Technical Document

Web-based Health Data Mapping Tools for San Diego County

Project conducted by:
San Diego State University
5500 Campanile Drive
San Diego, CA 92182-4493

Report prepared by:
Su Yeon Han
Department of Geography, San Diego State University

August. 2017

Abstract

This project consists of two web-mapping applications. One is San Diego County Cancer Mapping Tool. The other one is San Diego Public Health Data Mapping Tool. The program source code for both applications is basically same excepting some of the parameters in the source code.

This project aims to develop a highly-interactive and user-friendly Web-based mapping application to visualize mortality and hospitalization data for 19 cancer sites in each Subregional Areas (SRAs) of San Diego County during 2010 to 2013. The application allows users to synchronously explore cancer mortality rates on the left side and socioeconomic and demographic factors on the right side, and help them to explore relationships between cancer outcomes and socioeconomic factors by computing and visualizing spatial and statistical correlations.

The program that I developed originally aimed to visualize cancer data, but the source code can also be used to visualize any other data such as diabetes, alzheimers, asthma and etc.

Overview

This project consists of two web-mapping applications. One is San Diego County Cancer Mapping Tool. The other one is San Diego Public Health Data Mapping Tool.

The program starts from here:
http://vision.sdsu.edu/health/ This website is the starting point.

Github repository: https://github.com/HDMA-SDSU/HealthMapper (uploaded on July 12, 2017)

San Diego County Cancer Mapping Tool : http://vision.sdsu.edu/su42/sdcancermap/

San Diego Public Health Mapping Tool : http://vision.sdsu.edu/su42/sdcancermap/health.html

The javascript codes of cancer.html and health.html are the same. Only colors of the maps are different. Please note that cancer.js is imported for the cancer mapping tool, and health.js is imported for the health mapping tool.

The source code for changing the map color is available here:

```
697         var html_left = "<div style='float:left;font-size:15px;'> Advanced Option </div>"+
698
699         "<div id='menu_L5' style='float:left;font-size:15px;margin-top:0.2cm;'> " +
700             "<form name='classSelect1' action=''>" +
701                 "<span id='class_change'> </span>" +
702                 "<select name='classified' onChange='layerChange1()'>" +
703                     "<option value='equal'>Equal Classification</option>" +
704                     "<option value='quantile' selected>Quantile Classification</option>" +
705                 "</select>" +
706             "</form>    " +
707         "</div>" +
708
709         "<div id='menu_L6' style='float:left;font-size:15px;margin-top:0.1cm;'>" +
710             "<form name='colorNumSelect1' action=''>" +
711                 "<span id='colorNum_change'>Color : </span>" +
712                 "<select name='colorNum' onChange='layerChange1()'>" +
713                     "<option value='Green'>Green</option>" +
714                     "<option value='Blue'>Blue</option>" +
715                     "<option value='Orange' selected>Orange</option>" +
716                     "<option value='Red'>Red</option>" +
717                     "<option value='Pink'>Purple</option>" +
718                 "</select>" +
719             "</form>    " +
720         "</div>" +
721
722         "<div id='menu_L7' style='float:left;font-size:15px;margin-top:0.1cm;'>" +
723             "<form name='classNumSelect1' action=''>" +
724                 "<span id='classNum_change'>Class: </span>" +
725                 "<select name='classNum' onChange='layerChange1()'>" +
726                     "<option value='5'>5</option>" +
727                     "<option value='6'>6</option>" +
728                     "<option value='7'>7</option>" +
729                     "<option value='8' selected>8</option>" +
730                     "<option value='9'>9</option>" +
731                 "</select>" +
732             "</form>    " +
733         "</div>"
```

This program does not have server side program. All the data that has been used were saved in the format of Jason in javascript files. These are the data:

polygon.js (coordinates of all vertices of polygons in GeoJSON)
cancer.js (cancer data in JSON)
health.js (Diabetes, Alzheimers, Asthma, Heart Disease, Stroke and Chronic Obstructive Pulmonary Disease in JSON)
header_descriptions.js (header description of cancer data in JSON)
category_descriptions_master.js (need for grouping socioeconomic and demographic data in Array)
CENSUS2010_v2.js (socioeconomic and demographic data 2010 in JSON)
CENSUS2011_v2.js (socioeconomic and demographic data 2011 in JSON)
CENSUS2012_v2.js (socioeconomic and demographic data 2012 in JSON)
CENSUS2013_v2.js (socioeconomic and demographic data 2013 in JSON)
NoShown.js (remove some of variables that can be chosen on the top of the map)

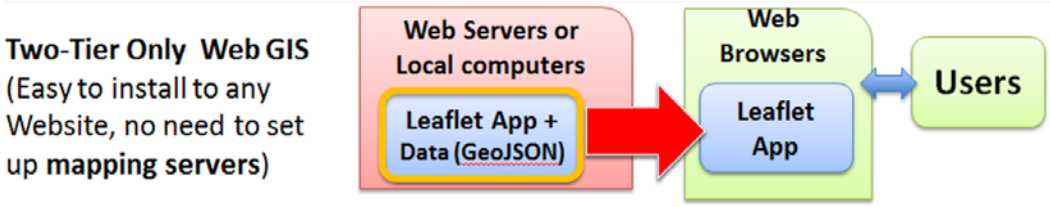


Figure 1. Input data are saved in GeoJSON format and included as javascript file

Links for the related libraries and websites

1. Dual Display mapping functions: Leaflet.Sync (https://github.com/turban/Leaflet.Sync).

```
659         var layer_1 = L.tileLayer('http://{s}.tile.stamen.com/toner-lite/{z}/{x}/{y}.png', stamenOptions);
660
661         var layer_2 = L.tileLayer('http://{s}.tile.stamen.com/toner-lite/{z}/{x}/{y}.png', stamenOptions);
662
663         var map1 = L.map('map1', {
664             layers: [layer_1],
665             center: center,
666             zoom: 10,
667         });
668
669         map1.scrollWheelZoom.disable();
670         map1.attributionControl.setPrefix('');
671
672         var map2 = L.map('map2', {
673             layers: [layer_2],
674             center: center,
675             zoomControl: false
676         });
677
678         map2.scrollWheelZoom.disable();
679         map2.attributionControl.setPrefix('');
680
681         map1.sync(map2);
682         map2.sync(map1);
```

2. Mouse Hover function: This comes from one of example from the leaflet library (http://leafletjs.com/examples/choropleth/)

```
        info1 = L.control();

        info1.onAdd = function (map1) {
            this._div = L.DomUtil.create('div', 'info');
            this.update();
            return this._div;
        };


        info1.update = function (props) {

            var dispItem = (item.length <= 40) ? item : item.substring(0,40-3) + " ...";

            this._div.innerHTML = (props ? dispItem + ' <br/>' +
                '<b>' + (!Number.isFinite(props[item]) || props[item] % 1 == 0 ? props[item] : props[item].toFixed(1)) +
                '</b><br/>'

                + '<br/>Area<br/>'
                + '<b>' + props.NAME + '</b>'
                : 'Hover over an area');
        };

        info1.addTo(map1);
```

3. Map Classification method functions

```
    if (classification == "equal") {

        var min = Number.MAX_VALUE;
        var max = Number.MIN_VALUE;
        for (var i=0; i<geojson.features.length; i++){
            if (geojson.features[i].properties[selectedCase] == "Data not shown") continue;
            try {
                value = parseFloat(geojson.features[i].properties[selectedCase]);
            } catch(e) {
                alert("cancer data invalid!!");
                return;
            }
            if (min > value) min = value;
            if (max < value) max = value;
        }

        if (min == Number.MAX_VALUE && max == Number.MIN_VALUE || min == max) classCount = 1;    // by

        var range = max - min;
        var interval = range / (classCount * 1.0);
        var intervals = new Array();
        intervals[0] = min * 1;
        for (var i=1; i<classCount; i++) {
            intervals[i] = intervals[i-1] * 1 + interval;
        }
        for (var i=0; i<classCount; i++) {

            intervals[i] = (Math.floor(intervals[i]*100)/100).toFixed(2);
        }

        ACSdata_render1(geojson, selectedCase, intervals, colorList);
    }

    if (classification == "quantile") {

        var values = new Array();
        var j=0;
        for (var i=0; i<geojson.features.length; i++) {
            if (geojson.features[i].properties[selectedCase] == "Data not shown") continue;
            value = parseFloat(geojson.features[i].properties[selectedCase]);
            values[j++] = value;
        }
        values.sort(function(a,b){return a-b});
        if (values.length < classCount) classCount = values.length;

        var interval = values.length / (classCount * 1.0);;

        var intervals = new Array();
        var next_interval = 0;

        intervals[0] = values[0];
        for (var i=1; i<classCount; i++) {
            next_interval += interval;
            j = Math.round(next_interval);

            intervals[i] = values[j];
        }
        for (var i=0; i<classCount; i++) {
            if (isFloat(intervals[i])) intervals[i] = (Math.floor(intervals[i]*100)/100).toFixed(2);
        }

        ACSdata_render1(geojson, selectedCase, intervals, colorList);
    }


}
```

4. correlation analysis display function: (https://gist.github.com/matt-west/6500993)

```
function getCorrleationValue(value1, value2) {
    var result = [0.0];
    var data = [];
    var data1 = [];
    var data2 = [];
    var idata1 = "";
    var idata2 = "";
    for (var key in value1) {
        idata1 += "'"+key+"':"+value1[key]+", ";
    }
    for (var key in value2) {
        idata2 += "'"+key+"':"+value2[key]+", ";
    }
    var sdata1 = "";
    var sdata2 = "";
    for (var key in value1) {
        if (key == 'selectedCase') {
            continue;
        }
        if (value1[key] == "Data not shown") continue;
        if (!(key in value2)) continue;
        if (value2[key] == "Data not shown") continue;
        sdata1 += "'"+key+"':"+value1[key]+", ";
        sdata2 += "'"+key+"':"+value2[key]+", ";
        data1.push(value1[key]);
        data2.push(value2[key]);
    }
    var data = [data1, data2];
    var correlationValue = pearsonCorrelation(data,0,1);
    result[0] = correlationValue.toFixed(2) * 1;
    result[1] = data1.length;
    result[2] = sdata1;
    result[3] = sdata2;
    result[4] = idata1;
    result[5] = idata2;
    return result;
}
```

5. export data function: AlaSQL (http://alasql.org/)

```
function exportData_left(dispValue1, dispValue2, allValue1, allValue2) {
    var csv_L_filename  = "";
    csv_L_filename  = document.CaseSelect.case.value + " of ";
    csv_L_filename += document.OutcomeSelect.outcome.value + " due to ";
    csv_L_filename += disease;
    csv_L_filename += " " + document.YearSelect1.year1.value + " and ";
    if ( document.NormalizationSelect.normalization.value == "none") {
        csv_L_filename += document.layerSelect.ACSdata.value  + " in ";
    }
    else if ( document.NormalizationSelect.normalization.value == "area") {
        csv_L_filename += document.layerSelect.ACSdata.value + " (mi²) ";
    }
    else {
        csv_L_filename += " " + document.layerSelect.ACSdata.value + " (%) ";
    }
    csv_L_filename += document.YearSelect2.year2.value + ".csv";
    //alert(csv_L_filename);
    alasql('SELECT * INTO CSV("'+csv_L_filename+'", {headers:false, separator:","}) FROM ?',[csv_L_list]);
}
```

6. Bar Chart: Google Bar Charts (https://developers.google.com/chart/interactive/docs/gallery/barchart)

7. Colorbrewer was used for the map color (http://colorbrewer2.org/)

8. Converter between JSON and CSV (http://www.convertcsv.com/json-to-csv.htm)

9. JQuery (https://jquery.com/)

Functionalities

Figure 2 and 3 show the interface of the web application. At the top of each map (Figure 2A), users can choose input data. With the data that the user selects in this section (Figure 2A), each of two maps (Figure

2B) and graphs (Figure 3B) are visualized. On the left panel, the user has an option to choose (1) 19 different cancer types, (2) outcomes (e.g. death or hospitalization), (3) the different types of mortality rate of the selected cancer such as total case, age adjusted rate, Hispanic case or White case, and (4) the year between 2010 to 2013 or 'Most up-to-date'. When the variable 'Most up-to-date' in the year option is selected, the map visualizes the most recently available cancer data during four years between 2010 and 2013. On the right panel, the user first has an option to choose one among the large seven categories of socioeconomic and demographic factors: gender, race, language, age group, education, economic factor and health factor, and within each seven categories, the user has an option to choose one specific variable. For example, when the user picks education among the large seven categories, in the category of education there are specific options such as population with master's degree, bachelor's degree, high school degree or etc. Categorizing 96 variables into seven groups allows users intuitively to choose one variable among 96 different socioeconomic and demographic data. At the top of the right panel, users also have an option for normalization using total population or the size of area. They also have an option to choose the year between 2010 and 2013, or the average of values of the selected socioeconomic and demographic variable in the same SRA region during the four years.

Map Visualization

Figure 2B is for map visualization. The left map is visualized with the user's option at the top of the left panel, and the right map is visualized with the user's option at the top of the right panel. In Figure 2B, as an example, the left map shows age-adjusted rate of lung cancer in 2013, and the right map shows the median total income of household($) in 2013. The data are separately visualized on the left panel and right panel, but the two maps are synchronized together. When the user moves one of each map by zooming in/out and panning, the two maps are zoomed in/out and move together. In addition, when users have mouse hovering on each of SRA region, the same regions on the two maps are highlighted together. The synchronized two maps allow users to examine the distributional relationship between cancer cases and socioeconomic and demographic factors.

On the left bottom corner of each map of Figure 2B, users have an option to differently design each choropleth map. The mapping function provides the two different classification methods – i.e. equal and quantile classification, different color scheme of the map and the number of classes of the choropleth mapping.

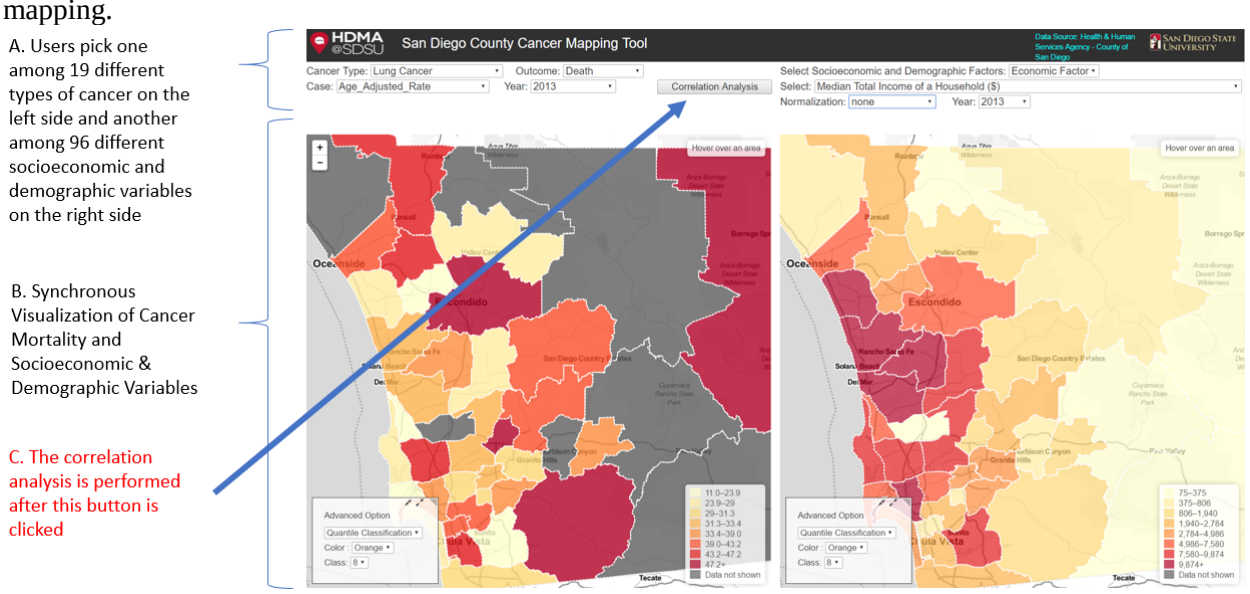


Figure2. The interface of the lightweight web mapping application. After users choose input data (A), the selected data are visualized on the maps (B) and the graphs (Figure 3). The map visualization of this image is one of examples. More data can be visualized on the maps by selecting one of 19 cancer types and one of 96 socioeconomic and demographic variables on the top of the interface.

Visualization of Correlation using Charts

The correlation (Pearson's r) is computed between one of cancer cases and the one of socioeconomic and demographic variables. For example, in case of Figure 3A, after the user clicks the button 'Correlation Analysis', the program computes the Pearson's r between the age-adjusted death rate of lung cancer in 2013 and the median total income of a household. The interface shows the correlation (Pearson's r) result in the alert box as shown in Figure 3A. After the user clicks the button 'CLICK TO SEE MORE CORRELATION RESULTS', graphs in Figure 3 is shown. In Figure 3, the left panel shows the correlation between age_adjusted death rate of each 19 different cancer type in 2013 and median total income of a household in 2013, and the right panel shows the correlation between age_adjusted death rate of lung cancer 2013 and each 96 different socioeconomic and demographic variable in 2013.

Figure 3. Visualization of correlation results. With the selected data, the correlations are visualized in in the graph. The result above is one of examples among results that can be found by using this application.

More data can be visualized on the graph by selecting each 19 different cancer types and different 96 socioeconomic and demographic variables on the top of the interface.

We developed a function to limit the options when users choose input variables at the top of the application (Figure 2A). We have 19 different cancer types and 96 different socioeconomic and demographic variables. In this case, users can technically have 19 x 96 = 1,824 combinations to examine the correlation. However, we found that in some cases, for example, it is not meaningful to compute the correlation between cancer death rate of population between age group between 0 to 14 and the number of population over age 25 with Master's degree. This application does not allow users to choose the meaningless combinations of input data. To achieve this, we made a function that reads elements of an array (Figure 4) that has variables of the meaningless combination and do not display them together on the top of the map. The first row of the array means that for all cancer type, in case of 'white cases', there is no option to choose different races such as Hispanic, Asian or Black. The last row of the array represents that for female breast cancer, there is no option to choose male population. In this case, male cannot have female breast cancer, so it is not meaningful to examine the correlation between the two variables.

```
var NO_SHOWNS = [
    ["*",                    "White_Cases",                  "Race",          "*",                "x"],
    ["*",                    "White_Rate",                   "Race",          "*",                "x"],
    ["*",                    "Black_Cases",                  "Race",          "*",                "x"],
    ["*",                    "Black_Rate",                   "Race",          "*",                "x"],
    ["*",                    "Hispanic_Cases",               "Race",          "*",                "x"],
    ["*",                    "Hispanic_Rate",                "Race",          "*",                "x"],
    ["*",                    "Asian_Pacific_Islander_Cases", "Race",          "*",                "x"],
    ["*",                    "Asian_Pacific_Islander_Rate",  "Race",          "*",                "x"],
    ["*",                    "Other_Race_Ethnicity_Cases",   "Race",          "*",                "x"],
    ["*",                    "Other_Race_Ethnicity_Rate",    "Race",          "*",                "x"],
    ["*",                    "Male_Cases",                   "Race",          "*",                "x"],
    ["*",                    "Male_Rate",                    "Race",          "*",                "x"],
    ["*",                    "Female_Cases",                 "Race",          "*",                "x"],
    ["*",                    "Female_Rate",                  "Race",          "*",                "x"],
    ["*",                    "Age_Group_0_to_14_Cases",      "Race",          "*",                "x"],
    ["*",                    "Age_Group_0_to_14_Rate",       "Race",          "*",                "x"],
    ["*",                    "Age_Group_15_to_24_Cases",     "Race",          "*",                "x"],
    ["*",                    "Age_Group_15_to_24_Rate",      "Race",          "*",                "x"],
    ["*",                    "Age_Group_25_to_44_Cases",     "Race",          "*",                "x"],
    ["*",                    "Age_Group_25_to_44_Rate",      "Race",          "*",                "x"],
    ["*",                    "Age_Group_45_to_64_Cases",     "Race",          "*",                "x"],
    ["*",                    "Age_Group_45_to_64_Rate",      "Race",          "*",                "x"],
    ["*",                    "Age_Group_65Plus_Cases",       "Race",          "*",                "x"],
    ["*",                    "Age_Group_65Plus_Rate",        "Race",          "*",                "x"],
    ["*",                    "Age_Adjusted_Rate",            "Age",           "*",                "x"],
    ["*",                    "Age_Group_0_to_14_Cases",      "Age",           "*",                "x"],
    ["*",                    "Age_Group_0_to_14_Rate",       "Age",           "*",                "x"],
    ["*",                    "Age_Group_15_to_24_Cases",     "Age",           "*",                "x"],
    ["*",                    "Age_Group_15_to_24_Rate",      "Age",           "*",                "x"],
    ["*",                    "Age_Group_25_to_44_Cases",     "Age",           "*",                "x"],
    ["*",                    "Age_Group_25_to_44_Rate",      "Age",           "*",                "x"],
    ["*",                    "Age_Group_45_to_64_Cases",     "Age",           "*",                "x"],
    ["*",                    "Age_Group_45_to_64_Rate",      "Age",           "*",                "x"],
    ["*",                    "Age_Group_65Plus_Cases",       "Age",           "*",                "x"],
    ["*",                    "Age_Group_65Plus_Rate",        "Age",           "*",                "x"],
    ["*",                    "Age_Group_0_to_14_Cases",      "Education",     "*",                "x"],
    ["*",                    "Age_Group_0_to_14_Rate",       "Education",     "*",                "x"],
    ["*",                    "Age_Group_15_to_24_Cases",     "Education",     "*",                "x"],
    ["*",                    "Age_Group_15_to_24_Rate",      "Education",     "*",                "x"],
    ["Female Breast Cancer", "*",                            "Gender",        "Male Population",  "x"],
];
```

Figure 4. The array showing meaningless combination of some of cancer cases and some of socioeconomic and demographic variables. Each column corresponds to each input parameter option in Figure 2A. The first column corresponds to the option for the 'Cancer Type', the second column corresponds to the option of 'Case, the third column correspond to the option of 'Select Socioeconomic and Demographic Factors', the fourth column corresponds to the option for 'Select', and the last column correspond to the option for 'Normalization'. "*" represents all variables within the option. "x" represents that it is wrong combination, so the combination of the selected input parameters will not be

displayed.

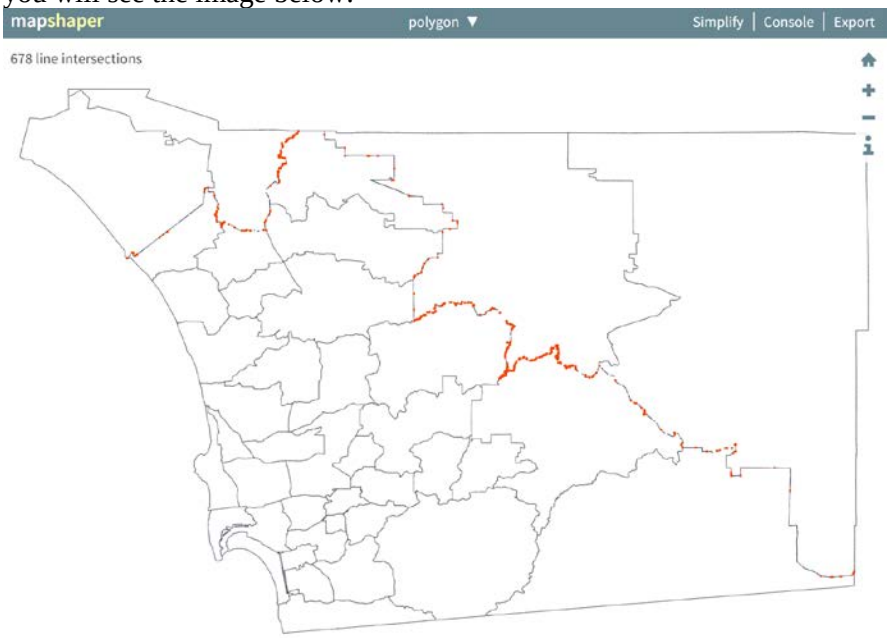
Data Conversion from Shapefile to GeoJSON

-polygon.js is San Diego Subregional Area. This file was converted from shapefile to Geojson. To convert from shapefile to geojson, you can use the Mapshaper: http://www.mapshaper.org/
The Mapshaper allows users (1) upload the data in Shapefile, GeoJSON, or TopoJSON, (2) simplify the polylines to reduce the file size, and (3) export the data in whatever the data format you want (Shapefile, GeoJSON, or TopoJSON). Therefore, you can covert shapefile to geojson by uploading and downloading data.

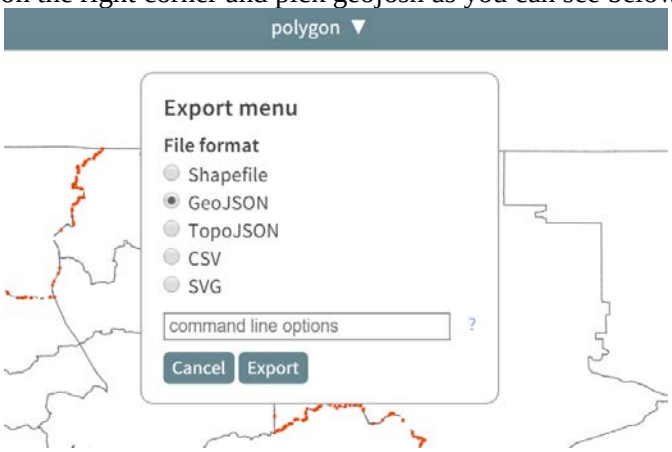
-I put the shapefile of San Diego SRA region (data/polygon.zip) in the folder, data. I used poplygon.zip and converted it to geojson and made popygon.js

-The procedure for the conversion is simple.

1. Drag and Drop polygon.zip (polygon.dbf, polygon.shp, polygon.shx files are together). Then, you will see the image below:



2. Click Export button on the right corner and pick geojosn as you can see below.



All the other files, cancer.js, header_descriptions.js, CENSUS2010_v2.js, CENSUS2011_v2.js ,CENSUS2012_v2.js ,CENSUS2013_v2.js has the data in JSON format. These files were converted from csv. In the folder, data, all the data are saved in CSV file. You can convert CSV file to geojson using the tool here: http://www.convertcsv.com/csv-to-json.htm

Depending on your data, you will have to modify the HTML code below and javascript code accordingly:
This is for the left map

```
<div id="menu_L1" style="background-color:#ffffff;width:365px;float:left;margin-bottom:0.1cm;">
    <form name="ConditionSelect" action="">
        <span id="condition_change">Cancer Type: </span>
        <select name="condition" id="cancer_type" onChange="layerChange1()">
        </select>
    </form>
</div>

<div id="menu_L2" style="background-color:#ffffff;width:245px;float:left;margin-bottom:0.1cm;">
    <form name="OutcomeSelect" action="">
        <span id="outcome_change">Outcome: </span>
        <select name="outcome" onChange="layerChange1()">
            <option value="Death">Death</option>
            <option value="Hospitalization">Hospitalization</option>
            <option value="ED Discharge">ED Discharge</option>
        </select>
    </form>
</div>

<div id="menu_L3" style="background-color:#ffffff;width:345px;float:left;margin-bottom:0.1cm;">
    <form name="CaseSelect" action="">
        <span id="case_change">Case: </span>
        <select name="case" id="cancer_case" onChange="layerChange1()">
        </select>
    </form>
</div>

<div id="menu_L4" style="background-color:#ffffff;width:250px;float:left;margin-bottom:0.1cm;">
    <form name="YearSelect1" action="">
        <span id="year_change">Year: </span>
        <select name="year1" onChange="layerChange1()">
            <option value="2010" >2010</option>
            <option value="2011">2011</option>
            <option value="2012">2012</option>
            <option value="2013" selected>2013</option>
            <option value="Most Up-to-Date" >Most up-to-date</option>
        </select>
    </form>
</div>
```

This is for the right map

```
409         <div id="menu_R0" style="background-color:#ffffff;margin-bottom:0.1cm;">
410             <form name="layerSelect0" action="">
411                 <span id="layer_change0">Select Socioeconomic and Demographic Factors:</span>
412                 <select name="ACSdata0" id="ACSdata_SDFgroup" onChange="yearChange2()">
413                 </select>
414             </form>
415         </div>
416
417         <div id="menu_R1" style="background-color:#ffffff;margin-bottom:0.1cm;">
418
419             <form name="layerSelect" action="">
420                 <span id="layer_change">Select: </span>
421                 <select name="ACSdata" id="ACSdata_options" onChange="layerChange2()">
422                 </select>
423             </form>
424         </div>
425
426         <div id="menu_R3" style="background-color:#ffffff;width:300px;float:left;margin-bottom:0.1cm;">
427             <form name="NormalizationSelect" action="">
428                 <span id="normalization_change">Normalization: </span>
429                 <select name="normalization" id="normalization_options" onChange="layerChange2()">
430                     <option value="none">none</option>
431                     <option value="Total Population" selected>Total Population</option>
432                     <option value="area">Area</option>
433                 </select>
434             </form>
435         </div>
436
437         <div id="menu_R2" style="background-color:#ffffff;width:140px;float:left;margin-bottom:0.1cm;">
438             <form name="YearSelect2" action="">
439                 <span id="year_change">Year: </span>
440                 <select name="year2" onChange="yearChange2()">
441                     <option value="2010" >2010</option>
442                     <option value="2011">2011</option>
443                     <option value="2012">2012</option>
444                     <option value="2013" selected>2013</option>
445                     <option value="Average">Average</option>
446                 </select>
447             </form>
448         </div>
```

Instructions below include some tutorials to show how you can reuse the source code and visualize your own data by using HealhMapper. As an example, I am going to visualize the data below.

ate_stage_dx_SD_Imperical_fake.js (fake cancer data in San Diego and Imperial county at MSA level)
MSSA_SD_Imperical.js (polygons representing the boundary of each MSA region in San Diego and Imperial County)
MSSA_ACS_SD_Imperical_simple.js (socioeconomic and demographic variables)

Please note that the data used in this section are not real data. The data were randomly created to show how to replace the data and reuse the source code.

First copy index.html and change the file name to SanDiego_Imperial.html. Start to modify SanDiego_imperial.html like the instructions below. The final version of this source code was saved with the name “SanDiego_Imperial_final.html”

1 Change the title at the top of the browser

```
7       <title>San Diego County Cancer Mapping Viewer</title>
```

San Diego County Imperial County Cancer Mapping Viewer

2 Change title at the top of the application

```
345     <div style = "margin:12px 0px 0px 0px;float:left;">
346         <font size="5" color="white">     San Diego County Cancer Mapping Tool</font>
347     </div>
```

3 Change the label of the top left panel in the image

To change the label of the top of the left panel as you can see right above, you need change from the source code below

```
<div id="menu_L1" style="background-color:#ffffff;width:365px;float:left;margin-bottom:0.1cm;">
    <form name="ConditionSelect" action="">
        <span id="condition_change">Cancer Type: </span>
        <select name="condition" id="cancer_type" onChange="layerChange1()">
        </select>
    </form>
</div>

<div id="menu_L2" style="background-color:#ffffff;width:245px;float:left;margin-bottom:0.1cm;">
    <form name="OutcomeSelect" action="">
        <span id="outcome_change">Outcome: </span>
        <select name="outcome" onChange="layerChange1()">
            <option value="Death">Death</option>
            <option value="Hospitalization">Hospitalization</option>
            <option value="ED Discharge">ED Discharge</option>
        </select>
    </form>
</div>

<div id="menu_L3" style="background-color:#ffffff;width:345px;float:left;margin-bottom:0.1cm;">
    <form name="CaseSelect" action="">
        <span id="case_change">Case: </span>
        <select name="case" id="cancer_case" onChange="layerChange1()">
        </select>
    </form>
</div>

<div id="menu_L4" style="background-color:#ffffff;width:250px;float:left;margin-bottom:0.1cm;">
    <form name="YearSelect1" action="">
        <span id="year_change">Year: </span>
        <select name="year1" onChange="layerChange1()">
            <option value="2010" >2010</option>
            <option value="2011">2011</option>
            <option value="2012">2012</option>
            <option value="2013" selected>2013</option>
            <option value="Most Up-to-Date" >Most up-to-date</option>
        </select>
    </form>
</div>
```

to the code below:

```
<div id="menu_L1" style="background-color:#ffffff;width:365px;float:left;margin-bottom:0.1cm;">
    <form name="ConditionSelect" action="">
        <span id="condition_change">Cancer Type: </span>
        <select name="condition" id="cancer_type" onChange="layerChange1()">
        </select>
    </form>
</div>

<div id="menu_L2" style="background-color:#ffffff;width:245px;float:left;margin-bottom:0.1cm;">
    <form name="OutcomeSelect" action="">
        <span id="outcome_change">Sex: </span>
        <select name="outcome" onChange="layerChange1()">
            <option value="Male" selected>Male</option>
            <option value="Female">Female</option>
        </select>
    </form>
</div>

<div id="menu_L3" style="background-color:#ffffff;width:345px;float:left;margin-bottom:0.1cm;">
    <form name="CaseSelect" action="">
        <span id="case_change">Case: </span>
        <select name="case" id="cancer_case" onChange="layerChange1()">
        </select>
    </form>
</div>

<div id="menu_L4" style="background-color:#ffffff;width:250px;float:left;margin-bottom:0.1cm;">
    <form name="YearSelect1" action="">
        <span id="year_change">Age group: </span>
        <select name="year1" onChange="layerChange1()">
            <option value="15-19">15-19</option>
            <option value="20-24">20-24</option>
            <option value="25-29">25-29</option>
            <option value="30-34">30-34</option>
            <option value="35-39">35-39</option>
            <option value="40-44">40-44</option>
            <option value="45-49">45-49</option>
            <option value="50-54">50-54</option>
            <option value="55-59">55-59</option>
            <option value="60-64" selected>60-64</option>
            <option value="65-69">65-69</option>
            <option value="70-74">70-74</option>
            <option value="75-79">75-79</option>
            <option value="80-84">80-84</option>
            <option value="85+">85+</option>
        </select>
    </form>
</div>
```

4 Change the label of the top of the right panel in the image below.

To change the label of the top of the right panel as you can see right above, you need change from the source code below:

```
<div id="menu_R2" style="background-color:#ffffff;width:140px;float:left;margin-bottom:0.1cm;">
    <form name="YearSelect2" action="">
        <span id="year_change">Year: </span>
        <select name="year2" onChange="yearChange2()">
            <option value="2010" >2010</option>
            <option value="2011">2011</option>
            <option value="2012">2012</option>
            <option value="2013" selected>2013</option>
            <option value="Average">Average</option>
        </select>
    </form>
</div>
```

To the code below:

```
484         <div id="menu_R2" style="background-color:#ffffff;width:140px;float:left;margin-bottom:0.1cm;display:none;">
485             <form name="YearSelect2" action="">
486                 <span id="year_change">Year: </span>
487                 <select name="year2" onChange="yearChange2()">
488                     <option value="2010">2010</option>
489                 </select>
490             </form>
491         </div>
```

Because ACS data have only one year. “menu_R2” does not need to be displayed. So “display:none” was inserted.

5 Change Input data from the code below:

```
487     <script type="text/javascript" src="polygon.js"></script>
488     <script type="text/javascript" src="cancer.js"></script>
489     <script type="text/javascript" src="header_descriptions.js"></script>
490     <script type="text/javascript" src="category_descriptions_master.js"></script>
491     <script type="text/javascript" src="CENSUS2010_v2.js"></script>
492     <script type="text/javascript" src="CENSUS2011_v2.js"></script>
493     <script type="text/javascript" src="CENSUS2012_v2.js"></script>
494     <script type="text/javascript" src="CENSUS2013_v2.js"></script>
495     <script type="text/javascript" src="NoShown.js"></script>
```

to the code below:

```
487     <script type="text/javascript" src="MSSA_SD_Imperical.js"></script>
488     <script type="text/javascript" src="late_stage_dx_SD_Imperical.js"></script>
489     <script type="text/javascript" src="header_descriptions.js"></script>
490     <script type="text/javascript" src="category_descriptions_master.js"></script>
491     <script type="text/javascript" src="MSSA_ACS_SD_Imperical_simple.js"></script>
492     <script type="text/javascript" src="NoShown.js"></script>
```

6 Change variable names of polygon data

Variables names in the polygon file are different between old version and new version. Thus, variable names in the input data need to be changed.

	Old Version	New Version
File name	Polygon.js	MSSA_SD_Imperical.js
Variable name	var CA=	var statesData =
FeatureCollection	"features":[{"type":"Feature","geometry":	"features":[{"type":"Feature","geometry":
properties	"properties":{	"properties":{
	"MSSA_ID":"161b"	"SRA":"63"
	"MSSA_NAME": "Brawley/Westmorland"	"SRA_Name": "Anza-Borrego Springs"

The source below changes from the variables in the old version to the variables in the new version in the table above.

```
498         var CA = statesData;        // from MSSA_SD_Imperical.js
499         for (var j=0; j<CA.features.length; j++) {
500             CA.features[j].properties["SRA"] = CA.features[j].properties["MSSA_ID"];
501             CA.features[j].properties["SRA_Name"] = CA.features[j].properties["MSSA_NAME"].split('/')[0]
502             //console.log(CA.features[j].properties.SRA + ":  " + CA.features[j].properties.SRA_Name);
503         }
```

7 Change variable names of cancer data

Variables names of cancer data are different between old version and new version. Thus, variable names in the input data need to be changed.

	Old Version	New Version
filename	Cancer.js	late_stage_dx_SD_Imperical.js
Variable name	var CANCER=	var CANCER_SD_Imperical =
	[0] CONDITION	[2] ca_type
	[1] OUTCOME	[1] SEX
	[2] YEAR	[4] agegrp
	[3] Geography	
	[4] SRANum	[0] mssa
	.	
	.	
	.	
	[13] Total_Cases	[6] case
	[14] Total_Rate	[7] proportion
	.	
	.	

The source below changes from the variables in the old version to the variables in the new version in the table above.

```
var CANCER = [];        // from late_stage_dx_SD_Imperical.js
CANCER.push(["CONDITION","OUTCOME","YEAR","","SRANum","","","","","","","","","case","proportion"]);
for (var j=1; j<CANCER_SD_Imperical.length; j++) {
    var col = CANCER_SD_Imperical[j];
    CANCER.push([col[2],col[1],col[4],"",col[0],"","","","","","","","",col[6],col[7]]);
}
```

7 Variables in the category_description_master.js need to be updated like below:

```
CATEGORY_DESCRIPTIONS_MASTER =
{
    "2010": {
        "Race": ["HISPANIC","WHITE","BLACK","NHS_BLACK","ASIAN"],
        "Age": ["AGE_65OVER","AGE_18_64","AGE_UNDR18","AGE_UNDER5"],
        "Health Factor": ["DENTIST"]
    },
    "2011": {},
    "2012": {},
    "2013": {},
    "Average": {},
}
```

8 Variables in the header_descriptions.js need to be updated like below:

```
HEADER_DESCRIPTIONS =
{
    "2010":["AREA_SQMI","POP","DENTIST","HISPANIC","WHITE","BLACK","NHS_BLACK","ASIAN","AGE_65OVER","AGE_18_64",
    "AGE_UNDER18","AGE_UNDER5"],
    "2011":[],
    "2012":[],
    "2013":[],
    "Definition":["area",
                  "Total Population",
                  "The number of dentists",
                  "The number of Hispanic Population",
                  "The number of WHITE Population",
                  "The number of BLACK Population",
                  "The number of non-Hispanic Population",
                  "The number of ASIAN Population",
                  "Population Age 65 and older",
                  "Population Age 18 to 64",
                  "Population Age under 18",
                  "Population Age under 5"
                  ]
}
```

9 Change variable names of census data

Variables names in the Census data are different between old version and new version. Thus, variable names in the input data need to be changed.

	Old Version	New Version
File name	CENSUS2010_v2.js, 2011, 2012, 2013	MSSA_ACS_SD_Imperical_simple.js
Variable name	var CENSUS2010=	var MSSA_ACS_SD_Imperical =
	[0] Year	“2010”
	[1] SRA Name	[4] MSSA_NAME
	[2] SRA Num	[0] MSSA_ID
	[3] Region Num	“”
	[4] Region Name	“”
	[5] T0610_TOTPOP	[6] AREA_SQMI
	[6] T0610_MALE	[7] POP
	.	[8] DENTIST
	.	[9] HISPANIC
	.	.
	.	.
	[end] area	[17] AGE_UNDER5

The source below changes from the variables in the old version to the variables in the new version in the table above.

```
556         var CENSUS2010 = [];        // from CENSUS2010_v2.js
557         var CENSUS2011 = [];        // from CENSUS2011_v2.js
558         var CENSUS2012 = [];        // from CENSUS2012_v2.js
559         var CENSUS2013 = [];        // from CENSUS2013_v2.js
560         var CENSUSaverage = [];
561         CENSUS2010.push(["Year","SRA Name","SRA Num","Region Num","Region Name","AREA_SQMI","POP","DENTIST","HISPANIC",
            "WHITE","BLACK","NHS_BLACK","ASIAN","AGE_65OVER","AGE_18_64","AGE_UNDER18","AGE_UNDER5"]);
562         CENSUS2011.push([]);        // no header
563         CENSUS2012.push([]);        // no header
564         CENSUS2013.push([]);        // no header
565         CENSUSaverage.push([]);     // no header
566         for (var j=1; j<MSSA_ACS_SD_Imperical.length; j++) {
567             var col = MSSA_ACS_SD_Imperical[j];
568             CENSUS2010.push(["2010",col[4],col[0],"","",col[6],col[7],col[8],col[9],col[10],col[11],col[12],col[13],col[
                14],col[15],col[16],col[17]]);
569         }
```

10 This program does not need to use the function “no show”. So two variables below need to be initialized.

```
575         MUST_REMOVE_SDFs = [];
576         NO_SHOWNS = [];
```

11 Comment out the code below because they do not need to be used in this program. The 4 year average does not need to be calculated in this program.

```
644         /*
645         // make most up-to-date cancer data of 4 year (2010, 2011, 2012, 2013) -> CANCER
646         makeUptodateCancerData();
647         // make average cancer data of 4 year (2010, 2011, 2012, 2013) -> CANCER
648         makeAverageCancerData();
649         // make average census data of 4 year (2010, 2011, 2012, 2013) -> CENSUSaverage
650         var CENSUSaverage = [];
651         makeAverageCensusData();
652         */
```

12 The code below shows how to create html tag in both left and left panel of the top of the map

```
667         var CANCER_DESCRIPTIONS = getNewCANCERs(document.layerSelect0.ACSdata0.value, document.layerSelect.ACSdata.value);
668         //setSelectField(Object.keys(CANCER_DESCRIPTIONS), 0, "cancer_type", "Lung Cancer");
669         setSelectField(Object.keys(CANCER_DESCRIPTIONS), 0, "cancer_type", "CRC");
670         //setSelectField(CANCER_DESCRIPTIONS[document.ConditionSelect.cancer_type.value], 0, "cancer_case", "Total_Rate");
671         setSelectField(CANCER_DESCRIPTIONS[document.ConditionSelect.cancer_type.value], 0, "cancer_case", "proportion");
672
673         // Reset Socio-economic and Demographic Factors (SDF) using Left_Map_Type and Left_Map_Case.
674         CATEGORY_DESCRIPTIONS = getNewSDFs(document.ConditionSelect.condition.value, document.CaseSelect.case.value);
675         //setSelectField(getSDFgroups(document.YearSelect2.year2.value), 0, "ACSdata_SDFgroup", "Economic Factor");
676         setSelectField(getSDFgroups(document.YearSelect2.year2.value), 0, "ACSdata_SDFgroup");
```

13. Lastly, you need to change the title of the two charts. The source of this part can be easily searched using the phrase “The Correlation (Pearson’s r) between”. You can change it in the javascript code, not in the html code