**Bubbles**

**Static JavaSript**

This is a JavaScript code snippet that creates a map with bubble markers to represent countries. The size of each marker is determined based on the country's population, and the color of the marker is based on the country's area. The data for the map is obtained from the "https://restcountries.com/v3.1/all" API endpoint.

Here's a breakdown of the code:

1. The `markerSize` function takes a country's population as input and returns the marker size based on the square root of the population multiplied by 50.

2. The `markerColor` function takes a country's area as input and returns a color based on the range of area values. Different colors are assigned based on different area ranges.

3. The `bubbleFunc` function is the main function that creates the map and the bubble markers. It uses Leaflet, a popular mapping library in JavaScript.

4. The map is created using Leaflet's `L.map` function, centered at latitude 34 and longitude 27, with an initial zoom level of 2.2.

5. The OpenStreetMap tile layer is added to the map using `L.tileLayer`.

6. The `d3.json` function is used to fetch data from the specified API endpoint.

7. The minimum and maximum values for area and population are calculated using the `ss.min` and `ss.max` functions from a statistical library (likely `simple-statistics` or similar).

8. The data is looped through, and for each country, a circle marker (`L.circle`) is created at the latitude and longitude specified in the `latlng` property of the country's data.

9. The marker's properties such as title, fill color, radius, etc., are set based on the country's area and population using the `markerColor` and `markerSize` functions.

10. A popup is added to each marker to display additional information about the country, including its name, population, and area.

11. A legend is created using `L.control` to display the color scale for the area ranges.

12. The legend is added to the map using the `addTo` method.

13. Finally, the `bubbleFunc` function is called to execute the map creation and rendering process.

Note: This code uses the Leaflet library and the `d3.json` function to fetch data. Make sure you have included these libraries in your HTML file before running this script. Additionally, it seems the code uses a third-party statistical library (e.g., `simple-statistics`) to calculate the min and max values for area and population. Ensure that this library is also included if you want to run the code as-is.

**HTML File**

The HTML file includes the necessary dependencies to create the Population-Bubble-Map. Let's go through each part of the code:

1. The `<!DOCTYPE html>` declaration defines the document type and version as HTML5.

2. Inside the `<head>` section, there are several meta tags that define the character encoding, viewport settings, and compatibility settings for Internet Explorer.

3. The title of the page is set to "Population-Bubble-Map."

4. The Leaflet CSS file is linked using the `<link>` tag. It provides the necessary styles for the Leaflet map.

5. An additional custom CSS file named "style.css" is included using the `<link>` tag. This file might contain additional styles to customize the appearance of the map or the page.

6. The main content of the page is within the `<body>` tags.

7. Inside the `<body>` section, there's a `<div>` element with the ID "map." This is the container where the map will be displayed.

8. The Leaflet JavaScript library is imported using the `<script>` tag. It provides the necessary functions to create interactive maps.

9. The D3 JavaScript library is imported using the `<script>` tag. It will be used to fetch and process data from the API.

10. The custom JavaScript file "bubble.js" is included using the `<script>` tag. This file contains the JavaScript code to create the bubble map using Leaflet and D3.

11. The Simple Statistics library is imported using the `<script>` tag. This library is likely used to calculate the min and max values for area and population in the "bubble.js" file.

Overall, this HTML file sets up the necessary dependencies and container for the Population-Bubble-Map. To complete the visualization, make sure the "bubble.js" file contains the JavaScript code provided in the previous response, including the functions `markerSize`, `markerColor`, and `bubbleFunc`. Also, ensure the API endpoint "https://restcountries.com/v3.1/all" is accessible and providing the required data. With all these pieces in place, the map should display bubble markers representing different countries with their population and area data.