**H517 Visualization Design, Analysis,  
& Evaluation**

 IUPUI

**Project 1**

Due: Tuesday, October 7 at 5:59 PM

*This is an individual assignment; While you may discuss the project with your peers, the work that submit must be yours.*

Dr. John Snow's map of [London's 1854 cholera epidemic](https://en.wikipedia.org/wiki/1854_Broad_Street_cholera_outbreak) was a true master piece. It was the first time that anyone had established a link between contaminated water and cholera. The visualization provided compelling evidence and made the case for public health policies to improve water sanitation. The map remains one of the most remarkable and insightful visualizations ever created.

In this project, you will recreate Dr. John Snow's map using today's tools of trade (D3, to be specific). In addition to recreating the original map in an interactive version, you will add additional charts to the side of the map, as well as interface elements to make the data more interactive.

You will be provided with several data files. These include a vector map of the area and the locations of the victims and a set of dates reflecting the number of victims relative to the beginning of the epidemic. I modified the format a bit and added in a randomized sex and age of the victims based on data from the Naples cholera outbreak in 1884.

The data files include:

* [streets.json](https://khreda.com/teaching/2020/H517/project1/cholera_data/streets.json) - gives the coordinates of the streets as a series of line segments. The file is formatted as a JSON array or arrays; The higer-level array is essentailly a list of all street segments. Each street segment is, in turn, an array containing X,Y tuples represnting vertices of the segment.
* [
* [
* // segment 1
* {x1,y1},
* {x2,y2},
* {x3,y3}
* ],
* [
* // segment 2
* {x1,y1},
* {x2,y2}
* ],
* ...
* ]

Here's how to visualize the street map: draw a series of lines connecting {x1,y1}, {x2,y2}, and {x3,y3}. This would give you one street segment. To draw the entire map, you would (individually) draw all of the segments in the file.

* [pumps.csv](https://khreda.com/teaching/2020/H517/project1/cholera_data/pumps.csv) - gives the locations of the 13 nearby pumps
* [deathdays.csv](https://khreda.com/teaching/2020/H517/project1/cholera_data/deathdays.csv) - gives the number of deaths for each day of the outbreak
* [deaths\_age\_sex.csv](https://khreda.com/teaching/2020/H517/project1/cholera_data/deaths_age_sex.csv) - gives the location of each death in the deathdays file, in order, plus information on the age and sex of the victim.

age 0-10 0

11-20 1

21-40 2

41-60 3

61-80 4

> 80 5

male 0, Female 1

Another piece of potentially useful information is the percentage of people in each age group in general during that time, as the percentages of people of different ages who died may not be the same as the general population. I haven't found any data for London at that time, but US census data from 1850 gives the following breakdown, so we will assume the same breakdown for London:

0-10 28%

11-20 25%

21-40 30%

41-60 13%

61-80 4%

> 80 1%

You will be writing your code to run on a modern Chrome browser. The application should show multiple linked visualizations of the data simultaneously. This will allow the user to dynamically filter based on age, sex, date of death, and see that filtering applied to all of the current visualizations at the same time. Your visualization should help someone (like John Snow) conduct an investigation.

For a C you need:

* A well drawn map of the area (streets and pumps) including the locations of the deaths
* a timeline graph showing the number of deaths per day
* ability to move the mouse over the timeline graph to choose which day to visualize - this should affect the deaths visible on the map and in the graph - at minimum all cholera deaths prior to that date should be shown on the map and the graph
* the graph and the map should be stable (i.e., they should not move when new data is added or removed)
* the graph and the map should be well labeled
* you should check your visualization with a color blindness simulation tool to see that its OK
* the map and graph should update quickly when the user interacts

an about screen with details on who wrote the project, where the data came from, etc.

For a B you need to add:

* more data to the map - major street names and the location of the brewery and the work house
* ability to easily see on the map whether each victim was male or female
* ability to easily see on the map the age of each victim
* additional graphs showing the distribution of deaths by sex and age overall

For an A you need to add:

* ability to cluster the data on the map into a grid that shows the number of dead in each grid cell (even with this small amount of data the points are starting to overwhelm) and the user should be able to vary the size of the clusters.
* ability to show data from a window of days on the timeline graph
* ability to zoom in/out and pan around the map

Tips on how to get started

**Start early!** I cannot emphasize this enough. Designing and implementing visualizations is an inherently iterative process, and you will likely want to revise your design a few times (whether on paper or in code). By starting early, you allow enough time to experiment with different design ideas, and ultimately arrive at a better design. It is also important to note that 'getting it to work' is just a prerequisite to using the visualization to find answers to your questions, or to discover interesting facts about the data. It is that usage that will give you ideas on how to improve your visualization, and how to make it easier and more intuitive.

I suggest that you start by getting D3 installed, running through some examples, and doing some initial tests to load in the data and start displaying it. Once you have a basic shell visualization working, you may want to draw some sketches (with pencil and paper) of what the visualization and interface might look like, and how you want to arrange and display the data. You can use other software to generate statistics about the data if you find that useful but be sure to document that process. Be careful of missing data when you generate statistics.

Documentation and deliverables

For your project deliverables, you will create a set of public web pages:

* **1 page for the visualization itself:** This should be a live, interactive version of the visualization you created. It is important that the visualization works in the Chrome web browser - this is the platform that we will be using for evaluating your work.
* **1 documentation page** that describes your work and documents the following:
  + Your design process: How did you go about designing the visualization? What are some of the initial designs / ideas you attempted in the beginning? A good way to document your design process is to scan your sketches and include them in the documentation page.
  + Rationale of your design choices: This should be a rigorous explanation of the design choices you made. For example, why did you use color to encode a particular variable? Why did you arrange your charts in a particular way?
  + Describe how you used your visualizations to discover facts or answer questions you had. Include evidence to support your findings as screenshots from the visualization. In this case, we have a clear hypothesis to start with, but are there other nuggets of insights one can uncover?
* **A 2-4 minutes YouTube video** showing the use of your visualization with narration. This video should be embedded in your documentation page. The video should be created using a screen-capture tool while interacting with the visualization. It should ideally be similar in style to the videos that accompany papers at the IEEE VIS or ACM CHI conferences. Here are a couple of good examples: [UpSet](https://www.youtube.com/watch?v=-IfF2wGw7Qk), and [GPLOM](https://www.youtube.com/watch?v=AtbIeCvB-E0)

The web pages should be live online, and your submission should consist of a single URL. The URL should open with the visualization itself. However, there should be a **clearly visible link** to the documentation page and the YouTube video.