

Bio-Diversity Dashboard

This project is a visualization Dashboard built with HyperText Markup Language (HTML)/Cascading Style Sheets (CSS) and JavaScript with the Plot.ly and D3 libraries, plotted from data in a JavaScript Object Notation (JSON) format.

Click below to view the Github pages deployment.

<https://bethanylindberg.github.io/Bio-DiversityDashboard/>

Introduction

The completed dashboard displays several visualizations of a data set containing results of a medical study of operational taxonomic units (OTUs) residing in the navels of test subjects. The dashboard is designed with a dropdown to select individuals in the study and display their details.

Dashboards like the one represented here are valuable for sharing data in an easy to understand way quickly and easily with internal and external stake holders.

This kind of dashboard can be achieved perhaps more easily by using various business intelligence tools, but building from scratch can provide more flexibility and allows for greater customization.

Description of Data

The data is accessed from data/samples.json in the Github repository. The data is divided by subjects and includes demographic data, OTU labels and OTU values. The study includes 153 individuals.

The keys included in the demographic data are below:

- ID
- Ethnicity
- Gender
- Age
- Location
- Belly Button Type (bbtype) - non-null values are either "I" or "O" to represent concave or convex navels, respectively
- Washing Frequency (wfreq) - non-null values represent the number of times the individual scrubs their navel per week

Both the keys and values are displayed in a table on the deployed dashboard.

Limitations of the Data

There are multiple fields with null values in the dataset which leaves certain factors unknown and prevents from a complete picture of every individual. There are also some locations that include only a state instead of city and state.

Additionally, the OTU labels are in a string format that does not lend to an easy read when hovering over the Plot.ly visualizations.

Finally, with a small sample set of 153 subjects it is difficult to draw any meaningful conclusions from the data.

Methodology

Plot.ly was used to plot the visualizations and D3 was used to manipulate the document object model (DOM). See below for the documentation for each library. Plotly.newplot is used in each code block to call the appropriate HTML tags for display. D3 is used to store tags, classes and IDs in variables to be used in the displayed data as well as reading the JSON file into app.js. While the bar graph and bubble graph are basic graphs included in Plot.ly functionality the gauge graph is a customized graph including the pie and scatter formats.

<https://plot.ly/javascript/>

<https://d3js.org/>

A simple front end was customized from a downloaded theme (link below). These files are stored in the assets folder in the Github repo. JavaScript functions were used to initiate the application and build the table and visualizations.

<https://www.creative-tim.com/product/light-bootstrap-dashboard>

A select tag in the HTML navigation bar header with a on change event listener calls to a JavaScript function to update the DOM every time a new ID is selected by the client.

The documents are deployed on Github Pages because it is a good option for projects with static data such as this Bio-Diversity Dashboard.

Post Mortem

The most difficult challenge for this dashboard was ensuring responsive behavior for every plot and the select dropdown. Plot.ly includes a responsive function but only the bar graph and bubble graph responded appropriately. After several ineffective experiments the gauge was successfully manipulated in the CSS sheet with an @media block.