**Population**

**Static JavaSript**

This JavaScript code reads data from a REST API and creates two pie charts using Plotly to visualize the population data by region and subregion. The code also includes a function to update the subregion pie chart based on user selection.

Here's a breakdown of what the code does:

1. It reads data from the URL "https://restcountries.com/v3.1/all" using `d3.json`.

2. It then processes the data to group the population by region and subregion and stores the results in the `result` array.

3. It initializes the default pie chart showing the population by region using `Plotly.newPlot`.

4. It sets up a listener on the DOM element with the ID "selDataset" to call the `getData` function when the dropdown menu is changed.

5. The `getData` function is responsible for updating the subregion pie chart based on the selected region.

6. The `init` function is called initially to display the default pie chart.

Here's a step-by-step explanation of the `init` function:

1. It initializes two arrays, `IDs` and `populations`, to store the region IDs and populations from the `result` array.

2. It creates a data object for the default region pie chart using `IDs` and `populations`.

3. It sets up a layout for the default region pie chart.

4. It creates the default region pie chart using `Plotly.newPlot` and the data and layout.

When the user selects a region from the dropdown menu, the `getData` function is called:

1. It retrieves the selected region from the dropdown menu.

2. It initializes two arrays, `subIDs` and `subpopulations`, to store the subregion IDs and populations for the selected region.

3. It filters the `result` array to get the relevant data for the selected region.

4. It populates `subIDs` and `subpopulations` with data from the subregions of the selected region.

5. It creates a data object for the subregion pie chart using `subIDs` and `subpopulations`.

6. It sets up a layout for the subregion pie chart.

7. It creates the subregion pie chart using `Plotly.newPlot` and the data and layout, updating the chart with the selected subregion data.

Overall, the code is designed to display population data for different regions and their subregions using interactive pie charts. The initial population data is shown for all regions, and when the user selects a specific region, the pie chart updates to display the population data for that region's subregions.

**HTML File**

This HTML file includes D3.js, Plotly, and custom JavaScript files to create a web page with two side-by-side pie charts showing population data by region and subregion. Let's break down the components of this code:

1. The HTML structure:

- The HTML document includes the necessary metadata, such as the charset and the page title.

- It imports the D3.js library and the Plotly library using script tags.

- A custom CSS file, "style.css," is linked to the page for custom styling.

- The body of the page contains a <select> element with the ID "selDataset," allowing users to choose a region from the dropdown menu.

- There is a <table> element with two <td> (table cell) elements, each containing a <div> element with IDs "pie" and "pie2." These <div> elements will be used to display the two pie charts side by side.

2. The content of the dropdown menu:

- The dropdown menu contains options for the five regions: Asia, Africa, Americas, Europe, and Oceania.

3. The custom JavaScript file:

- The JavaScript file "logicPopulation.js" is included at the end of the body.

- This file is responsible for fetching data from the API, processing it, and creating the pie charts using D3.js and Plotly. The provided JavaScript code in your previous message likely corresponds to this file.

4. The CSS file:

- The CSS file "style.css" is expected to be in the "staticPopulation/css/" directory relative to the HTML file. It's used to apply custom styles to the web page.

Overall, the HTML file sets up the structure of the web page with the dropdown menu and two side-by-side divs for the pie charts. The JavaScript file, along with the D3.js and Plotly libraries, handles the data processing and chart creation logic. The CSS file customizes the appearance of the page.