

# Plot.ly Homework - Belly Button Biodiversity



In this assignment, you will build an interactive dashboard to explore the [Belly Button Biodiversity DataSet](#).

## Before You Begin

1. Create a new repository for this project called `plotly-challenge`. **Do not add this homework to an existing repository.**

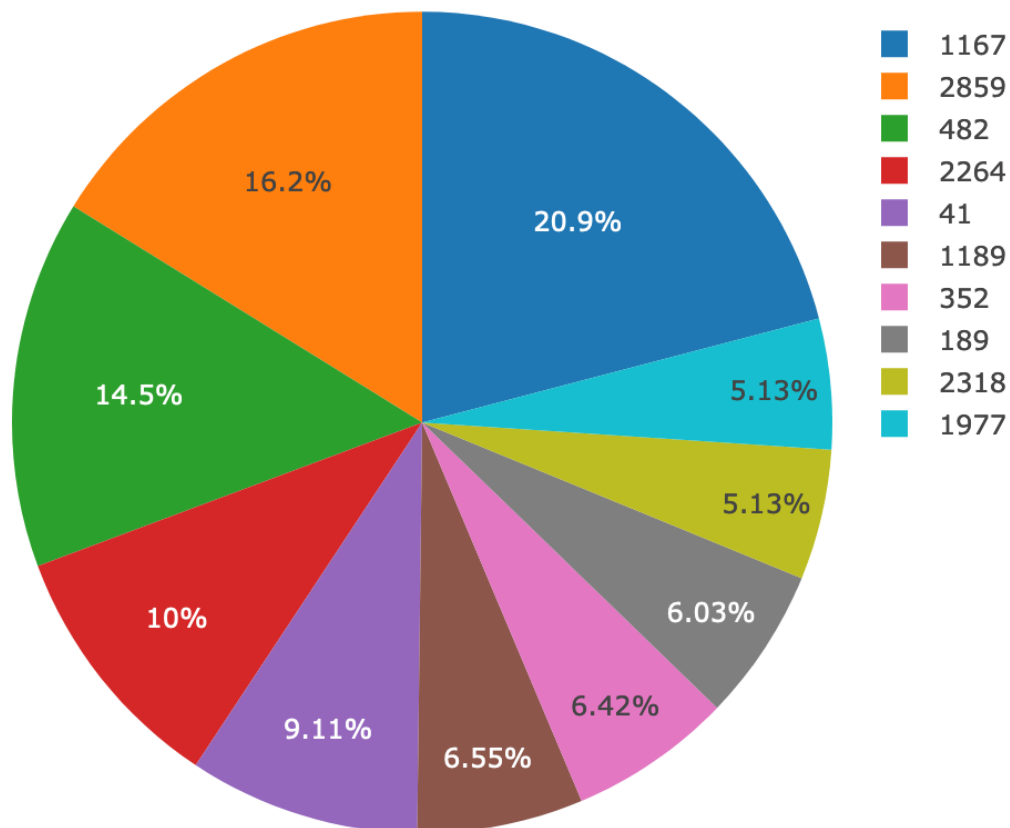
2. Clone the new repository to your computer.
3. Inside your local git repository, create a directory for the Plotly challenge. Use the folder name to correspond to the challenge: **Belly\_Button\_Diversity**.
4. This is a full stack app so add your html, js, css, python and sqlite files.
5. Push the above changes to GitHub or GitLab.

## Step 1 - Plotly.js

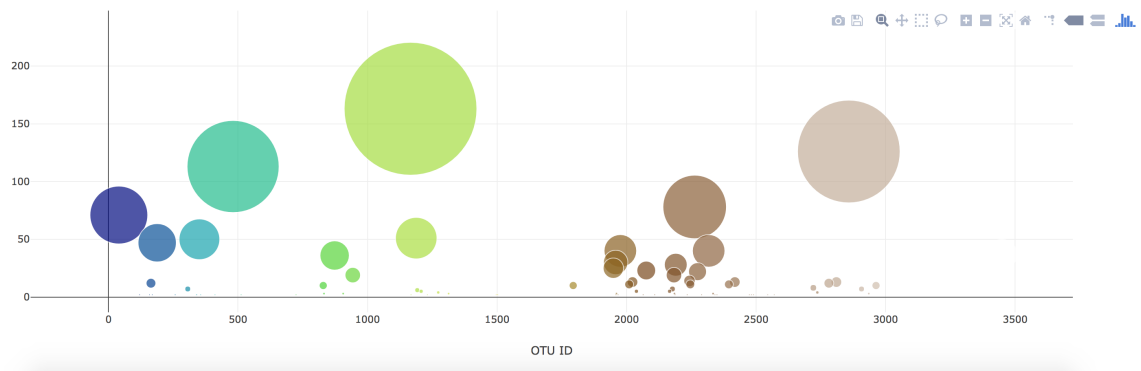
---

Use Plotly.js to build interactive charts for your dashboard.

- Create a PIE chart that uses data from your samples route ( `/samples/<sample>` ) to display the top 10 samples.
  - Use `sample_values` as the values for the PIE chart.
  - Use `otu_ids` as the labels for the pie chart.
  - Use `otu_labels` as the hovertext for the chart.



- Create a Bubble Chart that uses data from your samples route ( /samples/<sample> ) to display each sample.
  - Use `otu_ids` for the x values.
  - Use `sample_values` for the y values.
  - Use `sample_values` for the marker size.
  - Use `otu_ids` for the marker colors.
  - Use `otu_labels` for the text values.

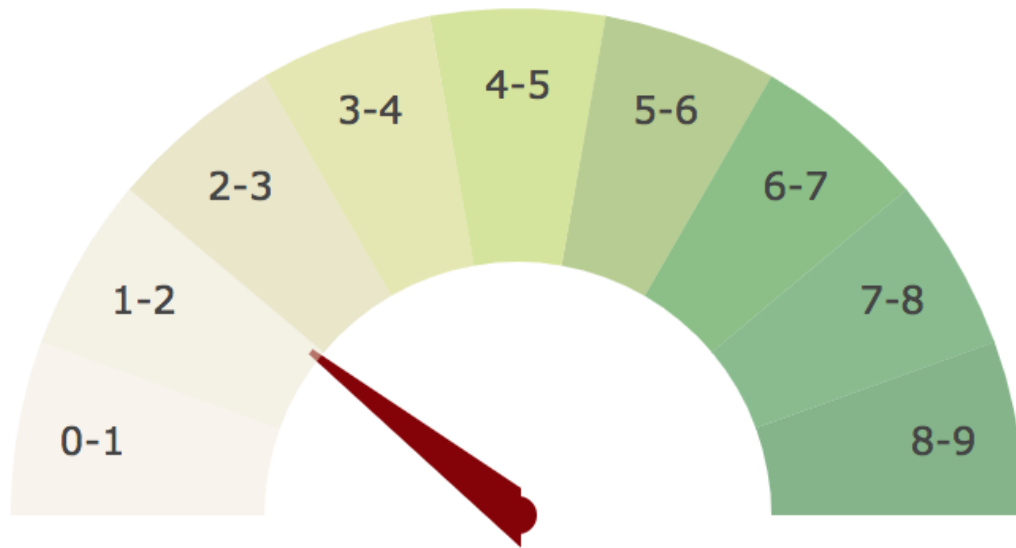


- Display the sample metadata from the route `/metadata/<sample>`
  - Display each key/value pair from the metadata JSON object somewhere on the page.
- Update all of the plots any time that a new sample is selected.
- You are welcome to create any layout that you would like for your dashboard. An example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



- You will need to modify the example gauge code to account for values ranging from 0 - 9.
- Update the chart whenever a new sample is selected.

## Belly Button Washing Frequency Scrubs per Week



## Flask API

---

Use Flask API starter code to serve the data needed for your plots.

- Test your routes by visiting each one in the browser.

## Hints

---

- Use `console.log` inside of your JavaScript code to see what your data looks like at each step.
- Refer to the [Plotly.js Documentation](#) when building the plots.

## Copyright

Trilogy Education Services © 2019. All Rights Reserved.