nacis | North American Cartographic Information Society

esri | THE SCIENCE OF WHERE

Visualize voting trends with more than 20 years of U.S. election data

Kristian Ekenes

NACIS Annual Meeting

Louisville, KY

October 16, 2025

Winner

Select a variable

Winner

Select an election year

2024

2004 2008 2012 2016 2020 2024

Legend

This map shows the winner of the election on the state and county level.

Winning party

- Democrat
- Republican
- Other

Margin of victory

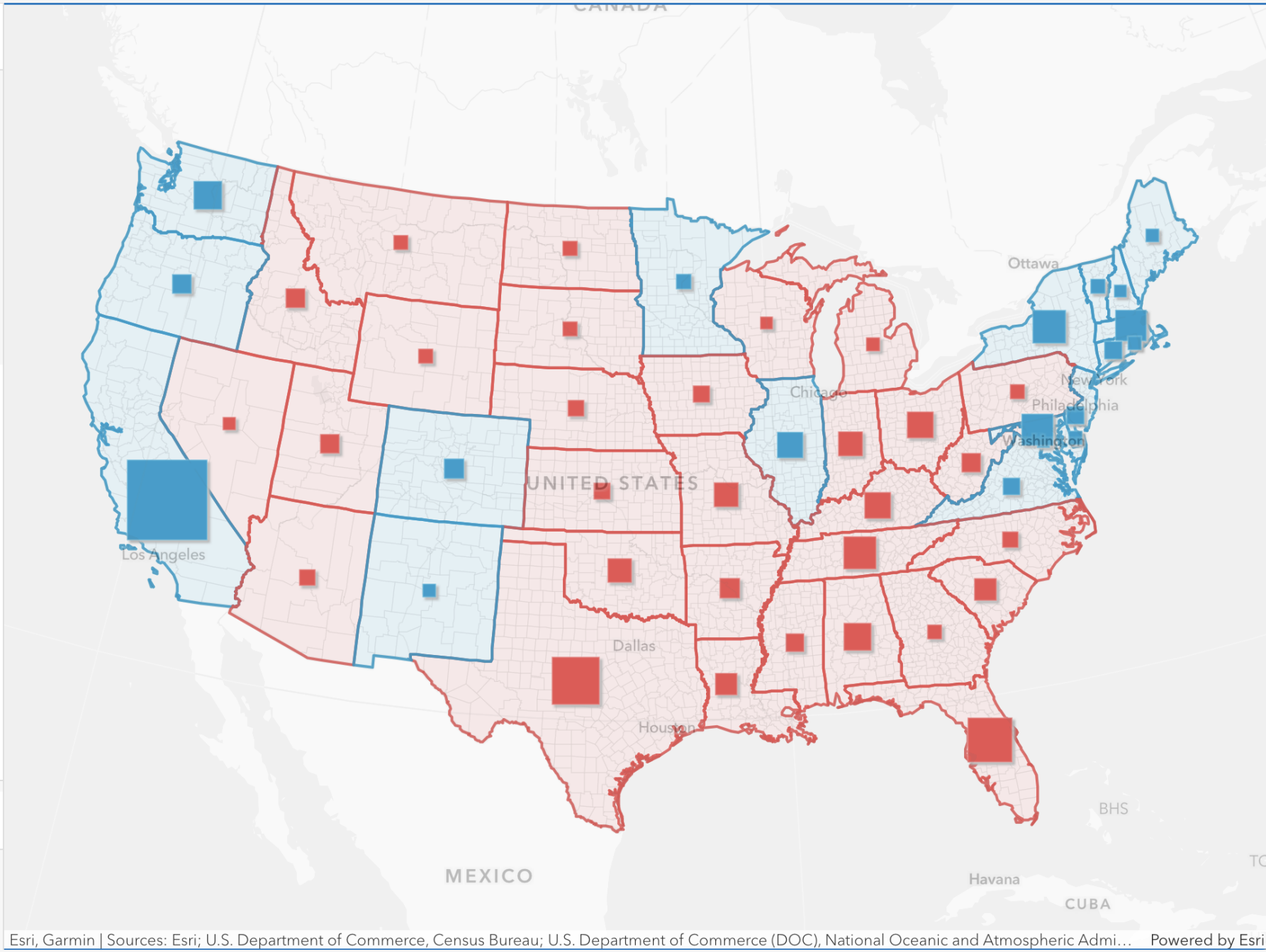
+3 million *States*

+1

+500,000 *Counties*

Symbol sizes vary by scale

Election results



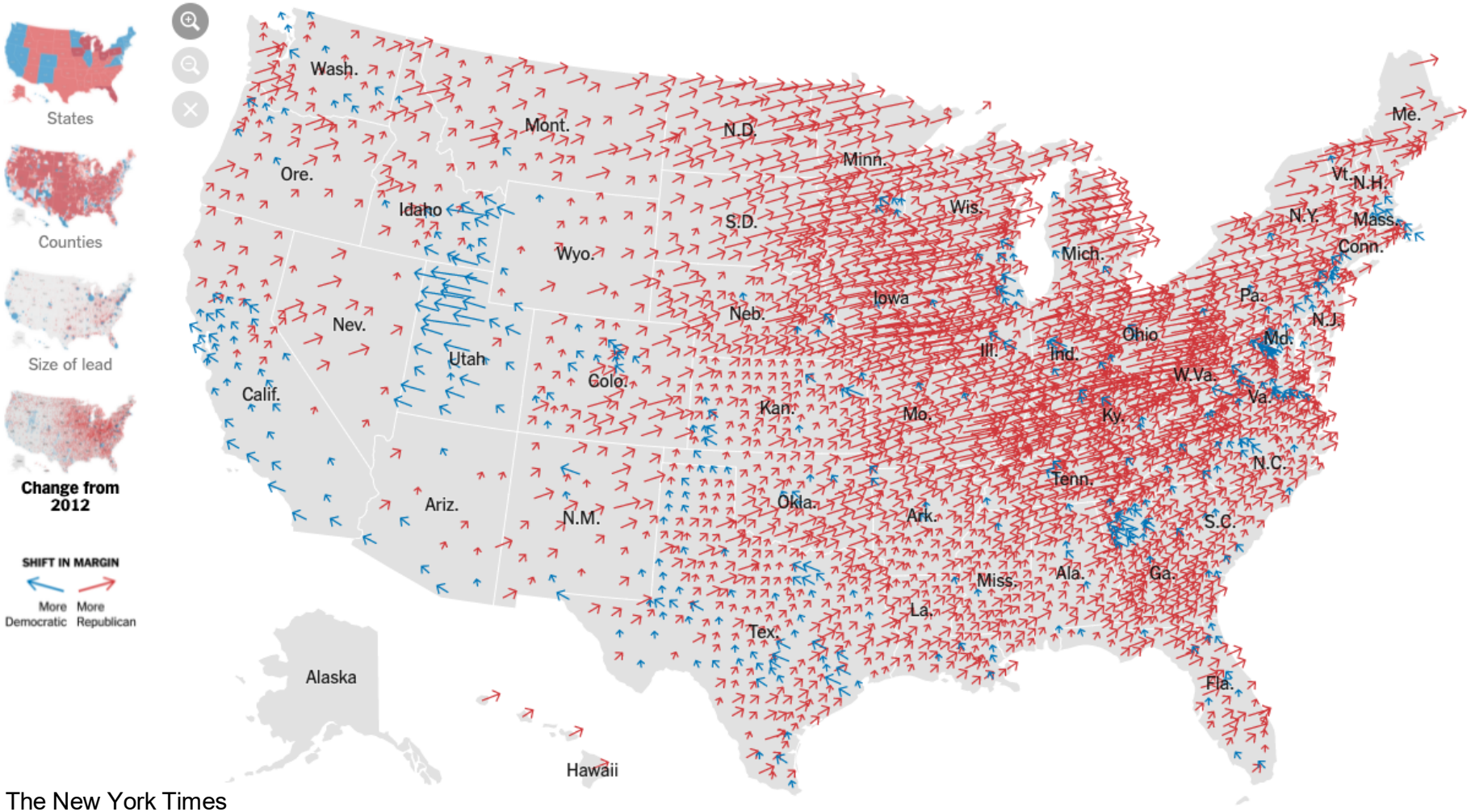
2016 Presidential Election Results

AUG. 9, 2017, 9:00 AM ET

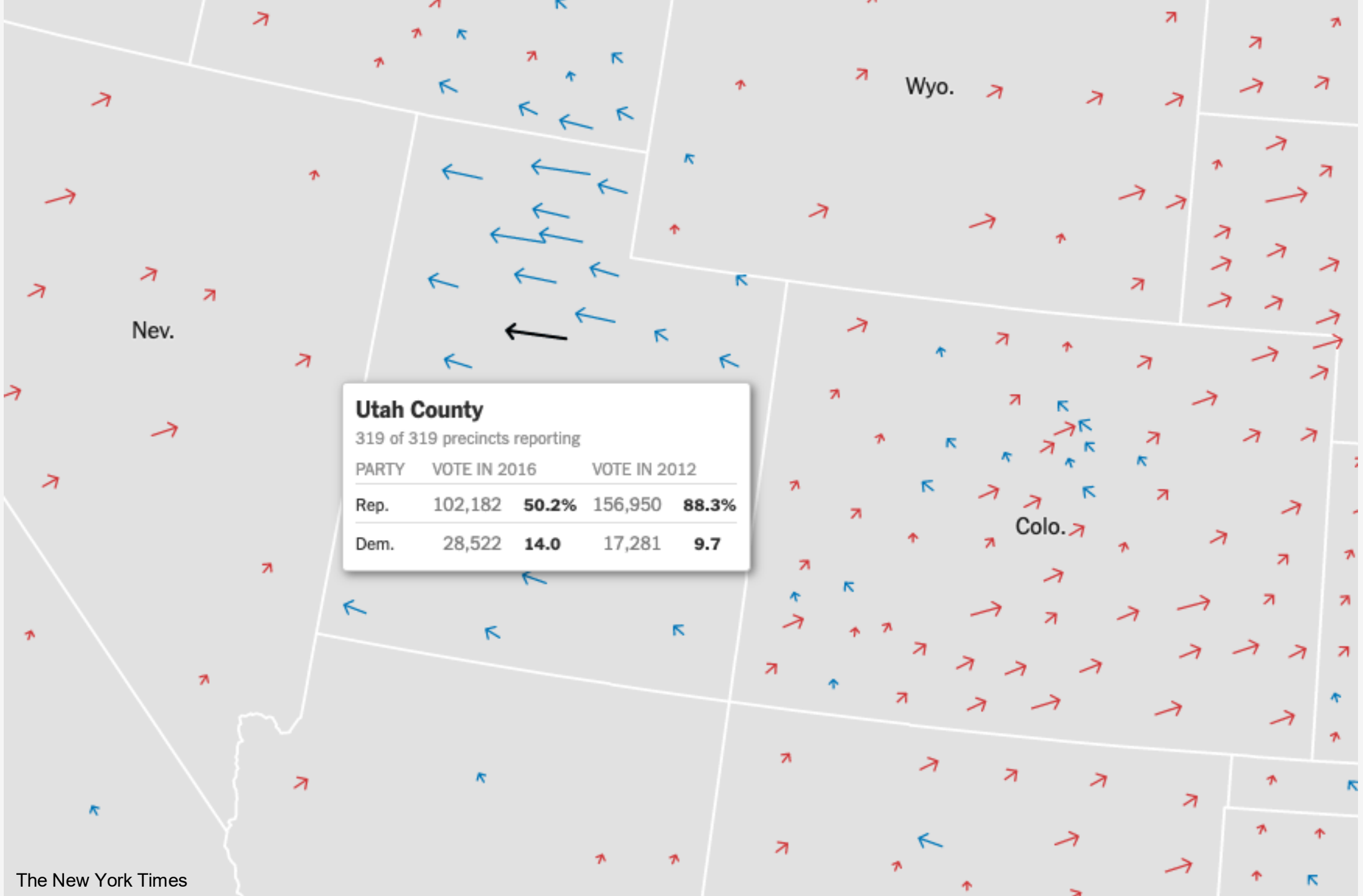
In 2016, Donald J. Trump won the Electoral College with 304 votes compared to 227 votes for Hillary Clinton. Seven electors voted for someone other than their party's candidate. Visit our 2020 election results pages for the latest updates.

232 Hillary Clinton — Donald J. Trump **306**

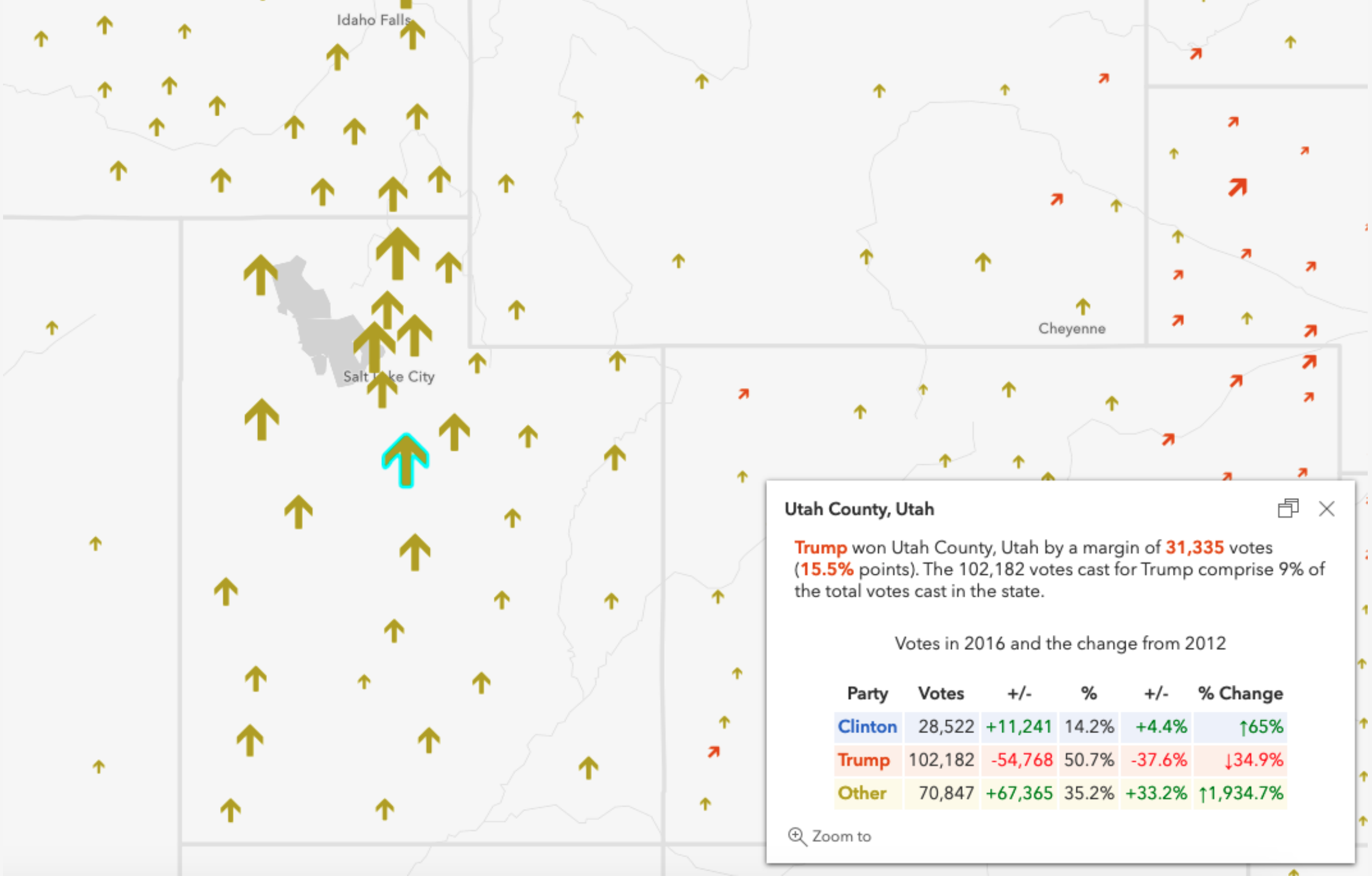
65,853,625 votes (48.0%) — 270 to win — 62,985,106 votes (45.9%)



The New York Times



The New York Times



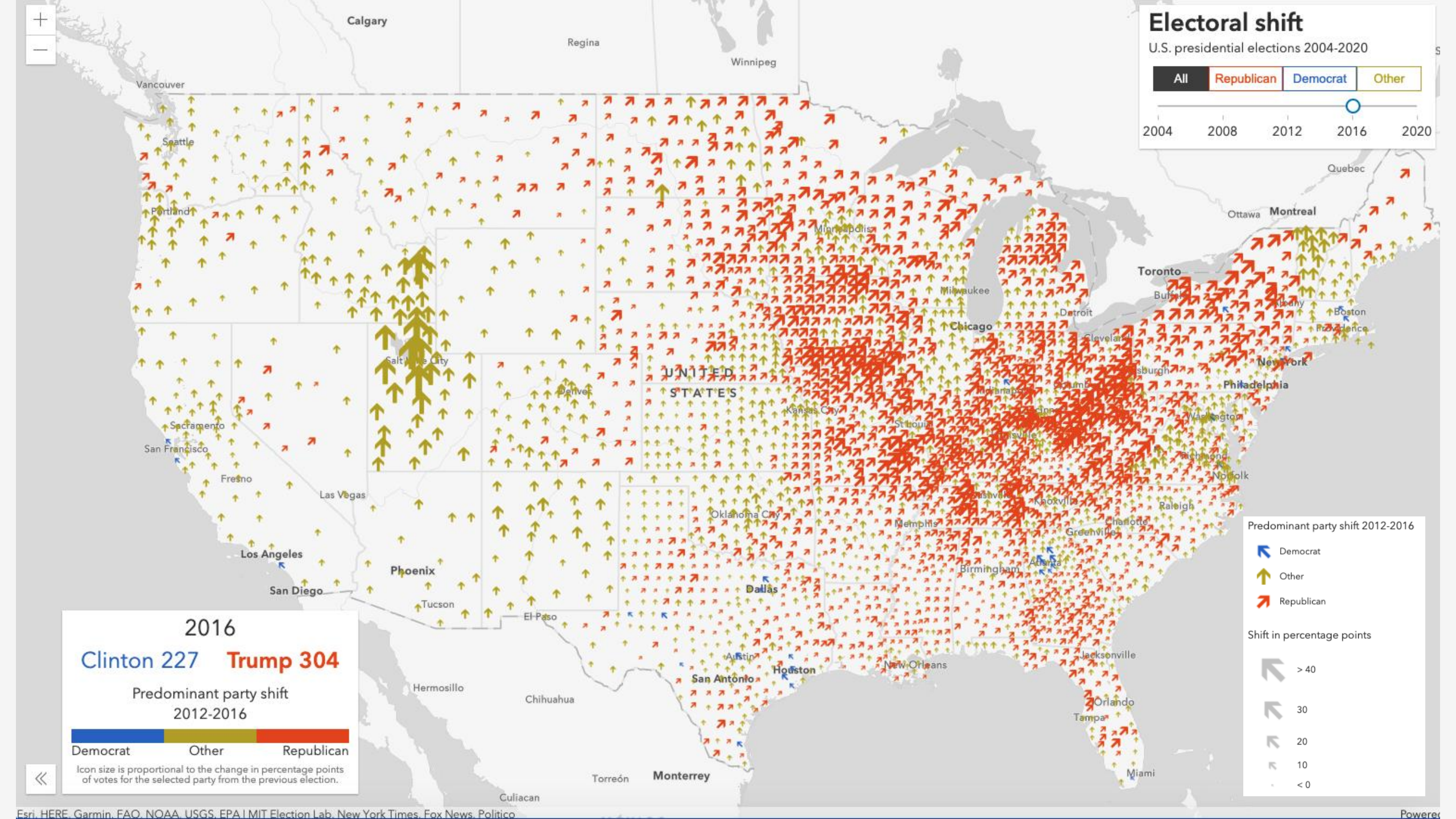
Utah County, Utah

Trump won Utah County, Utah by a margin of **31,335** votes (**15.5%** points). The 102,182 votes cast for Trump comprise 9% of the total votes cast in the state.

Votes in 2016 and the change from 2012

Party	Votes	+/-	%	+/-	% Change
Clinton	28,522	+11,241	14.2%	+4.4%	↑65%
Trump	102,182	-54,768	50.7%	-37.6%	↓34.9%
Other	70,847	+67,365	35.2%	+33.2%	↑1,934.7%

Zoom to


Electoral shift
U.S. presidential elections 2004-2020
All
Republican
Democrat
Other
2004
2008
2012
2016
2020
2016
Clinton 227
Trump 304
Predominant party shift
2012-2016
Democrat
Other
Republican
Icon size is proportional to the change in percentage points of votes for the selected party from the previous election.
Predominant party shift 2012-2016
Democrat
Other
Republican
Shift in percentage points
> 40
30
20
10
< 0
Calgary
Regina
Winnipeg
Vancouver
Seattle
Portland
Salt Lake City
Denver
Sacramento
San Francisco
Fresno
Las Vegas
Los Angeles
San Diego
Phoenix
Tucson
El Paso
Hermosillo
Chihuahua
Torreón
Monterrey
Culiacan
Oklahoma City
Dallas
Austin
San Antonio
Houston
Kansas City
Minneapolis
Chicago
Milwaukee
Detroit
Cleveland
Toronto
Buffalo
Ottawa
Montreal
Quebec
Boston
Providence
New York
Philadelphia
Pittsburgh
Indianapolis
Columbus
St. Louis
Louisville
Nashville
Memphis
Knoxville
Charlotte
Greenville
Raleigh
Richmond
Norfolk
Birmingham
Atlanta
New Orleans
Jacksonville
Orlando
Tampa
Miami
UNITED STATES
Esri, HERE, Garmin, FAO, NOAA, USGS, EPA | MIT Election Lab, New York Times, Fox News, Politico

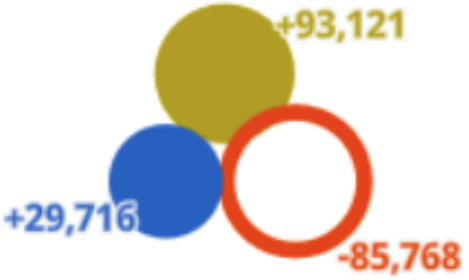
Change in U.S. presidential voting behavior from 2012-2016

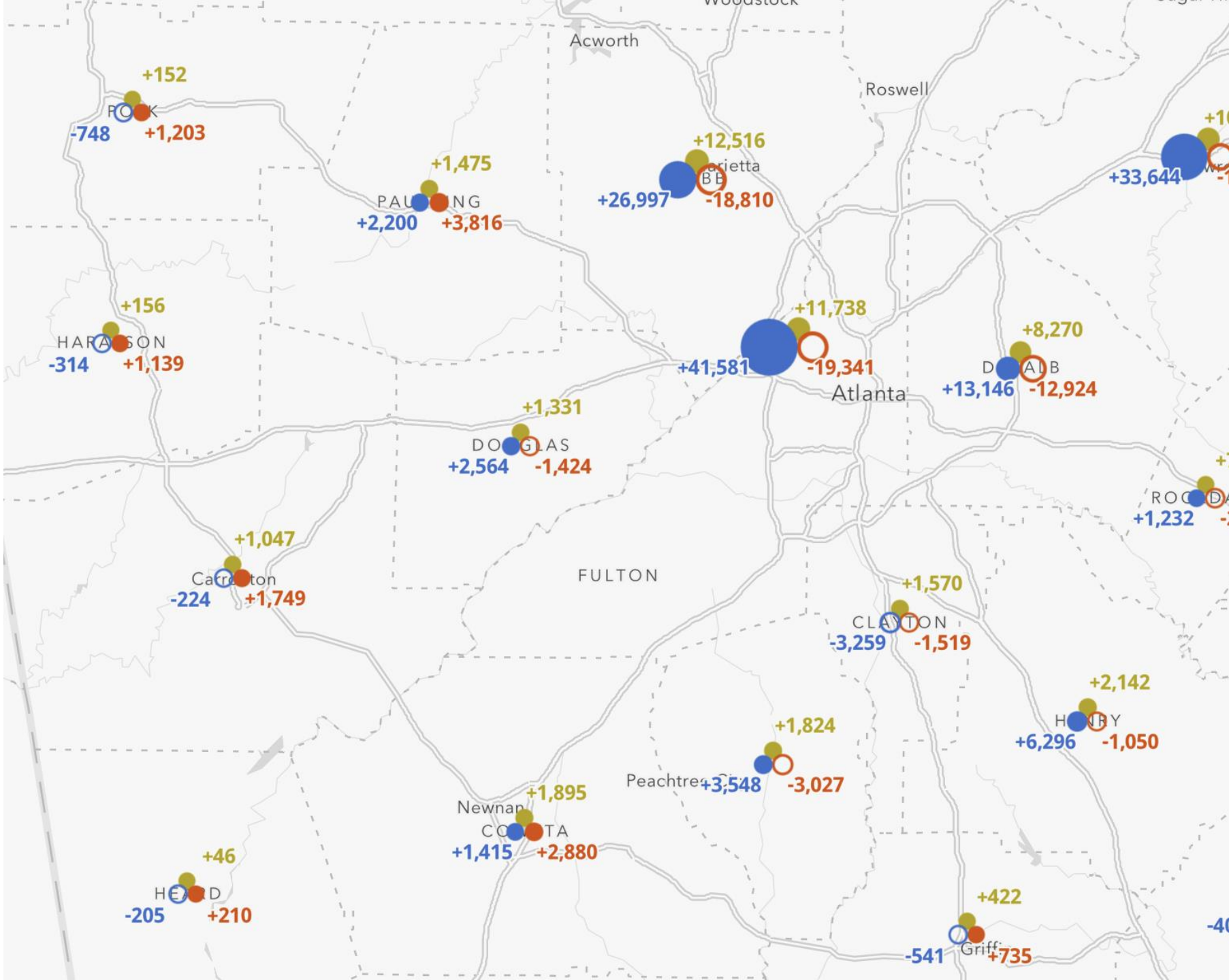
Votes gained or lost

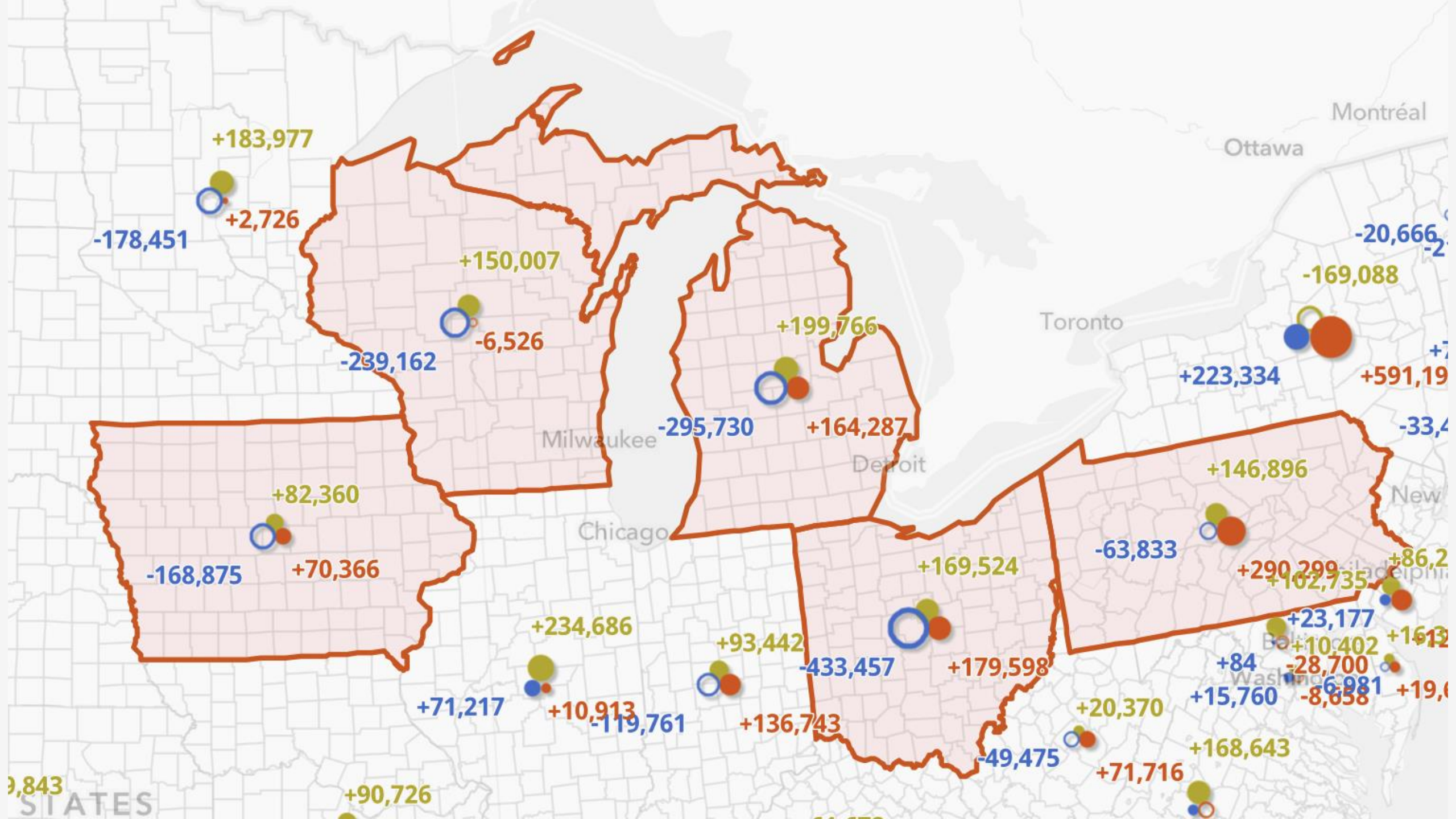
Increase in votes

Decrease in votes

This example from Utah County, Utah shows a significant decrease in Republican votes and a significant increase in third party votes.






+183,977
+2,726
-178,451
+150,007
-6,526
-239,162
+199,766
-295,730
+164,287
+82,360
-168,875
+70,366
+234,686
+71,217
+10,913
+93,442
-119,761
-433,457
+136,743
+169,524
+179,598
+146,896
-63,833
+290,299
+102,735
+23,177
+10,402
+84
-28,700
+15,760
-6,981
-8,658
+20,370
-49,475
+168,643
+71,716
+90,726
-20,666
-169,088
+223,334
+591,19
-33,4
+86,2
+19,6
Montréal
Ottawa
Toronto
Milwaukee
Detroit
Chicago
New
STATES

ArcGIS Blog | Mapping | ArcGIS Maps SDK for JavaScript | Oct 05, 2020

Visualize electoral swing using composite symbols

By Kristian Ekenes

The U.S. presidential election is only one month aw
showing the results of polls, projections, and ultima
election variables to explore is swing.

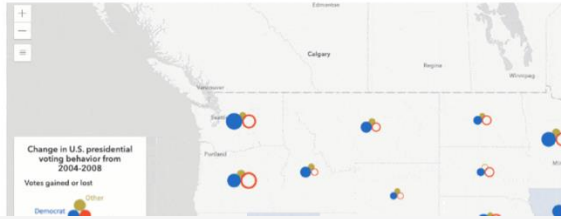
Electoral swing measures the change in votes for o
elections. The following map shows a traditional sw
presidential election using the ArcGIS API for JavaS

ArcGIS Blog | Mapping | ArcGIS Maps SDK for JavaScript | Jan 20, 2021

Electoral swing in the 2020 U.S. presidential election

By Kristian Eken

In the weeks leading up to the 2020 U.S. presidenti
visualizes electoral swing using data-driven multi-pa
JavaScript (JS API). I recently updated the app to us
added a slider, allowing you to explore all president
image below to open the app and explore the data



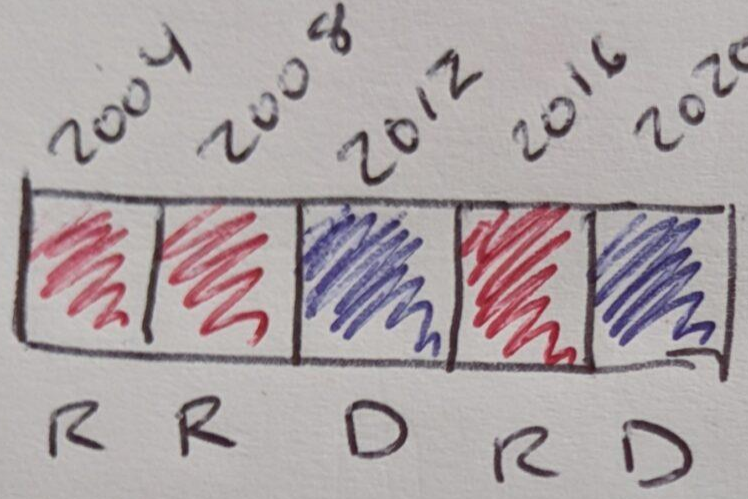
ArcGIS Blog | Mapping | ArcGIS Maps SDK for JavaScript | May 12, 2021

Six ways to visualize change over time in web maps

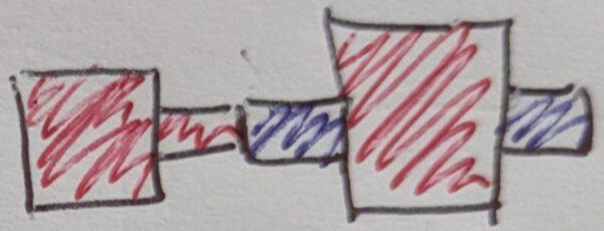
By Kristian Ekenes

Data visualization is an attempt to answer questions about raw data. These questions typically start with words like *what*, *how much*, *when*, and of course, *where*. In addition to these, people frequently ask questions about how a data variable changes over time. Some examples include:

- *How did population change from 2010 to 2020?*
- *Is the climate warmer on average over the last 30 years or cooler?*
- *How much did arctic ice increase and decrease over the last 10 years?*
- *Where has voter turnout increased since the last election? Where has it decreased?*



MAYBE VARY SIZE?



20-year trend

2016

2016 2020 2024

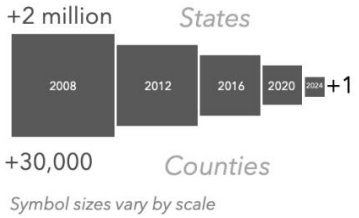
Legend

This map shows the results of each of the previous 5 U.S. presidential elections from 2000 to 2024. Each square represents the election winner for the given area in one year. The most recent election (2024) is represented as the right-most square. Each square's color represents the winner of the election; its size is proportional to the margin of victory for the winner. Smaller squares indicate a closer election. Larger squares indicate a larger margin of victory.

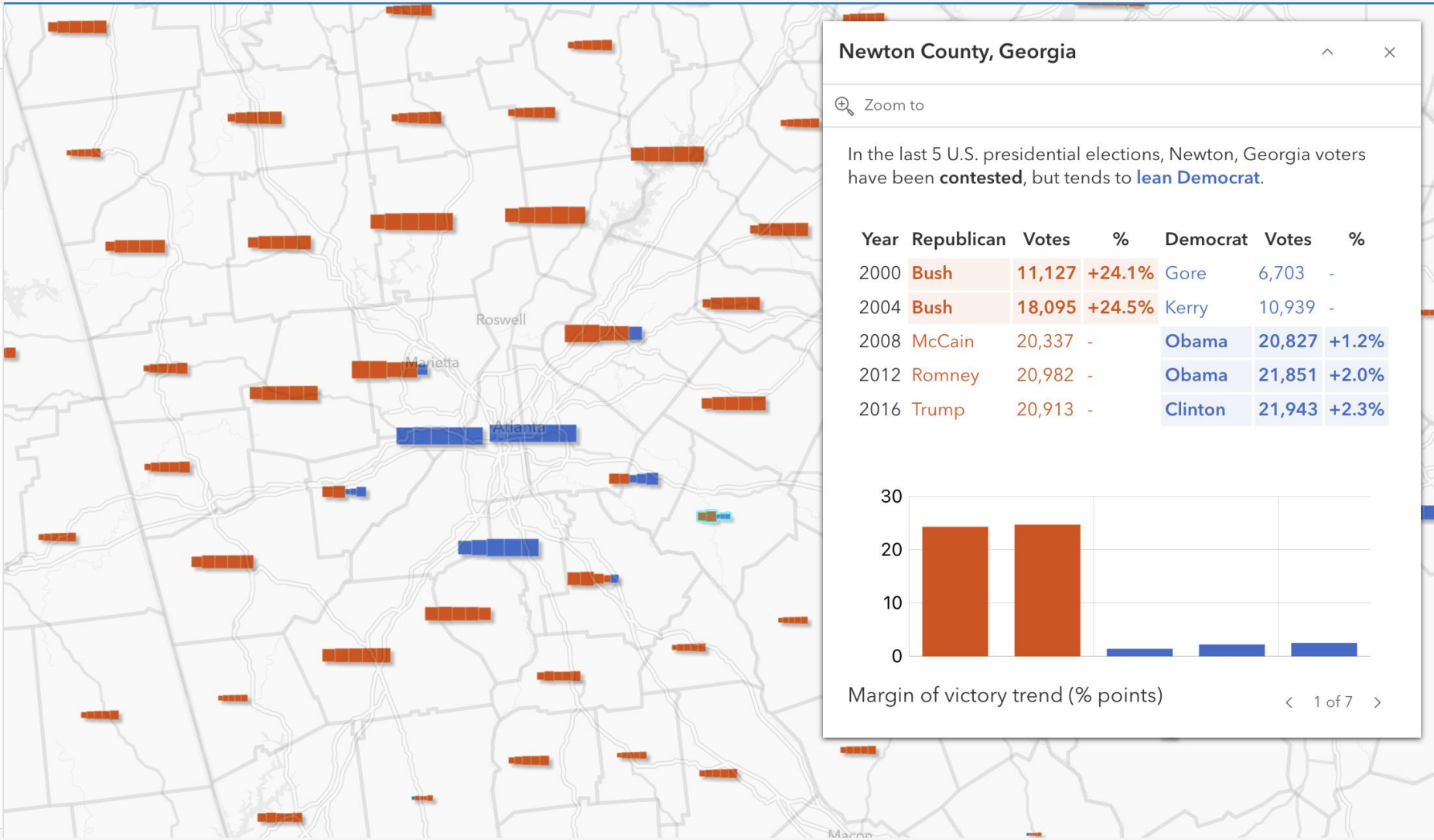
Winning party

- Democrat
- Republican
- Other

Margin of victory



Symbol sizes vary by scale



Newton County, Georgia

Zoom to

In the last 5 U.S. presidential elections, Newton, Georgia voters have been **contested**, but tends to **lean Democrat**.

Year	Republican	Votes	%	Democrat	Votes	%
2000	**Bush**	**11,127**	**+24.1%**	Gore	6,703	-
2004	**Bush**	**18,095**	**+24.5%**	Kerry	10,939	-
2008	McCain	20,337	-	**Obama**	**20,827**	**+1.2%**
2012	Romney	20,982	-	**Obama**	**21,851**	**+2.0%**
2016	Trump	20,913	-	**Clinton**	**21,943**	**+2.3%**

Margin of victory trend (% points)

1 of 7

Demo

How it’s made

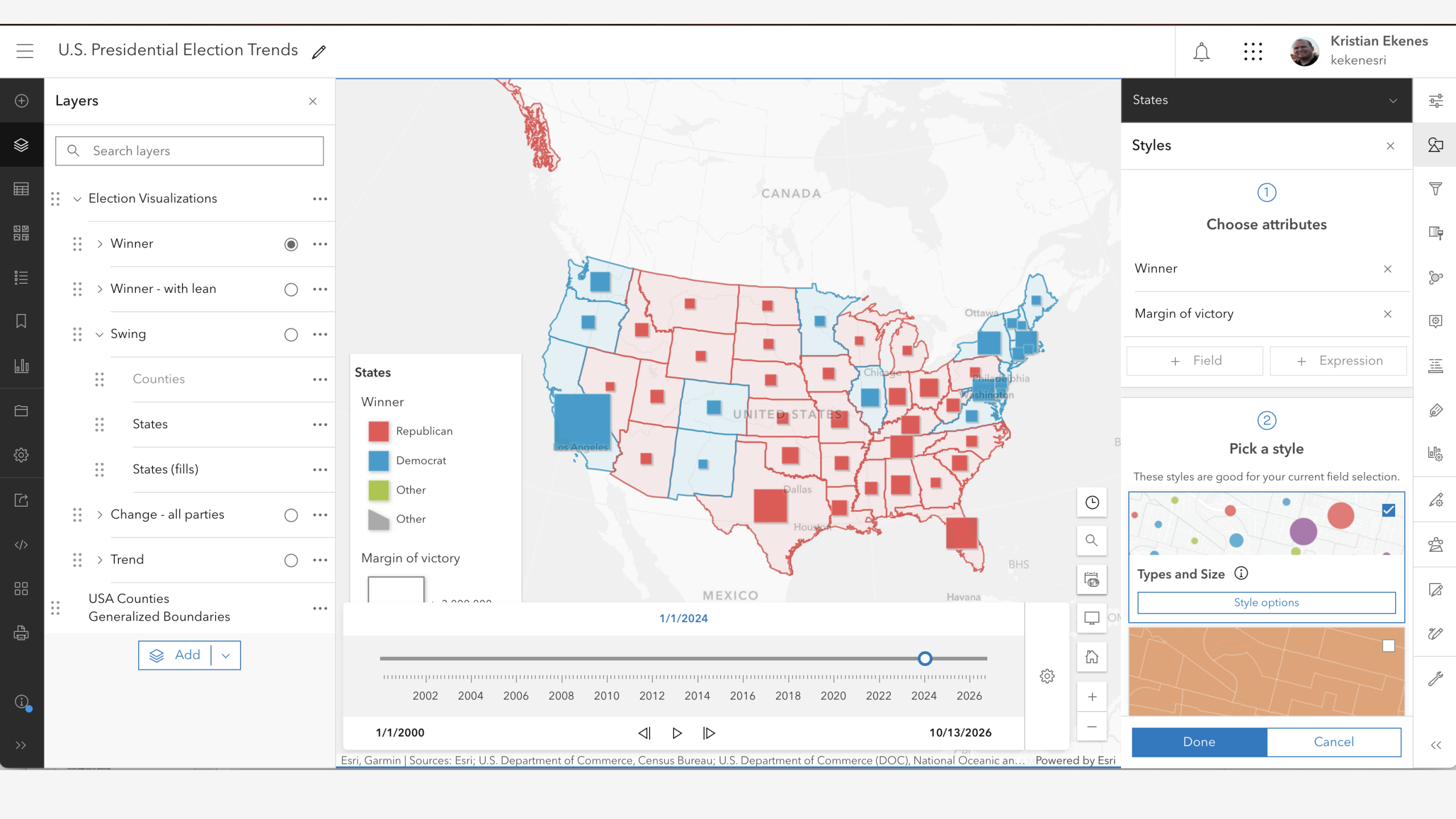
The web application

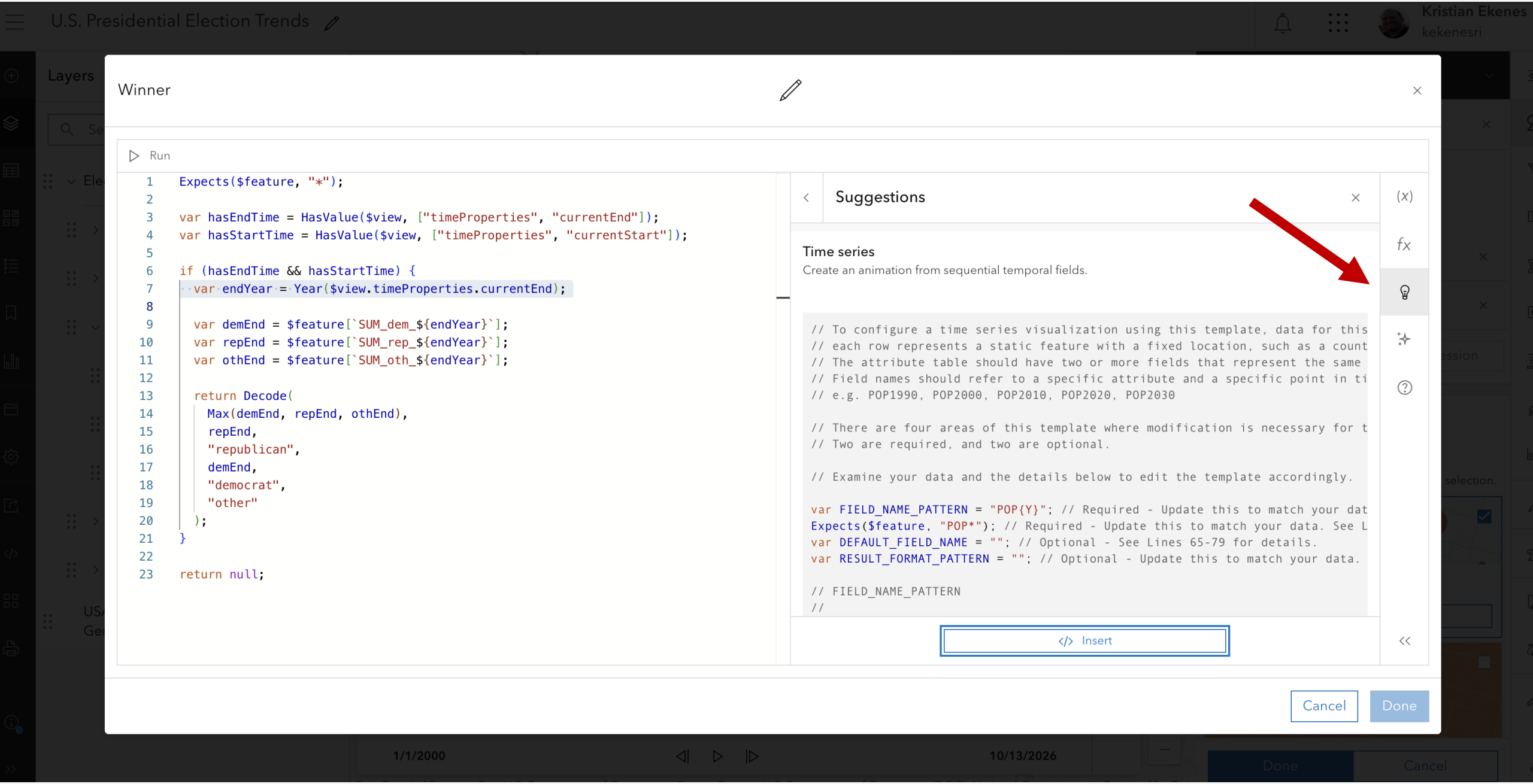
1. Create a web map in ArcGIS Online Map Viewer

- Winner + margin
  - Add **time series Arcade expression** returning winner
  - Add **time series Arcade expression** returning margin of victory
  - Configure unique values style
- Repeat process in step 1 for layers representing other variables
  - Winner + lean
  - Swing
- Add layers for composite symbols with minimal/default symbology (the UI for configuring this doesn't exist in Map Viewer)
  - **Vote change per party –** add layer, but with no renderer config
  - **Trend boxes** – add layer, but with no renderer config

Time series Arcade expressions allow you to update a layer's renderer using a time slider to point to fields representing data at different time periods that correspond to time slider values.

Opposed to traditional time series visualizations, this technique does NOT filter data or alter layer visibility.


U.S. Presidential Election Trends
Kristian Ekenes
kekenesri
Layers
Search layers
Election Visualizations
Winner
Winner - with lean
Swing
Counties
States
States (fills)
Change - all parties
Trend
USA Counties
Generalized Boundaries
Add
CANADA
UNITED STATES
MEXICO
Ottawa
Chicago
Philadelphia
Washington
Los Angeles
Dallas
Houston
Havana
BHS
States
Winner
Republican
Democrat
Other
Other
Margin of victory
1/1/2024
2002
2004
2006
2008
2010
2012
2014
2016
2018
2020
2022
2024
2026
1/1/2000
10/13/2026
Esri, Garmin | Sources: Esri; U.S. Department of Commerce, Census Bureau; U.S. Department of Commerce (DOC), National Oceanic an…
Powered by Esri
States
Styles
1
Choose attributes
Winner
Margin of victory
+ Field
+ Expression
2
Pick a style
These styles are good for your current field selection.
Types and Size
Style options
Done
Cancel


U.S. Presidential Election Trends
Kristian Ekenes
kekenesri
Layers
Winner
Run
1 Expects($feature, "*");
2
3 var hasEndTime = HasValue($view, ["timeProperties", "currentEnd"]);
4 var hasStartTime = HasValue($view, ["timeProperties", "currentStart"]);
5
6 if (hasEndTime && hasStartTime) {
7 var endYear = Year($view.timeProperties.currentEnd);
8
9 var demEnd = $feature[`SUM_dem_${endYear}`];
10 var repEnd = $feature[`SUM_rep_${endYear}`];
11 var othEnd = $feature[`SUM_oth_${endYear}`];
12
13 return Decode(
14 Max(demEnd, repEnd, othEnd),
15 repEnd,
16 "republican",
17 demEnd,
18 "democrat",
19 "other"
20);
21 }
22
23 return null;
Suggestions
Time series
Create an animation from sequential temporal fields.
// To configure a time series visualization using this template, data for this
// each row represents a static feature with a fixed location, such as a count
// The attribute table should have two or more fields that represent the same
// Field names should refer to a specific attribute and a specific point in ti
// e.g. POP1990, POP2000, POP2010, POP2020, POP2030
// There are four areas of this template where modification is necessary for t
// Two are required, and two are optional.
// Examine your data and the details below to edit the template accordingly.
var FIELD_NAME_PATTERN = "POP{Y}"; // Required - Update this to match your dat
Expects($feature, "POP*"); // Required - Update this to match your data. See L
var DEFAULT_FIELD_NAME = ""; // Optional - See Lines 65-79 for details.
var RESULT_FORMAT_PATTERN = ""; // Optional - Update this to match your data.
// FIELD_NAME_PATTERN
//
Insert
Cancel
Done
1/1/2000
10/13/2026
Done
Cancel

2. Create a web application using ArcGIS JS SDK

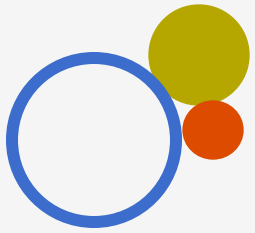
- **Load web map JSON** in Map component
- Build UI using Calcite components
  - Slider for animation
  - Custom legend for symbols not supported by OOB legend
- Use **@arcgis/core** to create custom symbology for layers not able to be configured in Map Viewer.

```
const webmap = new WebMap({
  portalItem: {
    id: webmapId,
  },
});
```

```
<CalciteShell contentBehind={true}>
  <h2 id="header-title" slot="header">
    {appTitle}
  </h2>
  <UIPanel
    onYearInput={(year) => {…
    }}
  />
  <ArcgisMap
    id="map"
    class="map-only"
    map={webmap}
    ref={mapRef}
    onArcgisViewReadyChange={initialize}
  ></ArcgisMap>
</CalciteShell>
```

3. Create custom styles not available in ArcGIS Online

Change in votes for each party



Trend

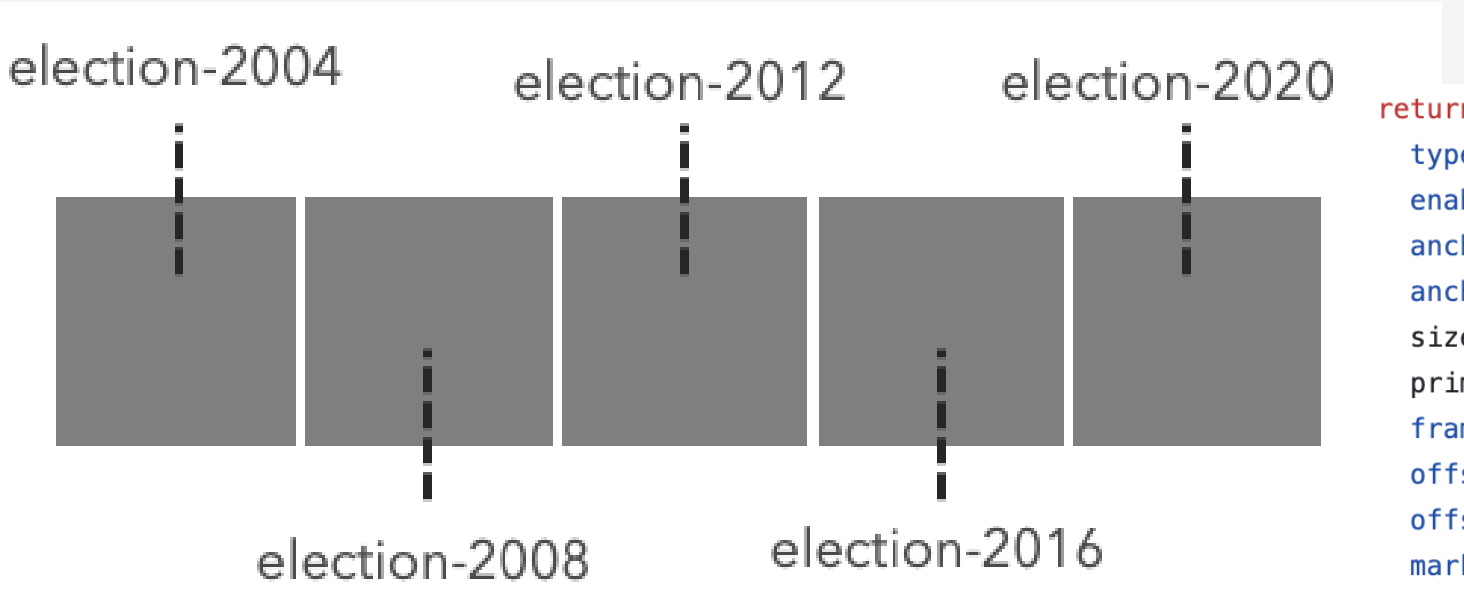
How it’s made

The trend box symbols

1. Create a renderer

```
const customRenderer = new SimpleRenderer({
  symbol: new CIMSymbol(), // define symbol here
  label: "Counties" // this will appear next to the symbol in the legend
});
```

2. Create a CIMSymbol



primitiveName	offsetX
election-2004	-2 * size
election-2008	-1 * size
election-2012	0
election-2016	1 * size
election-2020	2 * size

```
return {
  type: "CIMVectorMarker",
  enable: true,
  anchorPoint: { x: 0, y: 0 },
  anchorPointUnits: "Relative",
  size,
  primitiveName,
  frame: { xmin: 0.0, ymin: 0.0, xmax: 10.0, ymax: 10.0 },
  offsetX: size * offsetX,
  offsetY: 0,
  markerGraphics: [
    {
      type: "CIMMarkerGraphic",
      geometry: cimSquareGeometry,
      symbol
    }
  ],
  scaleSymbolsProportionally: true,
  respectFrame: true
} as __esri.CIMVectorMarker;
```

3. Define primitive overrides

Color – to show winner


```
const colorOverride = {
  type: "CIMPrimitiveOverride",
  primitiveName: "election-2016",
  propertyName: "Color",
  valueExpressionInfo: {
    type: "CIMExpressionInfo",
    title: "Winner",
    expression: colorOverrideExpression,
    returnType: "Default",
  }
};
```

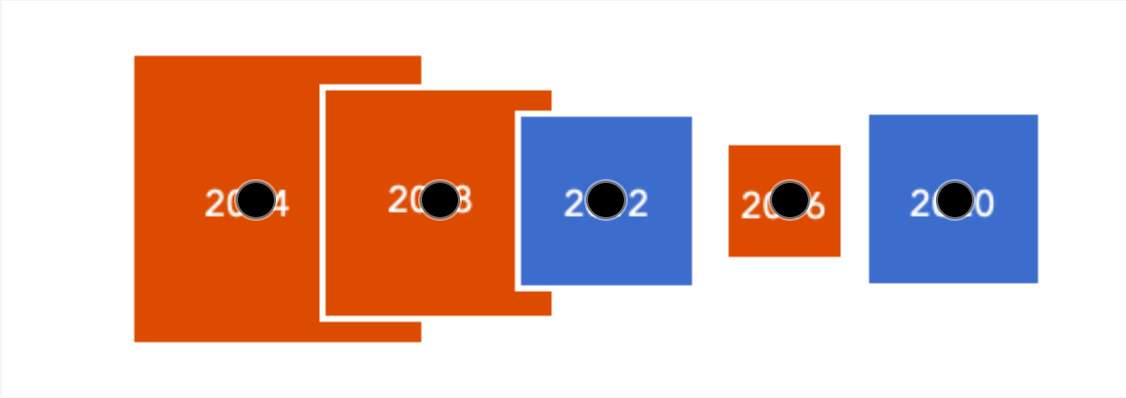
```
var rColor = "rgba(220, 75, 0, 1)";
var dColor = "rgba(60, 108, 204, 1)";
var oColor = "rgba(181, 166, 0, 1)";
var demVotes = $feature.SUM_dem_2016;
var repVotes = $feature.SUM_rep_2016;

var othVotes = $feature.SUM_oth_2016;
var allVotes = [demVotes, repVotes,
othVotes];

Decode( Max(allVotes), demVotes, dColor,
repVotes, rColor, oColor );
```

3. Define primitive overrides

Size – to show margin of victory

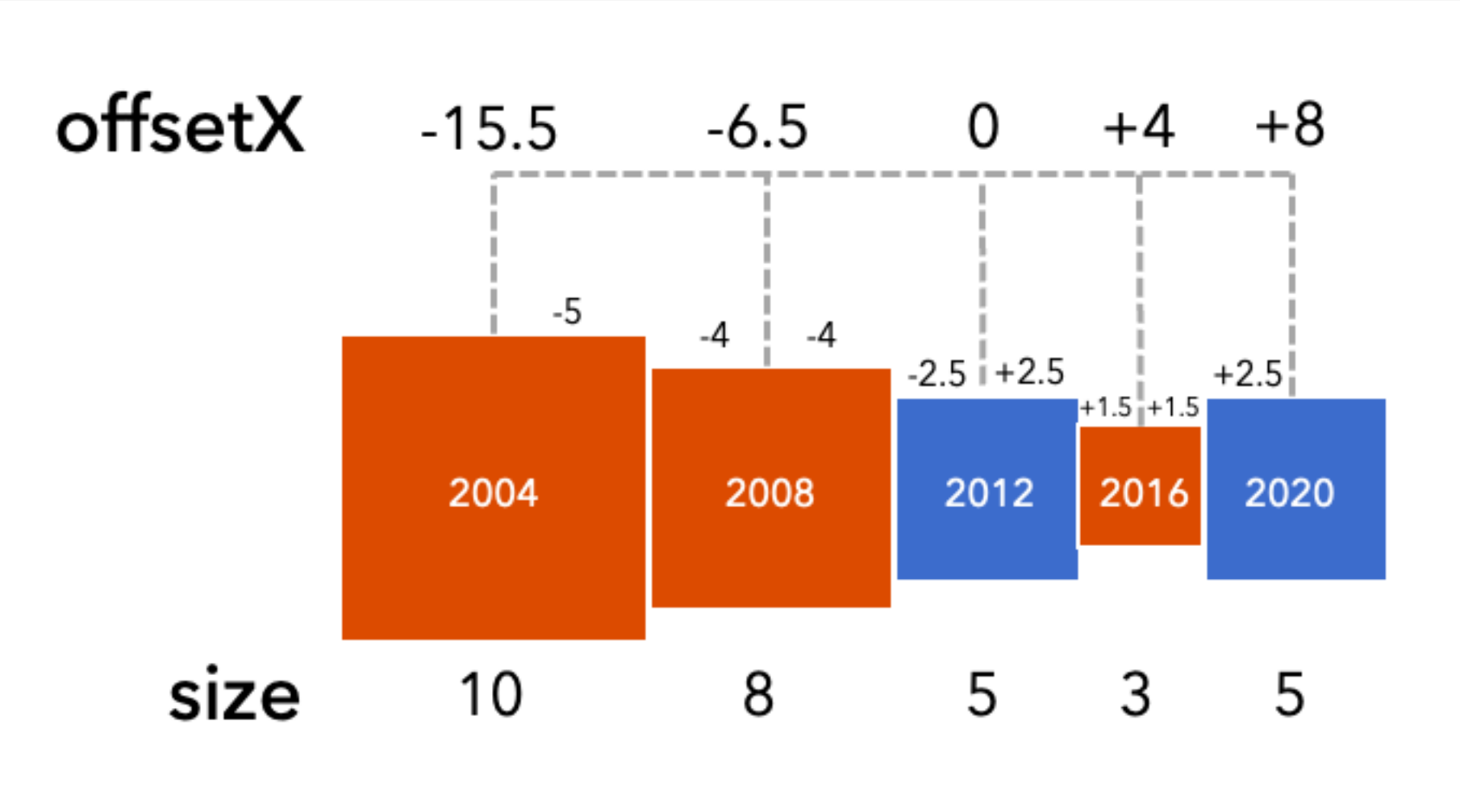


```
const sizeOverride = {
  type: "CIMPrimitiveOverride",
  primitiveName: "election-2016",
  propertyName: "Size",
  valueExpressionInfo: {
    type: "CIMExpressionInfo",
    title: "Margin of Victory",
    expression: sizeOverrideExpression,
    returnType: "Default",
  }
};
```

```
var demVotes = $feature.SUM_dem_2016;
var repVotes = $feature.SUM_rep_2016;
var othVotes = $feature.SUM_oth_2016;
var allVotes = Reverse(Sort([demVotes,
repVotes, othVotes]));
var value = allVotes[0] - allVotes[1];
var sizeFactor = When(
  value >= 2000000, 32,
  value >= 500000, 16 + (0.0000107 * (value -
500000)),
  value >= 100000, 8 + (0.00002 * (value -
100000)), value >= 10000, 4 + (0.000044 *
(value - 10000)),
  value > 0, 2 + (0.0002 * value), 0 );
var scaleFactorBase = (18489200 / $view.scale);
var scaleFactor = When(
  scaleFactorBase >= 0.5, scaleFactorBase *
0.6,
  scaleFactorBase >= 0.25, scaleFactorBase *
0.45,
  scaleFactorBase >= 0.125, scaleFactorBase *
0.3125,
  scaleFactorBase * 0.1875 );
return sizeFactor * scaleFactor;
```

3. Define primitive overrides

OffsetX – to properly place symbol layers so they are adjacent



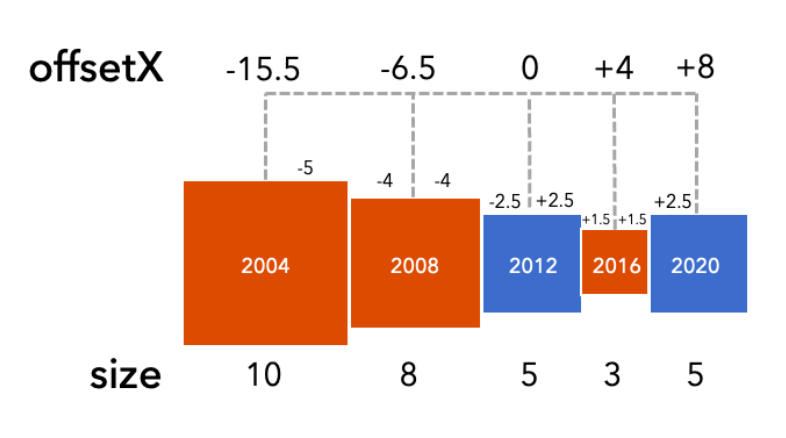
3. Define primitive overrides

OffsetX – to properly place symbol layers so they are adjacent

primitiveName	offsetX
election-2004	-1 * ((0.5 * election2004size) + election2008size + (0.5 * election2012size))
election-2008	-1 * ((0.5 * election2008size) + (0.5 * election2012size))
election-2012	0
election-2016	(0.5 * election2012size) + (0.5 * election2016size)
election-2020	(0.5 * election2012size) + election2016size + (0.5 * election2020size)

3. Define primitive overrides

OffsetX – to properly place symbol layers so they are

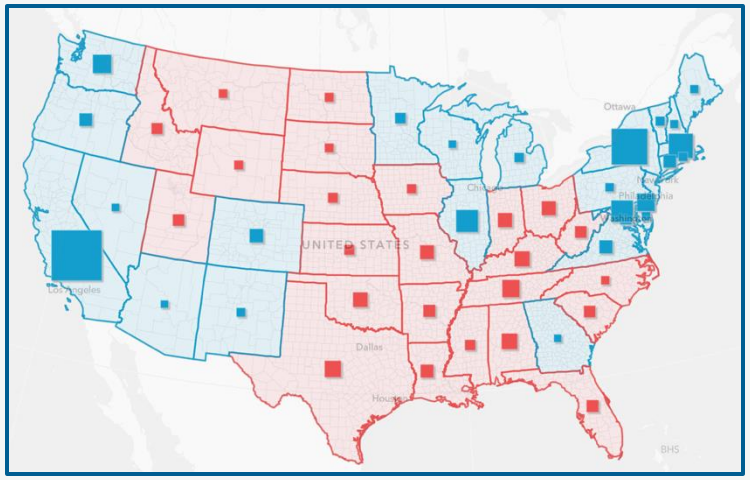


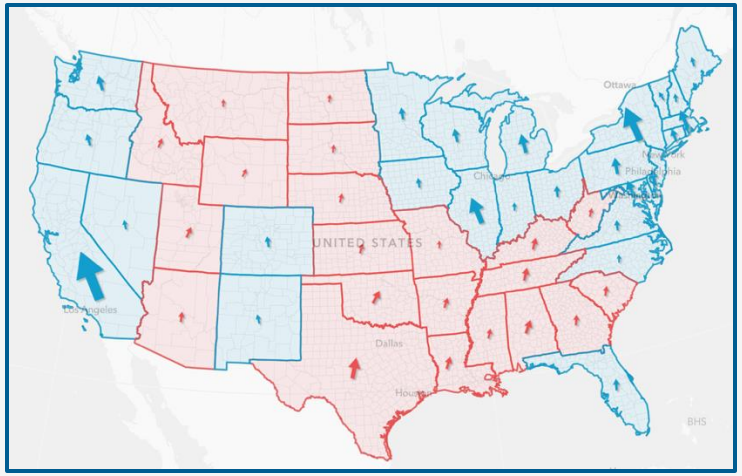

```
const colorOverride = {
  type: "CIMPrimitiveOverride",
  primitiveName: "election-2016",
  propertyName: ”OffsetX",
  valueExpressionInfo: {
    type: "CIMExpressionInfo",
    title: "Winner",
    expression: offsetXOverrideExpression,
    returnType: "Default",
  }
};
```

```
// a VERY abbreviated version of this…
var years = [2004, 2008, 2012, 2016, 2020];
var selectedYear = 2016;
var numItems = Count(years);
var isEvenItems = isEven(numItems);
var middleIndex = Floor(numItems / 2);
if (isEvenItems) { middleIndex -= 0.5; }
var index = IndexOf(years, selectedYear);
var direction = iif(index < middleIndex, -1,
1);
var yearStart = iif(direction < 0,
selectedYear, years[middleIndex]);
var yearEnd = iif(direction < 0,
years[middleIndex], selectedYear);
var sizes = [];
var offsetX = 0;
for (var year = yearStart; year <= yearEnd;
year += interval) {
  // insert size calculation here
  var factor = iif( (year == yearStart) ||
(year == yearEnd), 0.5, 1 );
  offsetX += (size * factor);
}
return offsetX * direction;
```

A review of each style

Winner + margin and winner + lean





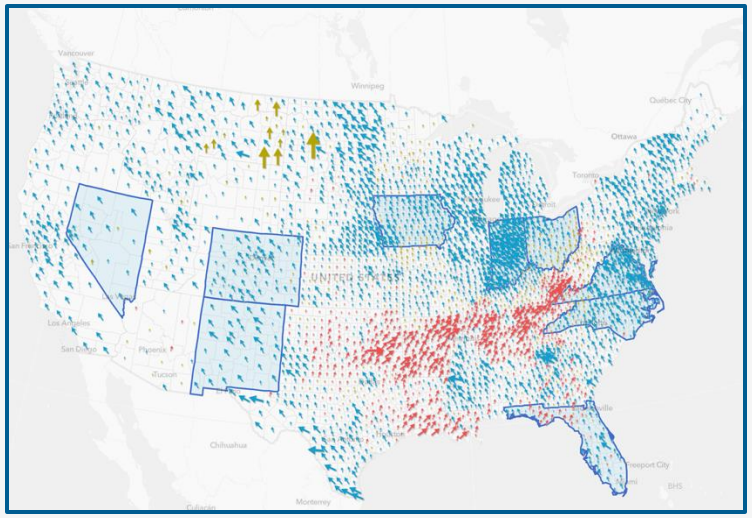
Strengths

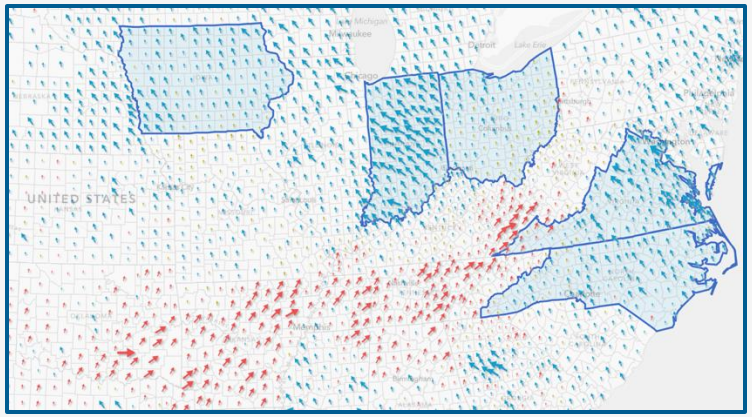
- Showing who won
- **Low cognitive effort** for seeing year-to-year **winner**
- Good for **short attention spans**
- Quick to grasp
  - Each frame has familiarity with general audiences

Weaknesses

- Showing overall trend
- Difficult to see slight regional changes that have major impact
  - e.g. individual counties influencing electoral votes
- One year visible at a time
- Seeing **trend** requires **high cognitive effort**
  - Requires toggling slider values back and forth at different stops

Swing





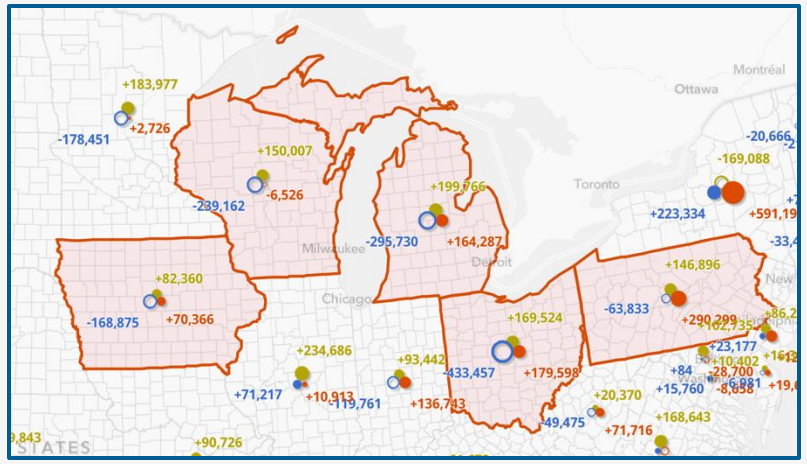
Strengths

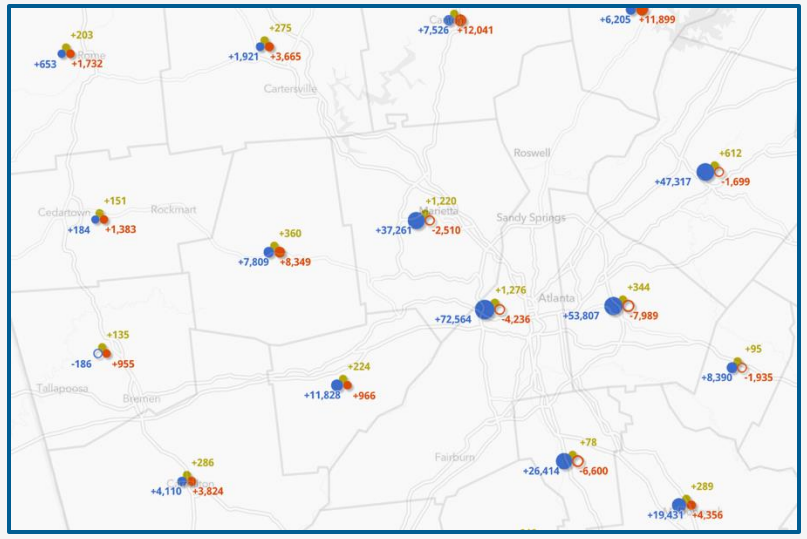
- Showing predominant change in voting behavior
- **Easy to see party swings** and dramatic changes in a given year
- Good for **short attention spans**

Weaknesses

- **High cognitive effort** for seeing year-to-year **winner**
- Difficult to see slight regional changes that have major impact
  - e.g. individual counties influencing electoral votes
- Seeing **trend** requires **high cognitive effort**
  - Requires toggling slider values back and forth at different stops

Composite symbols: change in votes for all parties





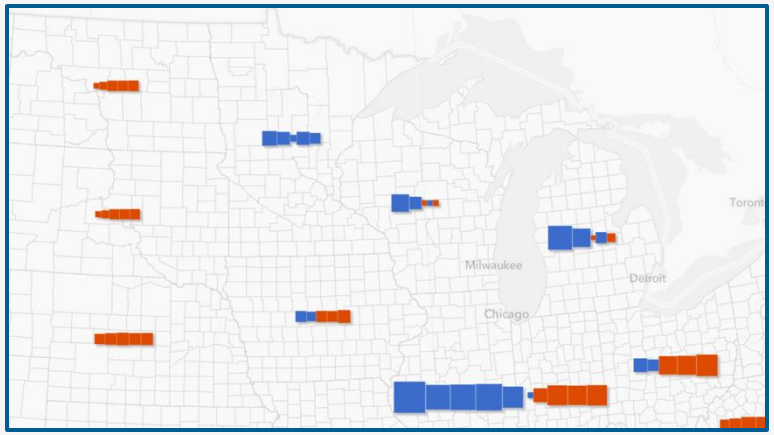
Strengths

- Showing **detailed nuances** of voting shifts in different parties year to year
  - e.g. individual counties influencing electoral votes
- Easy to see party swings, and dramatic changes using slider.

Weaknesses

- Showing who won
- **Initial view** requires **more cognitive effort**
  - e.g. “what am I looking at?”
- **High cognitive effort** for seeing year-to-year **winner**
- Seeing **trend** requires **high cognitive effort**
  - Requires toggling slider values back and forth at different stops
- Audiences with short attention spans

Trend boxes



Strengths

- Seeing **trend** requires **low cognitive effort**
  - Can see trend at first glance. No slider interaction required.
- Complements winner + margin style
  - Same foundational symbology in each symbol part
- Easy to see party swings, and **dramatic changes**
- Good at **showing outliers** that wouldn't be considered part of a trend

Weaknesses

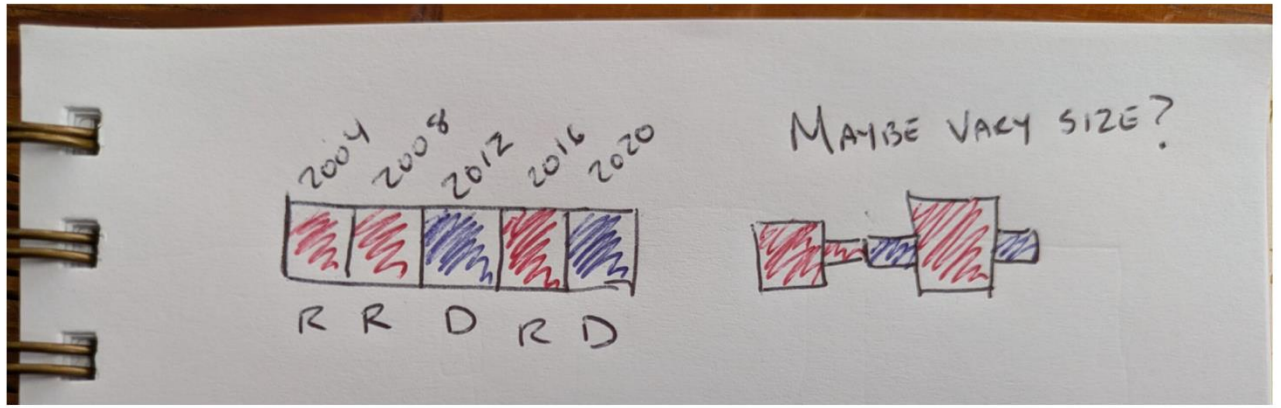
- **Initial view** requires **more cognitive effort**
  - e.g. "what am I looking at?"
- Audiences with short attention spans
- Symbol overlap – no deconfliction
- Size variations can be difficult to perceive (similar to graduated size)

Visualize voting trends in 20 years' worth of U.S. election data

By Kristian Ekenes

At the 2023 Esri Developer Summit, I demonstrated various ways to visualize data as it changed over time in this technical session. The session covered common approaches for visualizing change including animations, composite multivariate symbology, and small multiples. Many of these examples used election data to illustrate the advantages and disadvantages of each technique.

After the session, an attendee approached me with an idea she had sketched on a piece of paper. The idea was to visualize the results of multiple elections using square symbols where each square represented the winner of an election using color. The sketch looked something like the following.



The customer's sketch initially showed squares of equal size representing the results of multiple elections. As we conversed, we also discussed the possibility of varying the size of each square based on another attribute, like margin of victory.

In summary

- **Exploring a map is a lot like reading**
- Pamphlets, signs, news headlines, summaries, etc.
  - Use styles that require low cognitive effort
- Novels, long-form articles, textbooks
  - Requiring more cognitive effort is OK here
  - **Trend style** is best suited **for folks willing to take the time** to explore the data
  - Allows me to “get lost” in a map to learn and discover something new

Other ways of visualizing trends

- **Spark lines**
  - Not great for categorical data. Fine with a few features. Difficult for lots of features.
- **Waffle grids**
  - Good for categorical data, but a lot (more) to digest
- **Small multiples**
  - Good for print, hard as interactive web maps
- **Choropleth**
  - Diverging red-blue color ramp showing winner of most recent election with saturation of color varied by the number of elections that party won in a row.
  - Hides the nuances of the trend

Looking ahead – ArcGIS Online Map Viewer

- Add time series configuration UI
  - No need to write Arcade
  - Hopefully available next year
- Will evaluate potential of adding “Trend” as an out-of-the-box map style
  - Assess need, demand, design, level of effort, etc.

Thank you!

I welcome all feedback

Kristian Ekenes

kekenes@esri.com

kristian-ekenes

ekenes

nacis | North American Cartographic Information Society

esri | THE SCIENCE OF WHERE