BEAUTIFUL ONLINE MAPS IN MINUTES WITH LEAFLET.JS

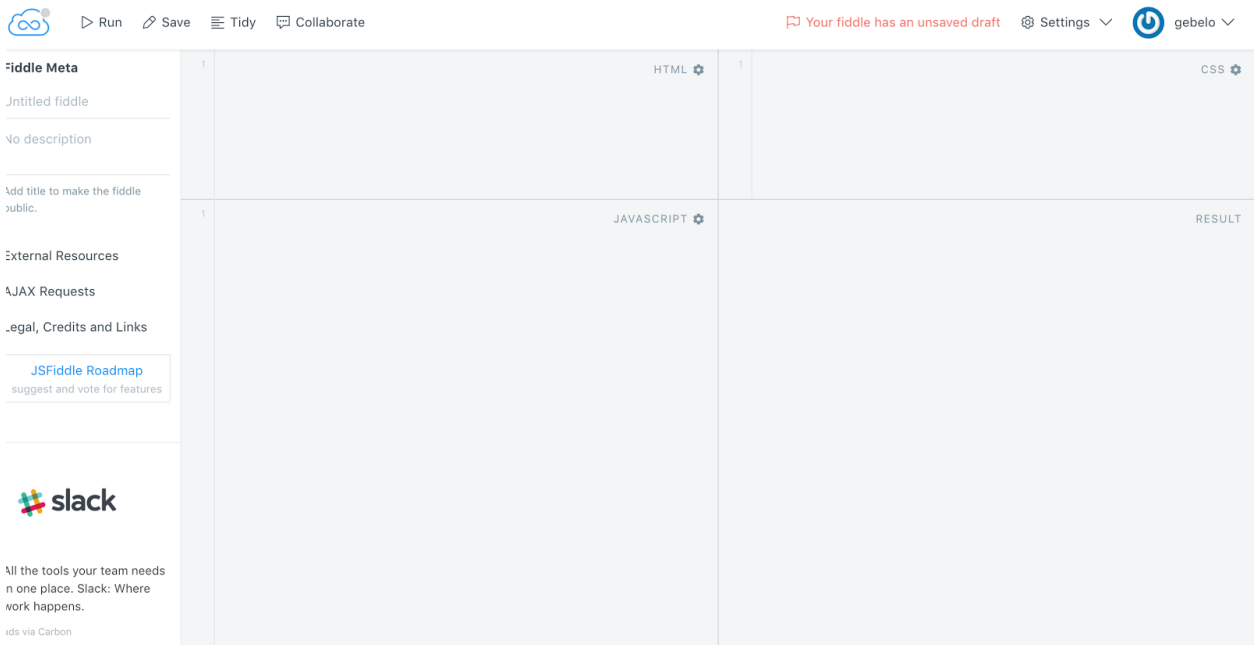
Robert Gebeloff
@gebeloffnyt
Dataharvest 2017

Over the last few years, the online mapping world has exploded, with a bevy of tools and techniques for sharing your geographic data online.

In this lesson, we will work with the free Leaflet javascript library. Leaflet is perhaps the easiest way for a non-programmer, non-designer to get a reasonable mapping app up and running quickly. People who are designers and programmers can leverage Leaflet in even more sophisticated ways, but in this class we'll stick with the basics.

In a true development environment, you'd have a local instance of your Web site running on your machine, with Web server software running your site's code. Because we cannot simulate that in this class, we will be working in the online Javascript development environment JSFiddle.

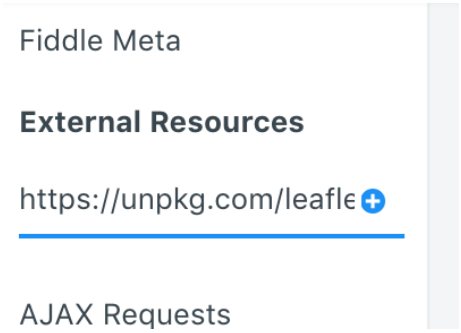
So to start with, load the JSfiddle Web site: https://jsfiddle.net/



JSFiddle helpfully organizes your code in sections so it's easier to keep track of what you're doing. In the top left, we will enter the HTML code our app needs. On the top right, we'll be entering some CSS style components. On the bottom left, we'll be

entering our Javascript code. And on the left side, we'll be entering some "external resources" -- links to the Leaflet.js libraries that your code will need to run.

So let's enter those resources first. Click on "External Resources" and enter this link to the main Javascript library: https://unpkg.com/leaflet@1.0.3/dist/leaflet.js



Hit the plus sign and then enter a link to the Leaflet CSS file: https://unpkg.com/leaflet@1.0.3/dist/leaflet.css

When you're done, it should look like this:

Now we're reading to program! In the HTML section, we need to create a container that will hold our map. So let's create a div:

```
<div id="map"></div>
```

In the CSS section, let's add some style to the map -- in this case, let's keep it simple and declare that our map should be 400 pixels tall:

```
#map { height: 400px; }
```

Now let's work on the Javascript. The first step is to create a Javascript variable that will contain our Leaflet map. If you're not familiar with Javascript, this means we need to declare a variable called "map", and then tell the program that we want the "map" variable to contain a new Leaflet map. We'll also tell the program that we want to disable a the user's scrollwheel from moving the map, which many users find annoying.

```
var map = new L.Map('map',{ scrollWheelZoom:false });
```

Next, we need to tell the program to fetch a basemap upon which we will add our own custom data. There are several options for free basemap tiles, but for this exercise, we'll use the tiles from openstreetmap.org. In this case, we're not creating a new variable, we're just issuing a command to add something to our existing map variable, so the syntax looks like this:

```
L.tileLayer('http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
          attribution: '&copy; <a
href="http://openstreetmap.org">OpenStreetMap</a> contributors',
          maxZoom: 18
       }).addTo(map);
```

And lastly, we need to give our map a starting point. In this case, the latitude and longitude of Mechelen:

```
map.setView([51.02574, 4.47762], 13);
```

Once you have these three command in the Javascript box, you should be able to hit the "Run" button and see this:

HTML

```
<div id="map"></div>
```

CSS

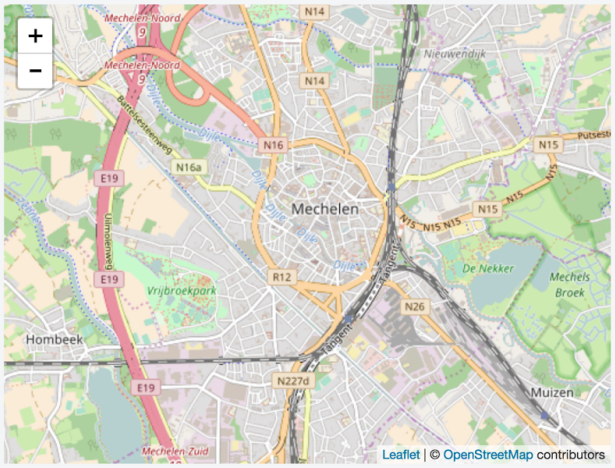
```
#map { height: 400px; }
```

JAVASCRIPT

```
var map = new L.Map('map',{ scrollWheelZoom:false });

 L.tileLayer('http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
          attribution: '&copy; <a
href="http://openstreetmap.org">OpenStreetMap</a> contributors',
          maxZoom: 18
       }).addTo(map);

map.setView([51.02574, 4.47762], 13);
```



Let’s add some data to the map to make it more interesting. We are right now sitting in the Thomas More Hogeschool -- let’s mark the spot:

```
var marker = L.marker([51.024270, 4.487897]).addTo(map);
```

If we want to slap a label on that marker, we can “bind” some HTML to it:

```
var marker = L.marker([51.024270, 4.487897]).addTo(map).bindPopup("<b>Thomas Moore</b><br>Hogeschool.").openPopup();
```

HTML

```
<div id="map"></div>
```

CSS

```
#map { height: 400px; }
```

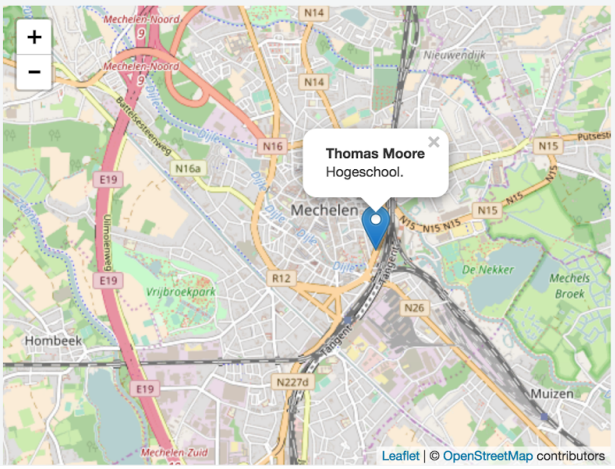
JAVASCRIPT

```
var map = new L.Map('map',{ scrollWheelZoom:false });

 L.tileLayer('http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
          attribution: '&copy; <a
href="http://openstreetmap.org">OpenStreetMap</a> contributors',
          maxZoom: 18
       }).addTo(map);

map.setView([51.02574, 4.47762], 13);

var marker = L.marker([51.024270, 4.487897]).addTo(map).bindPopup("
<b>Thomas Moore</b><br>Hogeschool.").openPopup();
```



When I make maps, I find I want to customize the markers. I want to color code them by category, or even change the size based on some underlying value. In Leaflet, it's easier to accomplish these tasks by using a marker class called circleMarker:

```
var marker = L.circleMarker([51.024270, 4.487897], {
    color: 'red',
    fillColor: 'red',
    fillOpacity: 0.5,
    radius: 8
}).addTo(map).bindPopup("<b>Thomas Moore</b><br>Hogeschool.");
```

Note how here I've also stripped out the "openPopup()" command -- when we plot multiple markers, it's usual style to keep the popups hidden until a user clicks on the point.

HTML

```
<div id="map"></div>
```

CSS

```
#map { height: 400px; }
```

JAVASCRIPT

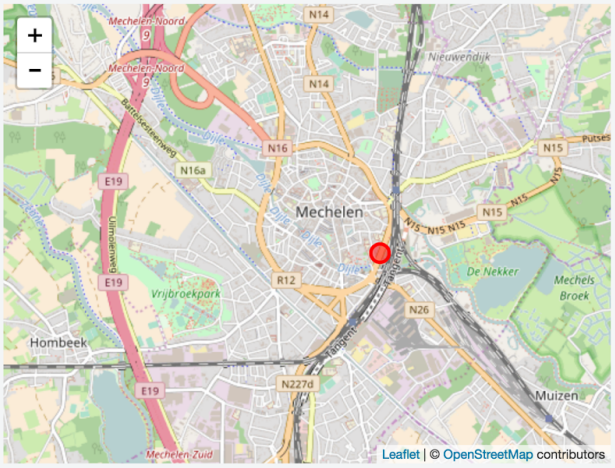
```
var map = new L.Map('map',{ scrollWheelZoom:false });

 L.tileLayer('http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
          attribution: '&copy; <a
href="http://openstreetmap.org">OpenStreetMap</a> contributors',
          maxZoom: 18
       }).addTo(map);

map.setView([51.02574, 4.47762], 13);

//var marker = L.marker([51.024270, 4.487897]).addTo(map).bindPopup("
<b>Thomas Moore</b><br>Hogeschool.").openPopup();

var marker = L.circleMarker([51.024270, 4.487897], {
    color: 'red',
    fillColor: 'red',
    fillOpacity: 0.5,
    radius: 8
}).addTo(map).bindPopup("<b>Thomas Moore</b><br>Hogeschool.");
```



Ok, let’s try a more complicated example, shifting our focus to London. Clear out everything below line 8 in the Javascript section and replace it with the London coordinates:

```
map.setView([51.507351, -0.127758], 9);
```

HTML

```
<div id="map"></div>
```

CSS

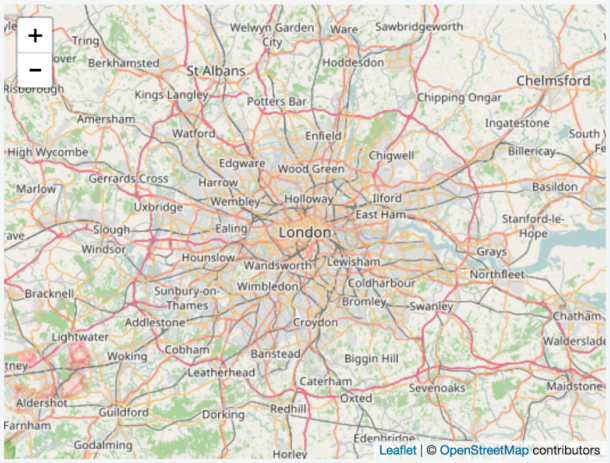
```
#map { height: 400px; }
```

JAVASCRIPT

```
var map = new L.Map('map',{ scrollWheelZoom:false });

 L.tileLayer('http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
          attribution: '&copy; <a
href="http://openstreetmap.org">OpenStreetMap</a> contributors',
          maxZoom: 18
       }).addTo(map);

map.setView([51.507351, -0.127758], 9);
```



I have a directory of schools in London, and I want to build a map that plots all of the schools and color codes the markers by gender.

Download the data here:
https://dl.dropboxusercontent.com/u/100051277/dataharvest/londonschools.csv

Open the file in Excel. Often, Excel serves as a great way to write code for your program, as we will illustrate here.

Note the key columns we want to use for our map. Column C has the school name. Column J tells us whether the school is for girls, boys or mixed. Column X is the latitude. Column W has the longitude.

In column Y, let's enter a formula that will assign a color based on the contents of column J. So in cell Y2, copy and paste this formula:

```
=IF(J2="girls","red",IF(J2="boys","blue","green"))
```

Copy and paste the formula down the column.

V	W	X	Y
Primary	x	y	
0	-0.378496	51.5074997	red
0	-0.241628	51.5789986	blue
0	-0.0425897	51.5940018	green
0	-0.150409	51.517601	blue
1	-0.193367	51.5404015	green
1	-0.0769998	51.5452995	green
1	-0.0717606	51.539299	green
1	-0.0776093	51.5527992	green
1	-0.0735189	51.5427017	green
1	-0.166066	51.4926987	green
1	0.176705	51.5630989	green
1	-0.351077	51.5612984	green
1	0.203542	51.5497017	green
1	0.0118644	51.5644989	green
0	-0.0659039	51.4889984	green
0	-0.420888	51.4528999	green
1	-0.155295	51.4390984	green
0	-0.0425897	51.5940018	red

Now the formula to create markers. Note: This is a long, complicated formula. Don't be afraid. Once you have some time to digest it, you'll see that we're simply creating the marker syntax Leaflet wants and using Excel to sub in the data we want.

In cell Z2:

```
="var marker = L.circleMarker(["&X2&","&W2&"], {color:'"&Y2&"', fillColor:'"&Y2&"',
fillOpacity:0.5,radius:3}).addTo(map).bindPopup("""&"<b>"&C2&"</b>"""&");"
```

Copy and paste that down the column.

	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI
997	red	var marker = L.circleMarker([51.5074997,-0.378496], {color:'red', fillColor:'red', fillOpacity:0.5,radius:3}).addTo(map).bindPopup("<b>									
986	blue	var marker = L.circleMarker([51.5789986,-0.241628], {color:'blue', fillColor:'blue', fillOpacity:0.5,radius:3}).addTo(map).bindPopup("<b									
018	green	var marker = L.circleMarker([51.5940018,-0.0425897], {color:'green', fillColor:'green', fillOpacity:0.5,radius:3}).addTo(map).bindPopup									
501	blue	var marker = L.circleMarker([51.517601,-0.150409], {color:'blue', fillColor:'blue', fillOpacity:0.5,radius:3}).addTo(map).bindPopup("<b:									
015	green	var marker = L.circleMarker([51.5404015,-0.193367], {color:'green', fillColor:'green', fillOpacity:0.5,radius:3}).addTo(map).bindPopup(									
995	green	var marker = L.circleMarker([51.5452995,-0.0769998], {color:'green', fillColor:'green', fillOpacity:0.5,radius:3}).addTo(map).bindPopup									
299	green	var marker = L.circleMarker([51.539299,-0.0717606], {color:'green', fillColor:'green', fillOpacity:0.5,radius:3}).addTo(map).bindPopup(									
992	green	var marker = L.circleMarker([51.5527992,-0.0776093], {color:'green', fillColor:'green', fillOpacity:0.5,radius:3}).addTo(map).bindPopup									
017	green	var marker = L.circleMarker([51.5427017,-0.0735189], {color:'green', fillColor:'green', fillOpacity:0.5,radius:3}).addTo(map).bindPopup									
987	green	var marker = L.circleMarker([51.4926987,-0.166066], {color:'green', fillColor:'green', fillOpacity:0.5,radius:3}).addTo(map).bindPopup(									
989	green	var marker = L.circleMarker([51.5630989,0.176705], {color:'green', fillColor:'green', fillOpacity:0.5,radius:3}).addTo(map).bindPopup("									
984	green	var marker = L.circleMarker([51.5612984,-0.351077], {color:'green', fillColor:'green', fillOpacity:0.5,radius:3}).addTo(map).bindPopup(									
017	green	var marker = L.circleMarker([51.5497017,0.203542], {color:'green', fillColor:'green', fillOpacity:0.5,radius:3}).addTo(map).bindPopup("									
989	green	var marker = L.circleMarker([51.5644989,0.0118644], {color:'green', fillColor:'green', fillOpacity:0.5,radius:3}).addTo(map).bindPopup									
984	green	var marker = L.circleMarker([51.4889984,-0.0659039], {color:'green', fillColor:'green', fillOpacity:0.5,radius:3}).addTo(map).bindPop									
999	green	var marker = L.circleMarker([51.4528999,-0.420888], {color:'green', fillColor:'green', fillOpacity:0.5,radius:3}).addTo(map).bindPopu									
984	green	var marker = L.circleMarker([51.4390984,-0.155295], {color:'green', fillColor:'green', fillOpacity:0.5,radius:3}).addTo(map).bindPopup(									
018	red	var marker = L.circleMarker([51.5940018,-0.0425897], {color:'red', fillColor:'red', fillOpacity:0.5,radius:3}).addTo(map).bindPopup("<b									
996	green	var marker = L.circleMarker([51.5326996,0.130539], {color:'green', fillColor:'green', fillOpacity:0.5,radius:3}).addTo(map).bindPopup("									
996	green	var marker = L.circleMarker([51.5326996,0.130539], {color:'green', fillColor:'green', fillOpacity:0.5,radius:3}).addTo(map).bindPopup("									
994	red	var marker = L.circleMarker([51.5191994,-0.0945782], {color:'red', fillColor:'red', fillOpacity:0.5,radius:3}).addTo(map).bindPopup("<b									
994	red	var marker = L.circleMarker([51.5191994,-0.0945782], {color:'red', fillColor:'red', fillOpacity:0.5,radius:3}).addTo(map).bindPopup("<b									
016	green	var marker = L.circleMarker([51.5138016,-0.0968321], {color:'green', fillColor:'green', fillOpacity:0.5,radius:3}).addTo(map).bindPopup									
016	green	var marker = L.circleMarker([51.5138016,-0.0968321], {color:'green', fillColor:'green', fillOpacity:0.5,radius:3}).addTo(map).bindPopup									
008	blue	var marker = L.circleMarker([51.5111008,-0.099525], {color:'blue', fillColor:'blue', fillOpacity:0.5,radius:3}).addTo(map).bindPopup("<b									
008	blue	var marker = L.circleMarker([51.5111008,-0.099525], {color:'blue', fillColor:'blue', fillOpacity:0.5,radius:3}).addTo(map).bindPopup("<b									
987	red	var marker = L.circleMarker([51.5509987,-0.170027], {color:'red', fillColor:'red', fillOpacity:0.5,radius:3}).addTo(map).bindPopup("<b>									
003	red	var marker = L.circleMarker([51.5578003,-0.190339], {color:'red', fillColor:'red', fillOpacity:0.5,radius:3}).addTo(map).bindPopup("<b>									
003	red	var marker = L.circleMarker([51.5578003,-0.190339], {color:'red', fillColor:'red', fillOpacity:0.5,radius:3}).addTo(map).bindPopup("<b>									
399	red	var marker = L.circleMarker([51.544899,-0.165759], {color:'red', fillColor:'red', fillOpacity:0.5,radius:3}).addTo(map).bindPopup("<b>S									
987	blue	var marker = L.circleMarker([51.5453987,-0.170914], {color:'blue', fillColor:'blue', fillOpacity:0.5,radius:3}).addTo(map).bindPopup("<b									
987	blue	var marker = L.circleMarker([51.5453987,-0.170914], {color:'blue', fillColor:'blue', fillOpacity:0.5,radius:3}).addTo(map).bindPopup("<b									

Now, starting at Z2, copy and paste all of these markers into our Javascript. When you run it, it works, but what a mess. Go back into your Excel and change the radius to 1, then re-copy the formula down the column. Copy all of the values and paste over the old markers.

HTML

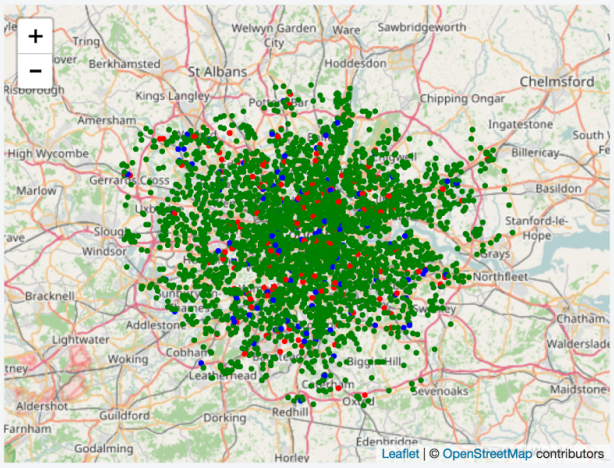
```
<div id="map"></div>
```

CSS

```
#map { height: 400px; }
```

JAVASCRIPT

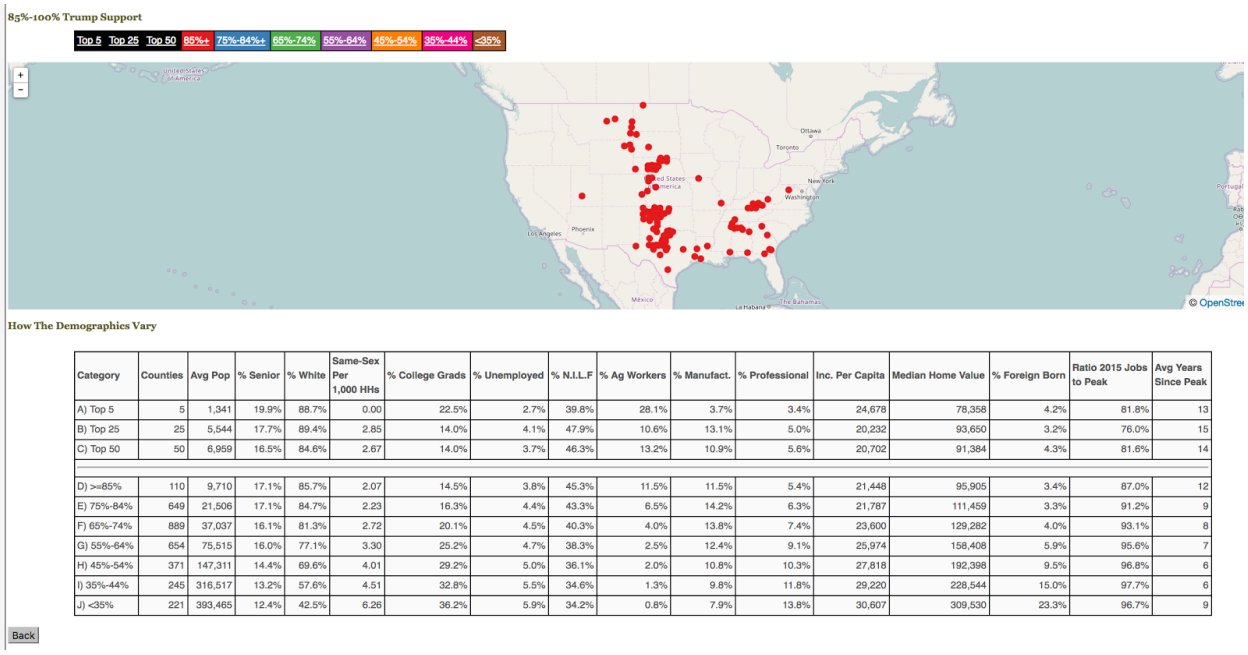
```
var marker = L.circleMarker([51.5363998,0.0240207], {color:'green',
fillColor:'green', fillOpacity:0.5,radius:1}).addTo(map).bindPopup("
<b>Selwyn Primary School</b>");
var marker = L.circleMarker([51.4431,0.211245], {color:'green',
fillColor:'green', fillOpacity:0.5,radius:1}).addTo(map).bindPopup("
<b>Westgate Primary School</b>");
var marker = L.circleMarker([51.5657005,-0.217671], {color:'green',
fillColor:'green', fillOpacity:0.5,radius:1}).addTo(map).bindPopup("
<b>Claremont Primary School</b>");
var marker = L.circleMarker([51.5339012,0.0178413], {color:'green',
fillColor:'green', fillOpacity:0.5,radius:1}).addTo(map).bindPopup("
<b>Portway Primary School</b>");
var marker = L.circleMarker([51.3699989,0.117722], {color:'blue',
fillColor:'blue', fillOpacity:0.5,radius:1}).addTo(map).bindPopup("
<b>Burwood School</b>");
var marker = L.circleMarker([51.3699989,0.117722], {color:'blue',
fillColor:'blue', fillOpacity:0.5,radius:1}).addTo(map).bindPopup("
<b>Burwood School</b>");
var marker = L.circleMarker([51.5600014,0.006027], {color:'green',
fillColor:'green', fillOpacity:0.5,radius:1}).addTo(map).bindPopup("
<b>Mayville Primary School</b>");
var marker = L.circleMarker([51.3843002,-0.080438], {color:'green',
fillColor:'green', fillOpacity:0.5,radius:1}).addTo(map).bindPopup("
<b>Davidson Primary School</b>");
```



The full code for this tutorial is here: https://jsfiddle.net/gebelo/b102sqms/

Where to go from here: The Leaflet site has a set of tutorials that covers additional functionality. See http://leafletjs.com/examples/quick-start/. Also, this is where Google is your friend. Google how to work with polygons in Leaflet. How to add a legend. How to cluster markers, etc.

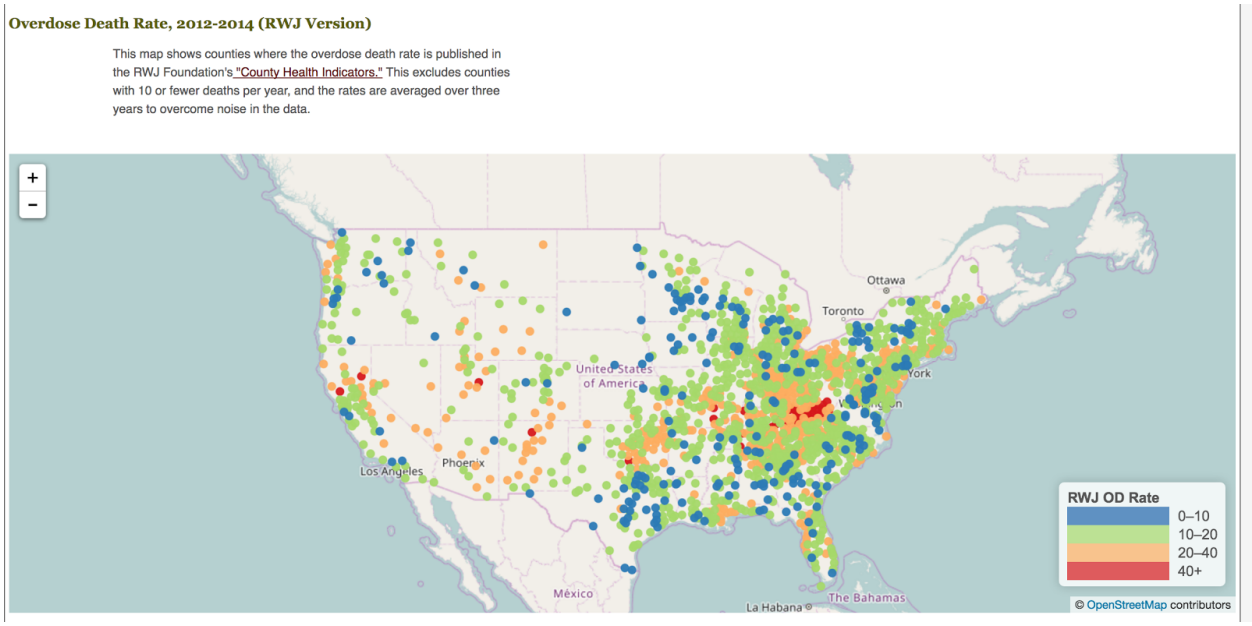
And to close, a few examples of how I use Leaflet in some internal apps I've built to help reporters track demographics in the U.S.



How The Demographics Vary

Category	Counties	Avg Pop	% Senior	% White	Same-Sex Per 1,000 HHs	% College Grads	% Unemployed	% N.I.L.F	% Ag Workers	% Manufact.	% Professional	Inc. Per Capita	Median Home Value	% Foreign Born	Ratio 2015 Jobs to Peak	Avg Years Since Peak
A) Top 5	5	1,341	19.9%	88.7%	0.00	22.5%	2.7%	39.8%	28.1%	3.7%	3.4%	24,678	78,358	4.2%	81.8%	13
B) Top 25	25	5,544	17.7%	89.4%	2.85	14.0%	4.1%	47.9%	10.6%	13.1%	5.0%	20,232	93,650	3.2%	76.0%	15
C) Top 50	50	6,959	16.5%	84.6%	2.67	14.0%	3.7%	46.3%	13.2%	10.9%	5.6%	20,702	91,384	4.3%	81.6%	14
D) >=85%	110	9,710	17.1%	85.7%	2.07	14.5%	3.8%	45.3%	11.5%	11.5%	5.4%	21,448	95,905	3.4%	87.0%	12
E) 75%-84%	649	21,506	17.1%	84.7%	2.23	16.3%	4.4%	43.3%	6.5%	14.2%	6.3%	21,787	111,459	3.3%	91.2%	9
F) 65%-74%	889	37,037	16.1%	81.3%	2.72	20.1%	4.5%	40.3%	4.0%	13.8%	7.4%	23,600	129,282	4.0%	93.1%	8
G) 55%-64%	654	75,515	16.0%	77.1%	3.30	25.2%	4.7%	38.3%	2.5%	12.4%	9.1%	25,974	158,408	5.9%	95.6%	7
H) 45%-54%	371	147,311	14.4%	69.6%	4.01	29.2%	5.0%	36.1%	2.0%	10.8%	10.3%	27,818	192,398	9.5%	96.8%	6
I) 35%-44%	245	316,517	13.2%	57.6%	4.51	32.8%	5.5%	34.6%	1.3%	9.8%	11.8%	29,220	228,544	15.0%	97.7%	6
J) <35%	221	393,465	12.4%	42.5%	6.26	36.2%	5.9%	34.2%	0.8%	7.9%	13.8%	30,607	309,530	23.3%	96.7%	9

Back



Birth to Death Ratio, 2015 -- 1,022 counties below 1

Blue counties had more deaths than births in the selected year. Red counties had the highest ratio of births to deaths.

2015 Change Year