All example files available at *github.com/northlandiguana/pcdRoxx*.
More in-depth lab tutorials are at *github.com/reroth/g575-2013*. Leaflet documentation at *leafletjs.com*.

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>PCD Roxx!</title>

    <meta name="viewport" content="width=device-width">

    <!-- links to Leaflet stylesheets -->
    <link rel="stylesheet" href="css/leaflet.css">
    <!--[if IE]>
        <link rel="stylesheet" href="css/leaflet.ie.css">
    <![endif]-->

</head>

<body>

    <!-- styles for html elements -->
    <style>
        body {
            font-family: sans-serif;
        }
        #wrapper {
            width: 960px;
            margin: 15px;
        }
        #map {
            width: 100%;
            height: 450px;
            margin: 15px;
        }
    </style>

    <!-- html elements to hold page content, map -->
    <div id="wrapper">
        <h1>Proportional Symbol Map for PCD</h1>
        <div id="map"></div>
    </div>

    <!-- script links to jQuery and Leaflet -->
    <script src="js/jquery.js"></script>
    <script src="js/leaflet.js"></script>

    <!-- main script -->
    <script>

        //use jQuery to load script after HTML
        $(document).ready(function(){

            //global variables
            var map,
                cities,
                tiles;

            //new Leaflet map
            map = L.map('map', {
                center: [38, -96],
                zoom: 4,
                minZoom: 4
            });
```

Glossary of Terms

HTML: Hypertext Markup Language; the language used to put elements on a web page.

DOM: Document Object Model; the standard for how web pages are organized, including how to get, change, add, or delete HTML elements.

Element: A piece of the web page with certain properties, such as body, div, p, span, etc.

Tag: The syntax used to define HTML elements, such as <body>, <div>, etc.

CSS: The language used to style HTML elements; can be in a linked .css stylesheet or between <style> tags.

JavaScript: The language used to make HTML pages dynamic and interactive; can be in a linked .js file or between <script> tags. (No relation to Java, a different programming language).

jQuery: A popular JavaScript library; very good for accessing HTML elements and loading external data resources in the script.

Function: A block of JavaScript code that does something and can be accessed (called) elsewhere in the script for easy task repetition. Functions may be written in the main script or contained within a library, such as Leaflet or jQuery.

Library: A collection of JavaScript functions useful for certain tasks that is shared with others by its creator(s).

Method: A function contained within a library, called through its prototype object.

Object: A set of key-value pairs, allowing the values to be accessed by reference to their keys. In JavaScript, objects are contained within curly braces; e.g. {key1: value1, key2:value2, …}

Array: A set of comma-separated values. In JavaScript, contained within braces; e.g. [value1, value2, value3, …]

Parameter: A value, object, or function passed into a function when the function is called.

Anonymous Function: A function without a name, usually passed as a parameter into another function.

```
            //new Leaflet tileLayer for background slippy tiles
            tiles = L.tileLayer('http://{s}.acetate.geoiq.com/tiles/acetate/{z}/{x}/{y}.png',
                {
                    attribution: 'GeoIQ layer'
                }).addTo(map); //<-don't forget this bit if you want to see the tiles!

            //use jQuery to load the data
            $.getJSON("data/cityData.json")
                .done(function(data){

                    //new Leaflet geoJson layer creates point features out of data objects
                    L.geoJson(data, {

                        //attach a popup with the feature name to each point feature
                        onEachFeature: function(feature, layer){
                            layer.bindPopup(feature.properties.name)
                        },

                        //change default png markers to dynamic circle markers
                        pointToLayer: function(feature, latlng){

                            //new Leaflet circleMarker layer for each marker
                            return L.circleMarker(latlng, {

                                //calculate each radius separately using an attribute
                                radius: calcPropRadius(feature.properties.yr2005),
                                fillColor: '#f00',
                                color: '#f00'
                            })
                        }
                    }).addTo(map); //<-add the geoJson layer to the map
                })

            //function to calculate the radius of each proportional symbol
            function calcPropRadius(attributeValue){
                var scaleFactor = 10;
                var area = attributeValue *scaleFactor;
                var radius = Math.sqrt(area/Math.PI)*2;
                return radius;
            }
        })

    </script>

</body>
</html>
```

OGR2OGR

To convert csv with latitude/longitude coordinate columns into geoJSON.

1) Create plain text .vrt file, containing the following (bolded items are variable):

```
<OGRVRTDataSource>
    <OGRVRTLayer name="cityData">
        <SrcDataSource>C:\location\cityData.csv</SrcDataSource>
        <GeometryType>wkbPoint</GeometryType>
        <LayerSRS>WGS84</LayerSRS>
        <GeometryField encoding="PointFromColumns" x="Longitude" y="Latitude"/>
    </OGRVRTLayer>
</OGRVRTDataSource>
```

2) In command shell running GDAL/OGR:

```
C:>ogr2ogr -f “geoJSON” C:\location\cityData.json C:\location\cityData.vrt
```