An interactive visualization of COVID-19 Time Series Data Using Flask/d3.js

By: Kollin Trujillo

Description of the Problem

- COVID-19 is a disease responsible by the novel coronavirus. In March 2020, we started into a pandemic and the US started collecting data related to deaths on a per-state basis.

Data Source

- https://data.cdc.gov/NCHS/Weekly-Provisional-Counts-of-Deaths-by-State-and-S/muzy-jte6
  - Can access the data source by download or via the SODA (Socrata Open Data API (SODA)) API
- Usually data is recorded on a weekly basis.
  - Some states are not as consistent with reporting.

Data Structure

- Consists originally of 35 columns of data that I trimmed down to 6.
- Data types of trimmed columns.

```
Index(['Jurisdiction of Occurrence', 'Week Ending Date', 'All Cause',
       'Natural Cause', 'COVID-19 (U071, Multiple Cause of Death)',
       'COVID-19 (U071, Underlying Cause of Death)'],
      dtype='object')
```

- We can see that there are four kinds of deaths reported in the data. I am interested in looking only at “COVID-19 (U071, Underlying Cause of Death)” values.

How to Process the Data

1. Load in the script with pd.read_csv.
2. Convert ‘Week Ending Date’ using pd.to_datetime to a datetime format.
3. Set index to ‘Week Ending Date’
4. Group the columns month and year and sum to get monthly aggregates.
5. Reset index.
6. Rename columns ‘Jurisdiction of Occurrence’ to ‘state’ and ‘COVID-19 (U071, Underlying Cause of Death)’ to ‘value’.
7. Construct a dataframe of only state and value.
8. Utilize this in d3.js to construct the visualizations.

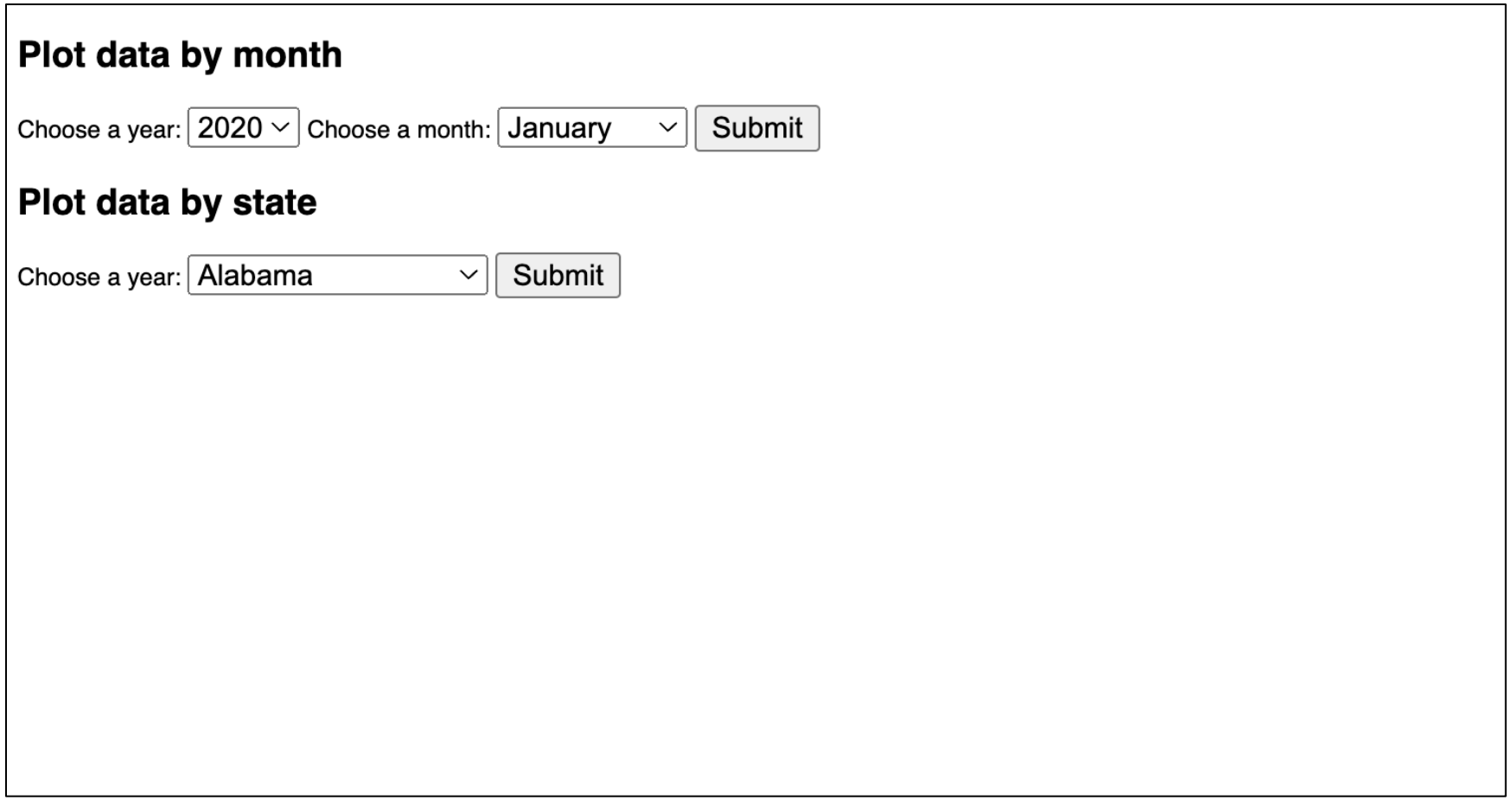
To impute or not to impute?

- Some state data does not have values for deaths.
  - In that case I decided to put 0's.
- Different ways to impute data… I will chose to do:
- Decided to not do anything. Any data with a value of 0 is typically assumed to be NaN, or have no reported data.

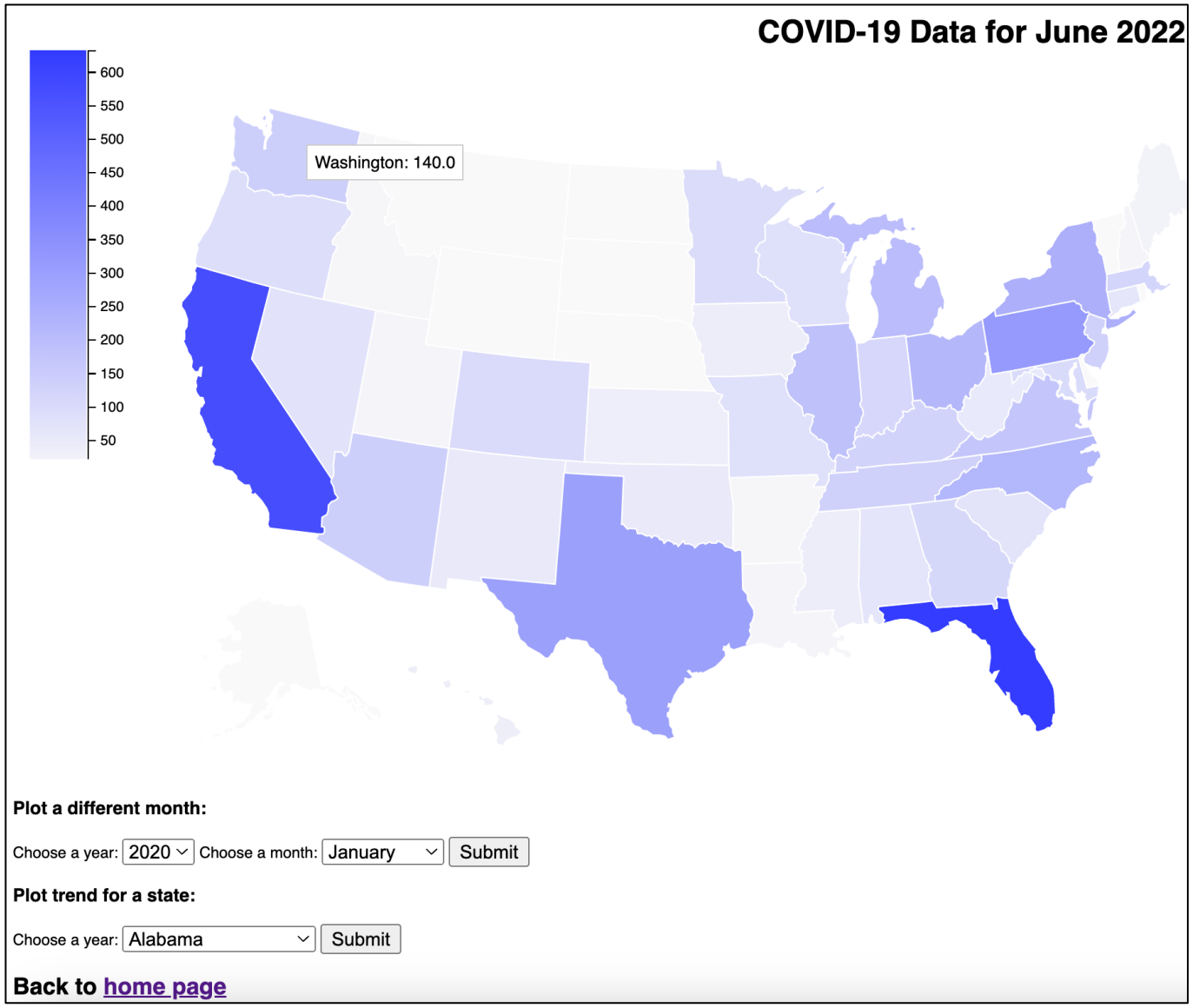
Prototyping of Data Visualization Design

- Utilized a chloropleth map and additionally a line graph.

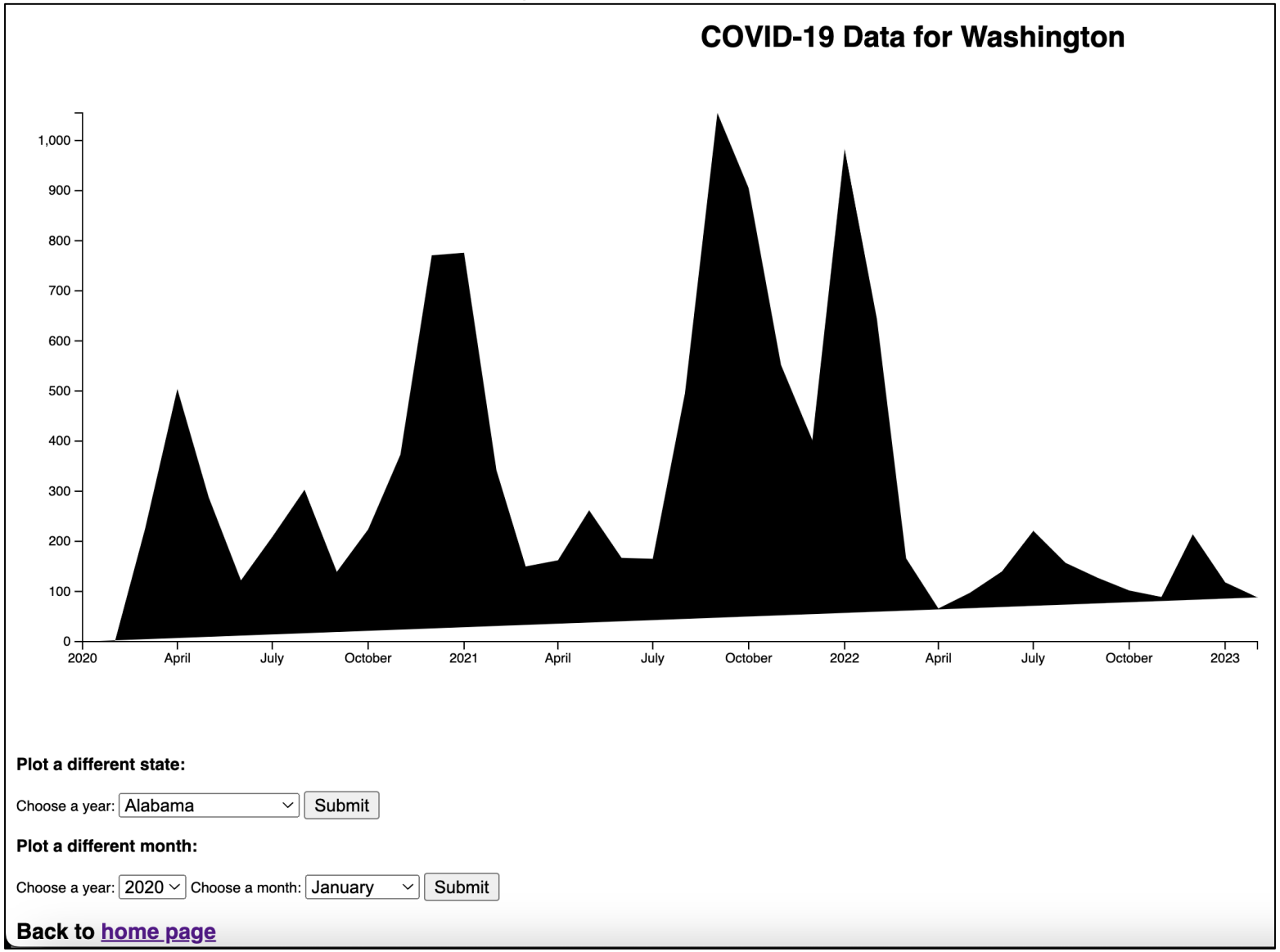
Outcomes (Front Page)



Outcomes (Choropleth)



Outcomes (Line Graph)

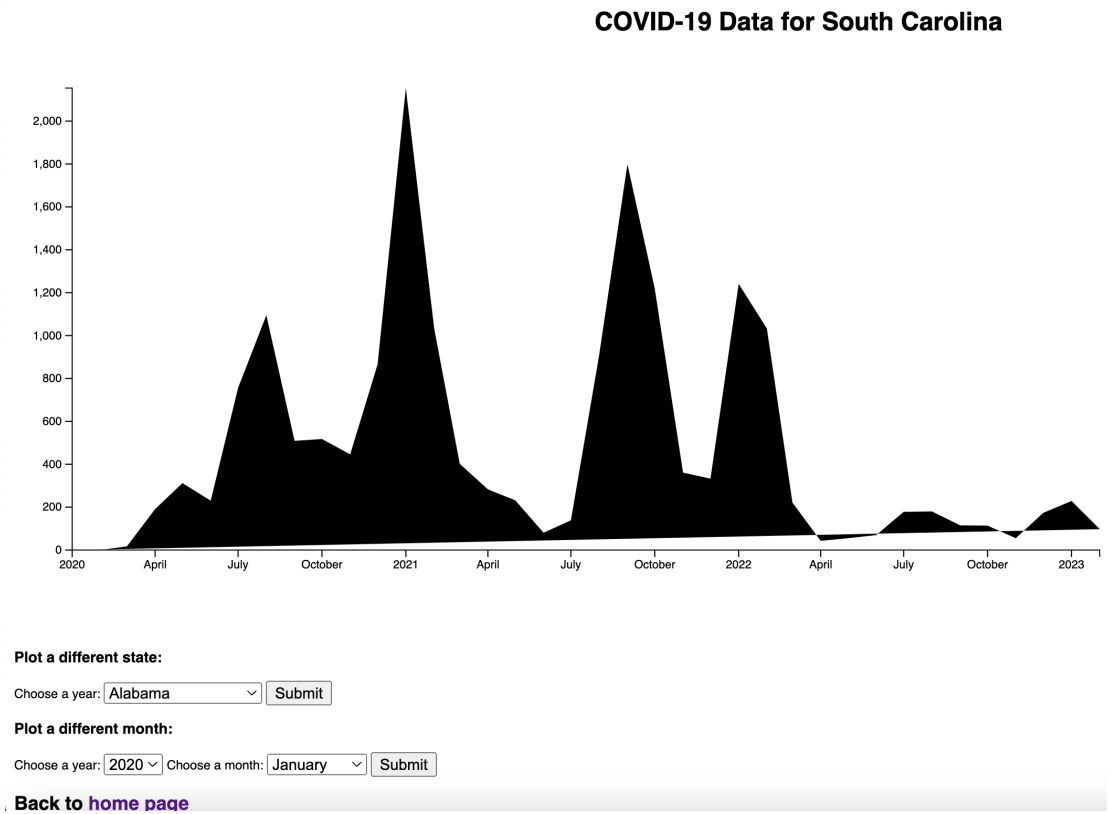


Difficulties

- Navigating the construction of communication between a server backend (Flask) and front-end (Javascript) was trickier than initially though.
  - HTTP Protocols are easy to learn but tricky to master.
- SVG graphics via d3.js are inherently finnicky. Debugging their non-cooperativeness can be difficult and tedious.
- Wanted to include animation of time series data and onClick plot the line graph visualization.

Problems

- Line graphs didn’t “reset”
  - Can see that the graphs didn’t go down to the baseline of the x-axis.



Conclusion

- Allowed for the visualization of COVID-19 data in both a “Parts” and “Whole” type of setup.

Bibliography

1. https://data.cdc.gov/NCHS/Weekly-Provisional-Counts-of-Deaths-by-State-and-S/muzy-jte6
2. https://d3-graph-gallery.com/graph/choropleth_hover_effect.html
3. https://d3-graph-gallery.com/graph/line_basic.html