GIS for Designers
VT | A+D | FA22

TUTORIAL 8 | JUMP INTO MAPBOX 2

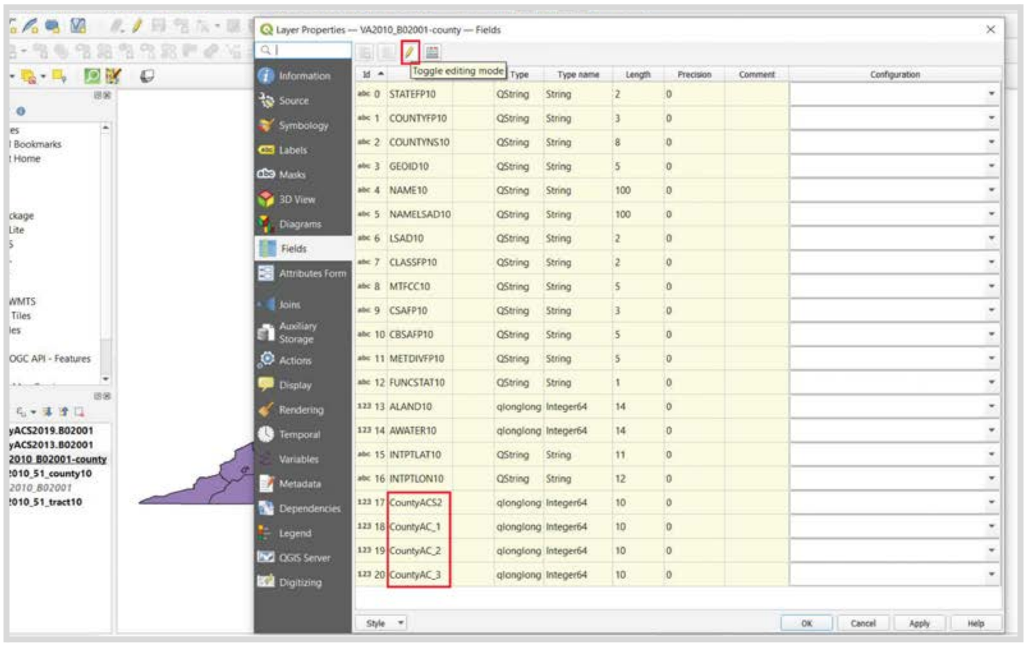
Goals

- Add a new field to QGIS Attribute Table.
- Replace QGIS layer in Mapbox.
- Style Mapbox data by QGIS layer data.
- Code interactive hover feature to display OSM feature attribute data.

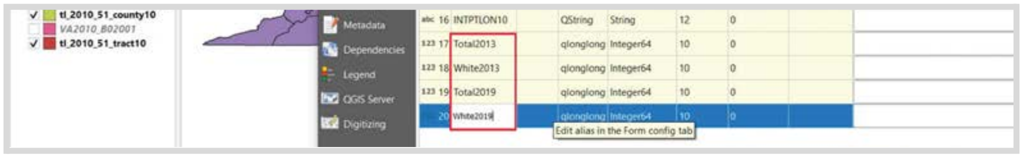
Step 1: In QGIS, create a new attribute column for percentage change in nonwhite population between 2013 and 2019.

1a **Open** the **Attribute Table** of your Mapbox census layer, with both ACS tables joined to it. You'll notice that your 4 new columns (total and white 2019, total and white 2013) have been renamed to ACS1, AC_2, AC_3, etc. This is NOT very helpful for knowing which column contains what data.

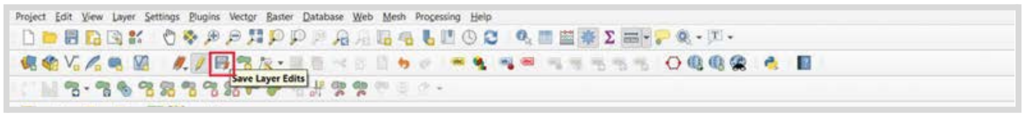
In **Layer Properties > Fields**, toggle on "**edit**" and rename these to **Total2013**, **White2013**, **Total2019**, and **White2019** respectively. **Save** your changes and **toggle off edit**.



Id	Name	Alias	Type	Type name	Length	Precision	Comment	Configuration
0	STATEFP10		QString	String	2	0		
1	COUNTYFP10		QString	String	3	0		
2	COUNTYNS10		QString	String	8	0		
3	GEOID10		QString	String	5	0		
4	NAME10		QString	String	100	0		
5	NAMELSAD10		QString	String	100	0		
6	LSAD10		QString	String	2	0		
7	CLASSFP10		QString	String	2	0		
8	MTFCC10		QString	String	5	0		
9	CSAFP10		QString	String	3	0		
10	CBSAFP10		QString	String	5	0		
11	METDIVFP10		QString	String	5	0		
12	FUNCSTAT10		QString	String	1	0		
13	ALAND10		qlonglong	Integer64	14	0		
14	AWATER10		qlonglong	Integer64	14	0		
15	INTPTLAT10		QString	String	11	0		
16	INTPTLON10		QString	String	12	0		
17	CountyACS2		qlonglong	Integer64	10	0		
18	CountyAC_1		qlonglong	Integer64	10	0		
19	CountyAC_2		qlonglong	Integer64	10	0		
20	CountyAC_3		qlonglong	Integer64	10	0		

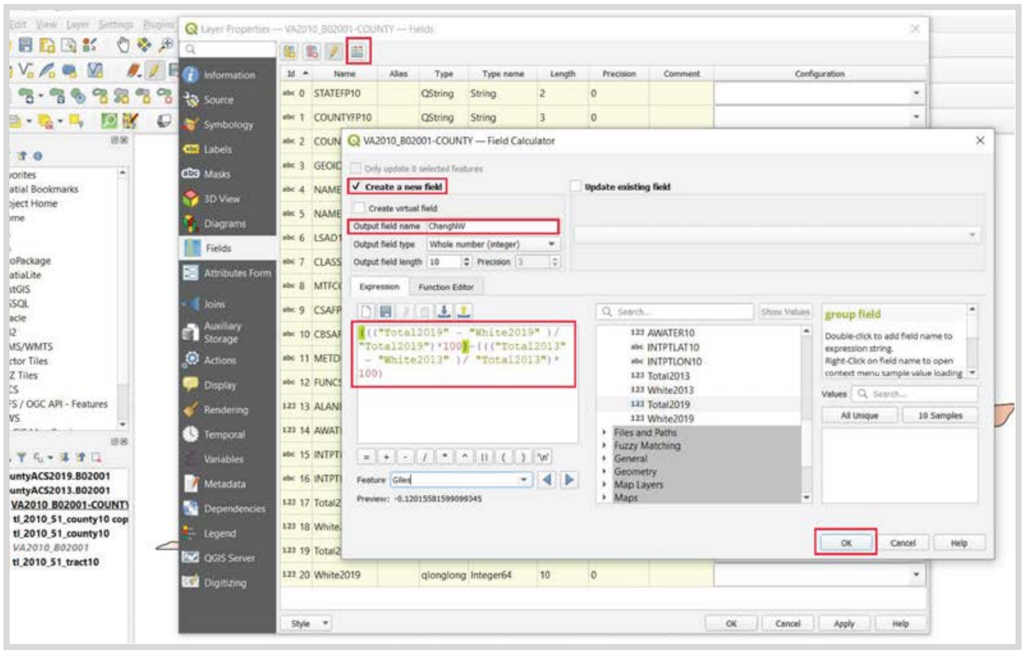


16	INTPTLON10	QString	String	12	0
17	Total2013	qlonglong	Integer64	10	0
18	White2013	qlonglong	Integer64	10	0
19	Total2019	qlonglong	Integer64	10	0
20	White2019	qlonglong	Integer64	10	0


Project Edit View Layer Settings Plugins Vector Raster Database Web Mesh Processing Help
Save Layer Edits

1b Next, add a new column ("field) using the **Field Calculator** (upper left abacus button in your Attribute Table). **Check "Add New Field"** and name it something short and simple like "**NWChange**".

Subtract the NonWhite percentage for 2013 from 2019 to get the change between the two years:
((("Total2019" - "White2019")/ "Total2019")- (("Total2013" - "White2013")/ "Total2013"))*100
Multiply by 100 at the end to get a percentage.


Layer Properties — VA2010_B02001-COUNTY — Fields
VA2010_B02001-COUNTY — Field Calculator
OK
Cancel
Help

✓ Create a new field

Output field name ChangNW
Output field type Whole number (integer)
Output field length 10 Precision 3

```
(("Total2019" - "White2019" )/
"Total2019")*100)-((("Total2013"
 - "White2013" )/ "Total2013")*
100)
```

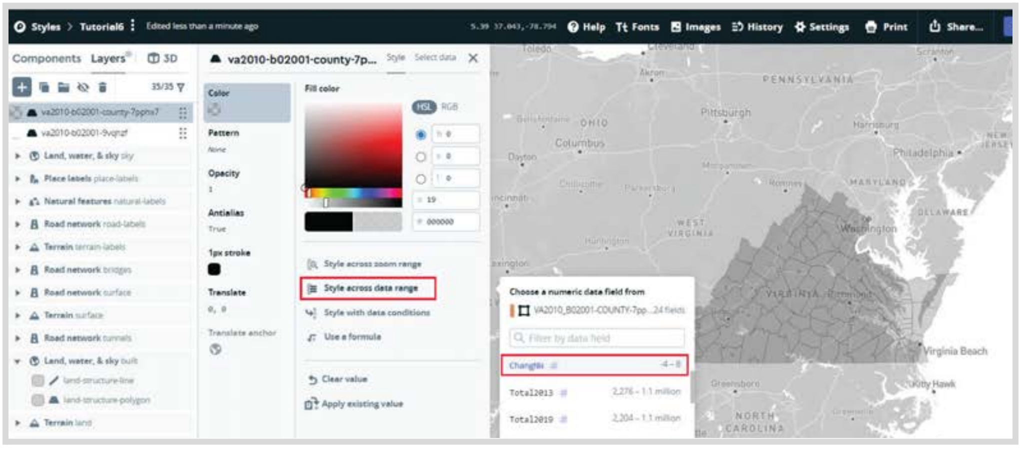
Feature Giles
Preview: -0.12015581599099345

1c **Re-export** your layer as an Esri shapefile with Pseudo-Mercator projection. **Rezip** and **reupload** your shapefile to Mapbox.

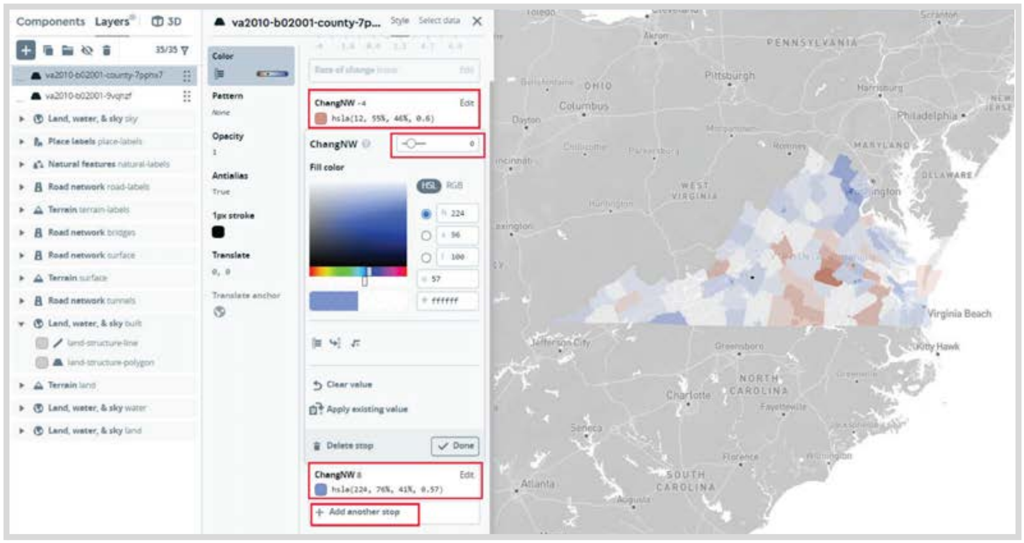
GIS for Designers
VT | A+D | FA22

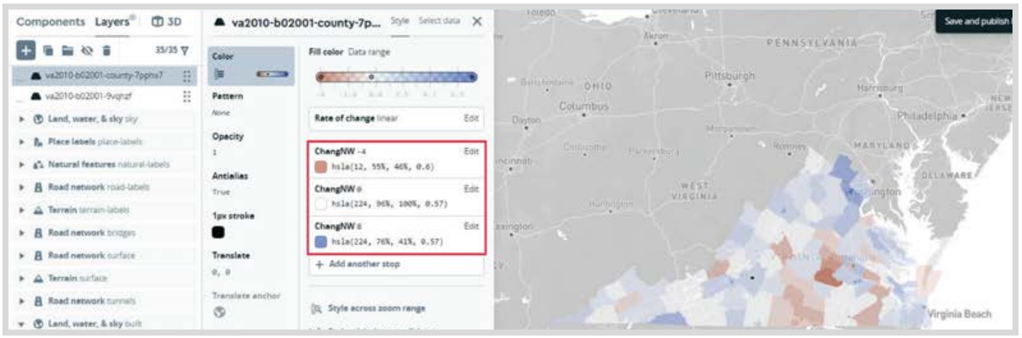
Step 2: In Mapbox, style your newly updated layer by data.

2a In the layer style, select “**Style Across Data Range**.” Select your new “**NWChange**” field from the list (scroll down)



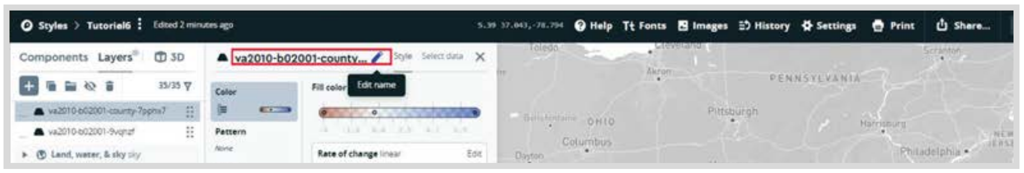
2b Choose some endpoint colors (I chose red and blue), and then “**Add Another Stop**” (below your two colors). Make this one white or some other neutral color, and set the slider at the top to “**0**” – this color will represent **no change** between the years (eg 0 percent change).

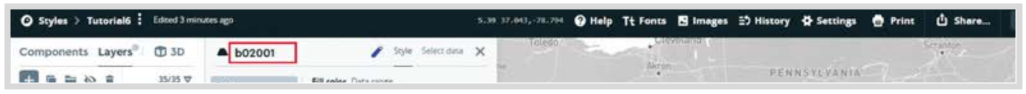




ChangNW -4 hsla(12, 55%, 46%, 0.6)
ChangNW 0 hsla(224, 96%, 100%, 0.57)
ChangNW 8 hsla(224, 76%, 41%, 0.57)

2c As a last step, **rename your layer** to something short and logical (eg “b02001”).





2d If all looks correct, **Publish** your map.

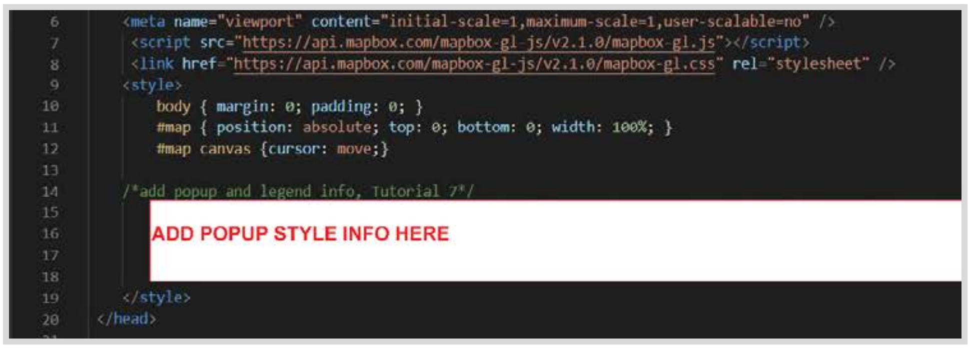
Step 3: Add an interactive hover feature to display OSM feature attribute data.

One very cool thing about the QGIS- Mapbox-HTML workflow is that you can call and display data from your Attribute Tables on your interactive map. For this next step, you'll add a **popup box** to display the **percent change in non-white population** that appears automatically **on hover** and disappears when you mouse off the feature.

> **Note**: you may not see the Tutorial 7 green comments in your template, but this will not change how the code works. Insert the new code in the place indicated.

3a First, add some styles for the new popup and its text. Copy the following into the ***<style>*** section:

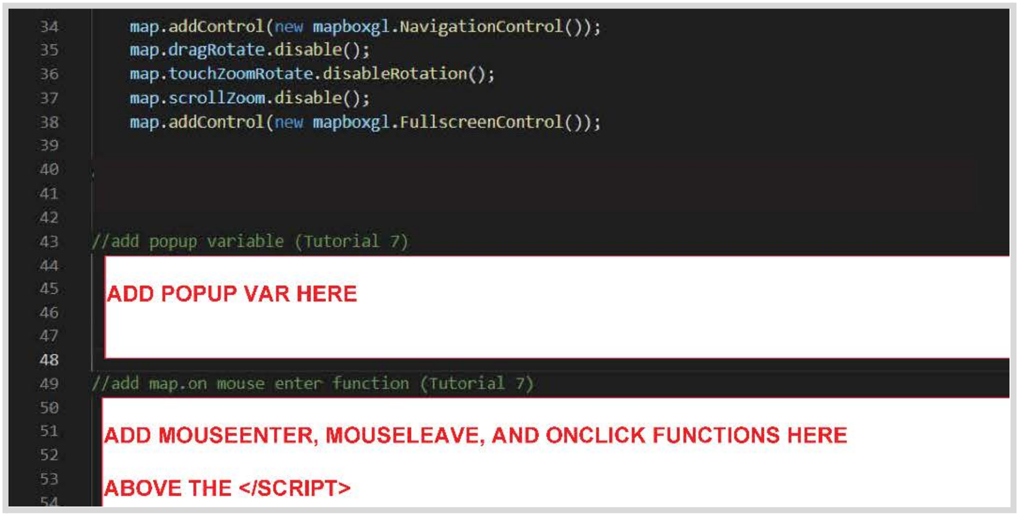
.mapboxgl-popup {max-width: 400px; font: 12px/20px 'Helvetica Neue', Arial, Helvetica, sans-serif;}
h1 {font: 12px sans-serif; font-weight: normal;}
h2 {font: 12px sans-serif; font-weight: bold; }



```
    <meta name="viewport" content="initial-scale=1,maximum-scale=1,user-scalable=no" />
     <script src="https://api.mapbox.com/mapbox-gl-js/v2.1.0/mapbox-gl.js"></script>
     <link href="https://api.mapbox.com/mapbox-gl-js/v2.1.0/mapbox-gl.css" rel="stylesheet" />
    <style>
        body { margin: 0; padding: 0; }
        #map { position: absolute; top: 0; bottom: 0; width: 100%; }
        #map canvas {cursor: move;}

    /*add popup and legend info, Tutorial 7*/

        ADD POPUP STYLE INFO HERE


    </style>
</head>
```

3b Second, add some code to the ***<script>*** section. Below your map controls, add a popup **variable**:

var popup = new mapboxgl.Popup({
closeButton: false,
closeOnClick: false
});



```
    map.addControl(new mapboxgl.NavigationControl());
    map.dragRotate.disable();
    map.touchZoomRotate.disableRotation();
    map.scrollZoom.disable();
    map.addControl(new mapboxgl.FullscreenControl());




//add popup variable (Tutorial 7)

 ADD POPUP VAR HERE


//add map.on mouse enter function (Tutorial 7)

 ADD MOUSEENTER, MOUSELEAVE, AND ONCLICK FUNCTIONS HERE

 ABOVE THE </SCRIPT>
```

Below this, add three new functions: one for ***mouseenter*** (when you mouse over a county), for ***mouseleave*** (when your mouse leaves a county), and for ***onclick*** (when you click a feature). The *mouseenter* will generate a new popup, *mouseleave* will remove it, and *onclick* will act as a backup to address some glitchy features with the mouseover function.

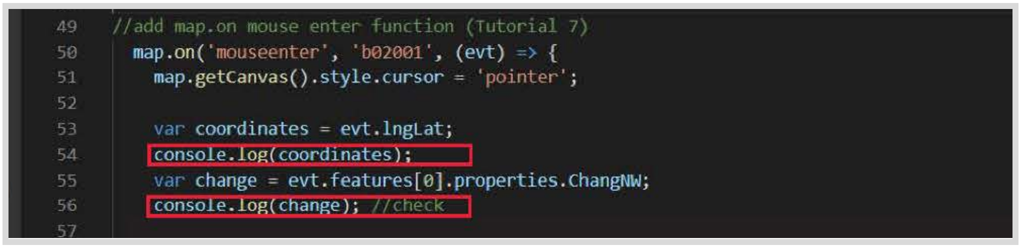
Note: make sure that you **replace EACH 'b02001'** with your own Shapefile **layer name** (as you changed it in Step 2c). Likewise, **replace EACH 'ChangeNW'** with the name of your **Change in Non-White attribute field**, exported from QGIS in Step 1b.

```
map.on('mouseenter', 'b02001', (evt) => {
  map.getCanvas().style.cursor = 'pointer';

  var coordinates = evt.lngLat;
  var change = evt.features[0].properties.ChangeNW;

  while (Math.abs(evt.lngLat.lng - coordinates[0]) > 180) {
      coordinates[0] += evt.lngLat.lng > coordinates[0] ? 360 : -360;
  }

  popup
  .setLngLat(coordinates)
  .setHTML('<h1>' + 'Non-white population change: ' + '<h2>' + change + '%' + '<h1>' + 'between 2013
     and 2019.') //change text here as you like
  .addTo(map);
  });

map.on('click', 'b02001', (evt) => {
  var change = evt.features[0].properties.ChangeNW;
  popup
  .setLngLat(evt.lngLat)
  .setHTML('<h1>' + 'Non-white population change: ' + '<h2>' + change + '%' + '<h1>' + 'between 2013
     and 2019.')
  .addTo(map);
  });

map.on('mouseleave', 'b02001', () => {
  map.getCanvas().style.cursor = 'move';
  popup.remove();
});
```

3c Add a few **console.log** messages to check code works as intended. This is a helpful trick to make sure that your variables are all carrying the right information.

Check that the code is correctly finding the Long/Lat and percent change information that the popup will use (to locate itself and display county information, respectively). To do this, add the following lines of code after their respective variables are defined, as shown:

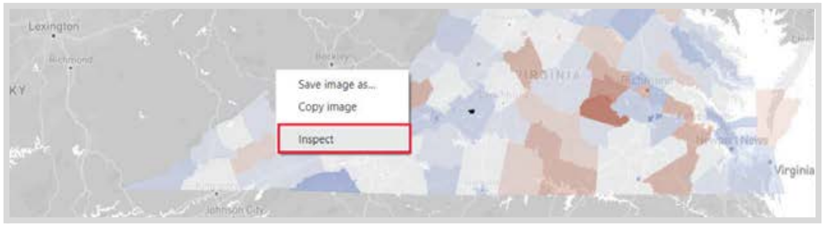
```
console.log(coordinates);

console.log(change);
```

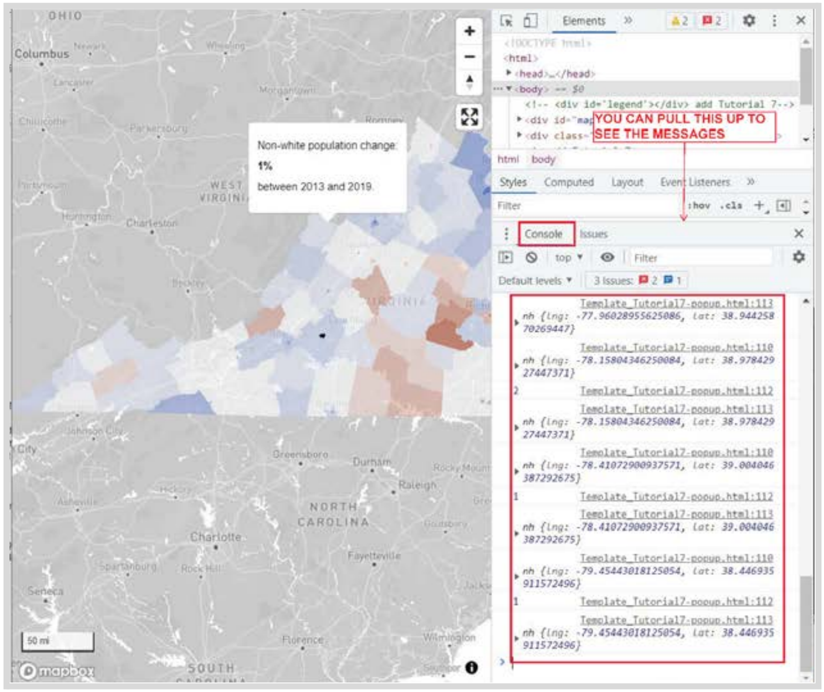


```
//add map.on mouse enter function (Tutorial 7)
  map.on('mouseenter', 'b02001', (evt) => {
    map.getCanvas().style.cursor = 'pointer';

    var coordinates = evt.lngLat;
    console.log(coordinates);
    var change = evt.features[0].properties.ChangNW;
    console.log(change); //check

```

We're only using this for reference, and so you can see the web page Inspector console, but keep in mind that it's a helpful trick for troubleshooting your code later on. You can add a console.log to check behavior before breakpoints and errors that you don't understand.

3d Save your code, and reload your Chrome preview page. To check the console messages, right click on the Chrome webpage that loads your map and "**Inspect**". In the bottom window, select the "**Console**" tab. Here, you should see log messages appear when you trigger the popup functions by hovering over counties.





Non-white population change
1%
between 2013 and 2019

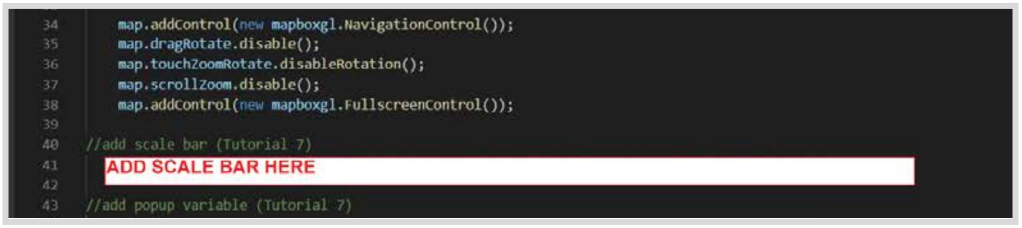
```
<!DOCTYPE html>
<html>
 ▸<head>…</head>
··· ▾<body> == $0
    <!-- <div id='legend'></div> add Tutorial 7-->
   ▸<div id="map
   ▸<div class=
```

```
                     Template_Tutorial7-popup.html:113
▸ nh {lng: -77.96028955625086, lat: 38.944258
  70269447}
                     Template_Tutorial7-popup.html:110
▸ nh {lng: -78.15804346250084, lat: 38.978429
  27447371}
2                    Template_Tutorial7-popup.html:112
                     Template_Tutorial7-popup.html:113
▸ nh {lng: -78.15804346250084, lat: 38.978429
  27447371}
                     Template_Tutorial7-popup.html:110
▸ nh {lng: -78.41072900937571, lat: 39.004046
  387292675}
1                    Template_Tutorial7-popup.html:112
                     Template_Tutorial7-popup.html:113
▸ nh {lng: -78.41072900937571, lat: 39.004046
  387292675}
                     Template_Tutorial7-popup.html:110
▸ nh {lng: -79.45443018125054, lat: 38.446935
  911572496}
1                    Template_Tutorial7-popup.html:112
                     Template_Tutorial7-popup.html:113
▸ nh {lng: -79.45443018125054, lat: 38.446935
  911572496}
```

Step 4: Add legend (via mapbox layer) and scale bar

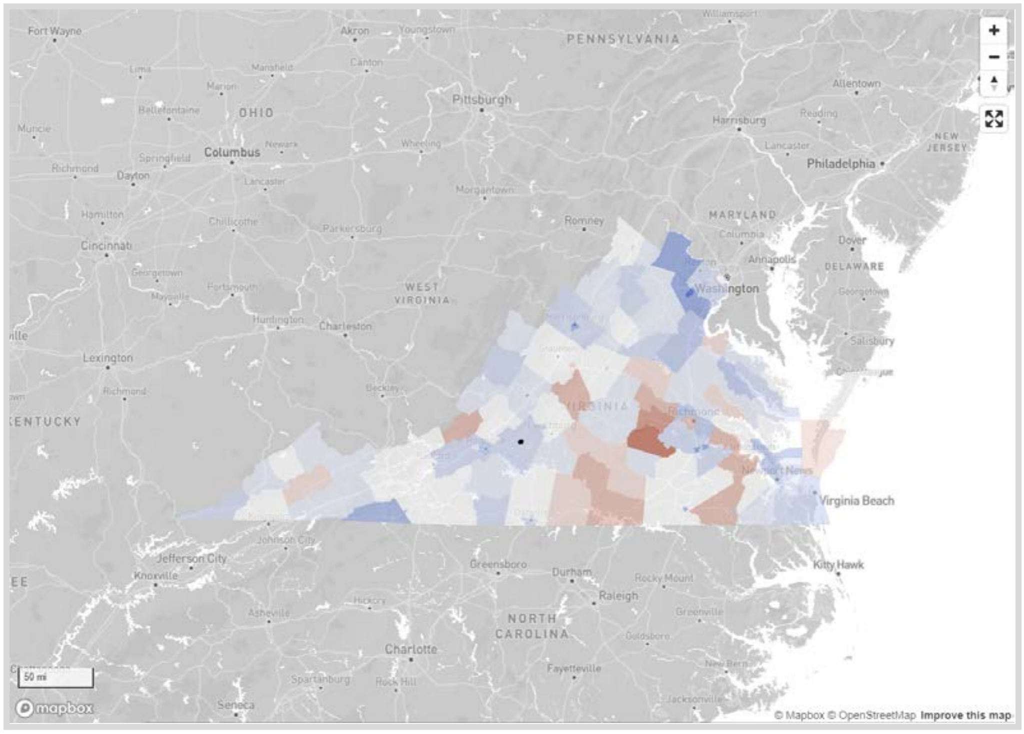
4a After the map controls (before the *var popup*), add:

map.addControl(new mapboxgl.ScaleControl({position: 'bottom-right',unit: 'imperial'}));



```
    map.addControl(new mapboxgl.NavigationControl());
    map.dragRotate.disable();
    map.touchZoomRotate.disableRotation();
    map.scrollZoom.disable();
    map.addControl(new mapboxgl.FullscreenControl());

//add scale bar (Tutorial 7)
    ADD SCALE BAR HERE

//add popup variable (Tutorial 7)
```

4b Reload and check out the new scale at the bottom left of your map.



- **Bonus -**

Step 5: Layers of those years can be toggled on and off (by default map only shows percentage change):

https://docs.mapbox.com/mapbox-gl-js/example/toggle-layers/