonstrate how visualization and/or web technology was used in the project, and summarize provisional findings or impacts.* It is imperative that presentations are polished and professional. For group projects, all contributors must have a substantial speaking role during the presentation.

**Write-up:** *The write-up should be a 4-page technical document detailing detailing the project's motivations, method or approach, provisional findings, and importance. It should include an abstract, figures, and formatted references.* The write-up should clearly accompany the work submitted in the codebase, and all scientific claims should be substantiated with demonstrated work or citations. Figures should be polished and should demonstrate design practices learned in this course. References must be credible sources. The reference format should be based on norms in the student's chosen area of research.

Above all, *students must present a clear narrative in technical writing.* This means that arguments should cohere, rely on valid logic, and avoid fallacies or baseless/unsubstantiated assertions. The style of writing should be formal and factual, while also presenting a story about the project. Storytelling can be difficult, so it may help to look at examples of academic papers and data journalism that make a complelling arguement and reflect on what they do well. Good academic writing in computer science often starts by identifying a problem, summarizing a solution or findings about the nature of that problem, then presenting the approach to the problem in depth, and concluding with a discussion of what was found. However, students do not need to follow this formula. We encourage students to be creative, and demonstrate what they've learned about how to do rigorous analysis and visualization.