Tag a release of your project with tag version v1.0-final by Wed Apr 14, 11:59pm. Your release must contain the final implementation and a thumbnail screenshot of your vis (thumbnail.png).

The team must submit a PDF with your writeup through Canvas, by the same deadline.

Each team member must fill out a peer review form, by the same deadline. A link to the form will be provided later in the term.

Reminder: Live demo

You will need to give us a short demo of your final visualization project (approx. 5-8 minutes per group) after the final submission. All members of your team should attend the face to face demo if at all possible, which will be a scheduled Zoom session with the TA. Exact time slots will be announced before the due date.

Implementation

Continue to develop your visualization project by using your git team repository (don't forget, we'll be considering your commits when grading your project).

Turn your project proposal into an interactive web-based visualization with D3 that is ready to be deployed and that can be consumed by your intended audience.

- Make sure that your code is well-organized, easy to read, and contains code comments. Use separate files for visualization components, use functions to promote code reuse.

- Recall the minimum requirements (from proposal):
  - At least 3 different views / visualization components (e.g. 3 bar charts only count as 1 component).
  - At least 1 of these views must be an innovative view that is either (a) an extension of an existing visualization type, or (b) a novel visualization type.
  - Multiple views coordinated with linked highlighting. A click/hover/selection interaction within one view must trigger a change in a different view. At least 2 views need to be linked. Ideally, these views are linked bidirectionally.
At least 2 UI widgets that allow users to filter the data or update certain views interactively (e.g. dropdown, radio button, range slider, calendar).

Interactive tooltips are shown when users hover over marks, in at least one view.

- We don't allow custom backends (Node.js, Python, etc) and database systems, such as Postgres or MySQL, although they could facilitate more powerful applications. Your web application should be stand-alone and have an `index.html` file as entry point. You can either store static (maybe preprocessed) datasets in your git repo or access live data through an API.

- Refine the features you started earlier and implement the rest of the features you outlined in your proposal, paying close attention to usability and information density.

- Your interface should be as self-documenting as possible, with appropriate labels for panes, axes, and widgets, a legend documenting the meaning of visual encodings, and a meaningful title and description for the app. Apply what you have learned in the color foundation lectures and choose appropriate color schemes.

Write-up

In addition to the implementation, you must submit a final report as PDF.

Your final report should be a standalone document that fully describes your project.

You can build upon your write-ups from Milestones 1 and 2. The report should be at least 5 pages of text (~2500 words) without screenshots. Please include the following sections in this order:

1. Overview
   - *Teaser image* (screenshot) of your visualization.
   - Concise summary of your project (max. 250 words).

2. Data
   - Description of your data in both domain-specific and abstract language (dataset type, scale/cardinality).
   - Include a URL linking to the source of your data.
   - Briefly describe your current data preprocessing pipeline, if there is one.

3. Goals and Tasks
   - Description of your intended task(s) in both domain-specific and abstract language. Do not discuss the visual encoding or interaction idioms chosen in this section.
4. Visualization

- Describe the visualization interface that you have built. What views are there and what do they allow users to do? For each view, describe your visual encoding choices and include the rationale for your design choices. How can users interact with your project within each view, and how are views linked? Include a usage scenario walking through how your visualization can be used during an interactive session, illustrated with screenshots of your system in action. It may be different than what you originally envisioned in your proposal, that's fine.
- Screenshots are mandatory.

5. Credits

- Indicate any sources of inspiration, including any specific D3 code blocks that you consulted or built upon. Explain what changes you made and their magnitude (e.g. unchanged vs. minor tweaks vs. major functionality additions) for any code that you built upon.

6. Reflection

- Describe how your project has developed from your initial proposal, through your first submission, to your final product.
- How have your visualization goals changed?
- How have your technical goals changed?
- How realistic was your original proposal in terms of what is technically possible in D3?
- Was there anything you wanted to implement that you ultimately couldn't figure out how to do? If so, then what workarounds did you employ, or did you abandon your original idea?
- If you were to make the project again from scratch (or any other interactive visualization), what would you do differently?

7. Project Management & Team Assessment

- Update your work breakdown and schedule with your final status. For each chunk of work that you carried out, include the actual numbers of how long it took to do in reality (put these next to your original estimates so you can see them side by side). Do this for both estimates: the number of hours and the date completed. Also add to your work breakdown any new chunks of work that you ended up needing to do that were not included in previous versions of your plan.
- Include brief descriptions of which team member worked on which tasks (and their responsibilities, if not equally divided).

Thumbnail

We would like to publicize the top projects! Please create a single representative screenshot of your project...
(size: 1280 x 720 px) named thumbnail.png and add it to the root directory of your repository.