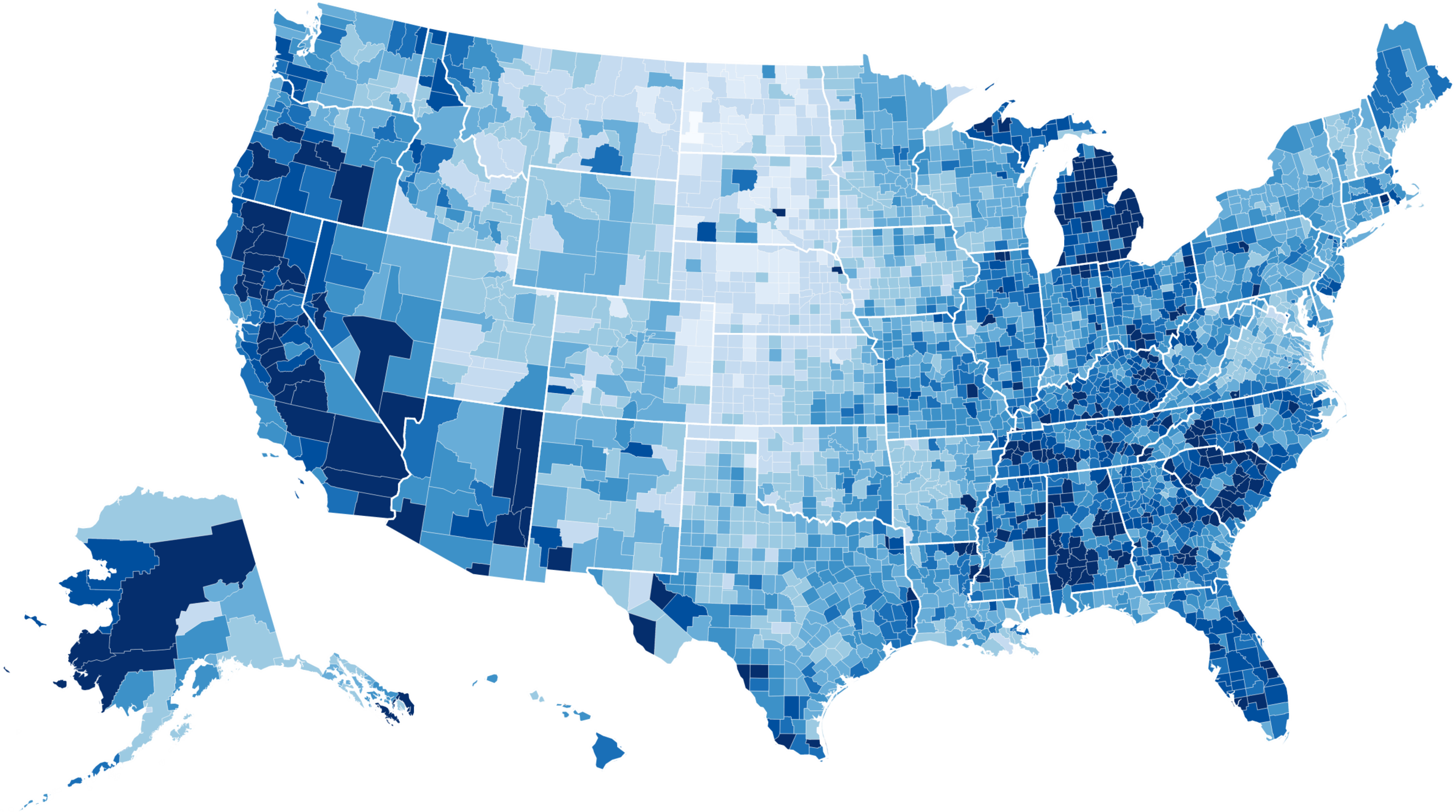
# Scalable Vector Graphics

IGME 230

# SVG (scalable vector graphics)

alternative to `<canvas>` and WebGL for 2D graphics in the browser.

# SVG : where is it used?



# SVG = XML

They look like HTML, but behave a bit differently.

# Drawing shapes

```
<!-- start tag and size attributes -->
<svg width='500' height='500'>

  <!-- draw some stuff -->
  <circle id='d' cx='100' cy='100' r='50' />
  <circle id='e' cx='200' cy='200' r='25' />

</svg><!-- end tag -->
```

# Formatting (with CSS)

```
<style>
    svg {
       position: absolute;
    }
    #d {
      fill: red;
      stroke-width:10;
      stroke:green;
    }
</style>
<svg width='500' height='500'>

  <circle id='d' cx='100' cy='100' r='50' />
  <circle id='e' cx='200' cy='200' r='25' />

</svg>
```

# SVG Element Reference

# JavaScript and SVG

Assuming a single SVG that fills the screen, draw one rectangle

```
var xmlns = 'http://www.w3.org/2000/svg';
var svg = document.querySelector( 'svg' );

var rect = document.createElementNS(xmlns, 'rect');
rect.setAttribute( 'x', 0 );
rect.setAttribute( 'y', 0 );
rect.setAttribute( 'width',  40 );
rect.setAttribute( 'height', 40 );

svg.appendChild( circle );
```

# JavaScript and SVG

Assuming a single SVG that fills the screen, draw a hundred randomly sized, randomly colored red circles.

```
var xmlns = 'http://www.w3.org/2000/svg';
var svg = document.querySelector( 'svg' );

var circles = [];
for( var i = 0; i < 100; i++ ) {
  var circle = document.createElementNS( xmlns, 'circle' );

  circle.setAttribute( 'cx', Math.random() * window.innerWidth );
  circle.setAttribute( 'cy', Math.random() * window.innerHeight );
  circle.setAttribute( 'r' , Math.random() * 40 );

  svg.appendChild( circle );
};
```

# Animating SVG elements

Assuming a single SVG that fills the screen, draw a thousand randomly sized, randomly colored circles that move across the screen.

- Create helper functions that randomly color, position, and set the speed of individual circle elements.
- Create a thousand circle elements and store them in an array
- Call our helper functions on each item our array of circles to create variation between individual elements
- Create a function to move a single circle according to its speed
- 60 times per second, call our circle moving function on every circle in our array.

# ICE: Abstract Animation

Create an abstract animation. Use at least four different types of SVG elements in your composition. All SVG elements (with the exception of the container `<svg>` tag) should be dynamically added via JavaScript. Try to create at least two different "scenes", moments where the animation changes in a significant way.

Place the resulting HTML file (and any external resources if needed) on Banjo and provide a link from your homepage.