**Briefly explain the logic for generating the base map.**

* Included necessary links to plugin (If we don’t desire to download onto our machines.)
* create the div container to hold the map on .html page with an element ‘id’ in order to be referenced by map object.
* To create a map object, it requires above div container id, its setView (could be lat and long), and zoom level set-default when landing on page. This could be stored into a variable.
* (ex. var myMap = L.map())
* Then adding a tile layer which is thought of our base map.
* The tile layer requires the URL to a tile server.
* Call the addTo() function and pass myMap into addTo().
* Up to this step, we should have the base map set up.

**Describe how the JSON was loaded and how was the data traversed. Explain how was the information from the JSON used to render data on the map.**

**Explain the logic for generating the circles and amending the size of them. What does this communicate**?

* Circles can be used to describe the density of a group of a data. The amending its size represent the differences size of data groups.

**Describe how the layer for the Tectonic plates was generated.**

**What are the components in the layer control? How were they generated?**

* Leaflet allows us to control which layers we would like to see on our map.
* There are two types of layers: 1) *base layers* that are mutually exclusive (only one can be visible on your map at a time), e.g. tile layers, and 2) *overlays*, which are all the other stuff you put over the base layers.
* To generate, first is to create the base layers as creating the base map explained earlier.
* Next, create two objects. One contains base layers and one contains overlays. These are just simple objects with key/value pairs. The key sets the text for the layer in the control (e.g. “Streets”), while the corresponding value is a reference to the layer (e.g. “Streets”).
* Now, create a layer control and add it to the map. The first argument passed when creating the layers control is the base layers object. The second argument is the overlays object. Both arguments are optional
* Code example : *L.control.layers(baseMaps, overlayMaps).addTo(map);*

**Explain the difference between the base map (tile layer) and the data layer(s).**

* Base map or Tile Layers is the main map working as a landing on map or a basic layer that allows other objects to put over. It can be visible one at a time; while, data layers are all other stuffs put over the base layers and can be visible multiply at the same time.

**Walk through the logic of how the legend was generated and rendered on the page.**

* To set up and show the legend section, we need to create a variable (named legend) in order to position the legend on map.
* Create an event of adding legends on map using — onAdd() method, which is from leaflet’s control class returns the container DOM element for the control and add listeners on relevant map events.
* Then *DomUtil* class has used. Here — *create()* method has used which is from leaflet’s DomUtil class. This method returns a HTML element and takes parameter as tag name and class name. Tag name to create that element on document and class name style that element. Example : *var div = L.DomUtil.create(‘div’, ‘info legend’)*
* There are few more variables created in order to assemble later as needed. From our project, we have — limits, colors, and labels.
* We use div.innerHTML to modify the content of an HTML element (Here is our *div*) is by using the innerHTML property.
* Lastly, add legend to the map using : *legend.addTo(map);*