Project 1: A world of data

We are going to create an interactive visualization application to help a general audience explore data about countries in the world.

The Data

You can find data for this project at Our World in Data: **https://ourworldindata.org/search** **(https://ourworldindata.org/search)**
Our World in Data has collected country-level data on poverty, health, education, energy, food, politics and more. You can explore the data through articles and visualizations in the link above.
Data pre-processing:
You will likely need to do some data pre-processing, using tools of your choice.
You will likely need to combine downloads, which requires pre-processing. We will be able to use both datasets because they share a common identifier for counties called 'FIPS' code. Is is the first column in the datasets.

Visualization goals:

We want to help a general audience understand and explore world data related to an issue or set of issues that you care about. For this project, you can pick a theme- such as 'exploring poverty and education', or 'understanding population change', or 'how do health and poverty intersect'. The theme is up to you and your interests. Spend some time exploring the data, and think about what might be an interesting theme to explore.

Project setup:

- This is an individual project. You may work with others to debug issues in your code or get suggestions. However, you must submit your own work. The material presented in the tutorials will give you a starting point.
- You will create an interactive web application using d3, javascript, html and css. This web application should be shared publicly on the web, using a hosting tool like Vercel.
- Your application will feature multiple views of the data.
- Code must be managed on your github account. Regular git commits + pushes to the repository are good practice. It also helps limit academic dishonesty. We are requiring students to regularly commit+push to their repository. Every feature you implement or problem you solve- please commit and push to the repo with a sensible comment.

Project deadlines:

The **code and published application** will be due **Tues Feb 24th 11:59pm**
Presentations for the project will be scheduled for: **Wed Feb 25th**
The **documentation** for your project will be due: **Thu Feb 26th 11:59pm**
We can't push these deadlines- the other projects would get compressed. Note 10% penalty per day late for each component.
We will have weekly check-in's on the project. This is to motivate you to make progress every week, so you aren't finishing the project at the last minute. A good pace would be to target **1 weeks working on the Level 1 and 2 goals, 1 week working on the Level 3 and 4 goals, and 1 week working on the Level 5 and 6 goals.**

Project grading breakdown:

App: 75%, Documentation 20%, Presentation 5%
Projects are grading holistically. We look at the overall quality of the application AND the richness of features. I strongly suggest aiming to submit an excellent project, even if it has fewer features - fewer levels as described below- instead of trying to pack everything in, even if it does not work well.
If you tried something, but couldn't get it working on-time, record your attempts and share in the documentation. We like to reward you for trying, even if you don't get all the way there. You do not need to include broken/not working things in your application for us to see this evidence of your efforts.

Project requirements:

Level 1 goals: Do these first:

- Create a visualization application using d3, javascript, html and css. One this page include a title for the project and you name. Indicate where the data came from.
- Select 2 quantitative (aka numbers) country-level data measures that interest you (e.g., poverty rate and edu - any 2, pick based on your theme above.)
  - Note - some data attributes have a time varying aspect (for example, a measure of unemployment in different years. Below, we have additional visualization design ideas that integrate time-varying data attributes, so you may want to consider this in your selection. For now, choose the most current year that is available for both attributes and indicate somewhere on the page what years these are.
  - Note- exclude any categorical or binary data attributes you find for now
- Download csv files from Our World In Data for your chosen attributes. You may want to preprocess your data so that your selected attributes are in one file. You can do this outside of your javascript project, using a language of your choice, such as python, or using Excel.
- **Visualization 1 and 2:** The user should be able to look at distributions for your your chosen attributes. Histograms or bar charts are a good approach here.
- **Visualization 3:** The user should be able to examine the correlation between these attributes - do counties that have a high value for one attribute also have a high value for the other? A scatterplot is a good option here.

Level 2: Do level 1 first, then do this:

- **Visualization 4 (and possibly 5):** The user should be able to understand how these attributes are distributed spatially in the world and compare them to each other. One solution is two choropleth maps, side-by-side. Or perhaps you want to do one choropleth map, and have a button so the user can toggle between attributes.
  - Tutorial on choropleth maps: **Tutorial 10: Choropleth Maps (https://docs.google.com/document/d/1FvAbNkAY6c80w7Swo_H7LYIIAgR6F1XTPYyzLUxxz8w/edit?tab=t.0)**
  - Choropleth shape data for the world: **https://raw.githubusercontent.com/holtzy/D3-graph-gallery/master/DATA/world.geojson (https://raw.githubusercontent.com/holtzy/D3-graph-gallery/master/DATA/world.geojson)**
  - Choropleth map example 1: **https://github.com/UBC-InfoVis/2021-436V-examples/tree/master/d3-choropleth-map (https://github.com/UBC-InfoVis/2021-436V-examples/tree/master/d3-choropleth-map)**
  - Choropleth map example 2: **https://d3-graph-gallery.com/graph/choropleth_basic.html (https://d3-graph-gallery.com/graph/choropleth_basic.html)**
- At this point you will need to think about how you are using color. We will have a class dedicated to color choices

in visualizations. Use this content to inform your choices. What is an appropriate color scheme? The reasoning behind your selection should be explained in your documentation.
- At this point you will need to think about how you are laying out your visualizations: Before coding, draw sketches of these visualizations, and decide how you want to lay them out on the screen. The layout should be intentional. Think about what would best allow the user to view the data in these different visualizations and compare attributes comfortably.
  - To enable comparisons, you should aim to have related visualizations for the two attributes near each other in some form. If you have to scroll between views, you can't very easily compare.
  - I believe most of my projects can be accomplished in a dashboard- one page view, no scrolling- layout. There are benefits to this layout- all views are consistently visible. But it is up to you.

Level 3 goals: Complete all the Level 1 and 2 goals, and do them well. Then:

- Expand beyond 2 data attributes: The user should be able to select from multiple attributes or measures from the dataset- using your theme as a guide-, and visualizations should update to show the new data. Here also, stick with quantitative measures for now.
  - Maybe you want to add additional economic indicators or health indicators? Choose attributes that fit a theme.
- There should be clear guidance for the user on how to change the attributes displayed in the visualizations. This could be UI elements or input actions from the user. Consider how to communicate these actions to the user. I suggest positioning buttons and controls in a logical location on the screen and labeling them clearly.
- For every data attribute you now show, use color appropriately. The reasoning behind your selection should be explained in your documentation.

Level 4 goals: Complete all the Level 1, 2 and 3 goals, and do them well. Then:

- Add detail-on-demand interactions to the map, showing information about the selected county.
- Add detail-on-demand interactions to the distribution visualizations. Showing the value and range of the selected bar.
- Add detail-on-demand interactions to the correlation visualizations. Showing information about the selected county.

Level 5 goals: Complete Level 1, 2, 3 and 4 goals, and do them well. Then:

- The user should be able to brush within the distribution visualizations, selecting a set of counties. All the visualizations in your application should update to either highlight the selected counties or filter to only show the selected counties. Decide which approach (filter or highlight) will help the user best see the brushed selection. You also need to decide whether to adjust the scales, based on the selection, or keep them the same. Justify your choices in the documentation.
- The user should be able to brush within the scatterplot, selecting a set of counties. All the visualizations in your application should update to only show the selected counties. As above, decide which approach to take- filter, highlight. Decide what to do about the scales. Justify your choices in the documentation.
- This is where it is helpful to have visualizations in a dashboard configuration- visible without scrolling- so that interactions update views on screen.

From here, you can choose your own adventure:

Choose your own adventure- complete the above levels and then choose any of these options:

- The user should be able to brush within the map, selecting a set of countries. All the visualizations in your application should update to show only the selected countries. As above, decide which approach to take- filter, highlight. Decide what to do about the scales. Justify your choices in the documentation. Note- historically this has been hard to do. But I believe in you!

- Show time-varying data:
  - In a new visualization. There are some data points for different time periods. You could make an additional visualization to show changes over time, such as a multi-line chart. This may be a lot of lines, so consider your representation carefully- perhaps you want to group countries in some way (grouping into high/medium/low categories for an attribute or grouping by region). Or perhaps you allow the user to select one or two countries to focus on. Use this data to make a visualization to show trends or change over time. This visualization should fit in with brushing-linking interactions that you accomplish for the goals above.
  - In another way, such as allowing the user to select a year, and see the data update for the chosen year.
- Do something else, that is interesting and valuable for your visualization application. A novel visualization type or approach for this data.

Documentation:

This is written documentation, posted on your portfolio site to describe your project. For documentation: assume that someone is encountering your project for the first time.

- Explain the motivation for your application. What can it allow someone to understand?
- 1 section on the data: Describe the data and include a link.
- 1 section on any sketches that you used to help design your visualization environment.
- 1 section on the visualization components: Explain each view of the data, the GUI, etc. Explain how you can interact with your application, and how the views update in response to these interactions. Please include screenshots to illustrate, and relate these screenshots to the text.
- 1 section on what your application enables you to discover: Present some findings you arrive at with your application, including screenshots.
- 1 section on your process- what libraries did you use? How did you structure your code? How can you access it and run it? Link to your code (and the live application, if it is deployed online).
- 1 section on challenges and future work: perhaps you struggled with the project and learned a lot, but maybe there wasn't quite time to create the project you hoped to have. It may feel like you don't have as much to write in your documentation, so you can include a 'future works' section where you describe what you wanted to do or a 'challenges' section, where you describe what technical difficulties you encountered while doing this project, and reflect on how you might approach future projects.
- 1 section on use of AI and collaboration with other students- Did you use AI? I want to learn from you how you used it. Note any peers who were helpful in debugging or learning as you worked on your project.
- Include a 2-3 minute demo video, showing your application in action. The easiest way to record this is with a screen capture tool, which also captures audio- such as Quicktime. Use a voiceover or video captions to explain your application. Demo videos should be sufficient on their own, but can reference your documentation. Include the name of the project, your name, the project components, and how your application works. You can present it on your webpage or on youtube, but linked on your webpage.

Presentation:

For this presentation, you will present your project to a small group in 5 minutes, using whatever presentation format you would like.

Notes, Suggestions, Details:

- Your page should be easy to understand and look professional, with appropriate use of labels, fonts, UI elements, colors and layouts. Usability will help your user explore the data. If your UI is not easy to use, it will be difficult for your user to understand the data.
- UI elements (check boxes, buttons, drop down menus, sliders) can be html UI elements. You can use libraries like Bootstrap. W3 schools is an excellent resource here: **https://www.w3schools.com/html/default.asp. (https://www.w3schools.com/html/default.asp)**
- Visualizations should be presented in a coherent layout, and sized appropriately. See the tutorial on layouts using css.
  - Wherever possible, a viewer using a modern laptop should be able to see related visualizations in one view, without scrolling or toggling between tabs.
  - A responsive layout- where elements resize when a page is small/big- it nice. However, you do not need to create a 'responsive layout' for this project. You can assume a certain page size. Note in documentation what this is.
- All visualizations should have appropriate titles, legends and labels, and these should be legible.
- Colors should be used to highlight the data, and should be chosen with care. Some guidance:
  - Use color to highlight your data. Unless you are very confident in your choices, it is easiest to minimize use of color elsewhere. For instance- I would suggest neutral background colors, such as white, grey or black.
  - Do not use sold red (255,0,0), green (0,255,0), blue (0,0,255) (or solid cyan, yellow, magenta, etc).
  - If you are using color to encode your data, look to d3 color scales or color brewer gradients .
    - **https://github.com/d3/d3-scale-chromaticLinks to an external site. (https://github.com/d3/d3-scale-chromatic)**
    - **https://colorbrewer2.org/#type=sequential&scheme=BuGn&n=3Links to an external site. (https://colorbrewer2.org/#type=sequential&scheme=BuGn&n=3)**
  - Otherwise, a color like steel blue is a safe choice for your visualizations.
    - **https://www.htmlcsscolor.com/hex/4682B4Links to an external site. (https://www.htmlcsscolor.com/hex/4682B4)**
  - More on colors in data visualization to follow in class
- You application will be tested in Chrome, through the public deployment.
- Documentation (descriptions, videos, screenshots) should be presented publicly on the portfolio page you created for homework previously.
  - There will be a deduction if documentation content is not presented publicly on the web.
  - There will be a deduction if documentation is not accessible to us when grading (e.g., if there is a private setting on images or videos).