Demographic Data Explorer

GEOG 678 PROJECT

FINAL PRESENTATION

APRIL 27, 2021

TRUPTI LOKHANDE
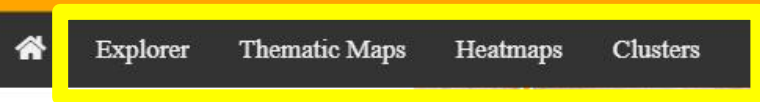
Demographic Data Explorer D2E

➤ D2E is an interactive web map application for visualization of demographic data of southeast Michigan region, built using Leaflet, an open source JavaScript library. It provides four options to visual data;

- ➤ Explorer
- ➤ Thematic Maps
- ➤ Heatmaps
- ➤ Clusters

➤ **PROJECT GOAL:**

The goal of this project is to create an interactive web map to display socio-economic data for cities in southeast Michigan. The thematic maps and heatmaps can help viewers to understand the pattern of socio-economic status in southeast Michigan.
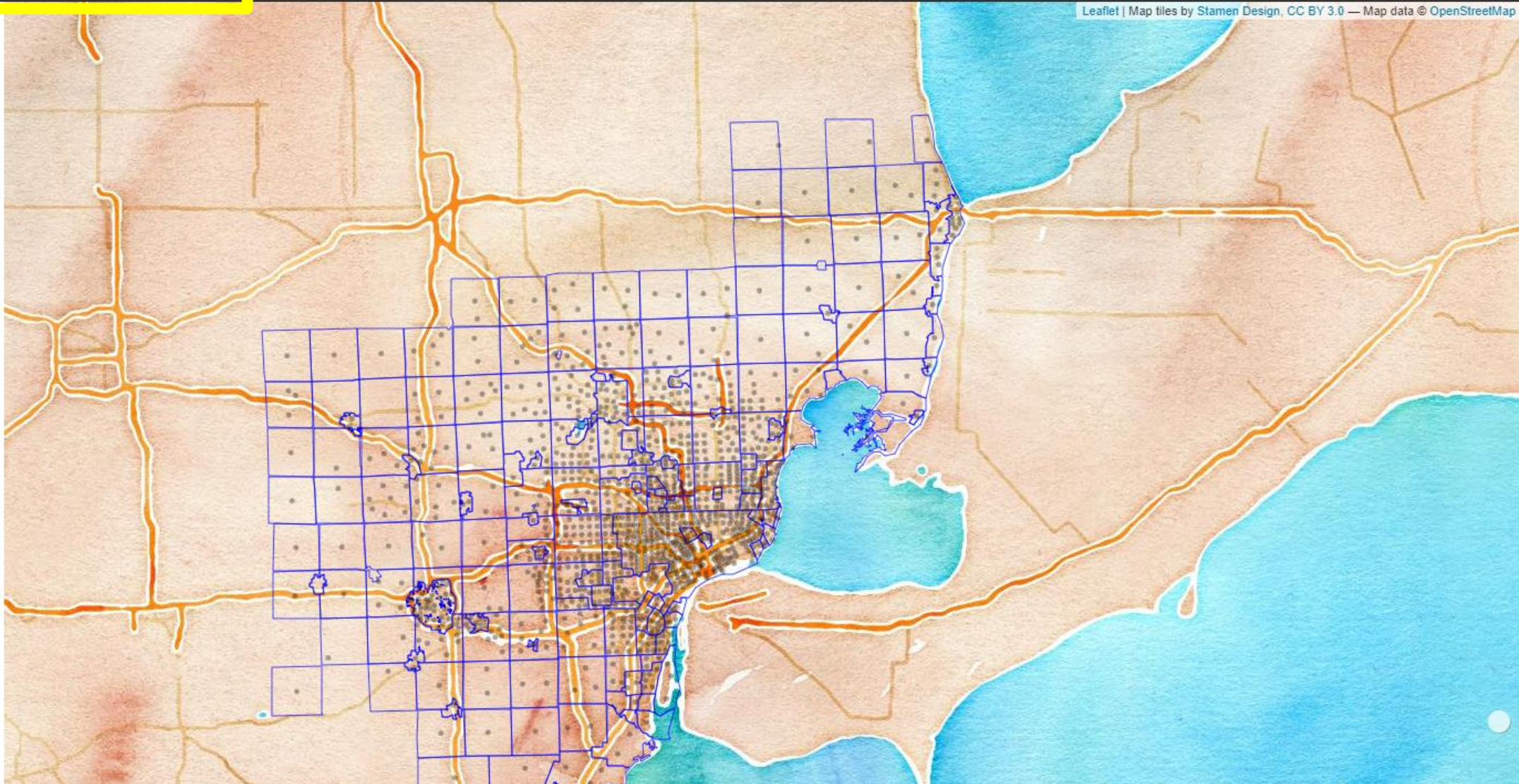
DEMOGRAPHIC DATA EXPLORER

Leaflet | Map tiles by Stamen Design, CC BY 3.0 — Map data © OpenStreetMap

Zoom to

Close

*Map does not show all cities in southeast Michigan.

(no city selected)

Technical Summary:

➢ Languages:
  ➢ Hypertext Markup Language (HTML)
  ➢ Cascading Style Sheets (CSS)
  ➢ JavaScript (JS)

➢ Data Formats:
  ➢ GeoJson
  ➢ Json

➢ Libraries:
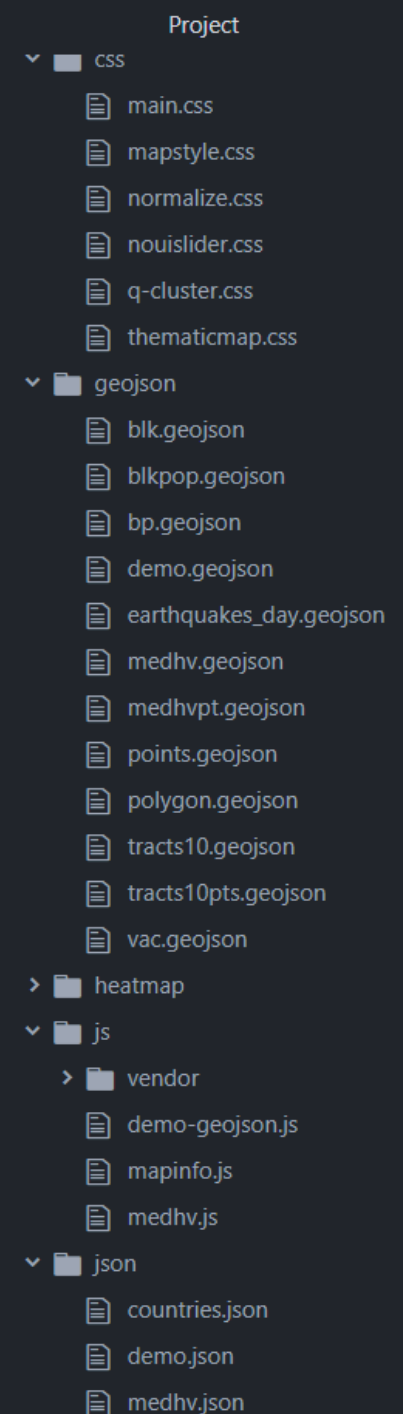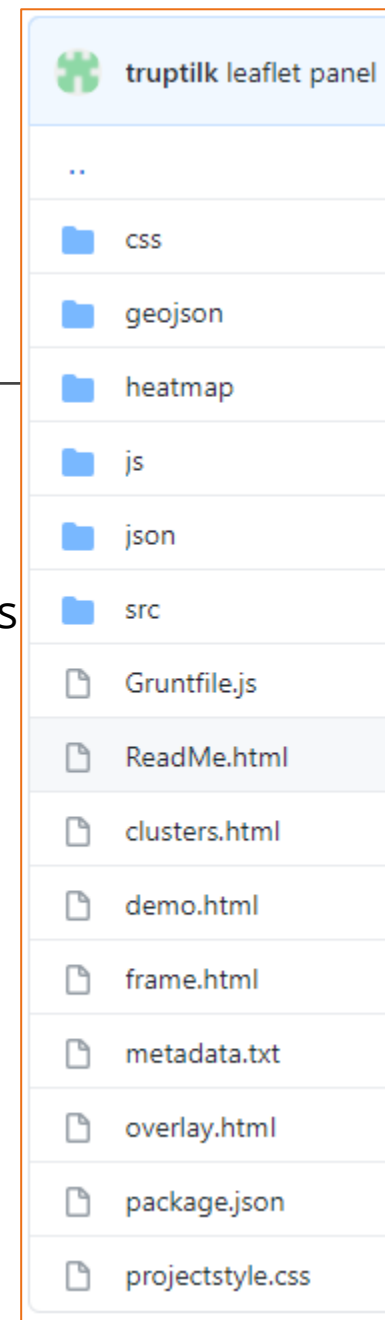 ➢ Leaflet

➢ Leaflet Plugins:
  ➢ Leaflet.markercluster
  ➢ Plugins for loading basemaps
  ➢ Heatmap.js

➢ Website Hosting Service:
  ➢ GitHub Pages

➢ Database:
  ➢ GitHub Repository

---

truptilk leaflet panel la

..

📁 css
📁 geojson
📁 heatmap
📁 js
📁 json
📁 src
📄 Gruntfile.js
📄 ReadMe.html
📄 clusters.html
📄 demo.html
📄 frame.html
📄 metadata.txt
📄 overlay.html
📄 package.json
📄 projectstyle.css

Project
∨ 📁 css
    📄 main.css
    📄 mapstyle.css
    📄 normalize.css
    📄 nouislider.css
    📄 q-cluster.css
    📄 thematicmap.css
∨ 📁 geojson
    📄 blk.geojson
    📄 blkpop.geojson
    📄 bp.geojson
    📄 demo.geojson
    📄 earthquakes_day.geojson
    📄 medhv.geojson
    📄 medhvpt.geojson
    📄 points.geojson
    📄 polygon.geojson
    📄 tracts10.geojson
    📄 tracts10pts.geojson
    📄 vac.geojson
  > 📁 heatmap
∨ 📁 js
  > 📁 vendor
    📄 demo-geojson.js
    📄 mapinfo.js
    📄 medhv.js
∨ 📁 json
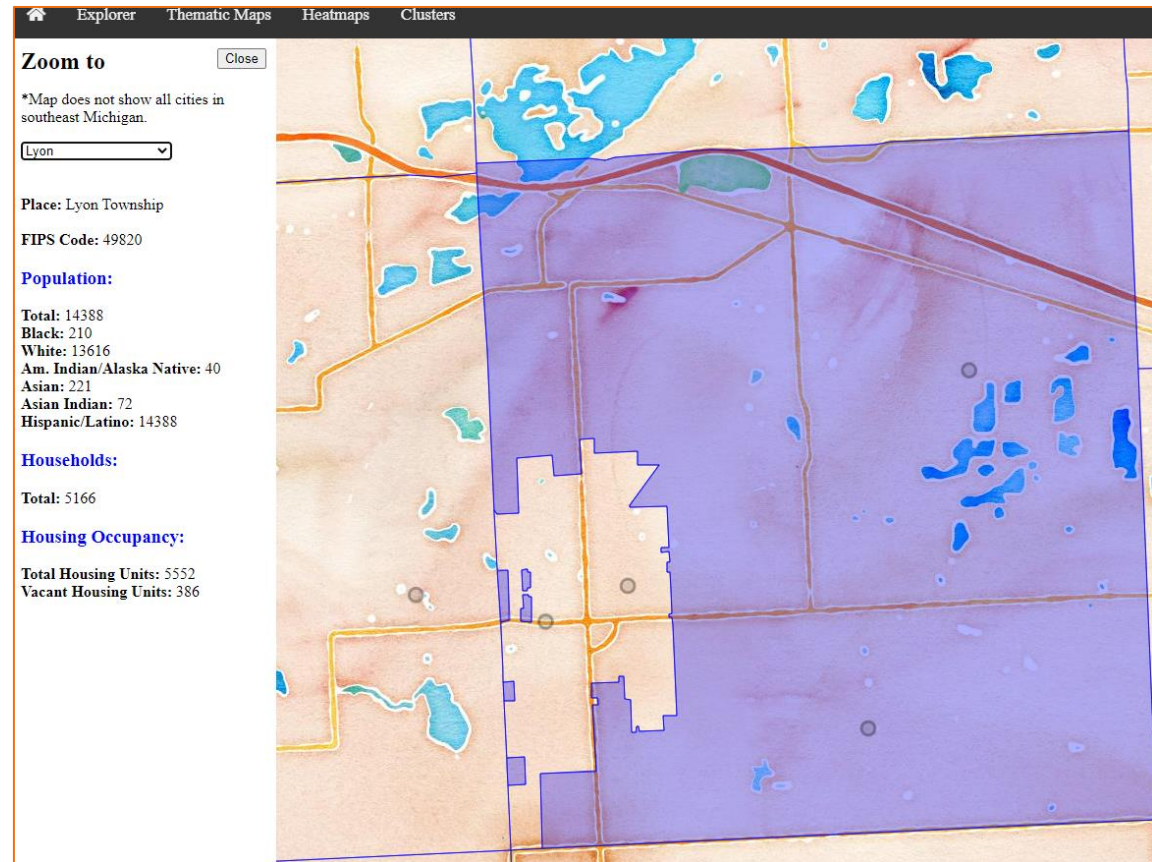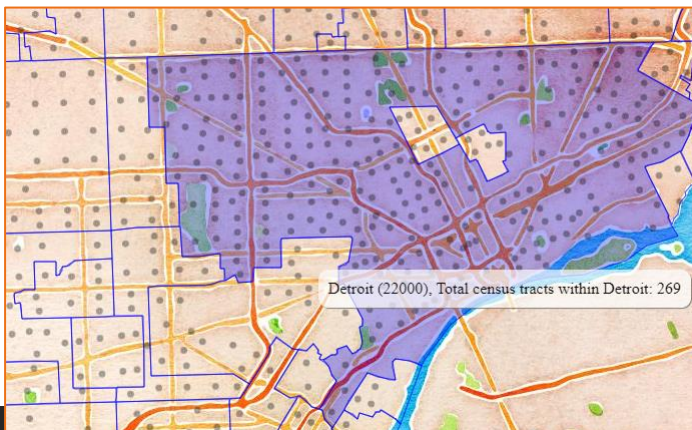   📄 countries.json
    📄 demo.json
    📄 medhv.json

Explorer

➢ Main Functions:

➢ Select City from Dropdown List:

➢ Map zooms to and highlight the selected city.

➢ Side panel displays the demographic information for the selected city.

➢ Hover over the Map:

➢ Get quick view of city name, FIPS code, and number of census tracts within the city.

Thematic Maps

➢ Main Functions:

➢ Use overlay control to switch between basemaps and thematic maps.

➢ The thematic maps are created using standardized data:

➢ Black/African American Population (BLKPOP) Thematic Map

➢ Data: BLKPOP as Proportion of total population of the census tract.

➢ Vacant Housing Units (VAC) Thematic Map

➢ Data: VAC as Proportion of total housing units within the census tract.
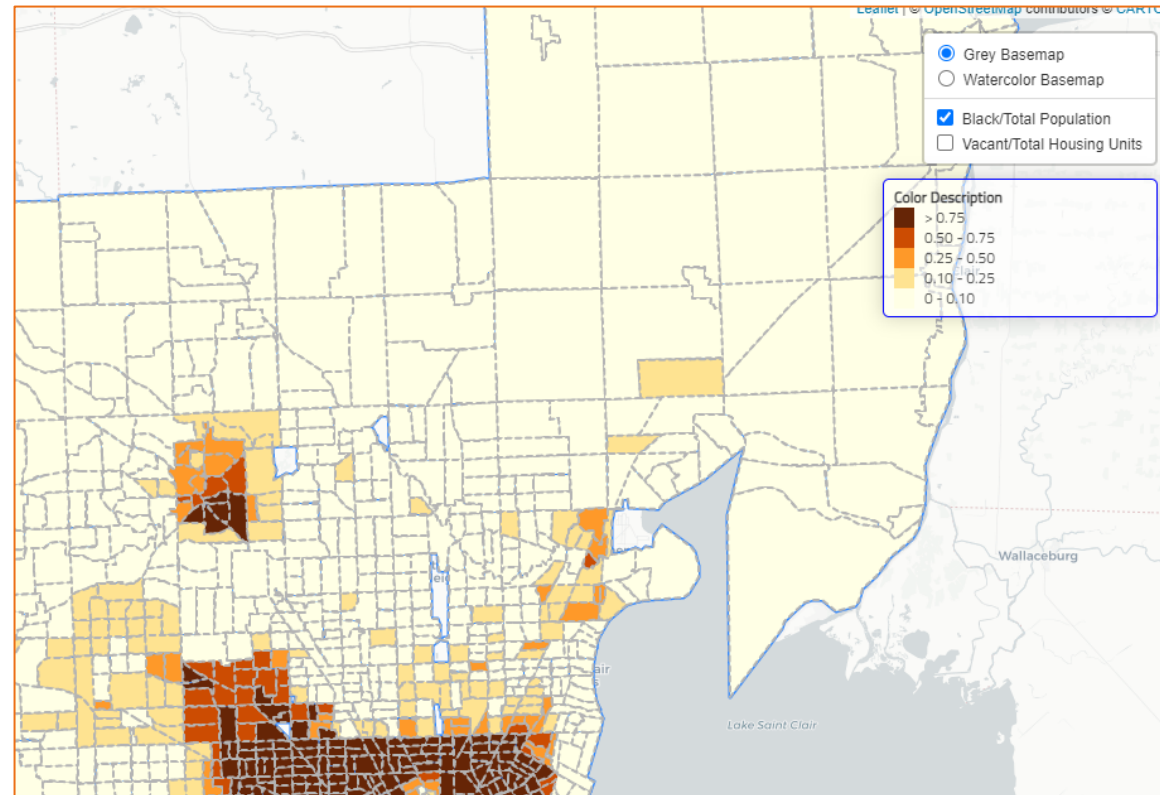
Heatmaps

➢ Main Functions:

  ➢ Use hyperlinks to switch between heatmaps.

 ➢ Use zoom and pan controls to navigate on maps.

Heatmap of Black/ African American Population (Census Tract 2010 Data)

LOW   HIGH
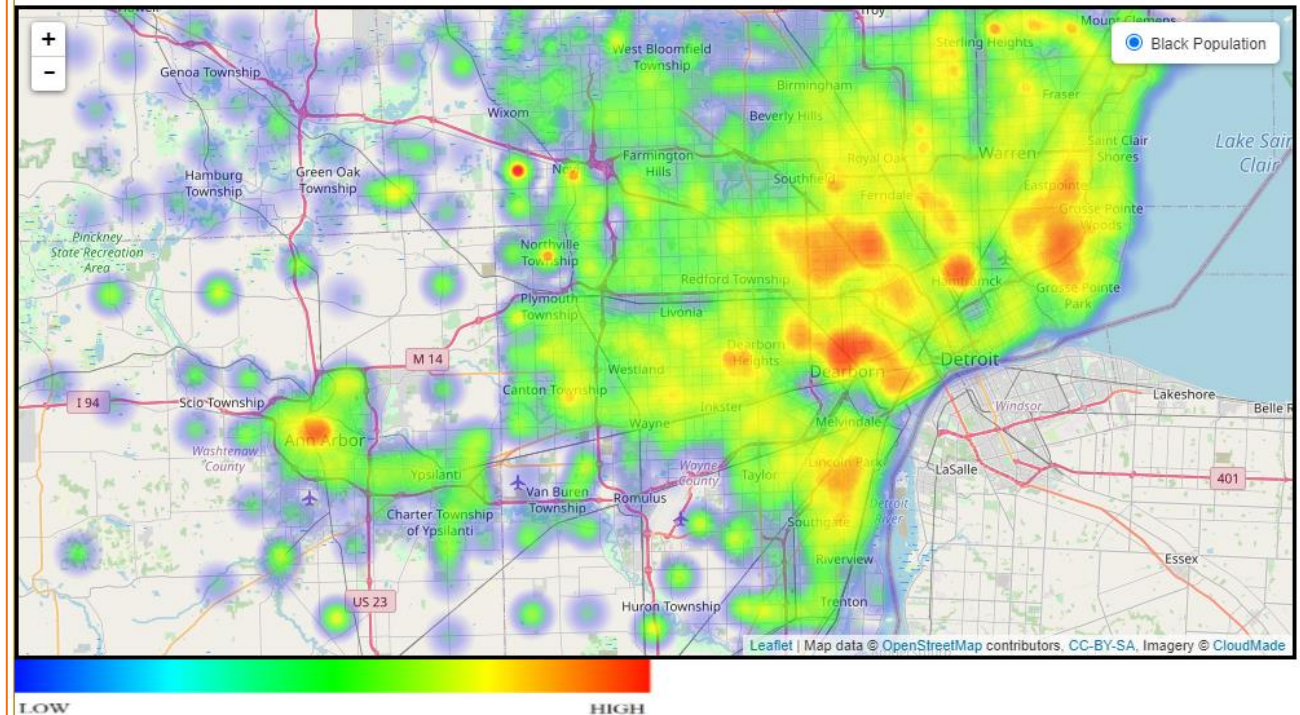
Check Heatmap of Below Poverty Level Proportion of Total Population (ACS 2006 - 2010 Data)

Clusters

➢ Main Functions:

➢ Click on clusters (circles) or zoom in – zoom out to see the cluster busting.

➢ Mouse over the cluster to see bounds.

➢ Click on a Marker to find the median home value.
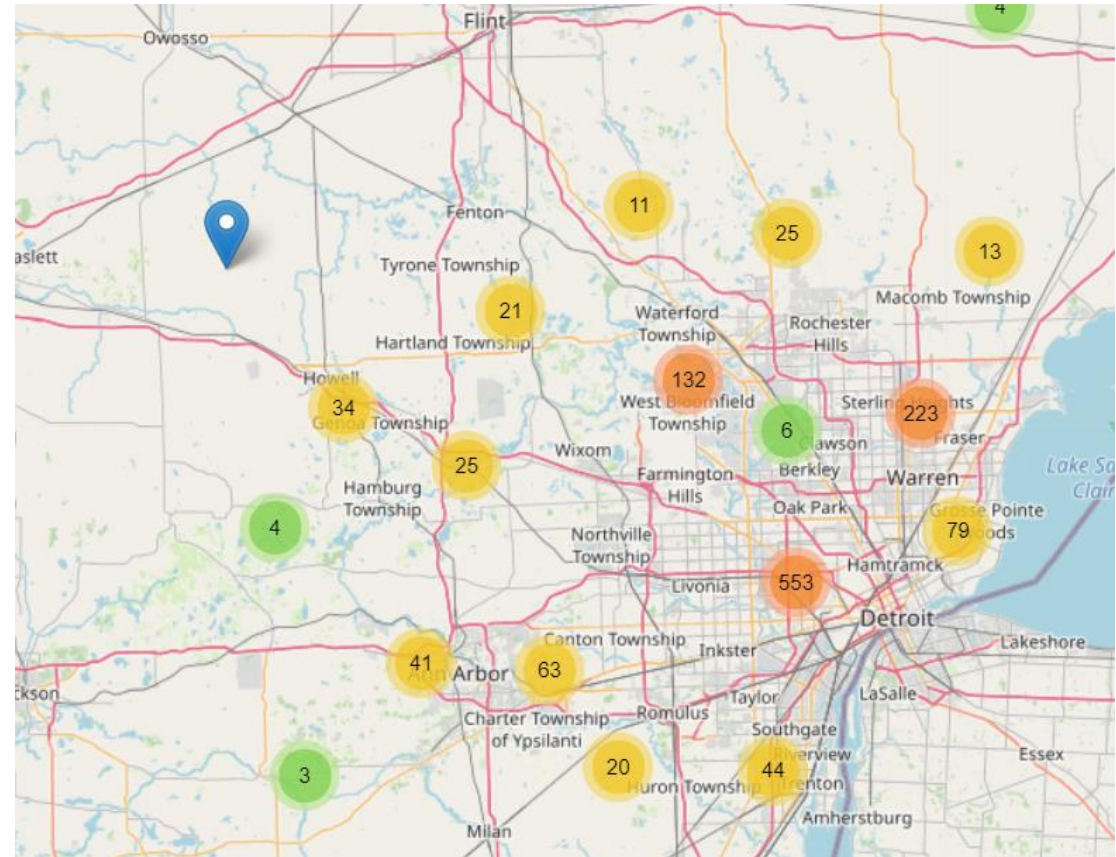
Read Me File

➢ Click on *More* option available on homepage to check read me file.

Description:

Demographic Data Explorer D2E is an interactive web map application for visualization of demographic data of southeast Michigan region, built using Leaflet, an open source JavaScript library. The web map provides a limited amount of data which includes, population, households, housing units information by cities; few thematic maps and heatmaps.

The application provides three options to view data: Explorer, Thematic Maps, and Heatmaps.

Explorer:

- Select a city from dropdown list to view data.
  - The map zooms to the selected city.
  - The side panel shows data such as
    - Place name
    - FIPS Code
    - Total Population
    - Population by Race
    - Total Households
    - Total Housing Units
    - Vacant Housing Units
- The circles represent census tracts within a city/village/township.
- Hover over the map to find out number of census tracts within a city/village/township.
- Use mouse navigations for zooming and panning.
- **Acknowledgement:** The web map is created by referecing a code obtained from 'Mapping in Leaflet JS', a data visualization course on Udemy.

Code Snippets for Explorer

➢ Sidebar

```
/*Click event for sidebar */

$(document).on('click','#advanced',function() {
  if($('#slide-in').hasClass('in')) {
    $('#slide-in').removeClass('in')
  } else {
    $('#slide-in').addClass('in')
  }
});
```

➢ Basemap Attribution

```
L.control.attribution({
  position: 'topright'
}).addTo(map);
```

➢ Calculation of number of census tracts within a city using *turf*, a JavaScript library for spatial analysis

```
var points = turf.points(PointsArray);
var totalPoints = 0;
console.log(layer.feature.geometry.coordinates)
if(layer.feature.geometry.coordinates[0].length===1) {
  layer.feature.geometry.coordinates.forEach(function(coords) {
    var searchWithin = turf.polygon(coords);
    var ptsWithin = turf.pointsWithinPolygon(points, searchWithin);
    totalPoints += ptsWithin.features.length;
  })
} else {
  var searchWithin = turf.polygon(layer.feature.geometry.coordinates);
  var ptsWithin = turf.pointsWithinPolygon(points, searchWithin);
  console.log(ptsWithin)
  totalPoints += ptsWithin.features.length;
}
```

Code Snippets for Explorer

➤ JavaScript to display data in side panel

```javascript
$(document).on('change','#city-select',function(e) {
  var newcity = e.target.value;
  if(newcity!=='') {
    citiesGeoJSON.eachLayer(function(layer) {
      if(layer.feature.properties.NAME===e.target.value) {
        layer.setStyle({color: 'Blue',fillOpacity: 0.3}),
        $('#city-information').html('<br><br><strong>Place:</strong> ' +layer.feature.properties.LABEL
        + '<br><br><strong> FIPS Code: </strong>' + layer.feature.properties.FIPSCODE
        + '<h3 style="color:blue;"> Population: </h3>'
        + '<strong> Total: </strong>' + layer.feature.properties.SUM_DP0010001
        + '<br><strong> Black: </strong>' + layer.feature.properties.SUM_DP0080004
      + '<br><strong> White: </strong>' + layer.feature.properties.SUM_DP0080003
      + '<br><strong> Am. Indian/Alaska Native: </strong>' + layer.feature.properties.SUM_DP0080005
      + '<br><strong> Asian: </strong>' + layer.feature.properties.SUM_DP0080006
      + '<br><strong> Asian Indian: </strong>' + layer.feature.properties.SUM_DP0080007
      + '<br><strong> Hispanic/Latino: </strong>' + layer.feature.properties.SUM_DP0100001
      + '<h3 style="color:blue;"> Households: </h3>'
      + '<strong> Total: </strong>' + layer.feature.properties.SUM_DP0130001
      + '<h3 style="color:blue;"> Housing Occupancy: </h3>'
      + '<strong> Total Housing Units: </strong>' + layer.feature.properties.SUM_DP0180001
      + '<br><strong> Vacant Housing Units: </strong>' + layer.feature.properties.SUM_DP0180003
      );
        map.fitBounds(layer.getBounds())
      }
    });
  } else {
    $('#city-information').html('');
  }
});
```

Code Snippets for Thematic Maps

➤ Overlay Controls

```
var baseLayers = {
  "Grey Basemap": basemap1,
  "Watercolor Basemap": basemap2

};


var overlays = {

  "Black/Total Population": bp,
  "Vacant/Total Housing Units": vac
};

L.control.layers(baseLayers, overlays,{collapsed: false}).addTo(mymap1);
```

Code Snippets for Heatmaps

➢ Feature Properties for Heatmap   ➢

➢ Data Format

                              ➢

```
var testData = {
data:[
{lng: -83.25042521 , lat:  42.123524009146486, count:  3359 },
{lng: -83.24844166 , lat:  42.334484596916013, count:  4585 },
{lng: -83.24291583 , lat:  42.070657701357128, count:  3289 },
{lng: -83.30464777 , lat:  42.283651808004095, count:  2535 },
{lng: -83.30556776 , lat:  42.301730874658475, count:  2724 },
{lng: -83.29915154 , lat:  42.322803051927302, count:  3110 },
{lng: -83.24923936 , lat:  42.146187836404543, count:  3934 },
{lng: -83.26464626 , lat:  42.132957300430512, count:  6168 },
{lng: -83.2628737 , lat:  42.294302586594881, count:  2649 },
{lng: -83.2610868 , lat:  42.286473769916441, count:  3591 },
{lng: -83.27796346 , lat:  42.288923273218096, count:  4059 },
```

```
var cfg = {
  // radius should be small ONLY if scaleRadius is true (or small radius is intended)
  "radius": 0.02,
  "maxOpacity": .8,
  // scales the radius based on map zoom
  "scaleRadius": true,
  // if set to false the heatmap uses the global maximum for colorization
  // if activated: uses the data maximum within the current map boundaries
  //    (there will always be a red spot with useLocalExtremas true)
  "useLocalExtrema": true,
  // which field name in your data represents the latitude - default "lat"
  latField: 'lat',
  // which field name in your data represents the longitude - default "lng"
  lngField: 'lng',
  // which field name in your data represents the data value - default "value"
  valueField: 'count'
};
var heatmapLayer = new HeatmapOverlay(cfg);

var map = new L.Map('map-canvas', {
  center: new L.LatLng(42.3523699,-83.3793885),
  zoom: 10,
  layers: [baseLayer, heatmapLayer]
});

heatmapLayer.setData(testData);
```

Code Snippets for Cluster Map

➢ JavaScript: leaflet.markercluster-src.js

```html
<link rel="stylesheet" href="dist/MarkerCluster.css" />
<link rel="stylesheet" href="dist/MarkerCluster.Default.css" />
<script src="dist/leaflet.markercluster-src.js"></script>
<script src="js/medhvpt.js" type="text/javascript"></script>
```

➢ Data: medhvpt.js

```javascript
var addressPoints = [
[42.12352396,-83.25042521,138724],
[42.33448459,-83.24844168,141763],
[42.07065769,-83.24291583,116184],
[42.28365179,-83.30464777,58268],
[42.30173085,-83.30556775,71295],
[42.32280305,-83.29915156,218655],
```

DEMO

https://truptilk.github.io/emugis/links/projecthomepage.html

THANK YOU