HW2 Writeup

Vanessa Yin

A clear description of the goals of project

In recent years, racism related to US police shootings has become a hot ethical topic that is essential and meaningful for all people in the US. In this project, the question I am enabling users to answer is “Does US Police Shooting reflect racism?” by dynamically visualizing the US police shooting data from January 2015 to June 2020. The data is provided from https://www.kaggle.com/ahsen1330/us-police-shootings, which is a cleaned data where the raw data is also from kaggle.

A rationale for design decisions

My first thought on the approach to visualize such data is to make a barplot or pieplot to show the count or percentage of people shot by police in each state, grouped by people’s races. Although it is pretty intuitive, such a design is not sufficient enough since we didn’t consider the fact that the race population in each state can be very different. Take an extreme example here: for people shot by police in Pennsylvania, 70% of them are white people and 30% of them are black people. If we only see such a percentage, we may think there is no bias or racism against black people at all. However, assume extremely that 95% of the population in Pennsylvania are white and only 5% of the population are black. Then obviously a racism problem will be raised.

So I thought about how to present the percentage in the plot and decided to display the bias ratio = shooting percentage of people of race R at state S / population percentage of people of race R at state S. Thus, as bias ratio increases, it is more possible there exists the discrimination against R at state S. In the previous example, the bias ratio for white would be 70%/95% = ~0.74, while the bias ratio for black would be 30%/5% = ~6.

Since the bias ratio is classified by each state, I chose to use the map plot instead of a barplot or pieplot to show the visualization in a more intuitive way. I considered using only a scatter plot to show circles on map with different sizes representing how big the ratio is. However, I cannot find a library that easily let me do so, and I found a much more informative approach, which is my current approach: show two layers on the map plot, with one scatterplot layer representing each shot case, and another choropleth layer showing the ratio of each state by color (from yellow to blue, where yellow represents high ratio and blue represents low ratio). Users can mouse over the states or the red points to see further information in the mapbox. I also added race selection radio buttons to enable users to switch between races. Users can switch and observe the color change of the map easily to distinguish the ratio difference of races. For the button interactions, I also considered using filters (multiple-conditions) instead of the ratio buttons (single-condition). However, I feel people’s eyes tend to catch the change of map more easily if only one single condition runs the map. It’s also not giving much information if multiple races’ total ratio is displayed.

Finally, since the dataset also has a column “armed” to show the armed status when the person got shot. I think this is also a valuable independent variable. So I also added the ratio for that column.

An overview of development process

1. Browse data and define the question wanted to answer: 2 hours
    a. Covid19: too many data analysis/visualizations have been done
    b. US police shooting: seems good and has an appropriate scale. Question can be: Does US police shooting really reflect racism?
2. Play with Streamlit and study on interactive data visualization on map: 3 hours
    a. useful altair gallery: https://altair-viz.github.io/gallery/
    b. example interactive visualization: https://altair-viz.github.io/gallery/airport_connections.html
    c. st.pydeck_chart with layers supported by DeckGL: must include longitude and latitude. Need to convert the given city state to get longitude, latitude for each row.
3. Data cleaning and EDA:
    a. Conversion from city state to longitude, latitude by geopy.geocoder: 3 hours because it’s sending API requests. Ended up with mostly correct but some wrong longitude, latitude that is out of US.
    b. Data cleaning/simple visualization: 1 hour
4. Data visualization: 6 hours
    a. Implementation and design: 6 hours
    Details are explained in the rationale. The time consuming part is to find a library that can achieve my visualization effect. The process is pretty iterative. I tried one library, installed its environments, failed, and found another library and went over the process again and again.
    b. Polish the interface and add story-telling narrations: 3 hours
    Add introduction, explanation of the bias ratio, and the instructions of how to use the graph in the application to help users understand the question and the visualization easily.
5. Write the writeup: 1.5 hours
    Since I’m also keeping track of what I’m doing when I was building the project, it is relatively easy for me to write all the things down.

    Overall, the most time consuming part is data visualization and converting city state data to longitude and latitude.