Global Data Visualization of Olympics Data (1924-2006)

Sirish Kumar Munipalli

Motivation

The Olympics is an international sporting event held every 4 years and the most prestigious athletic sporting event. It brings thousands of athletes from around the globe to compete in various sporting events. Since it is a global-level event, a world map can be used to visualize the data effectively.

Data augmentation (if any)

No data augmentation has been used in this project. Instead, the data has been taken in raw format and used for colorizing different aspects of the data analysis. In addition, data has been categorized for generating some stats.

Tasks

- Develop a dynamic webpage using javascript.
- Use svg world map for creating a visualization. Analyze the SVG image format and the SVG path tags for manipulating the image dynamically using javascript.
- Use the web browser localstorage to communicate data between different script tags. This localstorage will help in creating global variables for the webpage.
- Fetch the JSON data file into the javascript and parse the data.
- Using bootstrap CSS for styling the webpage
- Use google fonts for styling the webpage fonts.
- Using d3js for generating the plots

Expressiveness of design

The webpage's top starts with a heading that has been generated using the bootstrap Jumbotron. This heading effectively captures the user's attention and conveys information about the demonstrated data visualization. In addition, the subscript has a small amount of information that will tell the user to click on a country or to hover over a country for further details.

Just from a glance, the user can get a macro view of the performance in the Olympics. From a high level, the data has been grouped into their respective countries. The user can also implicitly understand the countries that have yet to participate in the Olympics without reading any text. One of the main emphases of this visualization is to acquire the information quickly through visual aid and keep the textual information minimal. For this task, I chose different shades of green. The countries that have performed well (based on the number of medals won) will be colored dark green, and with terrible performances will be colored with a lighter shade of green. And the countries that have not participated will be grayed (default disabled color while developing web apps) out.

Once the user hovers the mouse over a country, they will get first-level stats(ref. interaction). After clicking on a country, the user will be shown a detailed analysis of each sport(ref. interaction).

Effectiveness of the solution

This visualization effectively conveys the highlights within a few seconds. And then helps the user focus on parts that require detailed data analysis. In general, the user can effectively navigate through the data and understand the data well.

Interaction

I have effectively used mouse hover and click functions to interact with the visualization.

Stage - 1
As soon as the user hovers the mouse pointer over a country, they will be presented with tabular data summarizing the following items -

- Number of gold, silver, and bronze medals
- Number of medals won by each gender

This information is in a clean bootstrap modal (a popup) and will disappear as soon as the user moves the mouse away from the country.

Stage - 2
When the user clicks on a country, they will be presented with the heatmap of the number of medals won in different sports concerning the timeline (year). This heatmap effectively conveys to the user the best-performing sport over a given period in the country.
The color range of the heatmap takes into consideration all the global data. This visual cue will also help the user compare the sport with other countries even when they are not viewing the other countries' heatmaps. The user will be able to understand this implicitly with a few clicks.

Conclusions

This webpage effectively conveys the information by hiding complex analysis at first glance. Then, it slowly reveals the details as the user navigates through the visualization by interacting with it.

Data communication between different script tags has been achieved by creating custom events and dispatching them. However, loading the JSON file into the script is inefficient and uses the localstorage to get the values from ‘promise.’

Several quick hacks have been used to generate this visualization, but using javascript libraries and frameworks like ReactJS and Angular, a production-ready web app can be made.