Carter Sifferman

DC2 Phase 2: Project Proposal

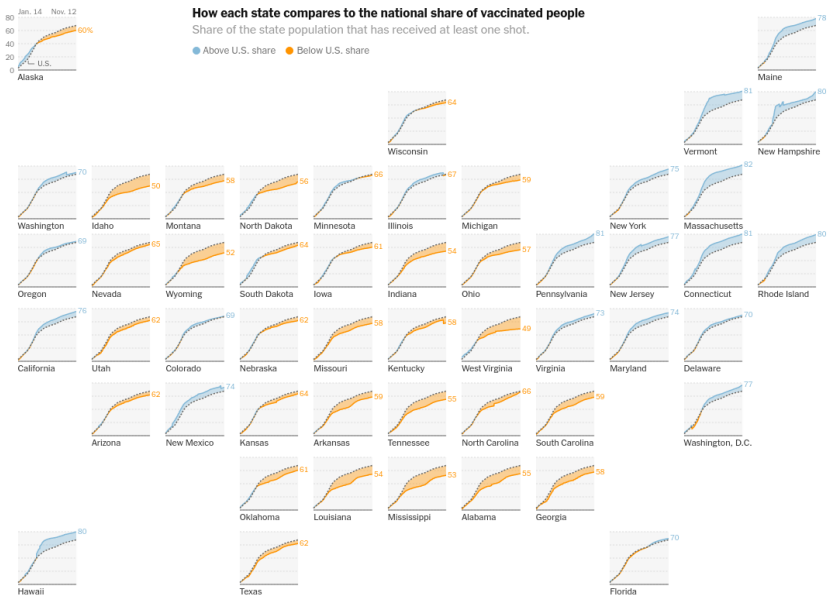
Changes in red. Main changes:

- Less specific about dataset
- Switched plan from D3 to recharts
- Changed evaluation plan slightly

Problem

I will focus on the problem of dynamically scaling stacked bar charts, especially how to preserve readability as the charts get small (<150px). As a “case study” of my solution, I will consider using a small version of the plot to represent some state-wide data, and arrange the plots to create a “map” of the 50 U.S. states. This was inspired by the below visualization from the New York Times. My “map” will, of course, be made of stacked bar charts rather than line charts.

The “map” presents a new question, which is, “when should we switch from displaying a “map” of 50 small charts to a single map chart?” This is a question that, as far as I can tell, hasn’t been addressed in the literature. Rather than just adding and removing elements from individual charts, I hope to combine many charts into one at small scales.



Solution

Design

Carter Sifferman

I plan to create two designs:

- A dynamically scaling stacked bar chart, similar to the one shown off in Semantic Resizing of Charts Through Generalization: A Case Study with Line Charts.This chart will contain an example set of data and be re-sizeable with a slider
- A "map", similar to the New York Times map above, but made of stacked bar charts and representing states' partisan vote share in the 2020 presidential election. This map will not be able to be dynamically resized, instead I will manually render it at different scales using the automatically resizing stacked bar chart.

Implementation

I plan to implement both of the above designs in an interactive web application which is deployed with GitHub pages to my personal website. I hope to make the application completely static so there's no need to connect a server. To accomplish this, I will use the JavaScript visualization library recharts, which is built on top of D3 and React. Compared to D3, recharts is easier to get started with, and because it includes many pre-built charts, the default charts can be used as a baseline for comparison.

Plan

Before Phase 3: Formulation (12/1)

- Get data in right format and reduce to what I need
- Be able to display the data with a default stacked bar chart in D3
  - If D3 proves to be more intimidating than I expected, I can decide to fall back to matplotlib, which I have a lot of experience with, at this point
- Have an idea of how to modify labels, chart properties, size

Before Phase 4: Initial Results (12/6)

- Be able to scale the bounds of a chart with the slider
- Make decisions about how to change chart elements as scale changes
- Investigate creating "map" of multiple charts
  - If this proves to be too difficult to do natively in the browser, I'll consider doing it just by combining exported PNG images

Before Phase 5: Final Handin (12/13)

- Finalize dynamically scaling chart with slider
- Have "map" done (lower priority)
- Deploy
  - If a web server is needed, I can submit a GitHub repository with instructions to run a local web server, rather than deploying to my GitHub pages
- Write report

- Create video demo

Evaluation

My project will certainly be successful if I'm able to make a stacked bar chart version of the country map that provides some benefit over a baseline encoding, which would be just color-encoding a map of the U.S.

If a map proves impractical, I will focus instead on improving the visualization that scales dynamically with a slider by including versions with different and more difficult to visualize data sets. Success in this case would mean my solution is capable of reasonable dynamic scaling for every data set. At the very least, my solution should be better (more readable) than the default recharts implementation in some way at some scale.

In my report, I plan to include side-by-side comparisons between my plot and the default recharts implementation as a baseline, at various scales and with multiple datasets. I will also include tasks which the reader can attempt on their own with both visualizations, to make their own comparisons.