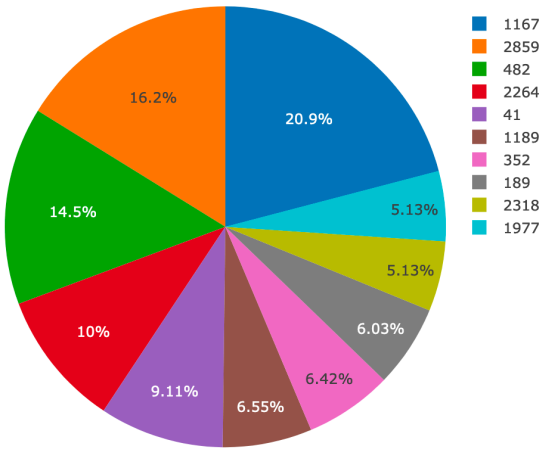
Piruz Alemi
Date: Feb 14th, 2020
Subject: Plotly.js + Interactive Dashboarding

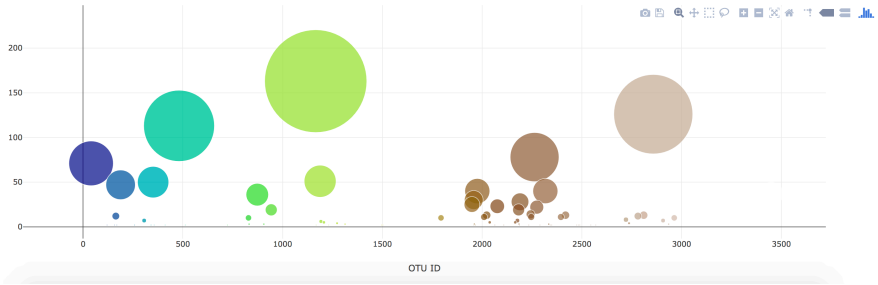
Step 1 - Plotly.js

Used Plotly.js to build an interactive chart for my dashboard.

1. Created a PIE chart that used data from my samples route (`/samples/<sample>`) to display the top 10 samples.
    - Used `sample_values` as the values for the PIE chart.
    - Used `otu_ids` as the labels for the pie chart.
    - Used `otu_labels` as the hovertext for the chart.



2. Created a Bubble Chart that used data from my samples route (`/samples/<sample>`) to display each sample.
    - Used `otu_ids` for the x values.
    - Used `sample_values` for the y values.
    - Used `sample_values` for the marker size.
    - Used `otu_ids` for the marker colors.
    - Used `otu_labels` for the text values.



3. Displayed the sample metadata from the route `/metadata/<sample>`
    - Displayed each key/value pair from the metadata JSON object somewhere on the page.
4. Updated all of the plots any time a new sample is selected.
5. You are welcome to create any layout that you would like for your dashboard. An example dashboard page might look something like the following.

Step 2 – Heroku
https://en.wikipedia.org/wiki/Heroku

Deployed my Flask app to Heroku.

- I used sqlite file for my database.

Advanced Challenge:

The following task was completely optional and quite advance. Done!!!

- Adapted the Gauge