A2: Interactive Visualization

Amindu Abeydeera, Divjot Muchhal, Nikhil Paruchuri

Introduction:

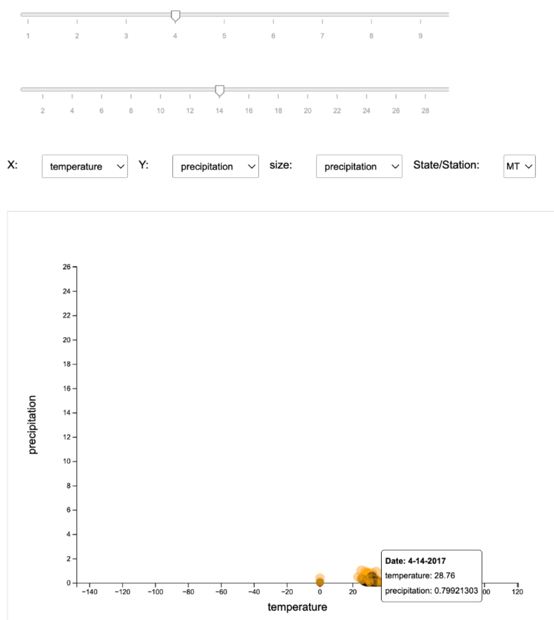
For our project, we decided to conduct a visualization on the "Daily Weather in the United States, 2017" dataset. The dataset consists of weather measurements in the United States gathered by the NOAA Daily Global Historical Climatology Network. The data contained all kinds of valuable geographical information such as station name, state, latitude, longitude, and elevation. The data spanned many weather variables including minimum (TMIN), maximum (TMAX), and average temperatures (TAVG), wind speed (AWND), wind direction (WDF5), and extreme wind speed recorded in 5 seconds (WSF5). It also traced precipitation (PRCP), snowfall (SNOW), and snow depth (SNWD). For our visualization, we chose to emphasize the impact of precipitation, temperature, wind direction, and wind speed, and how each influences each other differently in various regions of the United States. This approach allows us to uncover the relationships between regions in the United States, offering a clear understanding of how geographical location can affect weather patterns specifically during the year of 2017 in the United States. This analysis in one year could provide insight into the climate changes for the years to come.

Rationale, Design Decisions, & Data Source:

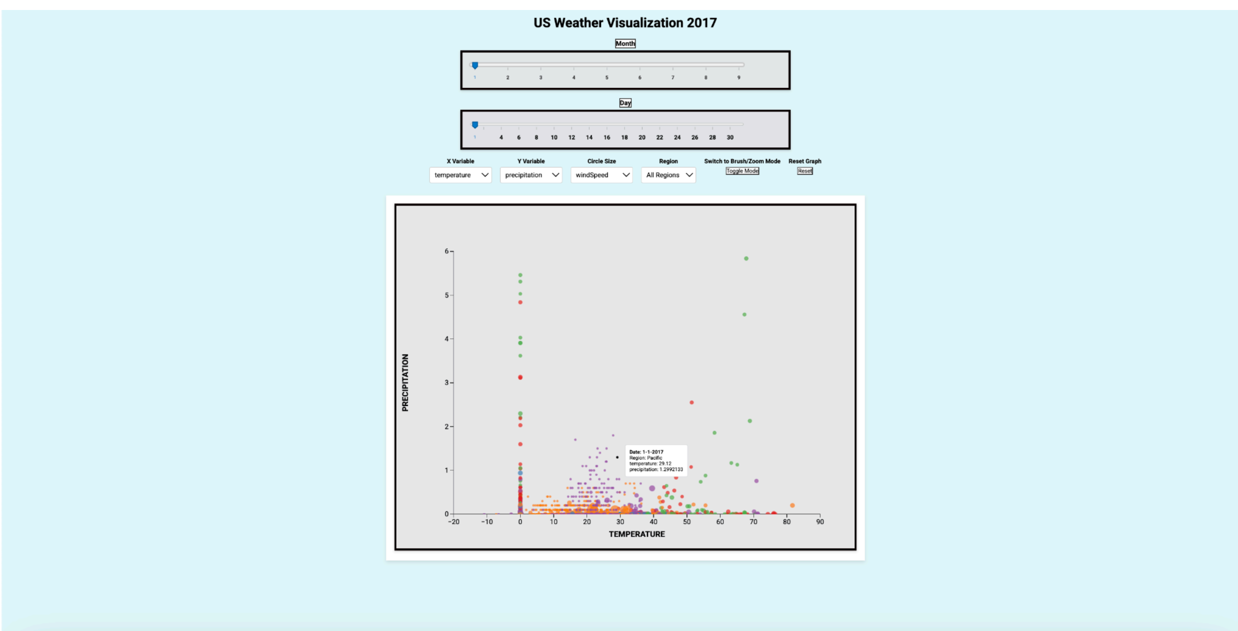
Our rationale and design for our visualization stemmed from our data source of NOAA's Daily Global Historical Climatology Network and lab 4-5 we completed last week. In lab 4-5, we saw how to apply D3 via interaction and transitions with data that could be adjusted with different parameters. Our dataset was similar to that, so we improved our visualization compared to the basic implementation from lab 4-5. We ultimately started from the inspiration from lab 4-5, but as we continued to make rational decisions about our design we used our intuition and creativity to help reveal a compelling narrative about how weather patterns evolve and vary across regions. We wanted to make sure our users could uncover and understand these various relationships through a dynamic interaction. Our data consisted of scatter plot points, and to facilitate this dynamic interaction we used sliders, drop downs, tooltips, brushing, and zooming. This design is usable and interactive to enable users to easily manipulate and navigate complex weather data. The sliders allowed users to view data over different periods over the year 2017, and the various drop-down menus allowed users to change parameters to their liking, whether it be precipitation, wind speed, wind direction, or temperature. For the metrics to measure the different variables, we used the x and y, but we also allowed users to pick the size of each of the data points, from one of the drop-down parameters, to measure the variables as well. This allowed us to compare 3 different variables against each other to find a correlation. The last drop-down allowed users to focus on what regions they wanted, either individual regions of the East Coast, Central, Midwest, and Pacific. Lastly, our design included buttons to toggle between our user preferences of either having a tooltip available or brushing, allowing users to be able to see individual point information with the tooltip, and also navigate through large clusters of data with brushing. This visualization design provided a clean, simple-to-use experience while retaining the data visually informative and interactive.

Development Process:

As far as our development process, the task was divided amongst us fairly evenly. Initially, we had 2 group members, were able to get a third member, which allowed the workload to be distributed better and relieved the stress from just 2 people. First, as a group we collectively decided to extend our deadline using our slack days. We decided this as none of us had used one yet, and all of us had multiple midterms & projects this past week, and we did not want to rush our visualization and report because of the clashing due dates. Extending our deadline allowed us to implement an amazing design and not rush any part of the process. It gave us time to debug and adjust components to our liking without the stress of the original deadline. We divided our workflow into 4 categories: creation of our visualization on Github, the write-up, demo video, and the actual implementation of our visualization. Nikhil created the GitHub for our project, and added Divjot & Amindu as collaborators. Divjot was responsible for the report, and Amindu was responsible for the demo video. All 3 of us contributed to the implementation of our visualization. Even though we divided the work into different tasks for each other, each person also took time to thoroughly read each segment of the assignment to give a second opinion and make sure key points weren't omitted and that everything is up to standards. We decided to plan what we wanted to accomplish, starting with which dataset to implement. We collectively agreed on the weather dataset from 2017. After this, Nikhil created the skeleton for our visualization. He created the initial SVG canvas, the sliders, and loaded the data from the csv into the graph for initial visualization. He also implemented the basics of the drop-down menus and our initial approach included points filtered with every single state/station. Here is the initial skeleton implementation:



This skeleton created by Nikhil consisted of functionality for all our key components, the sliders, the drop-down menu selection for our parameters, the data being displayed onto our graph, the axes, and the ability to hover over the points and display a tooltip. The functionality worked as hoped, which was very important. However, we wanted to portray our data in a meaningful way and make it visually appealing. The graph from the data didn't represent itself in a way for users to understand what was going on, especially because it was all clumped together because of so many different points. Additionally, Nikhil pointed out the axes weren't reflective of the data and the data would showcase towards the bottom as the axis intervals were not correct and some of the states did not have data so when selected from the drop-down it didn't show anything. Furthermore, there were some bugs with the hover as when the points were hovered, it stayed orange. As a group, we met during class and outside of class to discuss how to move forward with our initial skeleton. We decided to not portray data by individual states but group by timezone region, east coast, central, midwest, pacific, to allow the data to be more understandable for insight and fix an issue with null or empty data for any particular state. Since there were some non-U.S. states but U.S territories that fall under as part of the "other" category, we decided not to represent them in our drop-down. These data points did not give a big picture to our visualization as they were only 2 territories, which would not be helpful to compare against the other regions that had significantly more states. Filtering by region would make the visualization more appealing and also better for analysis. Amindu took the role to implement the filtering and mapped each state to its respective region. This is also when Amindu implemented two features of brushing and zooming in to make our visualization more dynamic and clearer to understand. As a group, we noticed how there were sometimes a lot of clusters with the data and so sometimes using the tooltip individually was hard and difficult to read and understand for our users. To combat this, we decided to implement the brushing and zooming features. This allows our users to be able to select a cluster of points and zoom into them for clearer analysis. However, this is the part where we ran into our biggest issues. We had already implemented the tooltip feature and we intended to keep that, but when the implementation of the brushing and zooming was added, it disabled our users from using the tooltips. This is not what we wanted as we wanted to have both, but now we're unsure how to implement both. This is where we decided to create a button that could be toggled to switch between tooltip mode and brushing/zooming in mode. Based on user preference, they could have tooltips on and then see the point data and if they wanted to zoom into a cluster, they could toggle the brush/zoom mode, and select the data points they wished. Once it zoomed into the selected cluster, they could switch back into tooltip mode and use the tooltip again. One additional issue encountered was that when the users were zooming in, some of the data points would go off the screen and to other points of the canvas, which did not make for an appealing visualization. We turned to external research/resources & AI to help debug this issue. Our uses of AI included this issue, any small debugs, and also ideas on how to format our CSS like background color generation. We learned about this concept of "clip-path" which allowed us to remove any graphical elements that fall outside of our defined display area. This helped combat the issue of data points being plotted all over the canvas when zoomed in. The final part was implementing CSS and making it a more appealing visualization. Divjot made this possible by changing the colors of each data point to its respective region. Each region got its color so it was clearer which point was part of which region, especially when all regions were displayed onto the graph. Furthermore, Divjot created formal and readable labels for the axes of our graphs, but also labels for the drop down menus, buttons and the sliders. The sliders were customized with sharpened tick values and the handle was also customized to be blue on the selected value. Additionally, sharp black borders on the buttons, sliders, and the svg canvas added a visually appealing look. Also, the background color of a soft blue was added to match the theme of weather. Originally, an image of the U.S. regions was the background image. but it was too distracting and did not appear well to our visualization, so a simple background color sufficed. Furthermore, the region name was added into the tooltip and tick marks were added based on the given data for user clarity and. Lastly, Divjot added a reset button to set the graph to its initial state in the event the user zooms into the graph and wants to revert the changes made. Here is our final visualization image(functionality shown in our demo video):



Overall, the assignment took time, but it was fair as we put in good effort. The initial github setup & skeleton took about 1 hour to complete, planning/discussing initial steps in and outside of class took about 30 minutes, the implementation of both the brushing & zooming features took about 1.5 hours, final CSS & layout implementation took about 1 hour, total debugging time around 30-45 minutes, the report took about 1.5 hours, and lastly the demo video, deployment, submission took about 30 minutes. Total, this project took about 5-6 hours total to complete, with 3 people working it was manageable. The features that took the most time were the brushing and zooming features, as there were bugs with them and required outside research. Another thing that took a decent amount of time was our initial planning steps on what dataset to choose and once we decided, we had to figure out which variables we would like to actually display in our visualization. There were so many different variables given to us in the dataset, as shown in the introduction, so we were a bit lost at first as to whether we would want to focus on visualizing all of the variables, or if we would want to focus on honing in on a few of the variables. After a lot of thinking, we decided it would be more beneficial to hone into a couple variables from the dataset rather than trying to analyze all of them, but the thought process behind that was one of the most time-consuming parts because there were pros and cons to both options. Other than that, we worked at a good pace, communicating progress to complete this visualization.