Module 12 Challenge

Start Assignment

Due Sunday by 11:59pm **Points** 100 **Submitting** a text entry box or a website url

Background

Roza has a partially completed dashboard that she needs to finish. She has a completed panel for demographic information and now needs to visualize the bacterial data for each volunteer. Specifically, her volunteers should be able to identify the top 10 bacterial species in their belly buttons. That way, if Improbable Beef identifies a species as a candidate to manufacture synthetic beef, Roza's volunteers will be able to identify whether that species is found in their navel.

What You're Creating

This new assignment consists of four technical analysis deliverables. You will submit the following:

- Deliverable 1: Create a Horizontal Bar Chart
- Deliverable 2: Create a Bubble Chart
- Deliverable 3: Create a Gauge Chart
- Deliverable 4: Customize the Dashboard

Files

Use the following links to download the Challenge starter codes.

[Download the bar chart starter code.](https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_12/BellyButton_bar_chart_starter_code.js)

[Download the bubble chart starter code.](https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_12/BellyButton_bubble_chart_starter_code.js)

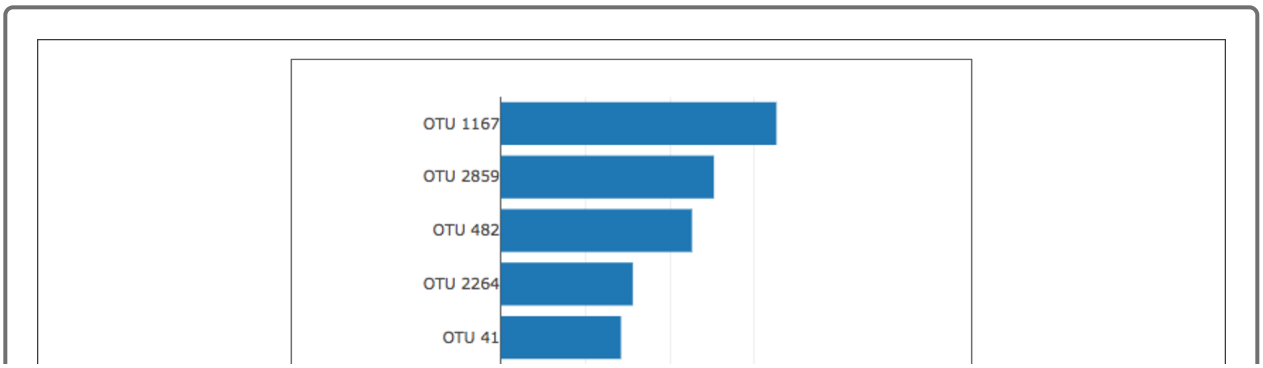
[Download the gauge chart starter code.](https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_12/BellyButton_gauge_starter_code.js)

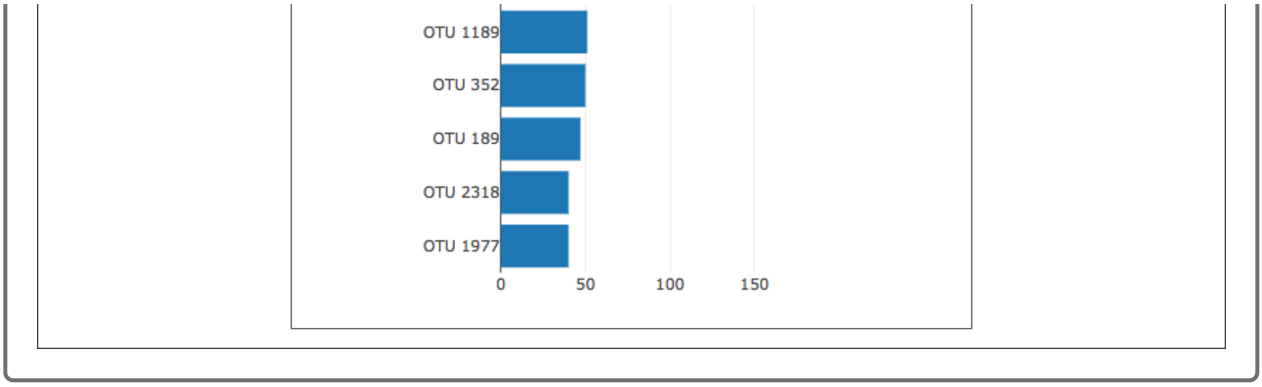
Deliverable 1: Create a Horizontal Bar Chart (35 points)

Deliverable 1 Instructions

Using your knowledge of JavaScript, Plotly, and D3.js, create a horizontal bar chart to display the top 10 bacterial species (OTUs) when an individual's ID is selected from the dropdown menu on the webpage. The horizontal bar chart will display the `sample_values` as the values, the `otu_ids` as the labels, and the `otu_labels` as the hover text for the bars on the chart.

Your bar chart should look like the following image:





REWIND

For this deliverable, you've already done the following in this module:

- **Lesson 12.1.3:** Create a bar chart
- **Lesson 12.2.1:** Use JavaScript functions and methods: `map()`, `filter()`, `reverse()`, and `slice()`.
- **Lesson 12.2.2:** Create a bar chart with filtered arrays
- **Lesson 12.3.2:** Load a JSON file with `d3.json()`
- **Lesson 12.4.3:** Create a function that reads in json data
- **Lesson 12.4.3:** Write code to use the ID number to create the sample's information on a panel or chart
- **Lesson 12.5.1:** Deploy your project to GitHub Pages

Download the `BellyButton_bar_chart_starter_code.js`, add it to the js folder of your GitHub pages (GitHub.io) folder, and rename the file `charts.js`. Use the instructions below to add code where indicated by the numbered-

step comments in the starter code file.

In Steps 3-6, you'll initialize variables that hold arrays for the sample that is selected from the dropdown menu on the webpage.

IMPORTANT

> Make sure that you use `console.log()` to help debug any issues.

1. In Step 1, we've provided the code for the `buildCharts();` function that contains the argument `sample`, which is the sample that is selected from the dropdown menu.
2. In Step 2, we've provided the code to retrieve the `samples.json` file using the `d3.json().then()` method.
3. In Step 3, create a variable that has the array for all the samples.
4. In Step 4, create a variable that will hold an array that contains all the data from the new sample that is chosen from the dropdown menu. To retrieve the data from the new sample, filter the variable created in Step 3 for the sample `id` that matches the new sample `id` chosen from the dropdown menu and passed into the `buildCharts()` function as the argument.
5. In Step 5, create a variable that holds the first sample in the array.

NOTE

> You can combine Steps 4 and 5 as one line of code, but make sure you use the correct variable name for Step 6 when retrieving the array data.

6. In Step 6, create variables that have arrays for `otu_ids` `otu_labels`

6. In Step 6, create variables that have arrays for `otu_ids`, `otu_labels`, and `sample_values`.

7. In Step 7, create the `yticks` for the bar chart.

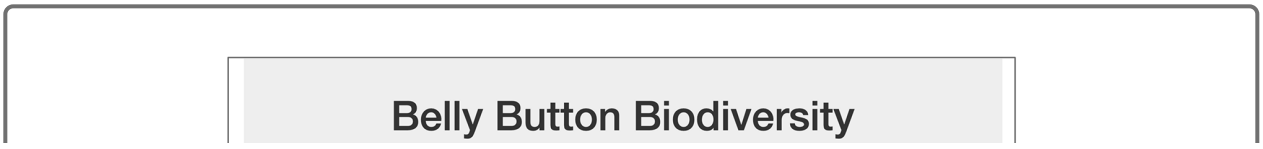
HIDE HINT

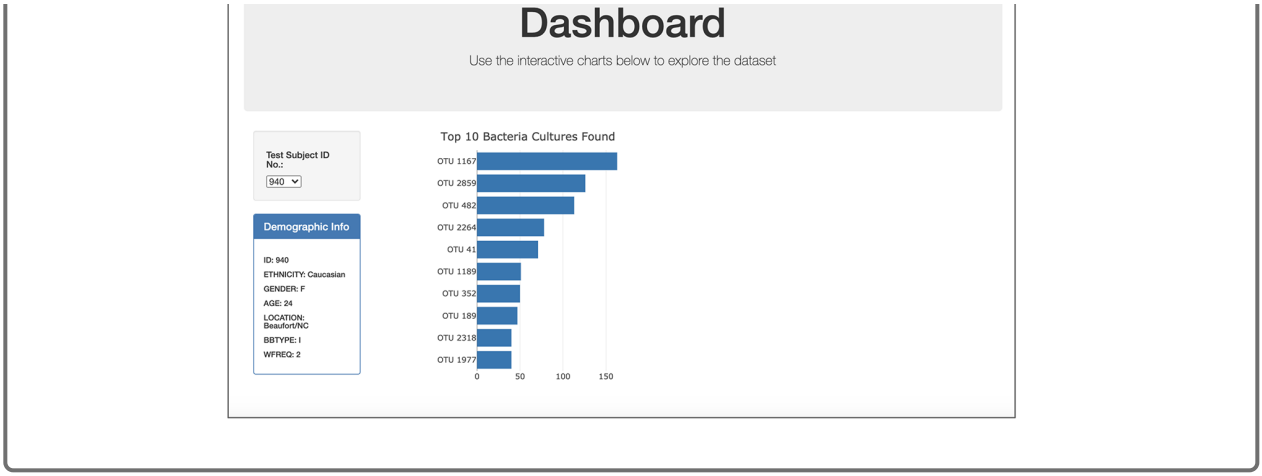
Chain the `slice()` method with the `map()` and `reverse()` functions to retrieve the top 10 `otu_ids` sorted in descending order.

In Steps 8-10, create the `trace` object, the layout, and `Plotly.newPlot()` function for the horizontal bar chart.

8. In Step 8, create the `trace` object for the bar chart, where the x values are the `sample_values` and the hover text for the bars are the `otu_labels` in descending order.
9. In Step 9, create the layout for the bar chart that includes a title.
10. In Step 10, use the `Plotly.newPlot()` function to plot the trace object with the layout.

After you have completed the coding requirements, your dashboard will look like this image when it loads for the first time:





Deliverable 1 Requirements

You will earn a perfect score for Deliverable 1 by completing all requirements below:

- Code is written to create the arrays when a sample is selected from the dropdown menu **(10 pt)**
- Code is written to create the trace object in the `buildCharts()` function, and it contains the following: **(10 pt)**
  - The y values are the `otu_ids` in descending order
  - The x values are the `sample_values` in descending order
  - The hover text is the `otu_labels` in descending order.
- Code is written to create the layout array in the `buildCharts()` function that creates a title for the chart **(5 pt)**
- When the dashboard is first opened in a browser, ID 940's data should be displayed in the dashboard, and the bar chart has the following: **(10 pt)**
  - The top 10 `sample_values` are sorted in descending order
  - The top 10 `sample_values` as values

- The `otu_ids` as the labels

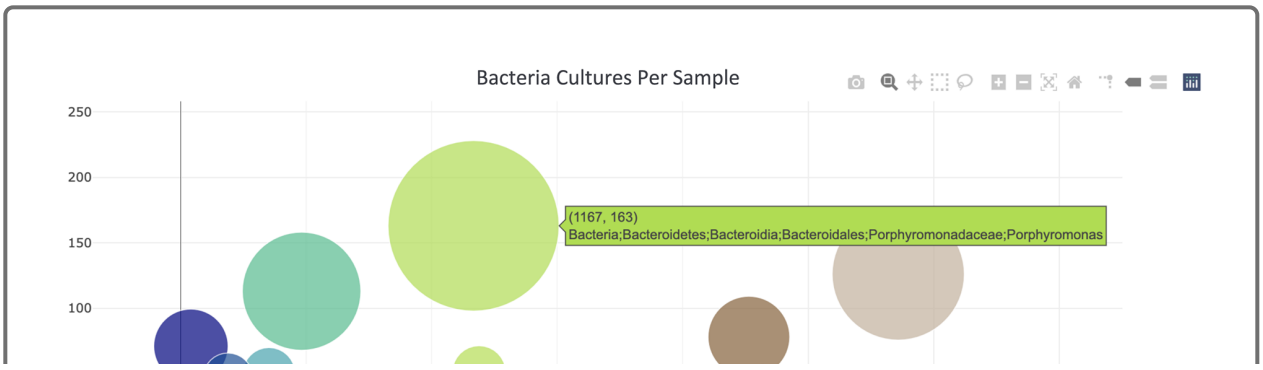
Deliverable 2: Create a Bubble Chart (30 points)

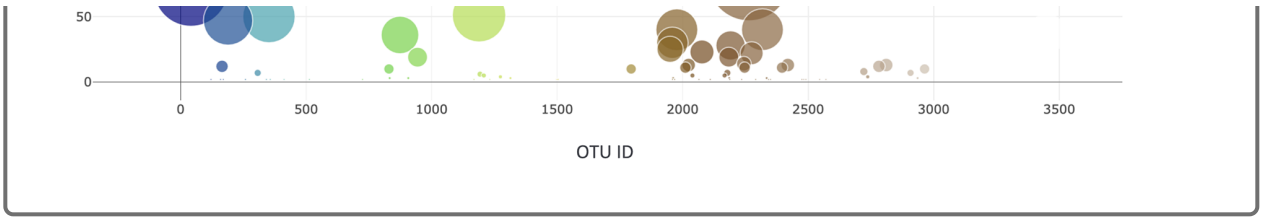
Deliverable 2 Instructions

Using your knowledge of JavaScript, Plotly, and D3.js, create a bubble chart that will display the following when an individual's ID is selected from the dropdown menu webpage:

- The `otu_ids` as the x-axis values.
- The `sample_values` as the y-axis values.
- The `sample_values` as the marker size.
- The `otu_ids` as the marker colors.
- The `otu_labels` as the hover-text values.

Your bubble chart should look like the following image:





REWIND

For this deliverable, you've already done the following in this module:

- **Lesson 12.2.2** Create a `trace` object and layout for a chart
- **Lesson 12.5.1:** Deploy your project to GitHub Pages

Download the `BellyButton_bubble_chart_starter_code.js` file, copy the starter code from Steps 1-3, and add it to your `charts.js` file after Step 10 for Deliverable 1.

Use the variables that were created in Deliverable 1 to populate the bubble chart. Then, use the instructions below to write the code for the `trace` object, the layout, and `Plotly.newPlot()` function to create the bubble chart.

1. To create the `trace` object for the bubble chart do the following:
    - Assign the `otu_ids`, `sample_values`, and `otu_labels` to the x, y, and text properties, respectively.
    - For the `mode` and `marker` properties, the mode is "markers" and the `marker` property is a dictionary that defines the `size`, `color`,

and `colorscale` of the markers.

If you'd like a hint on how to create a `trace` object for a bubble chart, that's totally okay. If not, that's great too. You can always revisit this later if you change your mind.

HIDE HINT

Using `d3.select()`, you can select the element that has changed and retrieve the property and HTML id that have changed.

Check out the Plotly **bubble chart documentation (https://plotly.com/javascript/bubble-charts/#hover-text-on-bubble-charts)** .

2. To create the layout for the bubble chart, add a title, a label for the x-axis, margins, and the `hovermode` property. The `hovermode` should show the text of the bubble on the chart when you hover near that bubble.

If you'd like a hint on how to create a layout for a bubble chart, that's totally okay. If not, that's great too. You can always revisit this later if you change your mind.

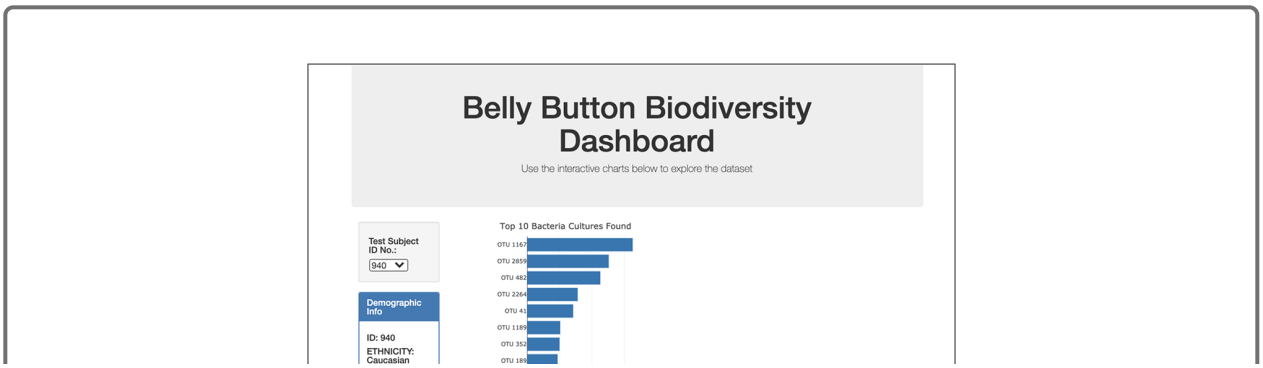
HIDE HINT

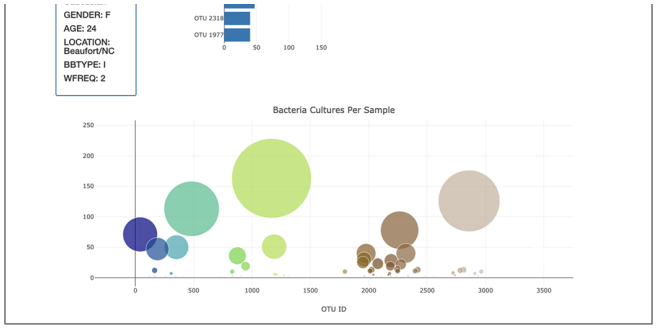
Using `d3.select()`, you can select the element that has changed and retrieve the property and HTML id that have changed.

Check out the Plotly **layout object documentation (https://plotly.com/python-api-reference/generated/plotly.graph_objects.Layout.html)** .

3. Lastly, use the given `Plotly.newPlot()` function to plot the `trace` object and layout.

After you have completed the coding requirements, your dashboard will look like the image below when it loads for the first time, with the bar chart you created in Deliverable 1 and the bubble chart.





Deliverable 2 Requirements

You will earn a perfect score for Deliverable 2 by completing all requirements below:

- The code for the trace object in the `buildCharts();` function does the following: **(10 pt)**
  - Sets the `otu_ids` as the x-axis values
  - Sets the `sample_values` as the y-axis values
  - Sets the `otu_labels` as the hover-text values
  - Sets the `sample_values` as the marker size
  - Sets the `otu_ids` as the marker colors
- The code for the layout in the `buildCharts();` function does the following: **(10 pt)**
  - Creates a title
  - Creates a label for the x-axis
  - The text for a bubble is shown when hovered over
- When the dashboard is first opened in a browser, ID 940's data should be displayed in the dashboard. All three charts should also be working according to their requirements when a sample is selected from the

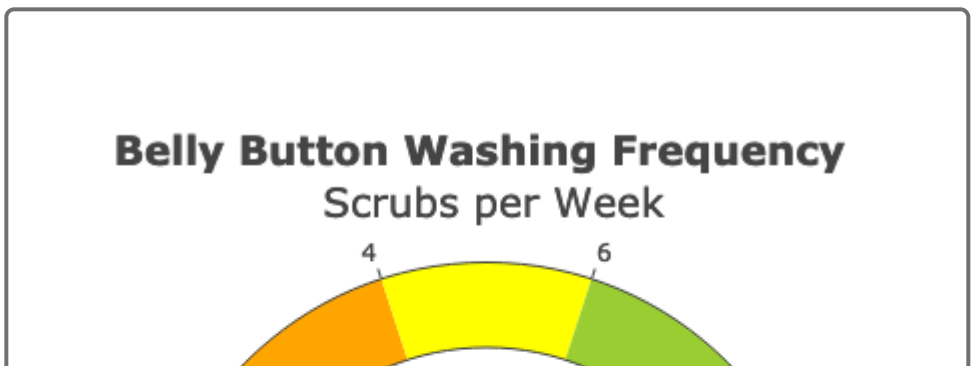
dropdown menu **(10 pt)**

Deliverable 3: Create a Gauge Chart (20 points)

Deliverable 3 Instructions

Using your knowledge of JavaScript, Plotly, and D3.js, create a gauge chart that displays the weekly washing frequency's value, and display the value as a measure from 0-10 on the progress bar in the gauge chart when an individual ID is selected from the dropdown menu.

Your gauge chart should look similar to the following image:



REWIND

For this deliverable, you've already done the following in this module:

- **Lesson 12.2.2:** Create a `trace` object and layout for a chart
- **Lesson 12.5.1:** Deploy your project to GitHub Pages

Download the `BellyButton_gauge_starter_code.js`, using Steps 1-3 in the `buildCharts()` function initialize variables that hold arrays for the sample that is selected from the dropdown menu on the webpage.

1. In Step 1, create a variable that filters the metadata array for an object in the array whose id property matches the ID number passed into `buildCharts()` function as the argument.
2. In Step 2, create a variable that holds the first sample in the array created in Step 2.

NOTE

> You can combine Steps 1 and 2 as one line of code, but make sure you use the correct variable name for Step 3 when retrieving the washing frequency value.

3. In Step 3, create a variable that converts the washing frequency to a floating point number.
4. In Step 4, create the trace object for the gauge chart.

If you'd like a hint on how to create a gauge chart, that's totally okay. If not, that's great too. You can always revisit this later if you change your mind.

HIDE HINT

Using `d3.select()`, you can select the element that has changed and retrieve the property and HTML id that has changed.

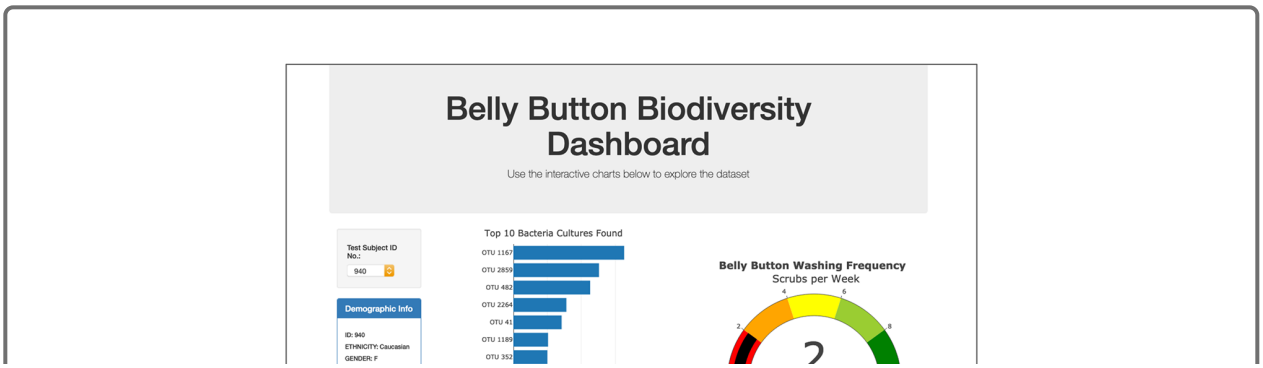
Check out the Plotly **gauge charts in JavaScript (https://plotly.com/javascript/gauge-charts/)** documentation and use these hints.

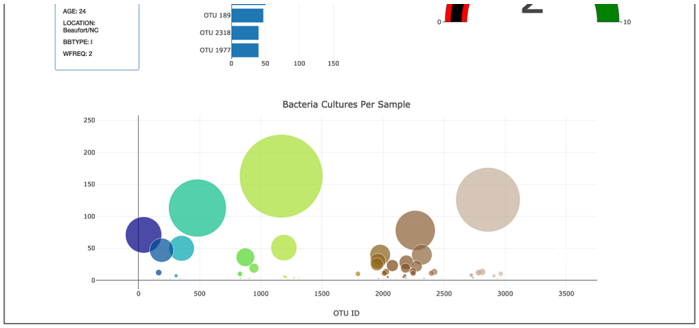
- Assign the variable created in Step 3 to the `value` property.
- The `type` property should be "indicator".
- The `mode` property should be "gauge+number".
- For the `title` object, assign the title as a string using HTML syntax to the `text` property.
- For maximum range for the gauge should be 10.

- Set the bar color of the gauge to black or a dark color to contrast against the range colors.
- Assign different colors as string values in increments of 2 for the steps object. The colors can be named colors as in the **Matplotlib colors (https://matplotlib.org/3.1.0/gallery/color/named_colors.html)** or rgba values.

5. In Step 5, create the layout for the gauge chart making sure that it fits in the `<div></div>` tag for the gauge id.
6. In Step 6, use the `Plotly.newPlot()` function to plot the `trace` object and the layout.

After you have completed the coding requirements, your dashboard will look like this image when it loads for the first time, with the bar chart you created in Deliverable 1, the bubble chart created in Deliverable 2, and the gauge chart:





Deliverable 3 Requirements

You will earn a perfect score for Deliverable 3 by completing all requirements below:

- The code to build the gauge chart does the following: **(10 pt)**
  - Creates a title for the chart.
  - Creates the ranges for the gauge in increments of two, with a different color for each increment.
  - Adds the washing frequency value on the gauge chart.
  - The indicator shows the level for the washing frequency on the gauge.
  - The gauge is added to the dashboard.
  - The gauge fits in the margin of the `<div>` element.
- When the webpage loads, the bar and bubble chart are working according to the requirements in Deliverable 1 and 2, respectively, and the gauge chart is working according to the requirements listed for this Deliverable **(10 pt)**

Deliverable 4: Customize the Dashboard (20

points)

Deliverable 4 Instructions

Use your knowledge of HTML and Bootstrap to customize the webpage for your dashboard.

1. Customize your dashboard with three of the following:
    - Add an image to the jumbotron.
    - Add background color or a variety of compatible colors to the webpage.
    - Use a custom font with contrast for the colors.
    - Add more information about the project as a paragraph on the page.
    - Add information about what each graph visualizes, either under or next to each graph.
    - Make the webpage mobile-responsive.
    - Change the layout of the page.
    - Add a navigation bar that allows you to select the bar or bubble chart on the page.

2. When the dashboard is first opened in a browser, ID 940's data should be displayed in the dashboard, and the three charts should be working according to their requirements.

3. When a sample is selected, the dashboard should display the data in the panel and all three charts according to their requirements.

Deliverable 4 Requirements

You will earn a perfect score for Deliverable 4 by completing all

requirements below:

- The webpage has three customizations. **(10 pt)**
- When the dashboard is first opened in a browser, ID 940's data should be displayed in the dashboard, and all three charts should be working according to the requirements when a sample is selected from the dropdown menu **(5 pt)**

Submission

Once you're ready to submit, make sure to check your work against the rubric to ensure you are meeting the requirements for this Challenge one final time. It's easy to overlook items when you're in the zone!

As a reminder, the deliverables for this Challenge are as follows:

- Deliverable 1: Create a Horizontal Bar Chart
- Deliverable 2: Create a Bubble Chart
- Deliverable 3: Create a Gauge Chart
- Deliverable 4: Customize the Dashboard

Upload the following to your GitHub pages repository:

- The updated `index.html` file.

- The `charts.js` file, which should be in the js folder of the static folder.
- The `samples.json` file.
- A README.md that describes the purpose of the repository. Although there is no graded written analysis for this challenge, it is encouraged and good practice to add a brief description of your project.

To submit your challenge assignment for grading in Bootcamp Spot, click Start Assignment, click the Website URL tab, then provide the URL to your deployment and your GitHub repository, and then click Submit. Comments are disabled for graded submissions in BootCampSpot. If you have questions about your feedback, please notify your instructional staff or the Student Success Manager. If you would like to resubmit your work for an improved grade, you can use the **Re-Submit Assignment** button to upload new links. You may resubmit up to 3 times for a total of 4 submissions.

IMPORTANT

> Once you receive feedback on your Challenge, make any suggested updates or adjustments to your work. Then, add this week's Challenge to your professional portfolio.

NOTE

> You are allowed to miss up to two Challenge assignments and still earn your certificate. If you complete all Challenge assignments, your lowest two grades will be dropped. If you wish to skip this assignment, click Next, and move on to the next Module.

Module-12 Rubric

Criteria	Ratings					Pts
Deliverable 1: Create a Horizontal Bar Chart	**35 to >33.0 pts Demonstrating Proficiency** ✓Arrays are created when a new sample is selected from the dropdown menu. ✓All THREE elements of the trace object are created, and the data is in descending order. ✓The layout array creates a title for the chart. ✓When the webpage loads, the bar chart behaves according to the requirements.	**33 to >31.0 pts Approaching Proficiency** ✓Arrays are created when a new sample is selected from the dropdown menu. ✓ALL THREE elements of the trace object are created, and some of the data is in descending order. ✓The layout array creates a title for the chart. ✓When the webpage loads, the bar chart behaves according to the requirements.	**31 to >27.0 pts Developing Proficiency** ✓Arrays are created when a new sample is selected from the dropdown menu. ✓ALL THREE elements of the trace object are created, but the data is not in descending order. ✓The layout array creates a title for the chart. ✓When the webpage loads, the bar chart behaves according to the requirements.	**27 to >0.0 pts Emerging** ✓Arrays are created when a new sample is selected from the dropdown menu. ✓TWO of the THREE elements of the trace object are created. ✓The layout array creates a title for the chart. ✓When the webpage loads, the bar chart behaves according to the requirements.	**0 pts Incomplete**	35 pts
Deliverable 2: Create a Bubble Chart	**30 to >27.0 pts Demonstrating Proficiency** ✓All FIVE elements of the trace object are created, and data is properly displayed. ✓All THREE elements are created with the layout. ✓When the webpage loads, the bar and bubble chart initialize without error and update when a new sample is selected.	**27 to >23.0 pts Approaching Proficiency** ✓Code is written for FOUR of the FIVE elements of the trace object, and the data is displayed. ✓All THREE elements are created with the layout. ✓When the webpage loads, the bar chart initializes without error and is updated. ✓When the webpage loads, there is a minor error when the bubble chart	**23 to >20.0 pts Developing Proficiency** ✓Code is written for THREE of the FIVE elements for the bubble chart. ✓Code is written for TWO of THREE elements for the layout. ✓When the webpage loads, the bar chart initializes without error and is updated. ✓When the webpage loads, there are a few errors when the bubble chart	**20 to >0.0 pts Emerging** ✓Code is written for TWO of the FIVE elements for the bubble chart. ✓Code is written for ONE of THREE elements for the layout. ✓When the webpage loads, the bar chart initializes without error and is updated. ✓When the webpage loads, there are errors loading the bubble chart.	**0 pts Incomplete**	30 pts

Criteria	Ratings					Pts
Deliverable 3: Create a Gauge Chart	**20 to >17.0 pts** **Demonstrating Proficiency** ✓The gauge chart on the dashboard has ALL requirements. ✓When the webpage loads, the bar, bubble, and gauge chart initialize without error and update when a new sample is selected.	**17 to >14.0 pts** **Approaching Proficiency** ✓The gauge chart on the dashboard has ALL requirements. ✓When the webpage loads, the bar and bubble charts initialize without error and are updated. ✓When the webpage loads, there is a minor error when the gauge chart initializes or is updated.	**14 to >11.0 pts** **Developing Proficiency** ✓The gauge chart has code written for THREE of the requirements, but there are a few errors when the gauge chart initializes or is updated. ✓When the webpage loads, the bar and bubble charts initialize without error and are updated.	**11 to >0.0 pts** **Emerging** ✓The gauge chart has code written for THREE of the requirements, but there are errors loading the gauge chart. ✓When the webpage loads, the bar and bubble charts initialize without error and are updated.	**0 pts** **Incomplete**	20 pts
Deliverable 4: Customize the Dashboard	**15 to >13.0 pts** **Demonstrating Proficiency** ✓The webpage has THREE customizations. ✓When the webpage loads, the bar and bubble chart initialize without error and update when a new sample is selected.	**13 to >11.0 pts** **Approaching Proficiency** ✓The webpage has THREE customizations. ✓When the webpage loads or is updated, there is a minor error when the bar or bubble charts initialize and are updated.	**11 to >9.0 pts** **Developing Proficiency** ✓The webpage has TWO of THREE customizations. ✓When the webpage loads or is updated, there is an error when the bar or bubble charts initialize and are updated.	**9 to >0.0 pts** **Emerging** ✓The webpage has ONE of THREE customizations. ✓When the webpage loads or is updated, there are errors when the bar or bubble charts initialize and are updated.	**0 pts** **Incomplete**	15 pts
						Total Points: 100