

## Project 2

INFO 526

Fall 2024

Your task for this project is to create *something* related to data visualization.

This is intentionally vague – part of the challenge is to design a project that showcases best your interests and strengths.

One requirement is that your project should feature some element that you had to learn on your own. This could be a package you use that we didn't teach in class (e.g., a package for building 3D visualizations) or a workflow (e.g., making a package) or anything else. If you're not sure if your "new" thing counts, just ask!

### Ideas

Your first task is to come up with a goal for your project. Here are a few ideas to help you get started thinking:

- Present and visualize a technical topic in statistics or mathematics, e.g., Gradient descent, quadrature, autoregressive (AR) models, etc.
- Create an R package that provides functionality for a set of ggplot2 themes and/or color palettes.
- Do a deep dive into accessibility for data visualization and build a lesson plan for creating accessible visualizations with ggplot2, Quarto, and generally within the R ecosystem.
- Create an interactive and/or animated spatio-temporal visualization on a topic of interest to you, e.g., redistricting, COVID-19, voter suppression, etc.
- Make a data visualization telling a story and convert it to an illustration, presenting both the computational and artistic piece side by side.
- Do a deep dive into an emerging topic for data visualization and build a lesson plan for creating reproducible visualizations with ggplot2, Quarto, and generally within the R ecosystem.

And, of course, your project can be about visualizing a dataset of interest to you (similar to your first project). The only rule about this dataset is that it can't be from TidyTuesday. Beyond that, it should be something truly of interest to your team, and a dataset that allows for a deep exploration.

Most importantly, be prepared to brainstorm a bunch of ideas and discard them until you settle on a topic that everyone in the team is happy with and feels like a good choice for showcasing what you've learned in the class and how you can use that to learn something new and implement for your project.

## Workflow

- Week 1 of project (week of Oct 28): Pick a focus for your project.
- Week 2 of project (week of Nov 04): Work on developing your project proposal and setting up the structure for your repository.
- Week 3 of project (week of Nov 11): Finalize your project proposal + Conduct peer review on project proposals
- Week 4 of project (week of Nov 18): Continue working on your project, and optionally, submit in an updated version of your proposal for Instructor review.
- Week 5 of project (week of Nov 25): Continue working on your project, incorporating feedback from proposals.
- Week 6 of project (week of Dec 02): Bring templates of prepared posters to class for peer review and feedback.
- Week 7 of project (week of Dec 09 - finals week): Present projects and turn in final write-up.

## Due dates

- Proposals for peer review: due Wed, Nov 13 at 2:00pm.
- Peer review of proposals: Wed, Nov 13 in class.
- Revised proposals for instructor review: due Mon, Nov 18 at 5pm.
- Peer review of poster: Wed, Dec 04 in class.
- Presentation will be during the iShowcase on Wed, Dec 11.
- Write up due by 11:59PM on Fri, Dec 13

## Deliverables

### Proposal

Your proposal should include:

- A one sentence description of your high-level goal (such as the ones listed under [Ideas](#) above.
- A one to two paragraph description of your goals, including your motivation. Depending on the focus of your project, the following might go in here.
  - If using particular dataset(s), a brief description of each dataset including the reason why you chose the particular dataset, its dataset, its dimensions and any other relevant metadata. (Make sure to load the data and use inline code for some of this information.)
  - If answering a particular research question, the question itself and the reason why you chose this question.

- A weekly “plan of attack” outlining your steps to complete your project and including the team member(s) assigned to that task.

- This should be in the following form:

Task Name	Status	Assignee	Due	Priority	Summary
				Low	
				Moderate	
				High	

- **Note** that this is a living document and should be updated at minimum once a week.
- The final organization of your project repository. This means describing the project organization in your proposal as well as implementing that organization in your repo. Create any folders needed and place a `README.md` in each folder explaining what goes in there.

## Peer review

The peer reviews will be completed during the lab sessions as indicated above. On those days teams will have access to the project repos of the teams whose work they’re reviewing. The first peer review will be of proposals and the second peer review will be of code and organization.

### Reviewer tasks

Teams will develop the review together, with discussion among all team members, but only one team member will submit it as an issue on the project repo. To do so, go to the *Issues* tab, click on the green *New issue* button on the top right, and then click on the green *Get started* button for the issue template titled *Proposal peer review* or *Code peer review*.

This will start a new issue with a peer review form that you can fill out. Make sure to update the introductory paragraph with your team name and the names of the team members participating in the review. Then, answer each of the questions in the spaces provided underneath them. You’re expected to be thorough in your review, but this doesn’t necessarily require lengthy responses.

Remember, your goal is to help the team whose project you’re reviewing. The team will not lose points because of issues you point out, as long as they address them before I review their work. You should be critical, but respectful in your review. Also remember that you will be evaluated on the quality of your review. So that’s an additional incentive to do a good job.

### Reviewee tasks

Once you receive feedback from your peers, you should address them. You should do this by directly updating your proposal or making any other updates to your repo as needed. You can do these updates all in one commit or you can spread it across multiple commits.

Regardless, in the last commit that addresses the peer review comments, you should use a keyword in your commit message that will close the peer review issues. These words are close, closes, closed, fix, fixes, fixed, resolve, resolves, and resolved and they need to be followed by the issue number (which you can find next to the issue title). So, your commit message can say something like *“Finished updates based on peer review, fixes #1”*.

### Peer review assignments

The peer review assignments for **proposal peer reviews** are TBD:

The peer review assignments for **code peer reviews** are TBD:

### Write-up

You have a lot more freedom in how you structure your write-up for this project compared to Project 1. This also comes with responsibility. You should make sure you have a clear introduction and a clear conclusion. You should also have other interim section headings that help the reader. Your write-up should be somewhere between 1000 and 2000 words. There is no expectation that you get close to the upper limit, anywhere in that range is fine as long as you have clearly explained yourself. The limits are provided to help you, not to set stressful expectations. This work should be your team's original work. Connections to literature and course work are encouraged (make sure to cite your sources).

### Presentation

The presentation will be a poster that is presented at the iShowcase. Each member of the team is expected to have 2-4 sentences about a specific portion of the work. All members should be able to speak about the the entire project.

This presentation should be considered similar to a poster conference presentation. I will have more guidance on how to print posters for the iShowcase. Students will share a bulletin board with other peers. Meaning, 4 teams per poster board. Do not print your poster until after the peer review.

You should create your presentation in a reproducible way, e.g., using Quarto. However you don't have to use a package that is designed specifically for slides. If you prefer to build a dashboard or a Shiny app or a website, that's fine too. The only rule is that it's built reproducibly using R.

Your evaluation will be based on your content, code, professionalism, translation from a project to a poster, and ability to explain your work and interpret the visualizations.

### Repo organization

Since you have complete freedom on the focus of your project, you also have complete freedom on the organization of your repository. But... it should be organized in a logical way! Remove items that are unnecessary or cluttering. Make sure ALL sources are cited.

At a minimum each project should have an `index.qmd`. If your project is analyzing a dataset, you'll probably want to structure your repo similar to your Project 1 repo. If you're building a package, it should be structured like a package.

## Grading

Total	110 pts
Proposal <sup>1</sup>	10 pts
Presentation <sup>2</sup>	30 pts
	(25 pts from teaching team, 5 pts from audience)
Write-up <sup>3</sup>	30 pts
Reproducibility, style, and organization <sup>4</sup>	10 pts
Within team peer evaluation <sup>5</sup>	20 pts
Between team peer evaluation <sup>6</sup>	10 pts

Some of the components are further detailed below.

Note that there will be points allocated to commits from each team member for the proposal, presentation, and write-up.

### Proposal (10 points)

- High level goal
- Expanded description
- Plan
- Repo organization
- Workflow - Peer review issues closed via commits. (1 point)
- Teamwork - All team members must contribute to the repo via commits. (1 point)

---

<sup>1</sup> Your proposal score will also take into consideration how much you've improved your proposal based on peer feedback.

<sup>2</sup> A more detailed grading rubric for this component to be added.

<sup>3</sup> A more detailed grading rubric for this component to be added.

<sup>4</sup> Style and format does count for this assignment, so please take the time to make sure everything looks good and your data and code are properly formatted.

<sup>5</sup> This relates to your contribution to the teamwork and how your team members evaluate this contribution. Scores for each teammate will be different for this component of the project grade.

<sup>6</sup> This relates to the quality and quantity of the peer review you provide for other teams.

## Presentation (30 points)

### Teaching team (25 points)

- Time management: Did the team divide the time well among themselves or got cut off going over time? (2 points)
- Professionalism: How well did the team present? Does the presentation appear to be well practiced? Did everyone get a chance to say something meaningful about the project? (2 points)
- Teamwork: Did the team present a unified story, or did it seem like independent pieces of work patched together? (3 points)
- Poster: Is the poster well organized, readable, not full of text, featuring figures with legible labels, legends, etc.? (3 points)
- Creativity / Critical Thought: Is the project carefully thought out? Does it appear that time and effort went into the planning and implementation of the project? (5 points)
- Content: Including, but not limited to the following: (10 points)
  - Is the question well articulated in the presentation?
  - Can the question be answered with the data?
  - Do(es) the data visualization(s) answer the question?
  - Do(es) the data visualization(s) follow good visualization practices?
  - Is/are the conclusion(s) made based on the visualization(s) justifiable?
  - Are the limitations carefully considered and articulated?

### Peers (5 points)

- Content: Is the project well designed and is the data being used relevant to the focus of the project? (1 point)
- Content: Did the team use appropriate visualizations and did they interpret them accurately? (1 point)
- Creativity and Critical Thought: Is the project carefully thought out? Are the limitations carefully considered? Does it appear that time and effort went into the planning and implementation of the project? (1 point)
- Slides: Are the slides (or other presentation medium) well organized, readable, not full of text, featuring figures with legible labels, legends, etc.? (1 point)
- Professionalism: How well did the team present? Does the presentation appear to be well practiced? Are they reading off of a script? Did everyone get a chance to say something meaningful about the project? (1 point)

### Write-up (30 points)

- Introduction: The introduction provides a clear explanation of the question and the dataset used to answer the question, including a description of all relevant variables in the dataset. (3 points)
- Justification of approach: The chosen approach and visualizations are clearly explained and justified. (3 points)
- Code: Code is correct, easy to read, properly formatted, and properly documented. (10 points)
- Visualization: The visualizations are appropriate, easy to read, and properly labeled. (10 points)
- Discussion: Discussion of results is clear and correct, and it has some depth without begin excessively long. (4 points)

### Reproducibility, style, and organization (10 points)

- All required files are provided. Quarto files render without issues and reproduce the necessary outputs. If building a package, the checks pass. (3 points)
- If there's a dataset, it's provided in a data folder, a codebook is provided, and a local copy of the data file is used where needed. (3 points)
- Documents are well structured and easy to follow. No extraneous materials. (2 points)
- All issues are closed, mostly with specific commits addressing them. (2 points)

### Guidelines

Please use the project repository that has been created for your team to complete your project. Everything should be done reproducibly. This means that I should be able to clone your repo and reproduce everything you've submitted as part of your project.

All results presented must have corresponding code. If you do calculations by hand instead of using R and then report the results from the calculations, you will not receive credit for those calculations. Any answers/results given without the corresponding R code that generated the result will not be considered. For example, if you're reporting the number of observations in your dataset, don't just write the number manually, use inline R code to calculate the number. All code reported in your final project document should work properly. Please do not include any extraneous code or code which produces error messages. Code which produces certain warnings and messages is acceptable, as long as you understand what the warnings mean. In such cases you can add `warning: false` and `message: false` in the relevant R chunks. Warnings that signal lifecycle changes (e.g., a function is deprecated and there's a newer/better function out there) should generally be addressed by updating your code, not just by hiding the warning.

### Tips

- We hope some of you will take the challenge to be adventurous and learn some new skills as part of this project. We're happy to support you along the way, so don't hesitate to ask as many questions as needed!

- You're working in the same repo as your teammates, so merge conflicts will happen, issues will arise, and that's fine! Commit and push often, and ask questions when stuck.
- Review the marking guidelines below and ask questions if any of the expectations are unclear.
- Make sure each team member is contributing, both in terms of quality and quantity of contribution (we will be reviewing commits from different team members).
- Set aside time to work together and apart (physically).
- Code:
  - If you want to highlight something specific about a piece of code, you're welcomed to show that portion. Comment often as to what your code is doing.
  - In your write-up your code should be visible.
- Teamwork: You are to complete the assignment as a team. All team members are expected to contribute equally to the completion of this assignment and team evaluations will be given at its completion - anyone judged to not have sufficiently contributed to the final product will have their grade penalized. While different team members may have different backgrounds and abilities, it is the responsibility of every team member to understand how and why all code and approaches in the assignment work.
- When you're done, review the documents on GitHub to make sure you're happy with the final state of your work. Then go get some rest!