CS/INFO 3300; INFO 5100
Project 2
Due 11:59pm Thursday, April 11

You are creating an **interactive data visualization**. Your interface should use **dynamic elements** to show more information than can be presented in a single view. An example of this is Shneiderman's formula: "overview first, zoom and filter, then details on demand." You are **required to integrate multiple data sources** for this project. Your project must be **responsive and performant** in our web browser, with no lengthy loading times or distracting latencies. Each team has been assigned a TA. Your TA is your best resource.

For this project, we will be assigning all teams an **undergraduate TA mentor**. All teams must use a **source code repository** such as Github. Each team must **give their undergraduate mentor access to their source code repository by 11:59pm on Thursday, March 14. All projects must obey a 50mB maximum file size limit for their final ZIP submission.**

Examples and sources:
You are encouraged to get data and inspiration from other sites. Make sure you acknowledge these in comments and in your written description. Any code that you did not write yourself should go in a separate .js file. While it is acceptable to link to the web repositories for d3.js and topoJSON, all other libraries must be bundled in your archive. **Unacknowledged code or concept reuse will be handled with standard academic integrity procedures.**

Meet with your undergraduate TA mentor to go over a written status report by 11:59pm on the following dates. **All teams must meet face-to-face with their mentor prior to the deadline; no CMS submissions will be permitted**. Additionally, **a different team member must meet with your mentor for each milestone** (except in the case of 2- or 1-person teams). On-time submission of your reports will be worth **5 points in total**.

- **Thursday, March 14**:
  Describe five ideas. For each member, list assigned tasks for the following week.
- **Friday, March 22**:
  Describe tasks accomplished by each team member. For each member, list assigned tasks for the next week. **Include scans or pictures of three to five hand-drawn designs or storyboards showing different project designs**. Take time with these, as more design thinking leads to easier and better coding. There was a strong correlation last year between careful design sketches and high grades.
- **Tuesday, April 9**:
  Describe tasks accomplished by each team member. For each member, list assigned tasks for the final submission. Have a working prototype reading in time for demo day.

Regarding grading:
This is an **open-ended assignment**. With homework we have a specific idea of what we want and we "take off" points when your work deviates from that. The reason project experience is the single most valuable asset you can bring to a new job is that we do *not* have specific ideas about what projects should look like: **it's up to you and your teammates**. As a result, think of the criteria below as an opportunity to "earn" points, not "lose" points. Our principle with projects is that better work should get better grades. That does not mean that we curve: there's no reason we wouldn't in theory give everyone 100s, but in practice "perfect" grades are extremely rare for this assignment.

Regarding teamwork and conflicts:
If you have concerns about how your team is working contact your TA individually as soon as possible. In rare circumstances we will differentiate grades with a group, but we are much happier to help a group succeed. Our options will be limited if you reach out to us only a few days before the final deadline.

Best practices:

- Start now.
- Talk to each other. Listen and value each other's different perspectives.
- If a group member will be unavailable for any period during the project, figure out in advance how you will work around that absence.
- Your ZIP file should contain a directory called project1. The web page should be in this directory and should be called index.html.
- Use relative paths for data, images, and other resources: do not start URLs with "/". Your project will be one directory among many, not the document root.
- Set up a code repository, like Github or Bitbucket, early in the process and regularly make use of it. This provides you with backups, an easy mechanism for accountability in team contributions, and a way for your undergraduate mentor to check in on your progress.
- Did I mention that you should start now?
- No, seriously, start now.

Grading criteria:
Your final submission has two parts, a **d3-based static data visualization (60 pts),** a **written description of your visualization (30 pts),** an **outline of team-member contributions to the project (5 pts)**. Turn in a .zip archive containing:

1. **An HTML page called index.html containing your visualization.** Include any additional script files (such as d3 or jQuery) and any additional data files, preferably in JSON form. You may import the d3 library from d3js.org, but all other libraries and data files must be contained in the zip. We will look for the following elements. You can show your TA a prototype at any time. This section will be graded on the following elements:

    a) **Complexity of the data**. Find a dataset that is manageable, however you ought to avoid trivial data. There should be more than two variables, for example. An advanced project combines multiple datasets to provide a unique, novel perspective. Editing is important, and be sure to preprocess your data! Beginning projects often have too little data or too much. Don't overwhelm us. **You must integrate at least two data sources in this project.** Advanced projects will deliver a seamless presentation of data, while beginning projects will show obvious splits in the interface between data sources.

    b) **Interactivity**. Advanced projects will provide clear, intuitive tools for exploring a complex data set. Each view should have an appropriate amount of information -- not too much, not too little. Projects that use motion or transitions to highlight contrasts and similarities are encouraged (though be careful with overuse of animation). Beginning projects might only add tooltips or similar descriptive elements to a fundamentally static interface. Do not rely solely on mouseover interactions – they are very hard to users to discover and often go unused. If you add an interactive affordance, signal to the user that it is there.

    c) **Technical correctness**. The code must actually do what you intend it to do. We also prefer good style in coding: use informative variable names, consistent indenting and whitespace, and informative comments.

    d) **Creativity**. Beginning projects often look like online examples or things we've seen before. Advanced projects will make us think "how did they do that?" or use something familiar in an unfamiliar way. Don't be boring.

    e) **Proper use of visual channels**. Use scales such as position, shape, color, and text appropriately for variables. Advanced projects give us accurate impressions of the underlying data values, allow us to make comparisons between relevant data points, and balance between focus and context. Beginning projects are often hard to interpret and make comparisons difficult.

    f) **Usability**. Someone viewing your work should be able to understand the data values represented in the visualization easily and accurately. Advanced projects make choices that are clear and intuitive and may walk us through specific examples. Beginning

projects often leave us wondering what we're looking at or make us read long descriptive paragraphs to figure out what's going on.

g) **Overall polish**. Beginning projects will look like a collection of parts, with default styles. Advanced projects will have a sense of unity, even if they have multiple sections.

h) **Motivation**. What's the point? What are you trying to say? Beginning projects will present information. More advanced projects will have a clear argument and use carefully chosen combinations of marks and channels to guide our attention to the evidence that supports that argument. Advanced projects deliver insights.

2. **A PDF file containing a written description of your project**. There are no specific page or word limits. This document should contain:

   a) **A description of the data**. Report where you got the data. Describe the variables. If you had to reformat the data or filter it in any way, provide enough details that someone could repeat your results. If you combined multiple datasets, specify how you integrated them. Mention any additional data that you used, such as shape files for maps. **Editing is important! You are not required to use every part of the dataset.** Selectively choosing a subset can improve usability. Describe any criteria you used for data selection.

   b) **An overview of your visual design rationale.** A good rule of thumb to follow is "**every pixel must be justified."** Instead of a 100,000-element breakdown, give us an overview of the design decisions you made and the trade-offs inherent in how you displayed the data. **This part ought to include a description of the mapping from data to visual elements**. Describe marks and channels you employ such as position, color, or shape. Mention any transformations you performed, such as log scales.

   c) **An overview of your interactive elements and their design rationale**. Give us an outline of the design decisions that went into the interaction affordances you added to your visualization. What process did you use to choose the interactions you developed? How did you make them discoverable, usable, and interesting?

   d) **The story**. What does your visualization tell us? What was surprising about it? What insights do you want to convey to the viewer of your visualization?

3. At the end of your PDF file, include an **outline of team contributions to the project**. Identify how work was broken down in the group and **explain each group member's contributions to the project**. Give a rough breakdown of how much time you spent developing and which parts of the project took the most time.