

# ArcGIS Maps SDK for JavaScript: Data Visualization

Anne Fitz & Kristian Ekenes

2023 ESRI USER CONFERENCE





Discover how to visualize geodata  
in **meaningful** ways using the  
ArcGIS Maps SDK for JavaScript

One visualization cannot tell  
the whole story





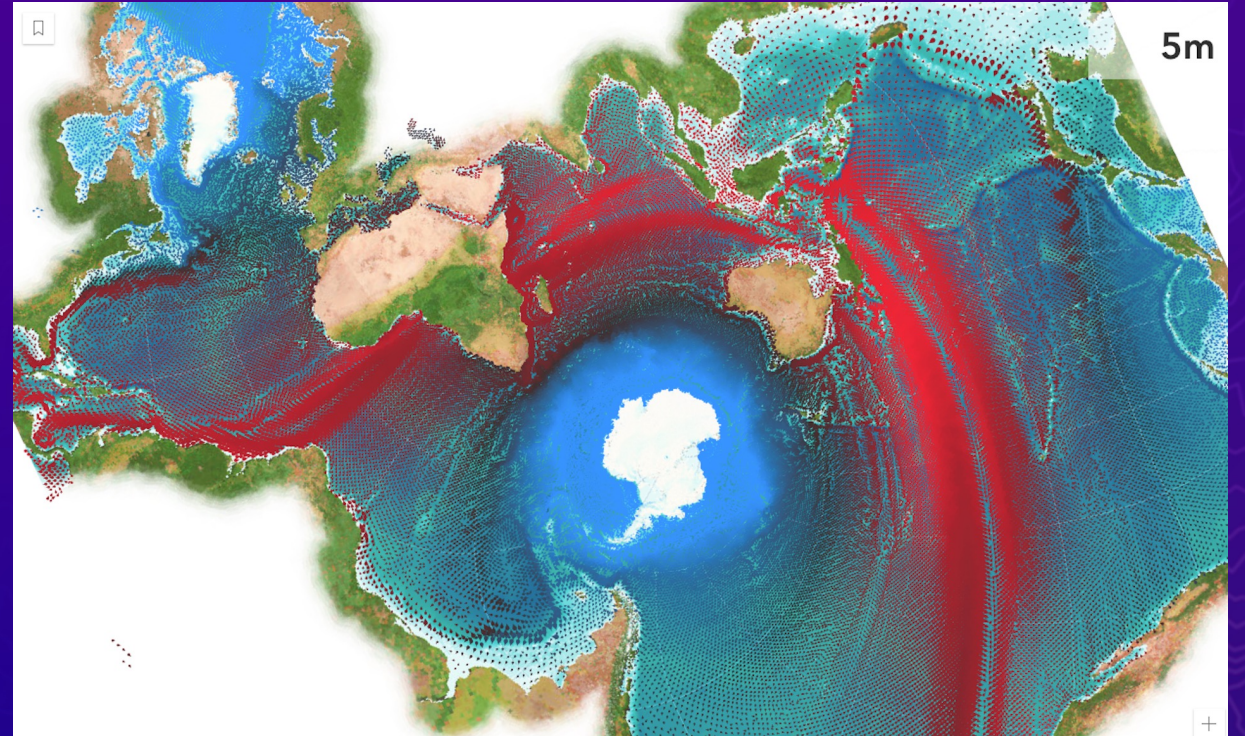
# API Overview

Renderers and symbols



What can we visualize?

- Where?
- What?
- How much?
- When?
- Multivariate



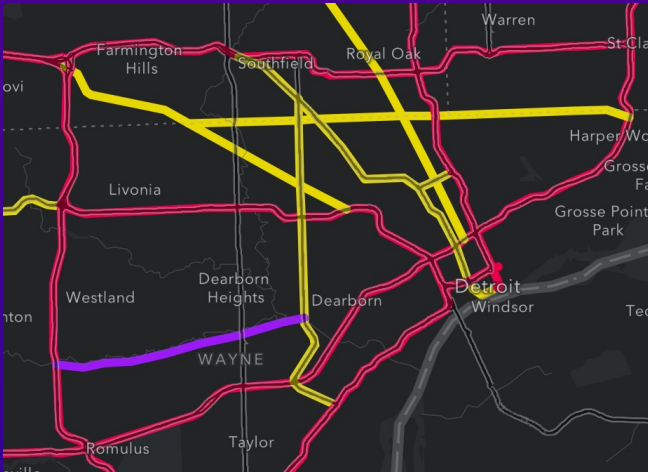


# Symbols

The background is a deep purple gradient. In the upper right, there are several glowing, flowing lines in shades of blue, purple, and orange, resembling light trails or energy flows. A faint, semi-transparent map of the Americas is visible in the lower half of the image, with the United States and Canada appearing as lighter purple shapes against the darker background.

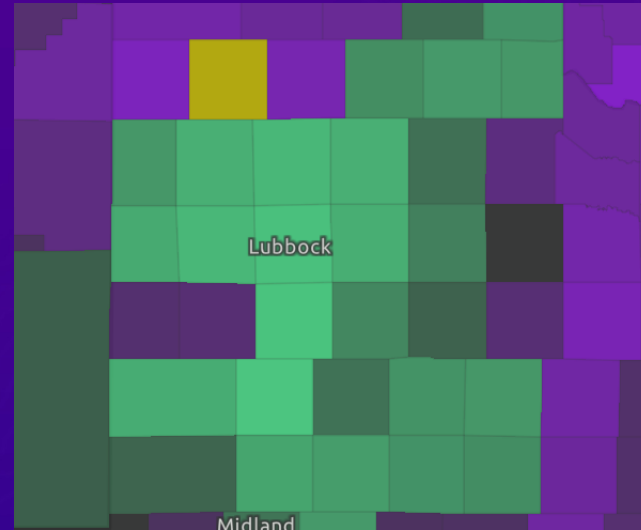


# Symbol primitives



## SimpleLineStyle

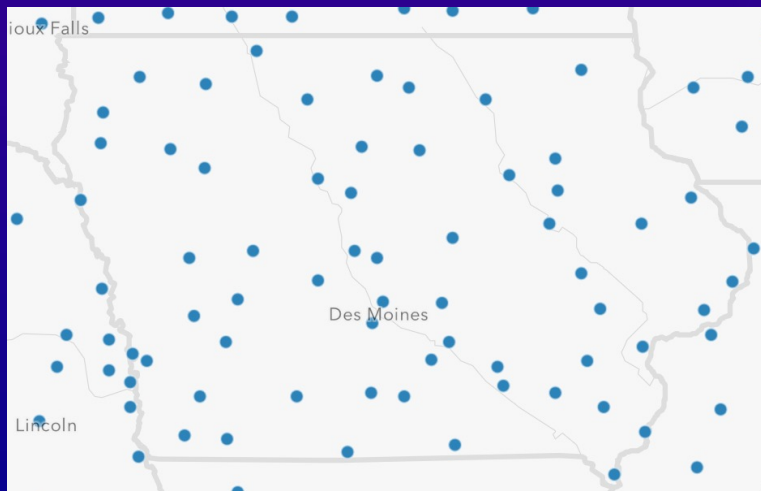
```
const sls = {  
  type: "simple-line",  
  width: 1,  
  color: [255, 255, 255, 1],  
  style: "solid",  
  cap: "round",  
  join: "round"  
}
```



## SimpleFillSymbol

```
const sfs = {  
  type: "simple-fill",  
  color: [0, 0, 0, 0.25],  
  style: "solid",  
  outline: {  
    width: 1,  
    color: [255, 255, 255, 1]  
  }  
}
```

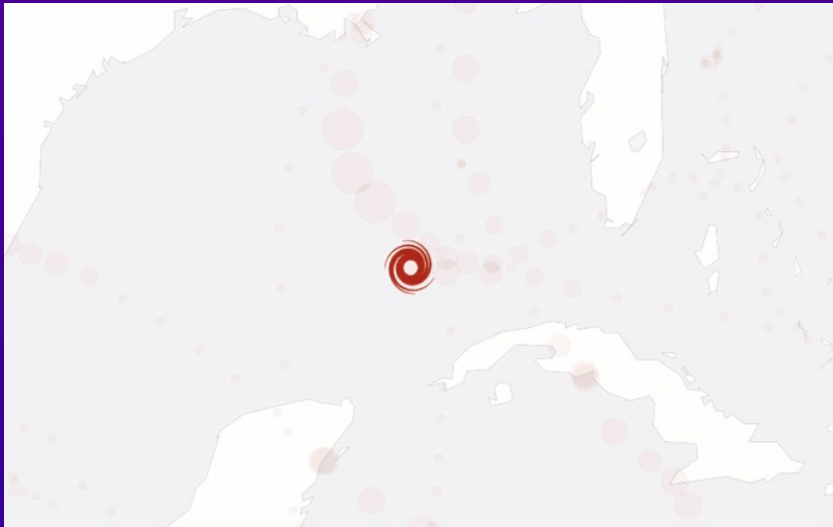
## SimpleMarkerSymbol



```
const sms = {  
  type: "simple-marker",  
  color: [255, 255, 255, 0.25],  
  size: 12,  
  style: "circle",  
  outline: {  
    width: 1,  
    color: [255, 255, 255, 1]  
  }  
}
```

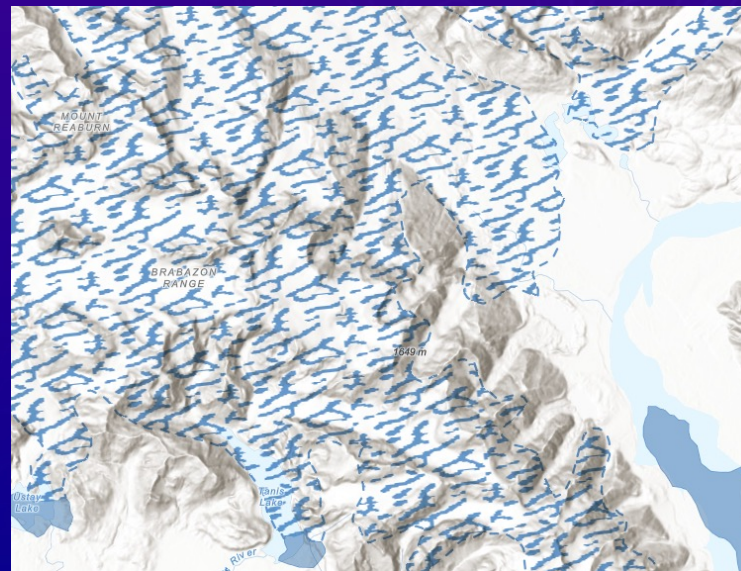


# Picture symbols



PictureMarkerSymbol

```
const pms = {  
  type: "picture-marker",  
  url: "image-url",  
  height: 12,  
  width: 12  
}
```



PictureFillSymbol

```
const pfs = {  
  type: "picture-fill",  
  url: "swamp.png",  
  width: 12,  
  height: 12,  
  xoffset: 0,  
  yoffset: 0  
}
```



# CIMSymbol

High quality, scalable



Scaled vector  
symbol



Scaled image

Symbol layers



Symbol



1

layer1



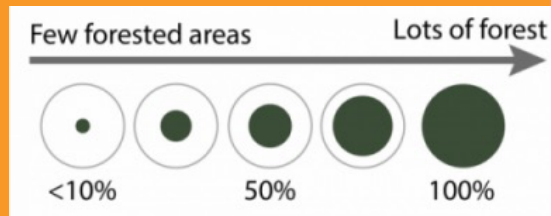
layer2



layer3

## Primitive Overrides

Dynamically update attributes of an individual symbol layer using Arcade



# How is it used in the ArcGIS Maps SDK for JavaScript?

```
// require(["esri/symbols/CIMSymbol"], function(CIMSymbol)
const cimSymbol = new CIMSymbol({
  data: {
    type: "CIMSymbolReference",
    symbol: {
      type: "CIMLineSymbol", // CIMPointSymbol or CIMPolygonSymbol
      symbolLayers: [{ ... }]
    },
    primitiveOverrides: [{ ... }]
  }
});
```

follows the [cim-spec](#)

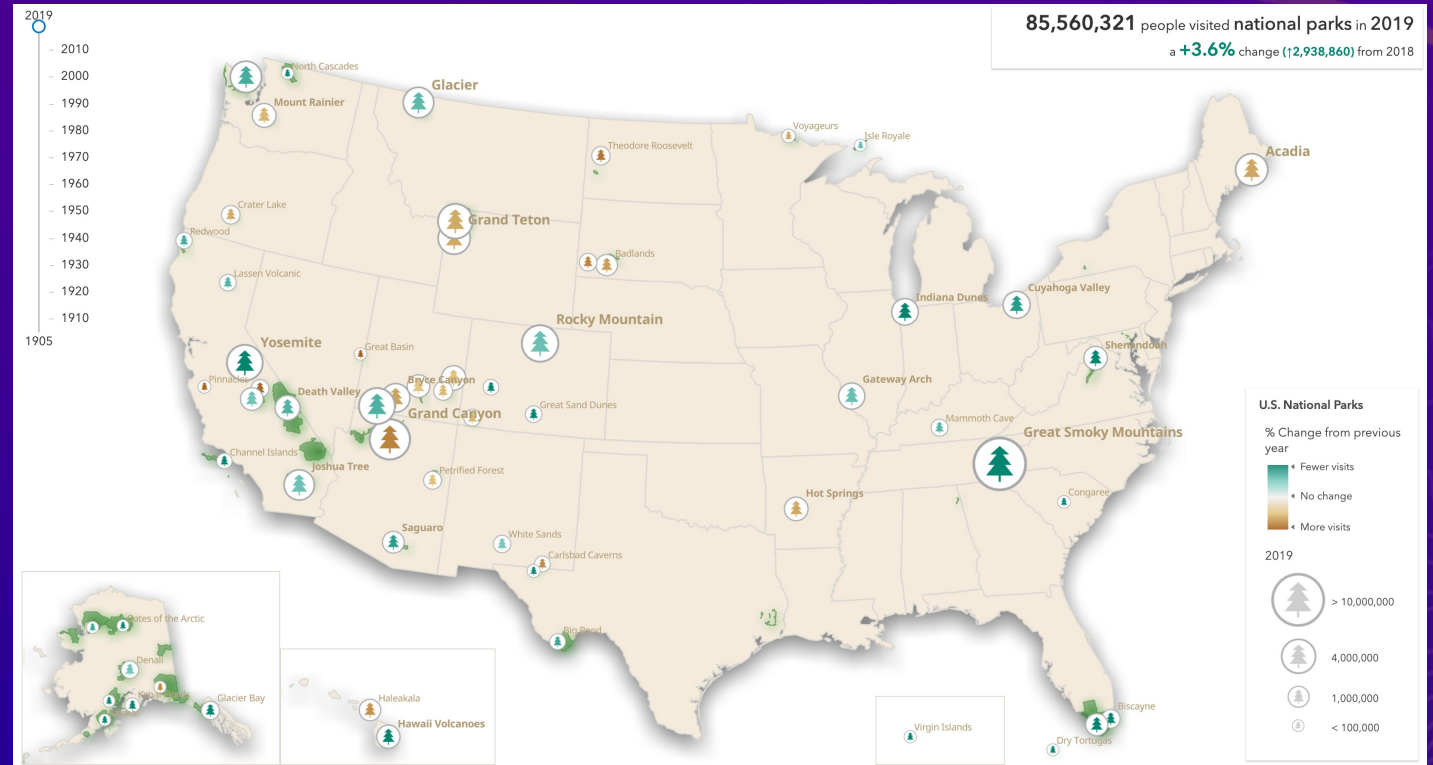


# WebStyleSymbol

## 2d styles

### Cartographic Information Model (CIM)

- Vectors
- Scalable
- Multi-layer
- Overrides



```
const webStyleSymbol = new WebStyleSymbol({  
  name: "park",  
  styleName: "Esri2DPointSymbolsStyle"  
});
```

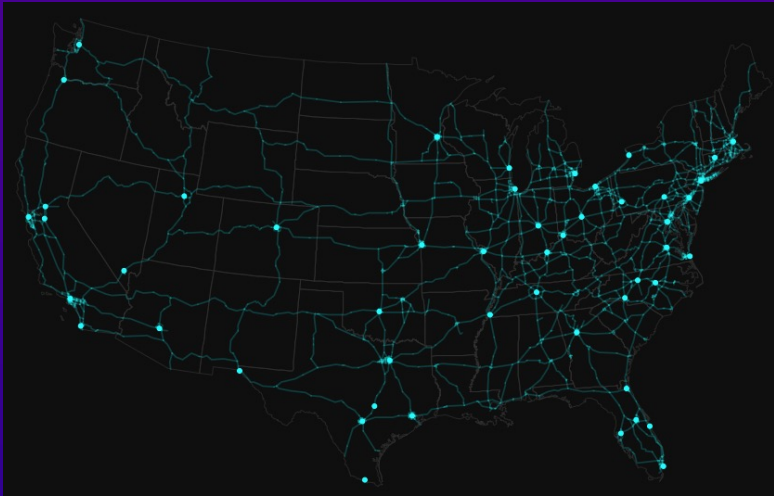
# Renderers



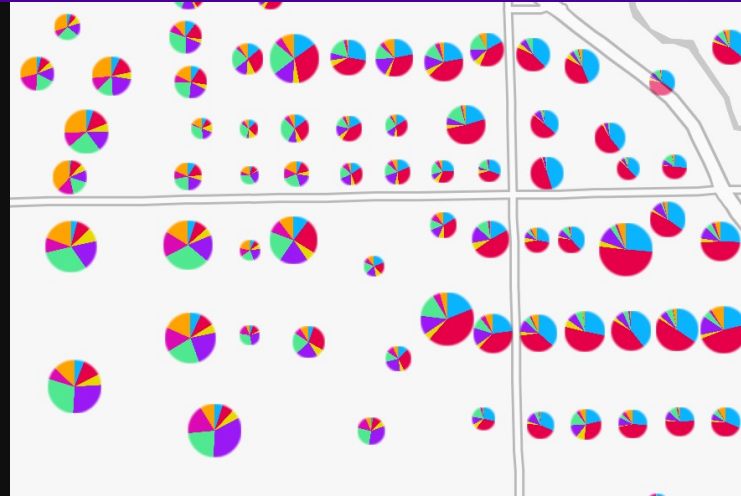


# Renderers

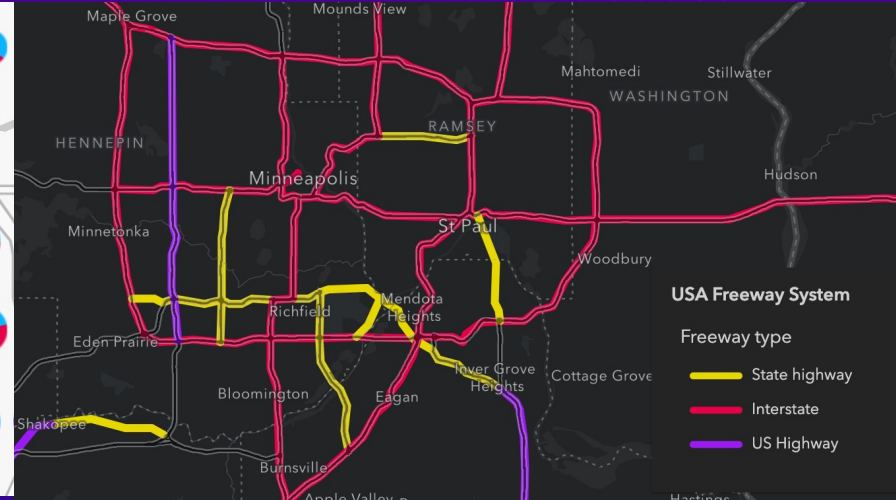
## SimpleRenderer



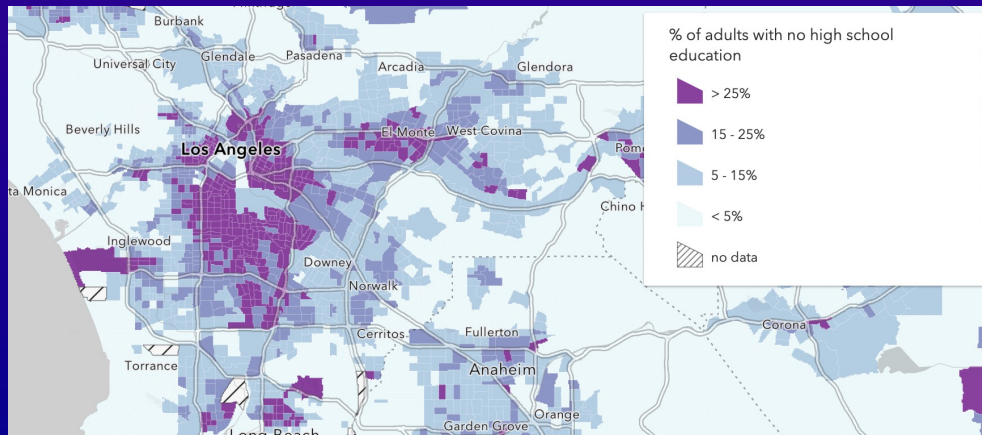
## PieChartRenderer



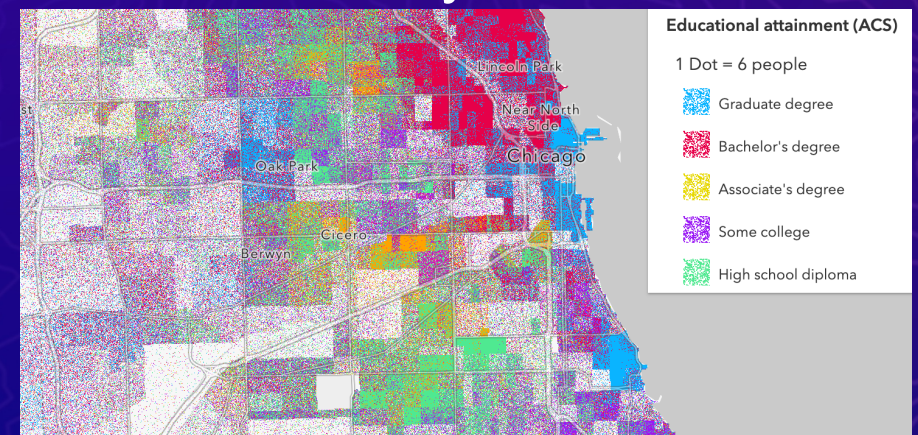
## UniqueValueRenderer



## ClassBreaksRenderer



## DotDensityRenderer

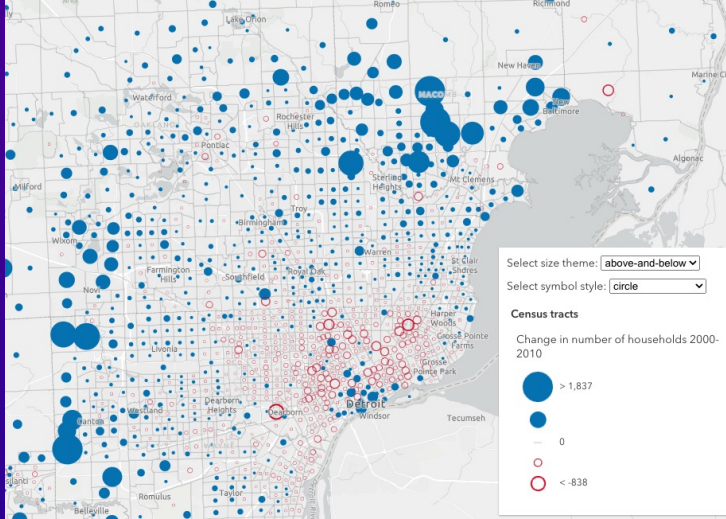




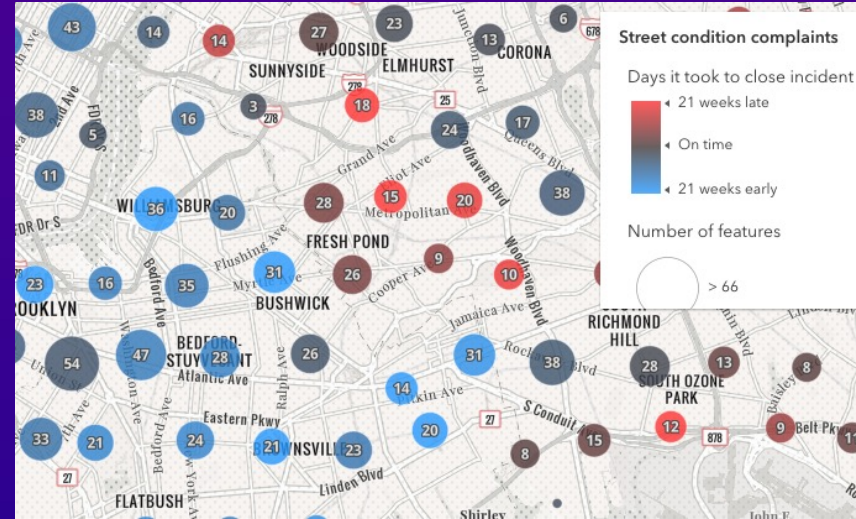
# Renderers

Derivative styles created from standard renderer types

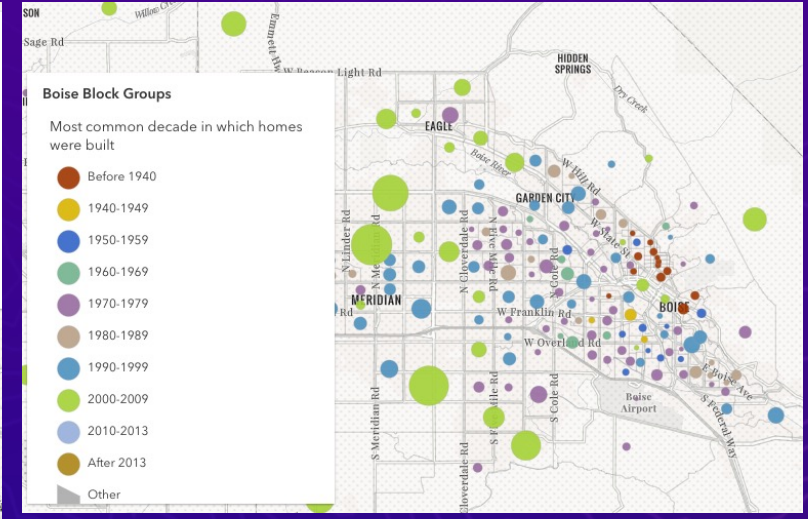
## Above and below



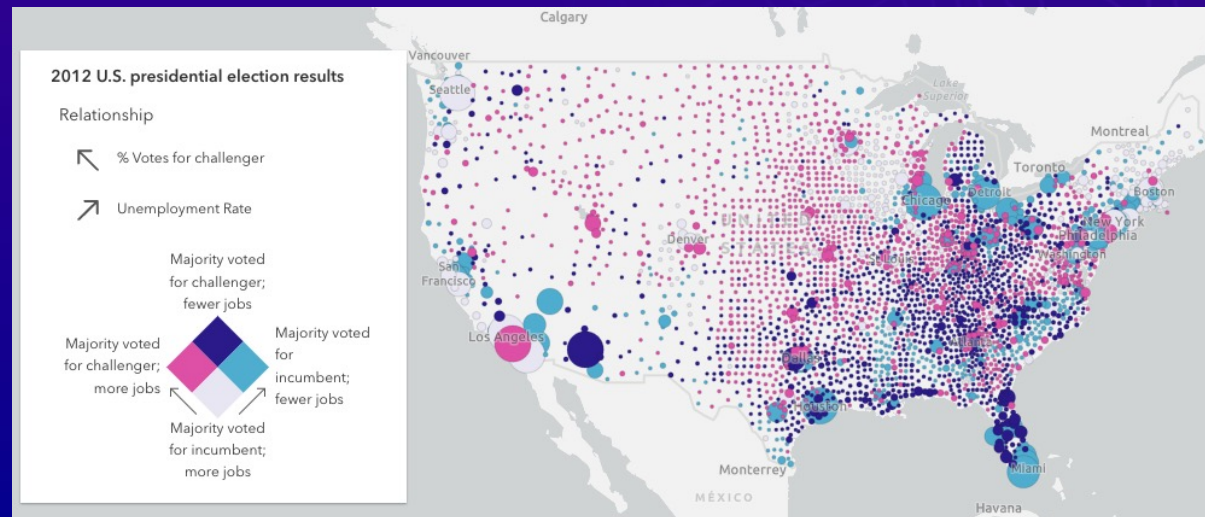
## Age



## Predominance



## Relationship

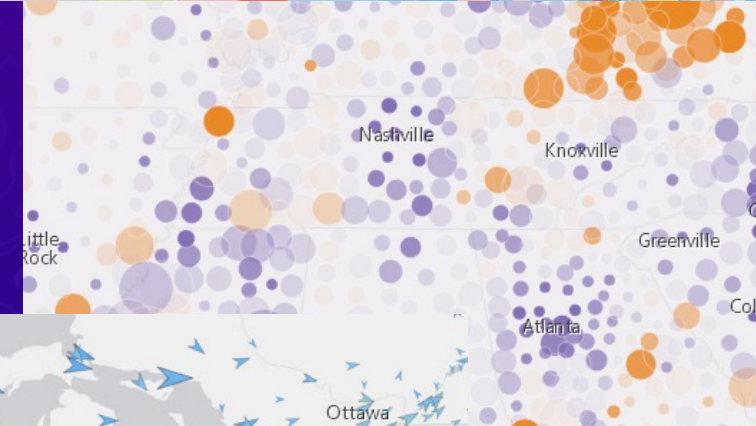
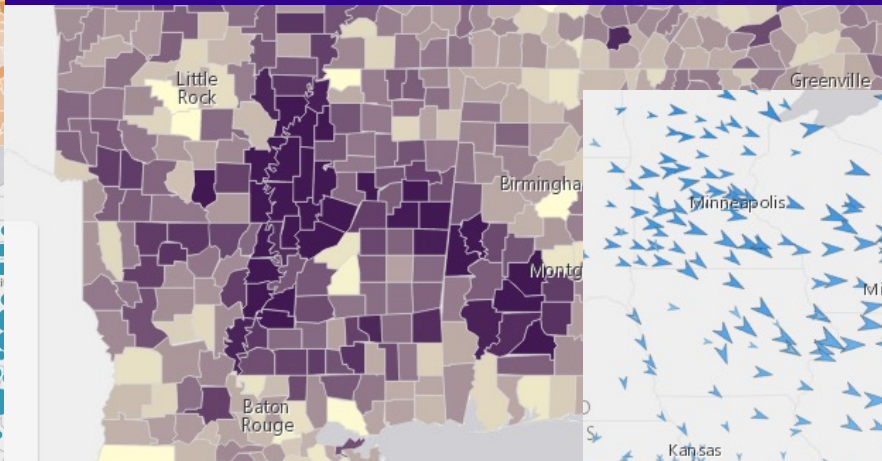
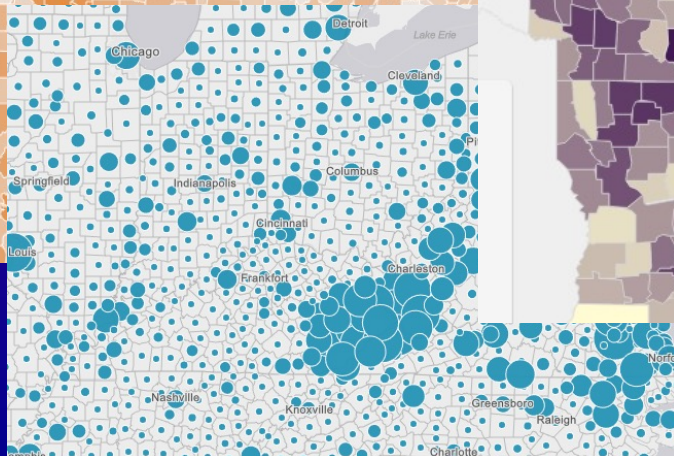
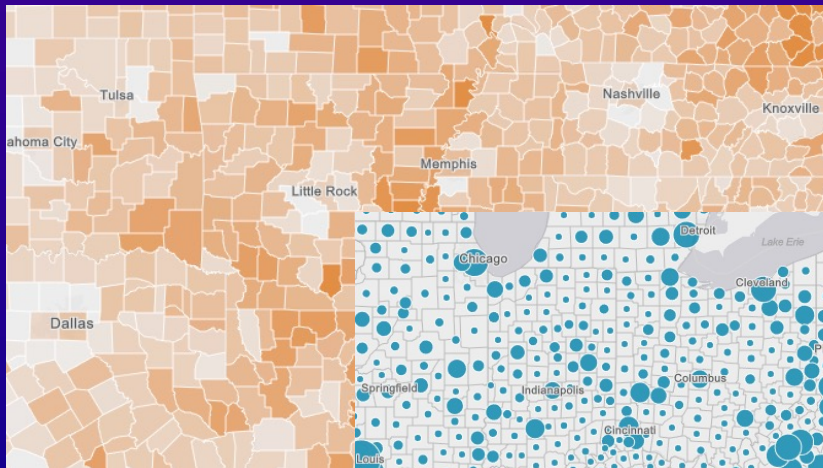
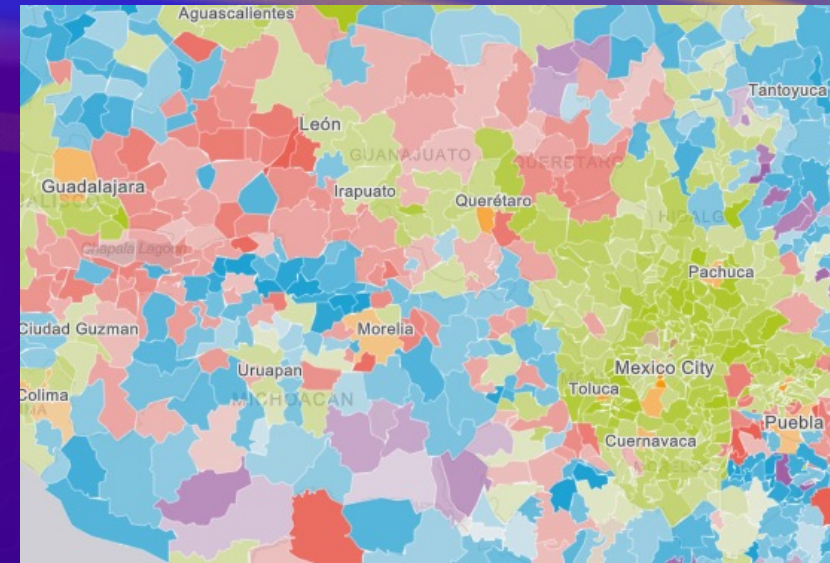




# Visual variables

- Color
- Size
- Opacity
- Rotation

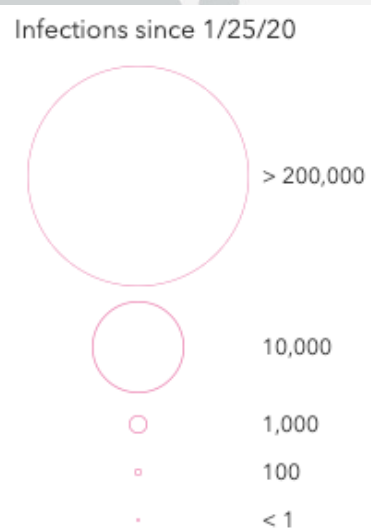
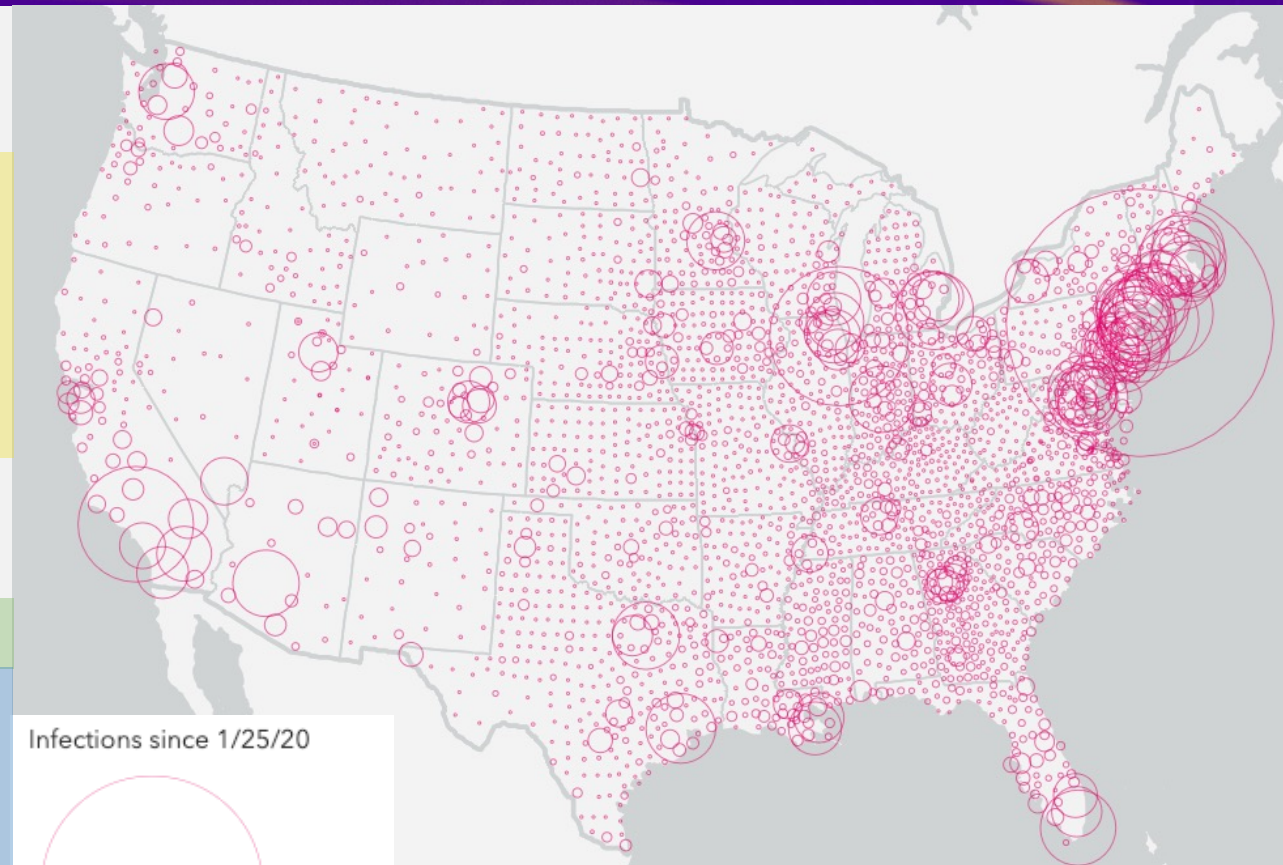
- A property of the renderer
- Overrides symbol properties with data
- For **numeric** data-driven continuous visualizations





# Data-driven visualization

```
layer.renderer = new SimpleRenderer({  
  symbol: new SimpleMarkerSymbol({  
    style: "none",  
    outline: new SimpleLineSymbol({  
      color: new Color("rgba(227, 0, 106)"),  
      width: 0.5  
    })  
  }  
}),  
visualVariables: [  
  new SizeVariable({  
    field: "INFECTIONS_6_1_2020",  
    stops: [  
      { value: 1, size: "2px" },  
      { value: 100, size: "4px" },  
      { value: 1000, size: "10px" },  
      { value: 10000, size: "50px" },  
      { value: 200000, size: "200px" }  
    ]  
  }  
])  
});
```



Field value



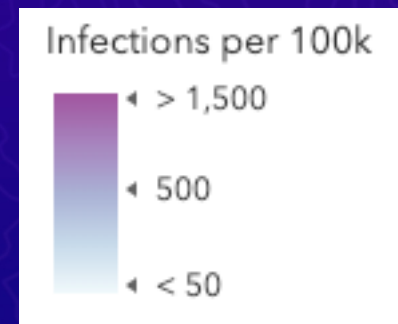
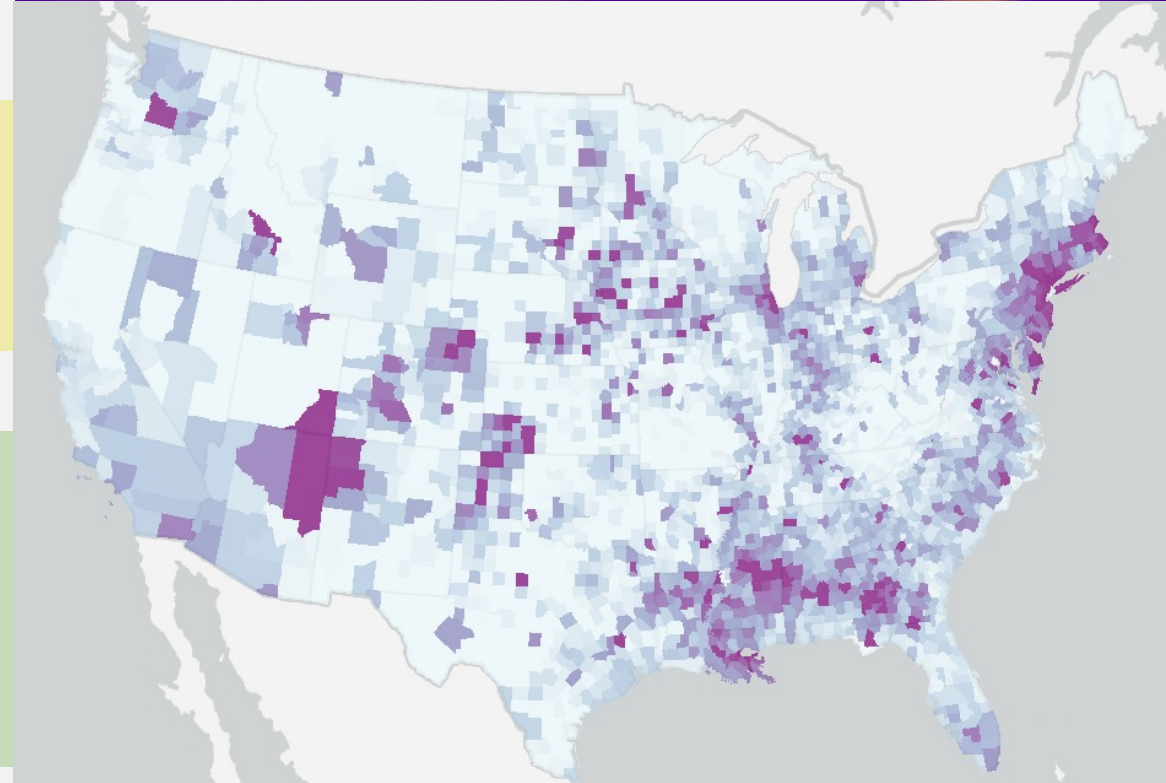
# Data-driven visualization

```
layer.renderer = new SimpleRenderer({
  symbol: new SimpleFillSymbol({
    outline: new SimpleLineSymbol({
      color: "rgba(128,128,128,0.4)",
      width: 0
    })
  })
}),
visualVariables: [
  new ColorVariable({
    valueExpression: `
      var currentDayValue = $feature["DAYSTRING_05_31_2020"];
      var currentDaySplit = Split(currentDayValue, "|");
      var infections = Number(currentDaySplit[0]);
      var deaths = Number(currentDaySplit[1]);
      var population = $feature.Population_1;
      return (infections / population ) * 100000;
    `,
    valueExpressionTitle: `Infections per 100k`,
    stops: [
      { value: 50, color: "#edf8fb" },
      { value: 200, color: "#b3cde3" },
      { value: 500, color: "#8c96c6" },
      { value: 1000, color: "#8856a7" },
      { value: 1500, color: "#810f7c" }
    ]
  })
]
});
```

Symbol

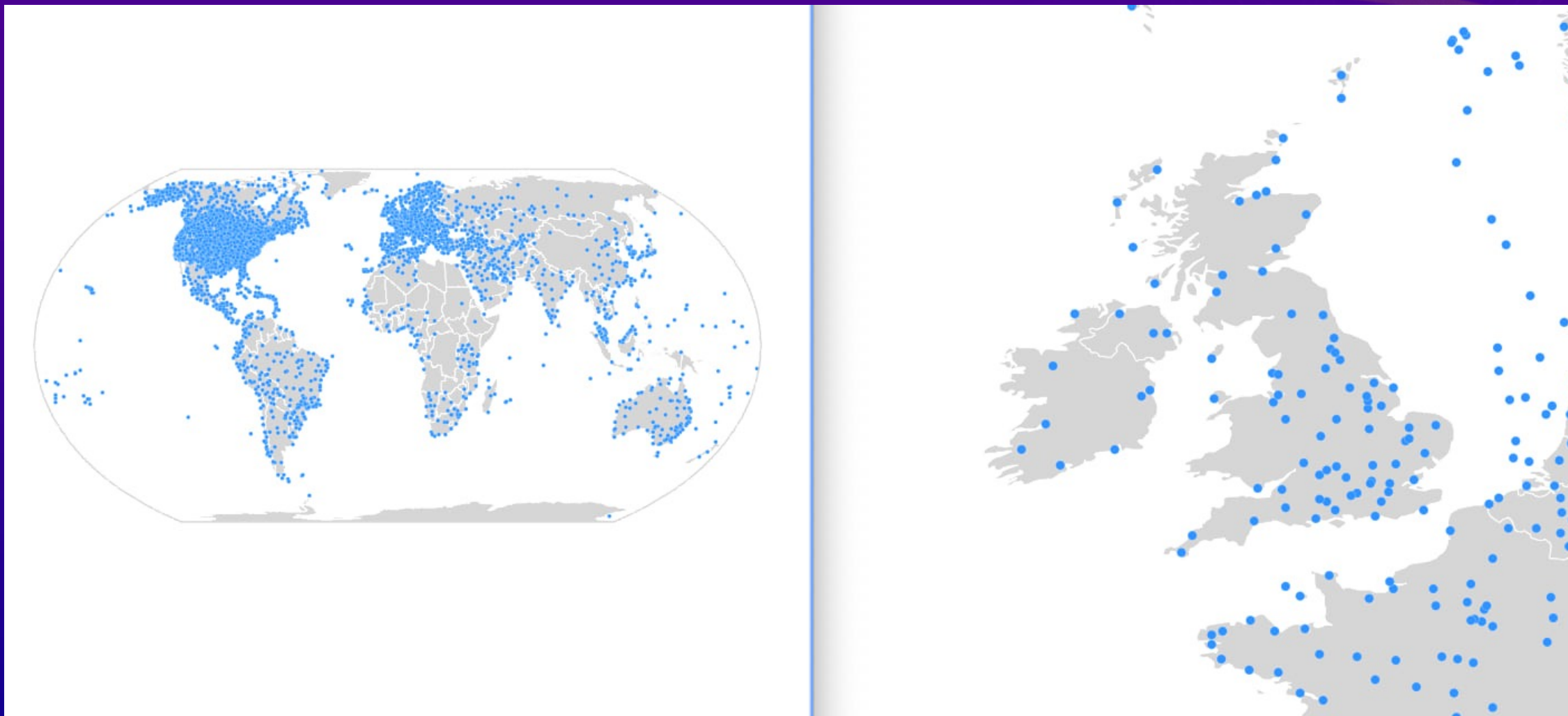
Data

Color  
driven  
by data



Arcade  
expression

# Symbol Size by Scale



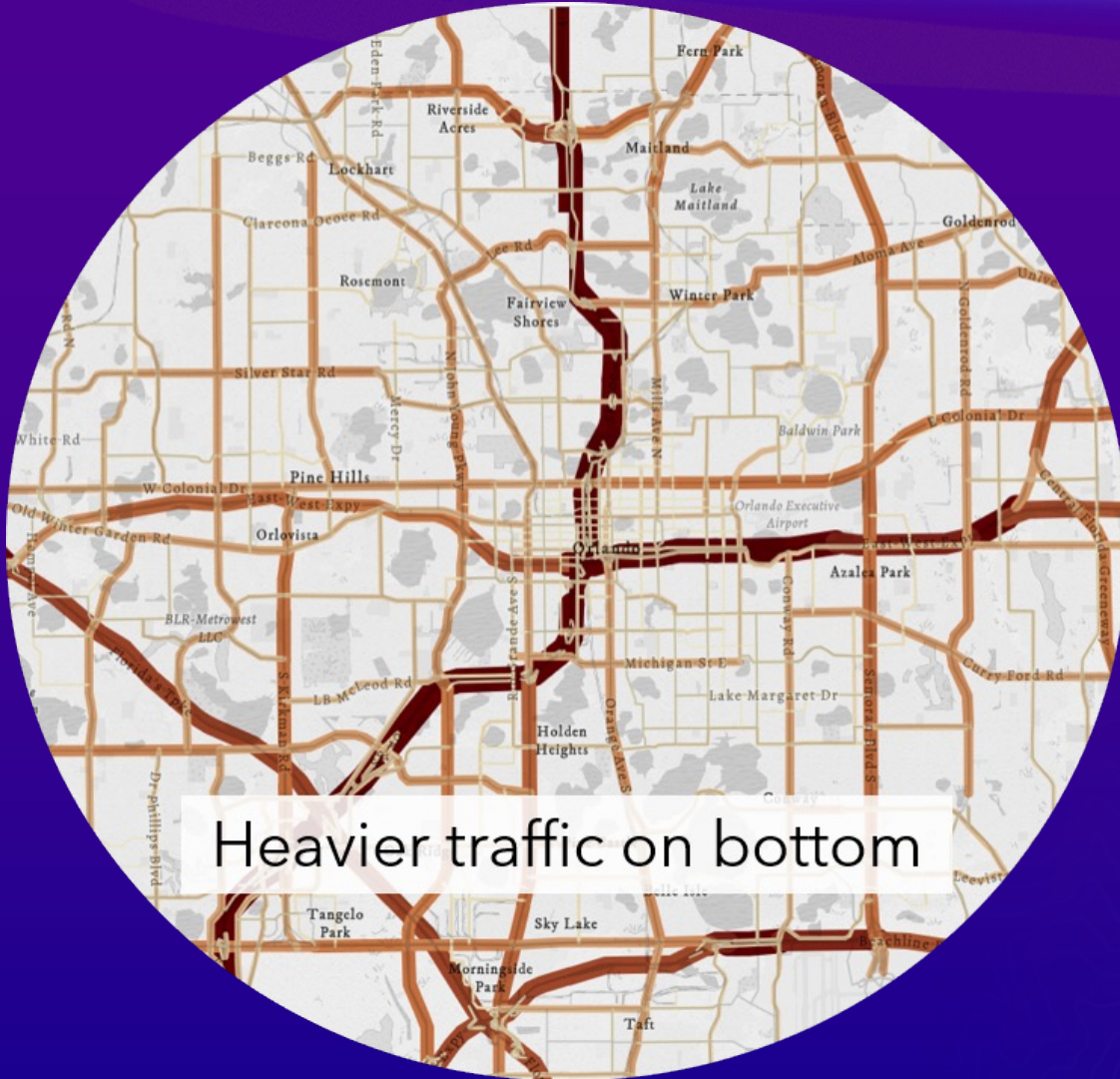
[ArcGIS blog: How and why to size symbols by scale in web maps](#)



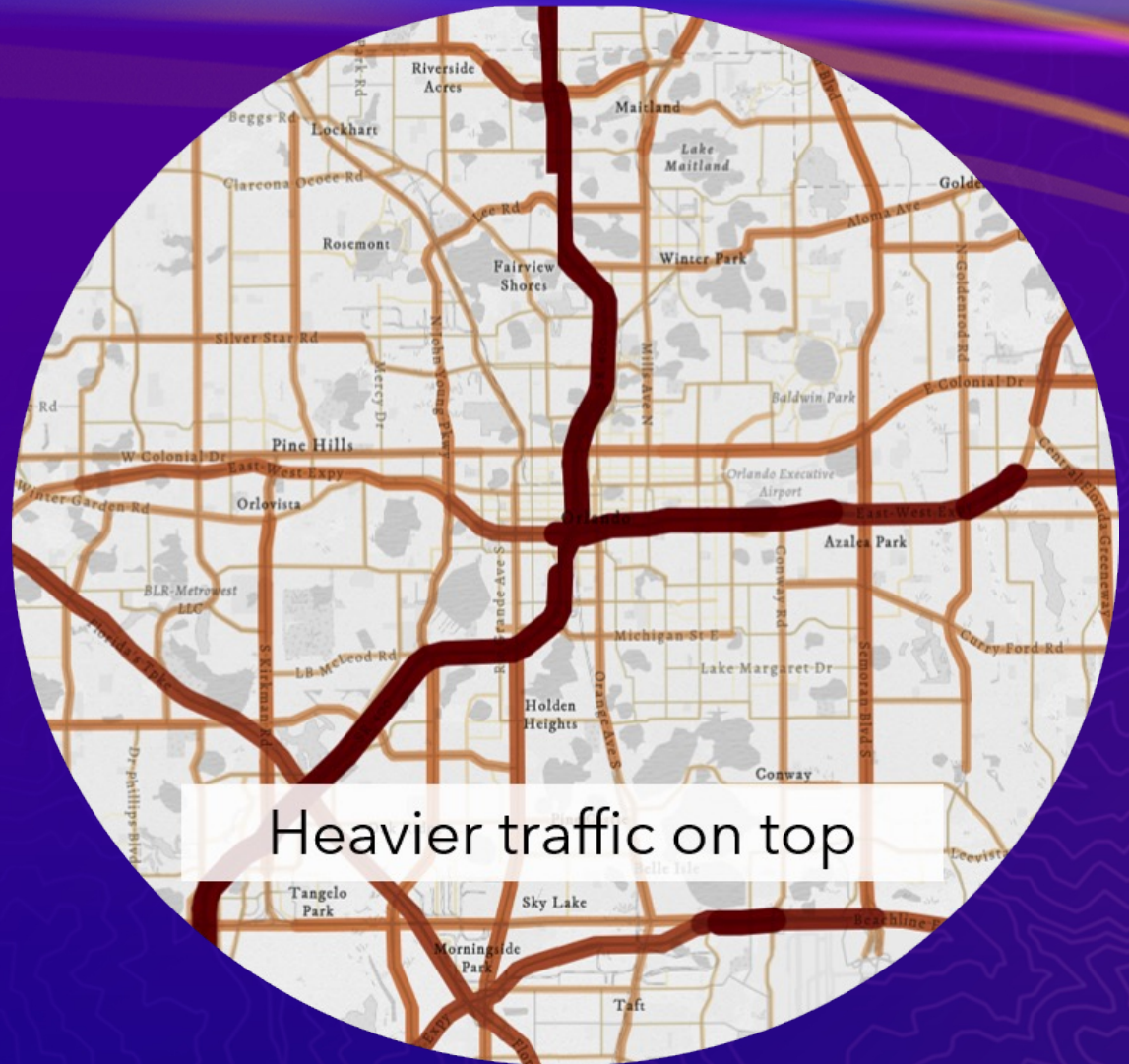
# Aggregation and Density

The background features a dark purple gradient. On the right side, there are several thick, wavy, glowing lines in shades of blue, purple, and orange. In the lower-left and central areas, a faint, light-colored map of Europe is visible, showing the outlines of the continent and its major landmasses.





Heavier traffic on bottom



Heavier traffic on top

# Drawing order control

Feature layers, CSV, GeoJSON,  
OGC Feature Layers

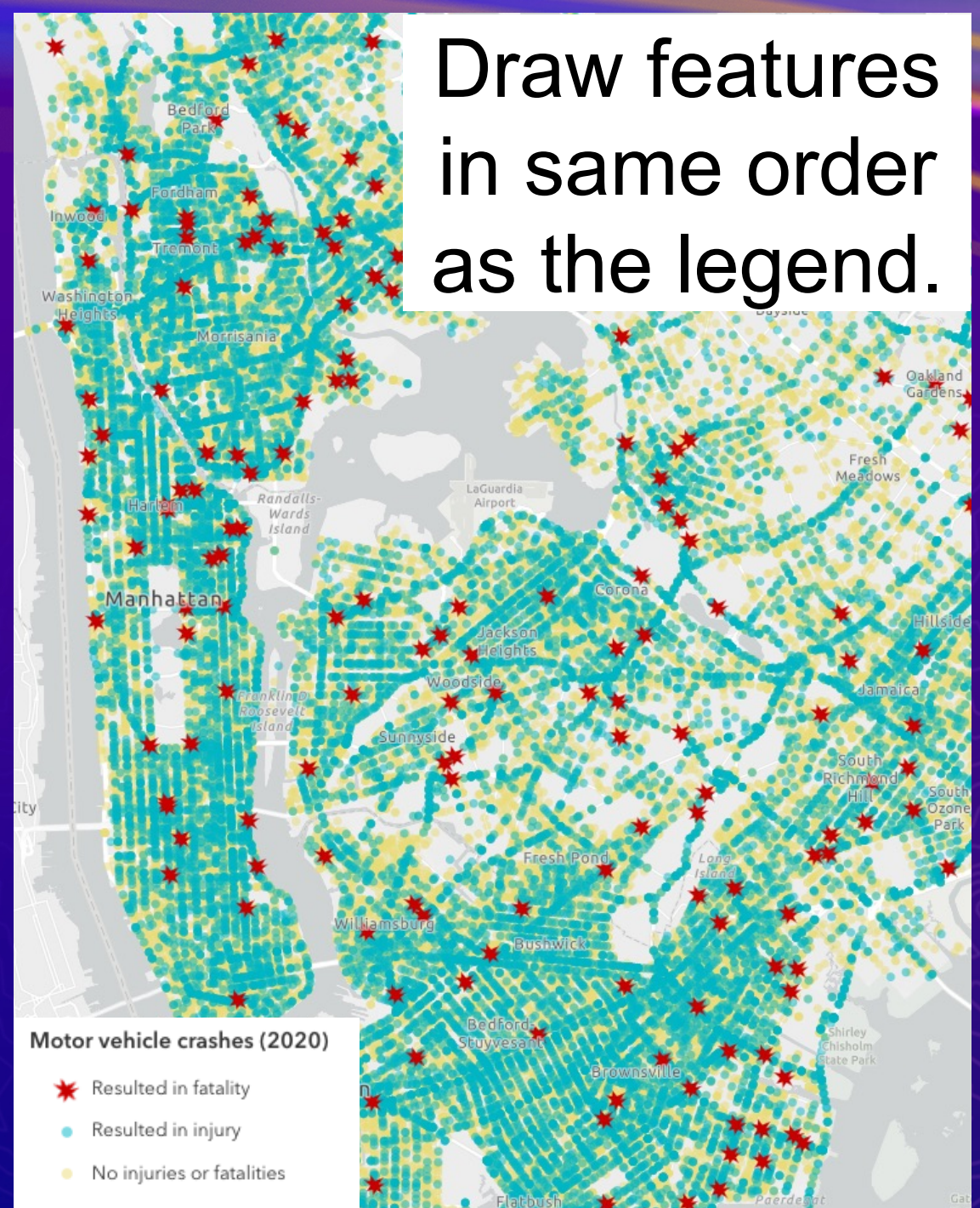


```

renderer: {
  type: "unique-value",
  labelingInfo: {
    title: "Crash location"
  },
  orderByClassesEnabled: true,
  valueExpression: `
    When(☞)
  `,
  uniqueValueInfos: [
    {
      value: "Fatality",
      label: "Resulted in fatality",
      symbol: {☞}
    },
    {
      value: "Injury",
      label: "Resulted in injury",
      symbol: {☞}
    },
    {
      value: "None",
      label: "No injuries or fatalities",
      symbol: {☞}
    }
  ]
}

```

Draw features  
in same order  
as the legend.



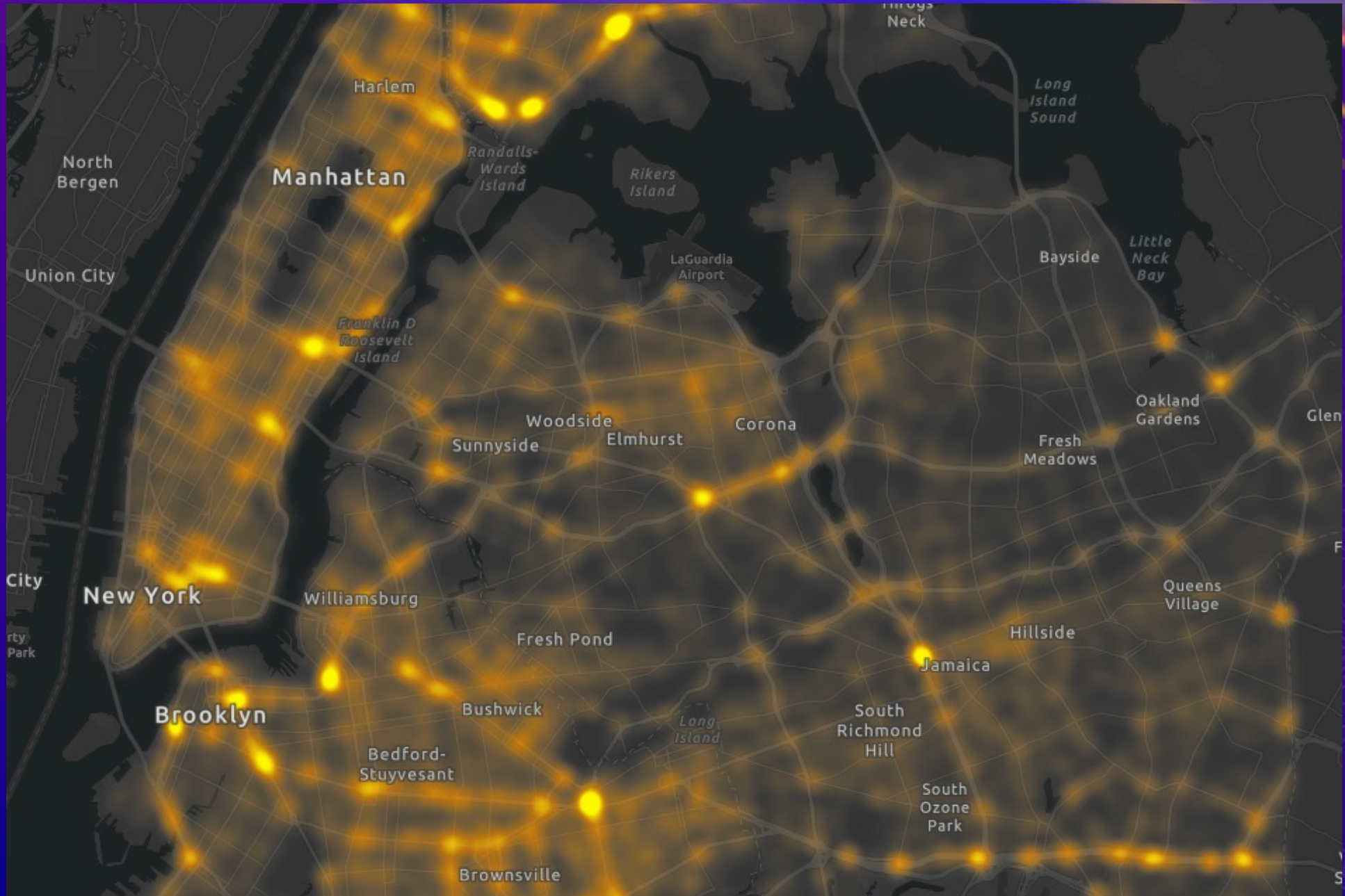






# Heat map

- Density as a continuous surface.
- Can be dynamic or static.
- Weighted heat maps.
- Supports popups and labels.

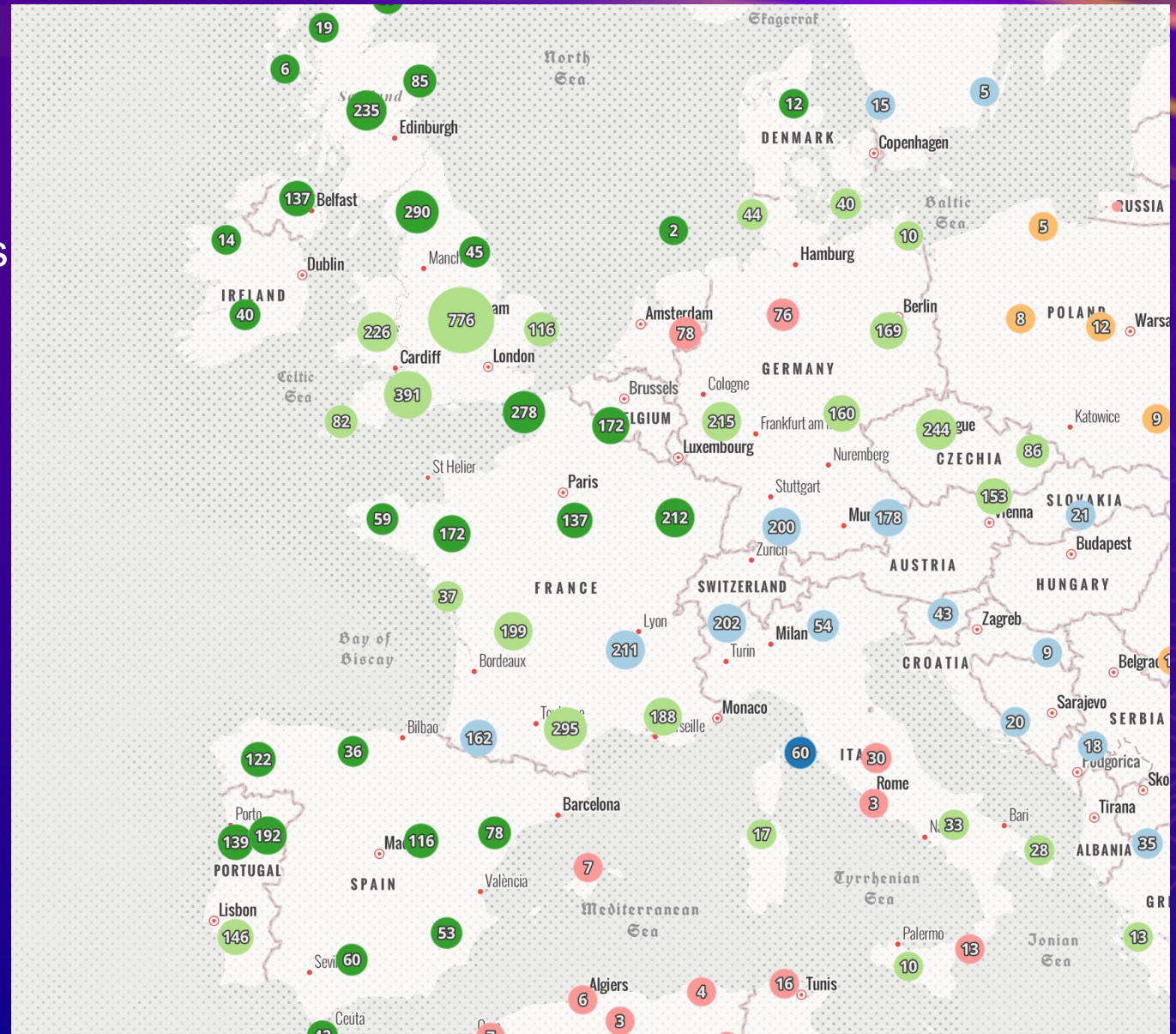




# Clustering

- Client-side aggregation dynamically updates at all scales
- Aggregates in **screen** space
- Summarizes the features within the cluster using the layer's renderer
- Aggregate fields
- Feature access with Arcade in the popup

```
layer.featureReduction = {  
  type: "cluster"  
};
```

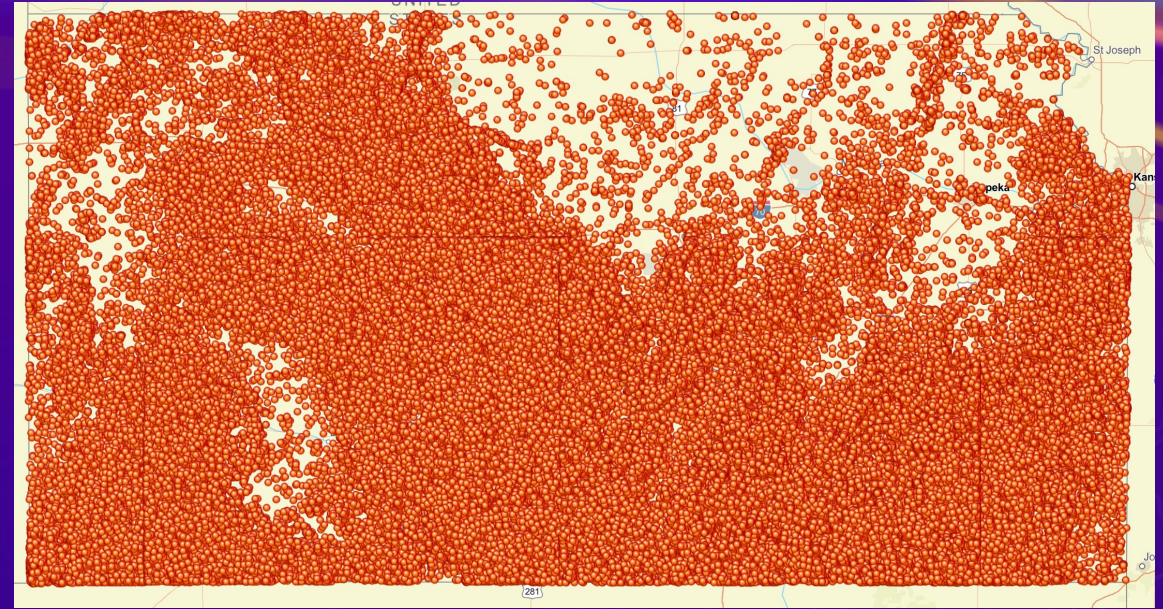




# Binning

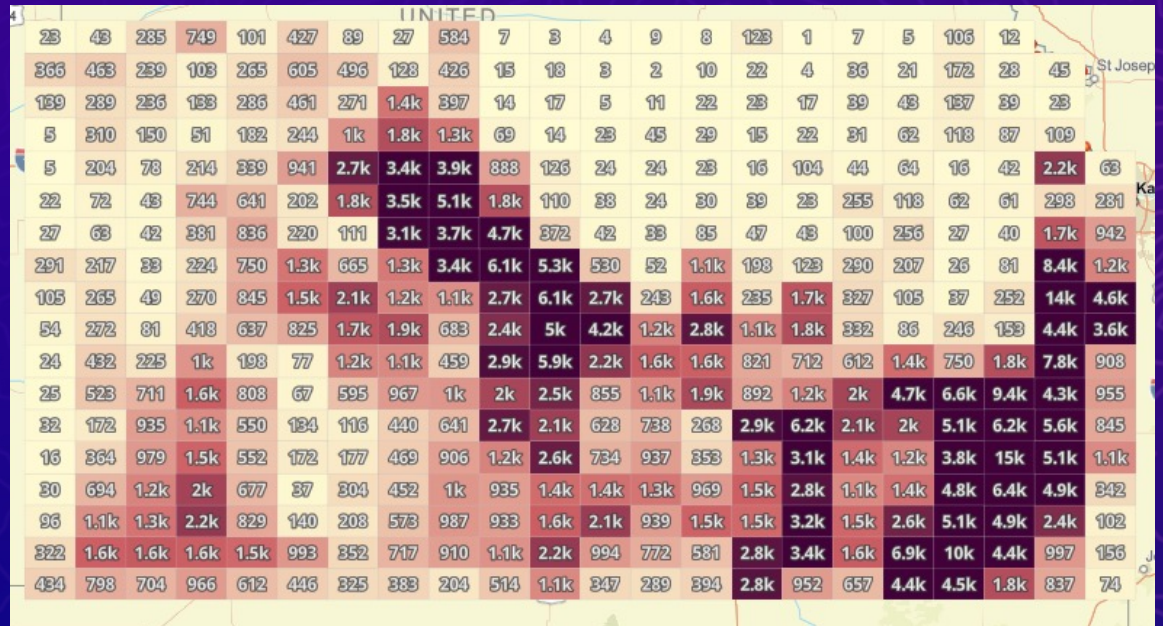
- Client-side aggregation at fixed bin level
- Aggregates in geographic space
- Can style with any renderer
- Aggregate fields
- Feature access with Arcade in the popup

Before



```
const layer = new FeatureLayer({
  featureReduction: {
    type: "binning",
    fixedBinLevel: 6,
    labelingInfo: [
      // labels configured here
    ],
    popupTemplate: {
      content: "{aggregateCount} car crashes occurred in this area."
    },
    renderer: {
      type: "simple",
      // other renderer properties
    }
  }
});
```

After



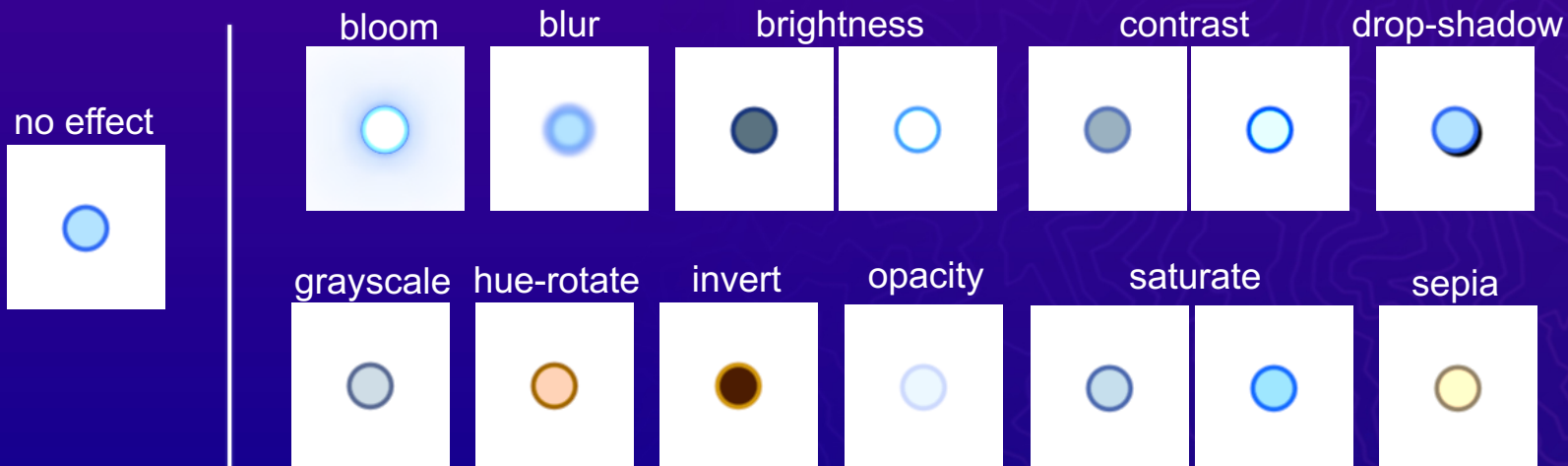


# Blending and effects

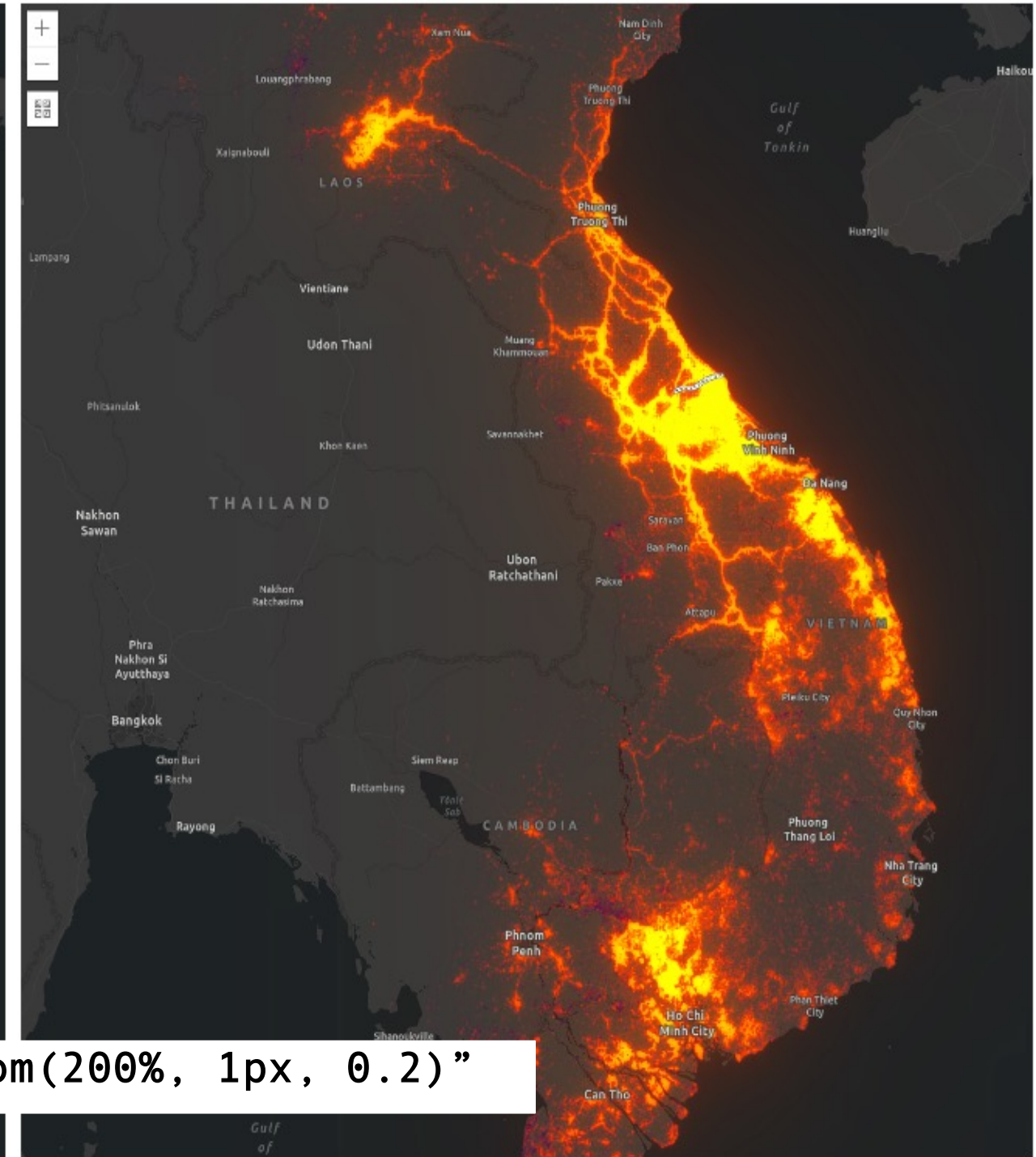
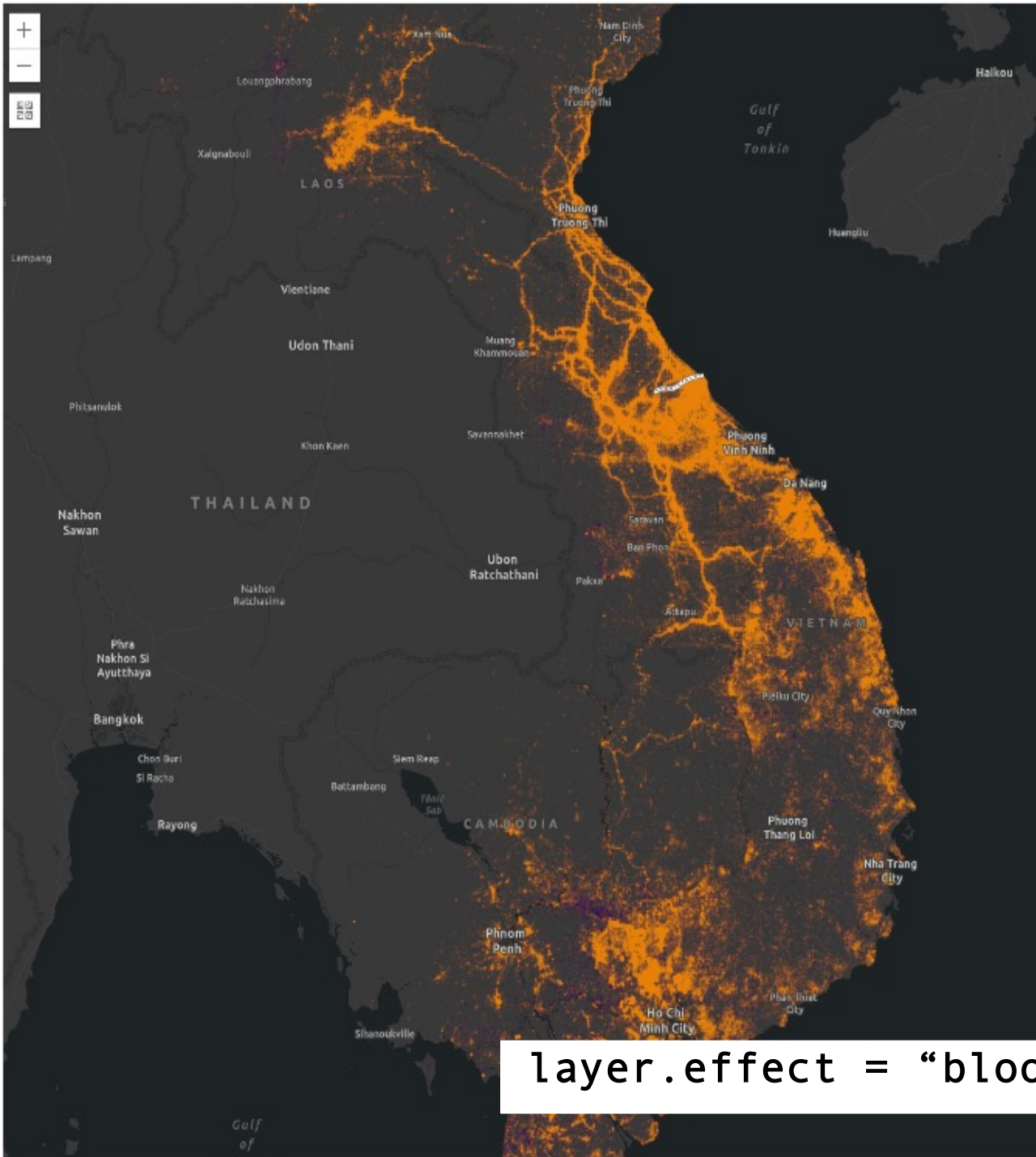
The background is a dark purple gradient. On the right side, there are several glowing, curved lines in shades of blue, purple, and orange, resembling light trails or stylized orbits. In the lower-left quadrant, a faint, light-colored map of Europe is visible, showing the outlines of the continent and its major landmasses.

# Layer & feature effects

- Apply CSS filter-like functions to all features, or based on a filter
- Can be scale dependent
- **Supported effects:**

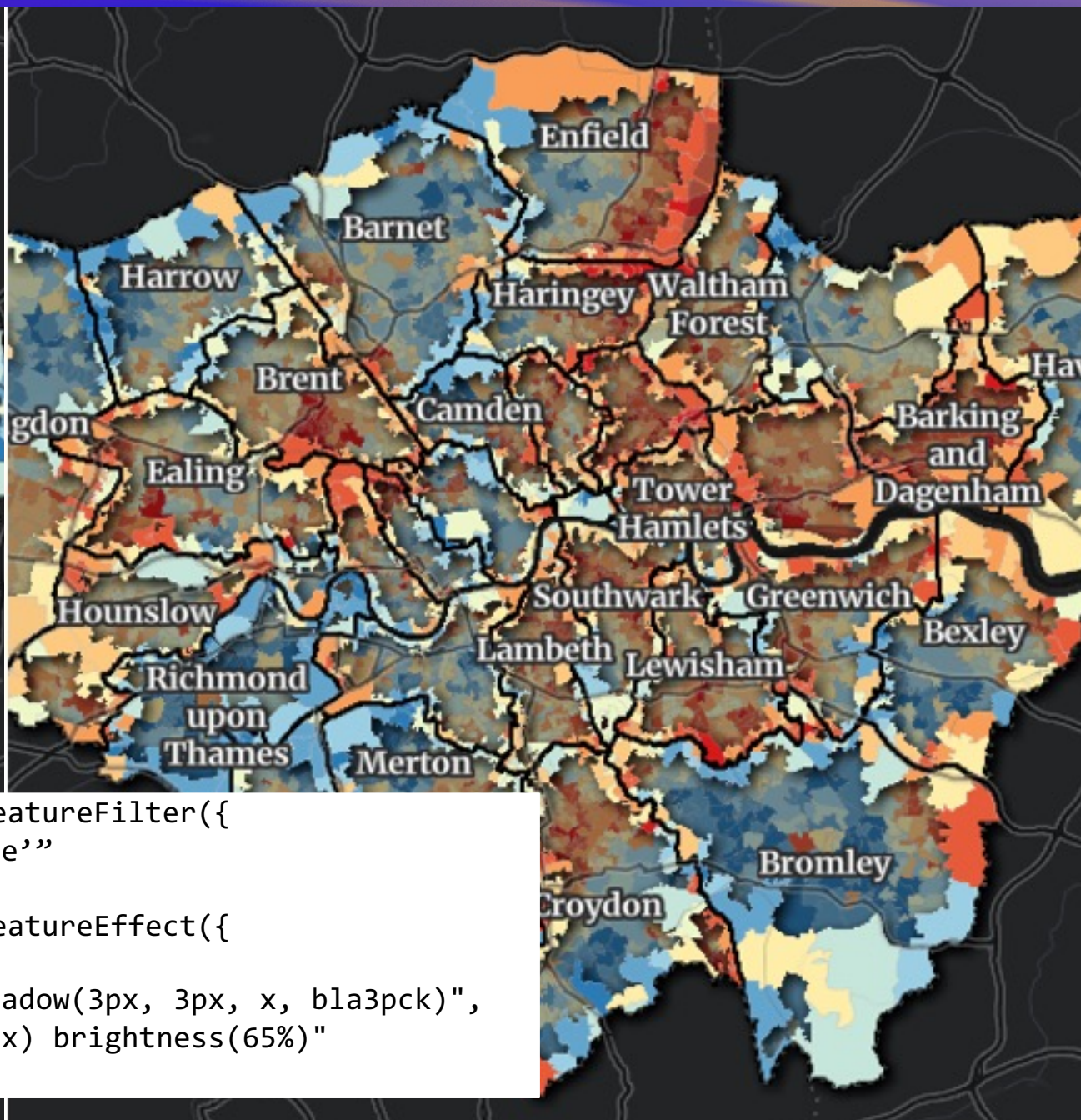
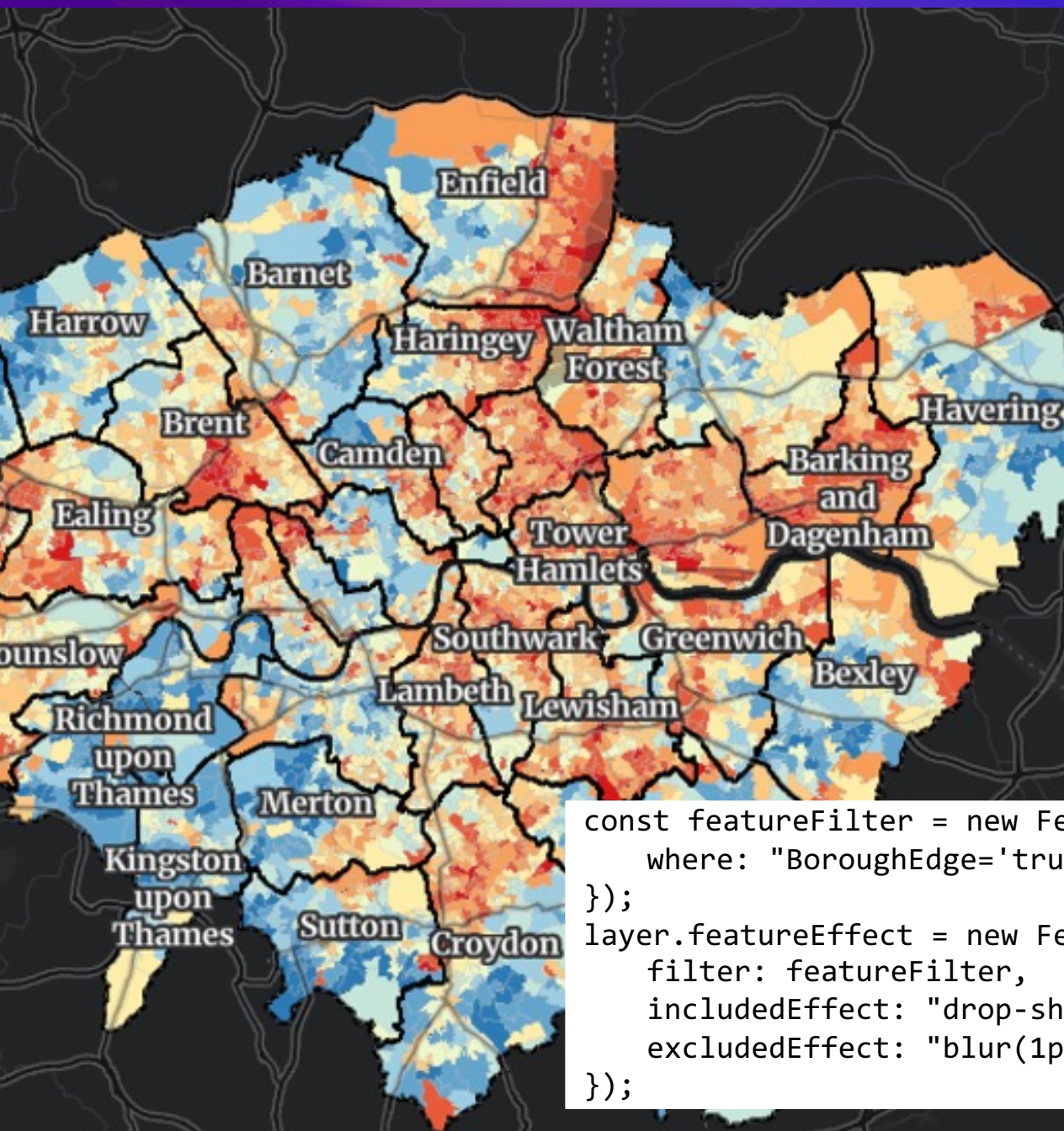






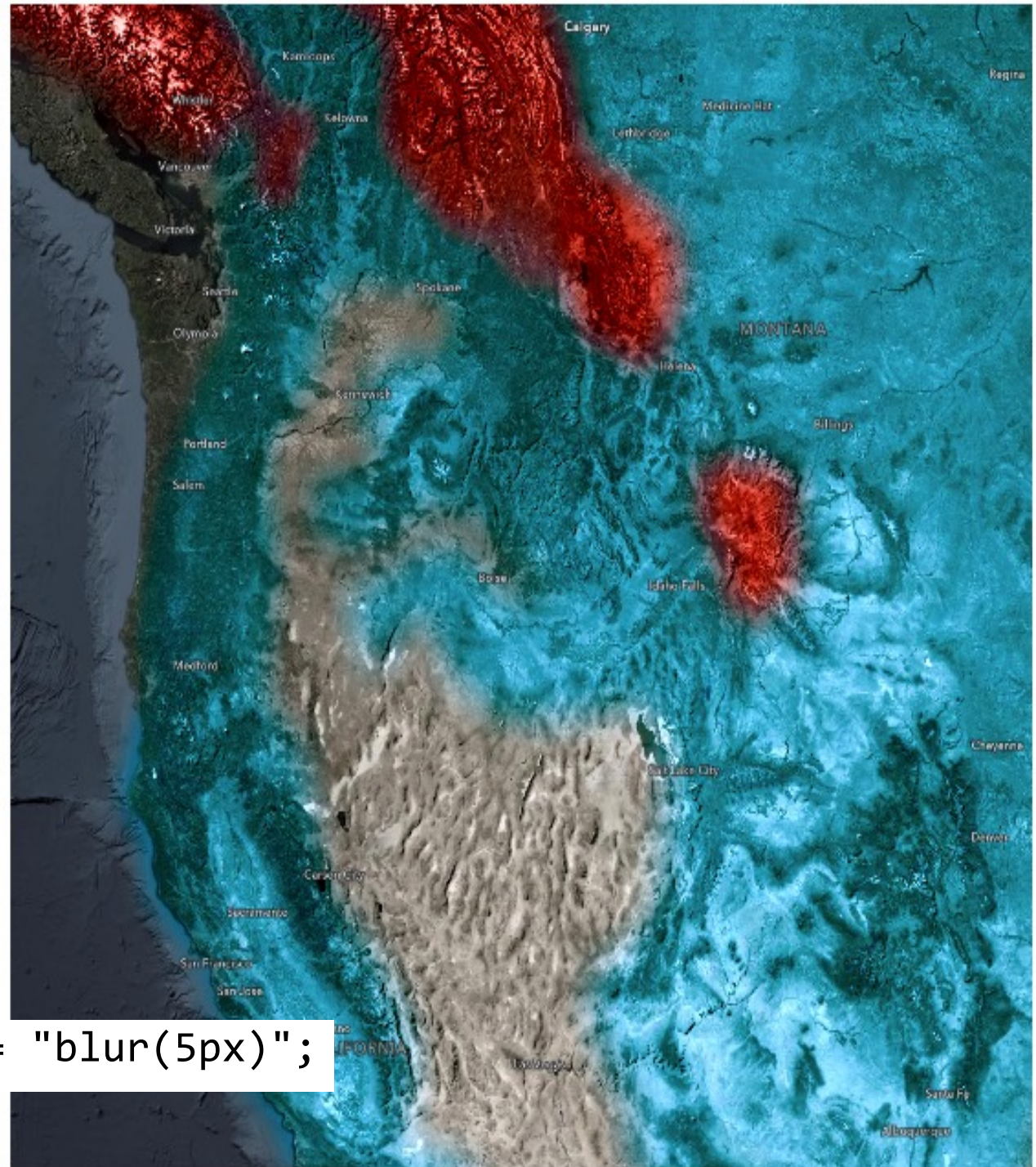
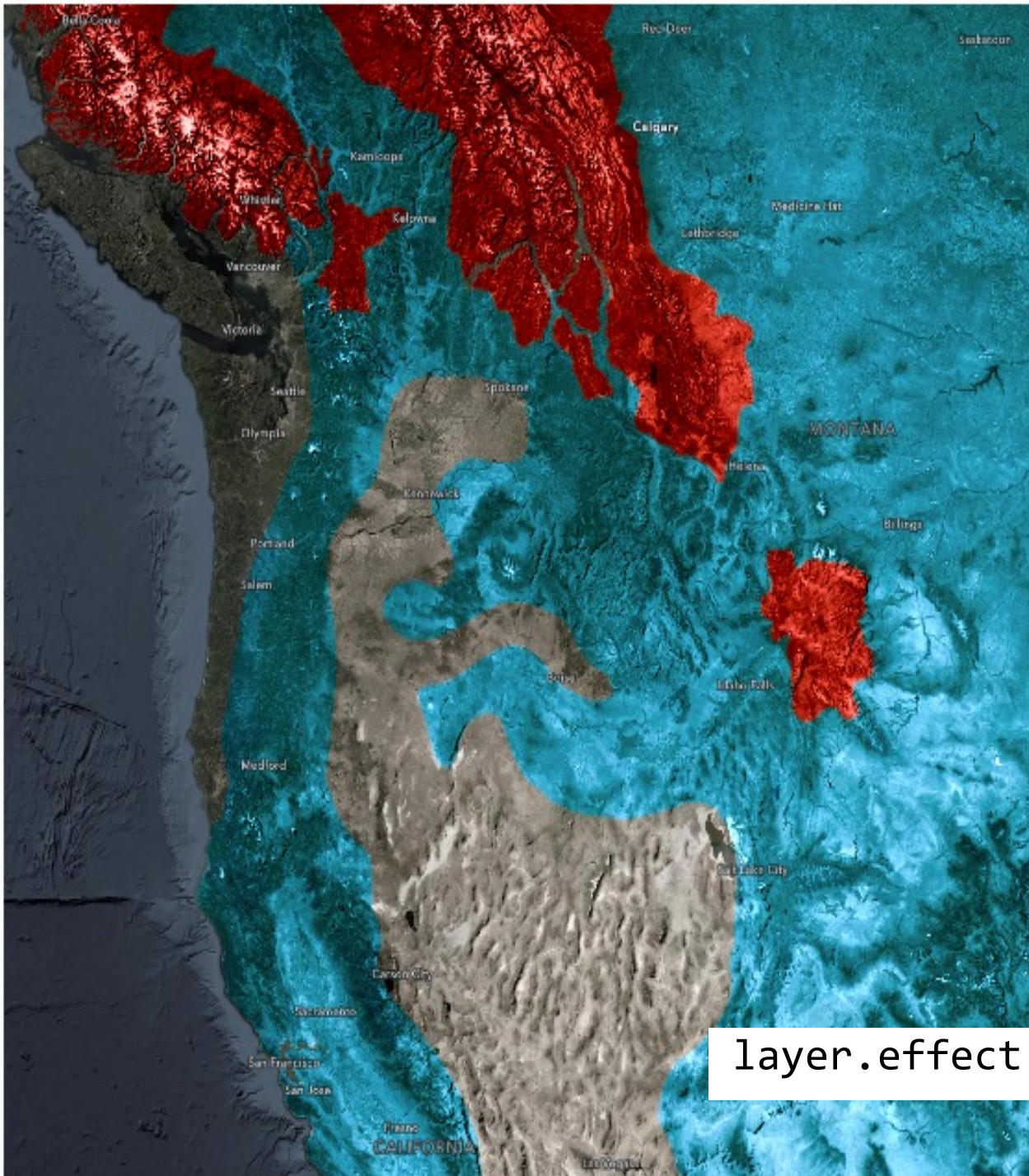
`layer.effect = "bloom(200%, 1px, 0.2)"`





```
const featureFilter = new FeatureFilter({
  where: "BoroughEdge='true'"
});
layer.featureEffect = new FeatureEffect({
  filter: featureFilter,
  includedEffect: "drop-shadow(3px, 3px, x, bla3pck)",
  excludedEffect: "blur(1px) brightness(65%)"
});
```



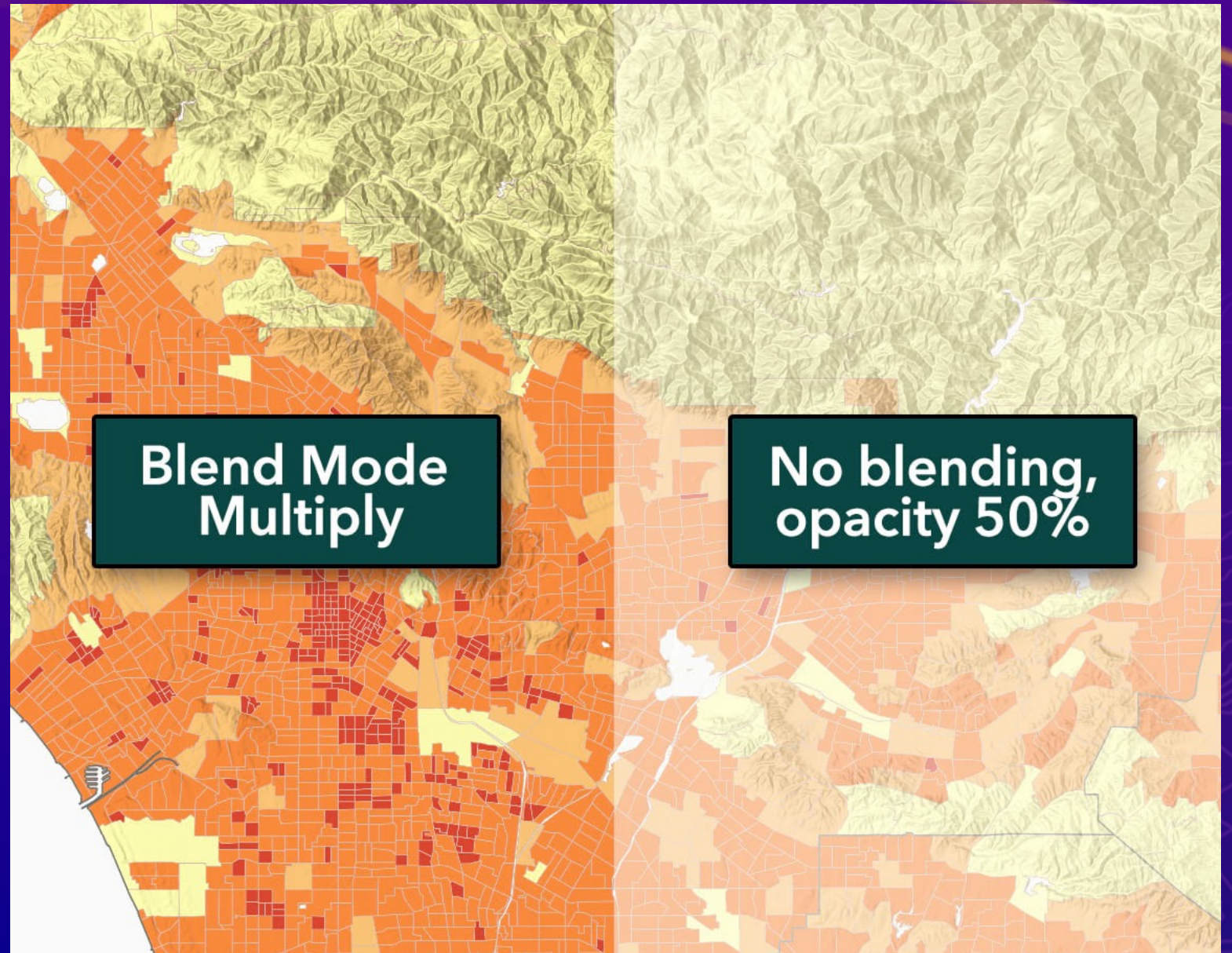


```
layer.effect = "blur(5px)";
```

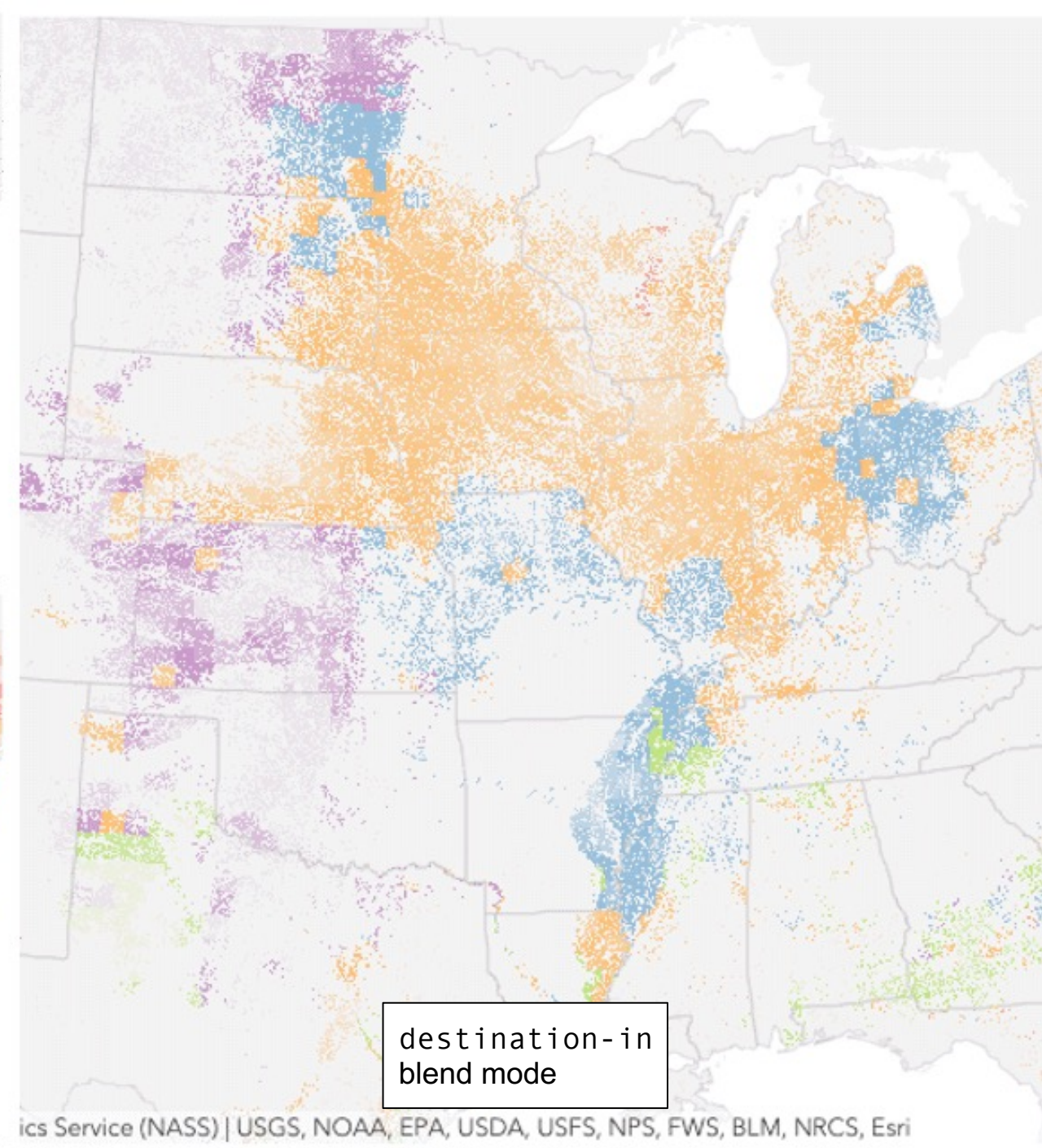
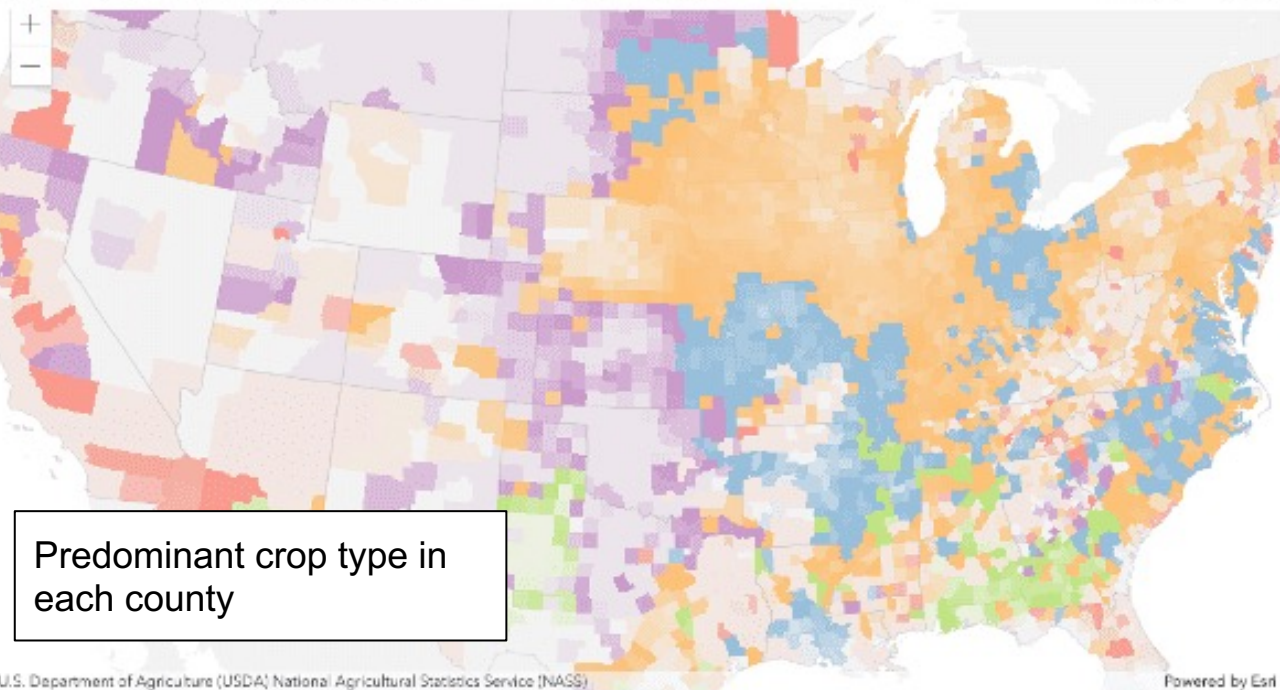
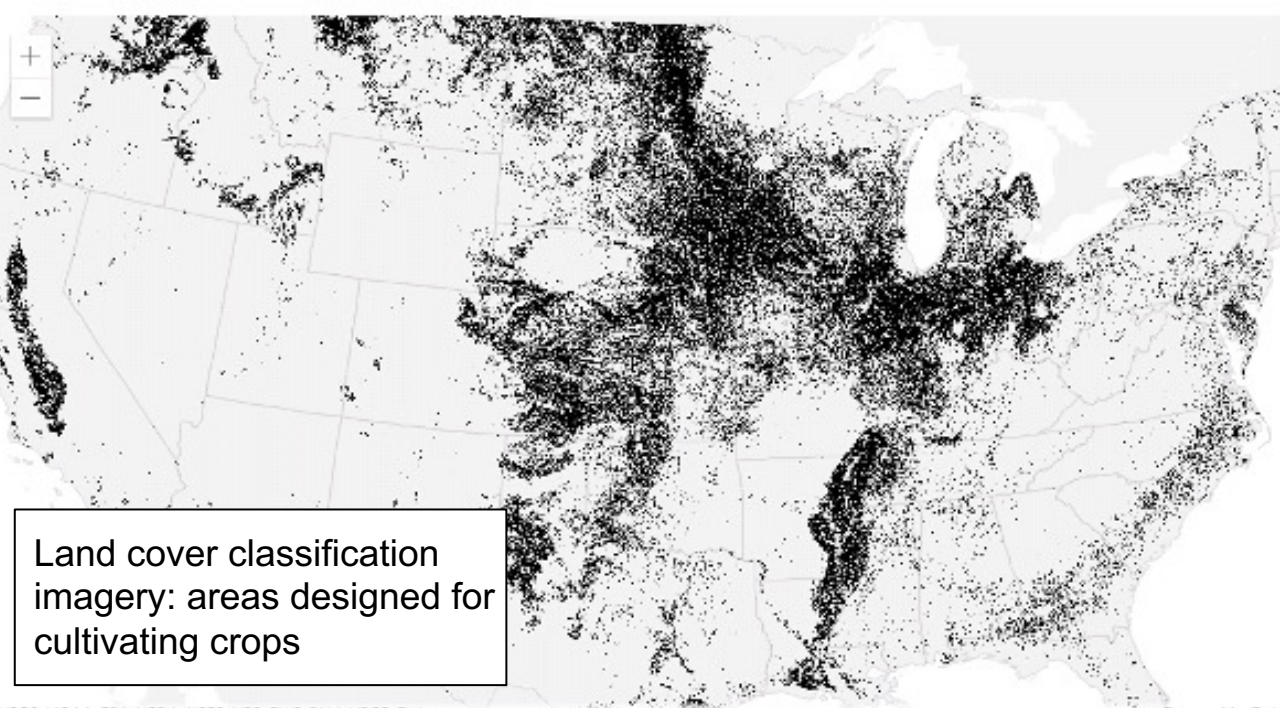


# Blend modes

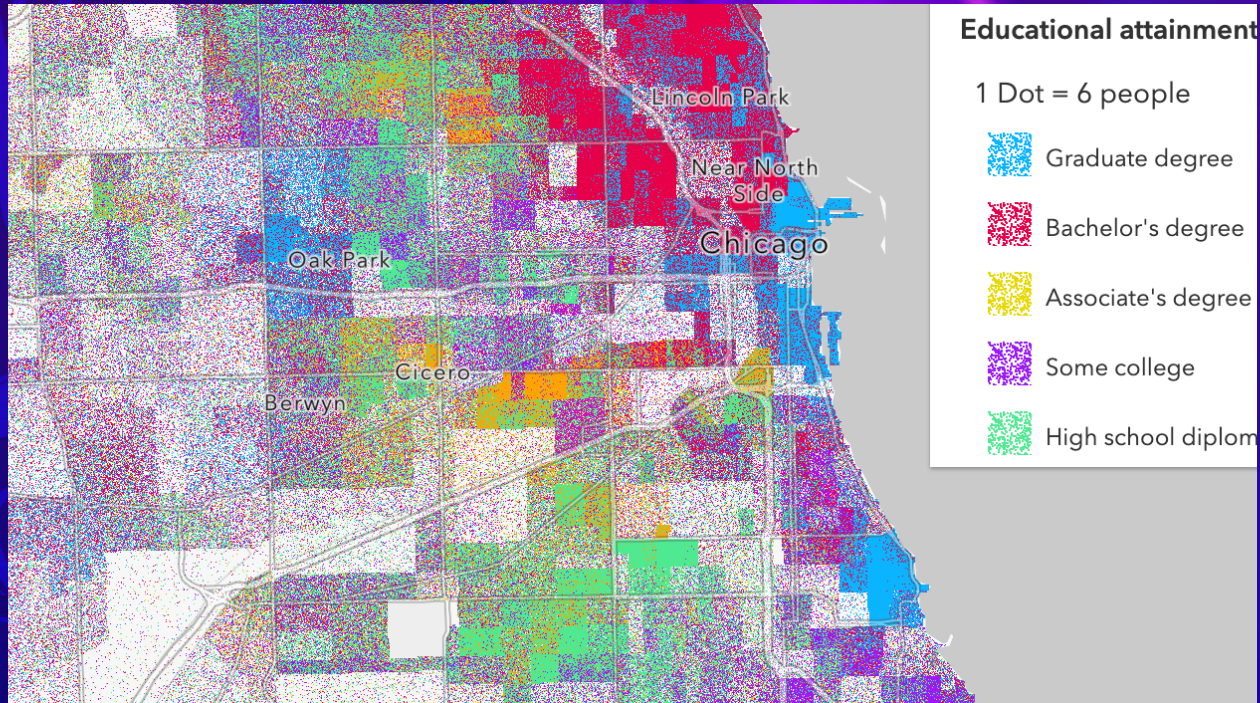
- Helps to combine/blend the visualization of two or more layers
- Blend mode themes:
  - Lighten
  - Darken
  - Contrast
  - Component
  - Compositing
  - Invert











# Demos



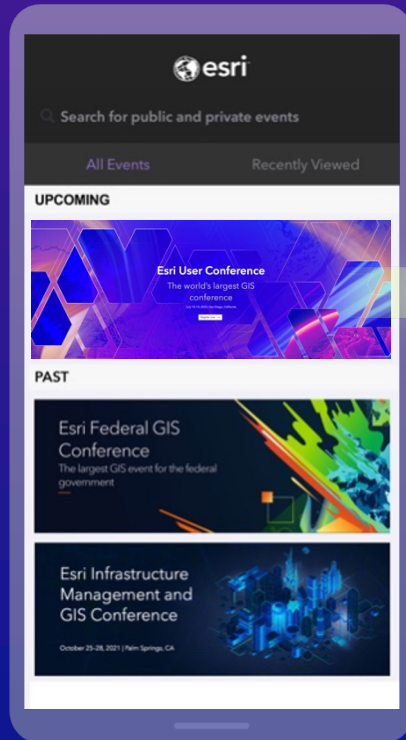
# Resources

- [Get started with visualization](#)
- [ArcGIS Blog](#)
- [Visualization Samples](#)
- [Documentation](#)
  - [Renderer](#)
  - [Symbol](#)

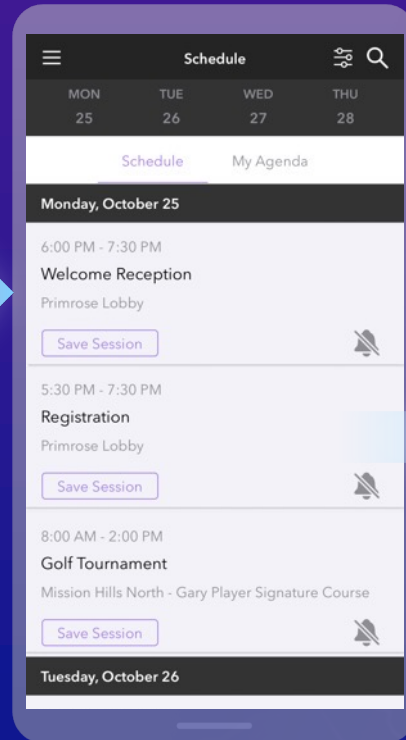


# Please Share Your Feedback in the App

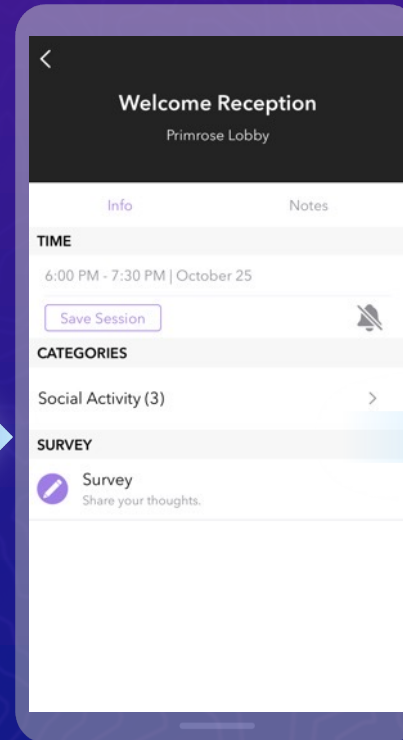
Download the Esri Events app and find your event



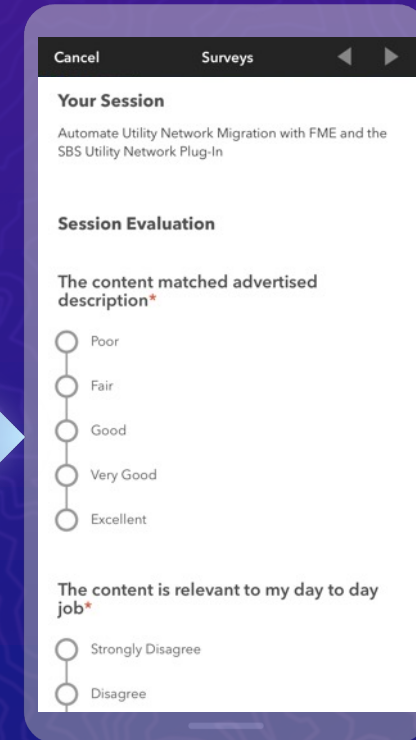
Select the session you attended



Scroll down to "Survey"



Log in to access the survey







**esri**<sup>®</sup>

**THE  
SCIENCE  
OF  
WHERE**<sup>®</sup>