Exercise

Labels as Symbols

Section 4 Exercise 3

February 2, 2021

esri

THE SCIENCE OF WHERE®
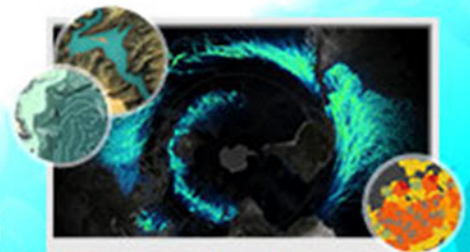
Labels as Symbols

Instructions

Use this guide and ArcGIS Pro to reproduce the results of the exercise on your own.

Note: The version of ArcGIS Pro that you are using for this course may produce slightly different results from the screen shots that you see in the course materials.

Time to complete

Approximately 50-60 minutes

Software requirements

ArcGIS Pro 2.7
ArcGIS Pro Standard license (or higher)

Note: The MOOC provides a separate ArcGIS account (user name and password) that you will need to use to license ArcGIS Pro and access other software applications used throughout the MOOC exercises. This account (user name ending with _cart) provides the appropriate ArcGIS Online role, ArcGIS Pro license, ArcGIS Pro extensions, and credits. We strongly recommend that you use the provided course ArcGIS account to ensure that you have the appropriate licensing to complete the exercises. Exercises may require credits. Using the provided course ArcGIS account ensures that you do not consume your organization's credits. Esri is not responsible for any credits consumed if you use a different account. Moreover, Esri will not provide technical support to students who use a different account.

Introduction

On a map, labels serve many purposes. Although labels can give cartographers headaches when map space is scarce, they are important because they designate features, names, and other characteristics.

In thematic mapping, you use symbols to show differences in amount or type. You might use proportional symbols to show differences in the quantity of something as it differs from one place to another. You might use a color scheme ranging from light to dark to show changes in magnitude between areas or use different hues to show differences in categorical information.

What if you combine all these approaches?

Labels can operate as literal symbols. For instance, you could position a label over an area to show the data value for that area. Then, there is no need to symbolize the map because the

label indicates data values and spares people from trying to interpret the data values through symbols. But maybe you could go further. You could give the labels different colors to signify certain characteristics and size them differently to show other information.

What will you learn?

In this exercise, you will create a map of proportional labels that have some additional meaning encoded through the use of color. You have previously created a series of different thematic maps in Section 3 and used attribute-driven symbology, so you already have a good understanding of creating thematic maps. You have also explored labeling options in the first exercise in Section 4. Now you will put these two ideas together.

In addition, you will be introduced to the use of Arcade Expressions in ArcGIS Pro, which allows you to control many aspects of symbology by creating scripted expressions.

Step 1: Download the exercise data files

In this step, you will download the exercise data files.

  a Open a new web browser tab or window.

  b Go to https://bit.ly/2SmoWDT and download the exercise data ZIP file.

Note: The complete URL to the exercise file is https://www.arcgis.com/home/item.html?id=b93cdd3a604a4835a77bcc4fef1eae05. The file is 23 MB.

  c Extract the files to a folder on your local computer, saving them in a location that you will remember.

## Step 2: Open an ArcGIS Pro project

Now, you will open the map and begin to familiarize yourself with its current symbology.

  a If necessary, start ArcGIS Pro and sign in using your provided course ArcGIS credentials (user name ending with _cart).

  b From the main ArcGIS Pro start page, click Open Another Project.

  c Browse to the location where you saved the exercise data files and open the LabelsAsSymbols.ppkx project package.

The project opens with the Proportional Labels map active. This map contains a simple polygon layer of the counties of the United States active. The Contents pane displays two

views of the data: one in the Diverging Hue layer, which is currently turned on, and another in the Blended Hue layer. Although the layers are symbolized differently, they depict the same data. This setting is just to give you some context for what you are mapping: the share of the vote for Republicans and Democrats in the 2016 presidential election.

The diverging and blended hue maps are just two different ways to look at the same dataset. They show you where greater and lesser shares of votes occurred for the two parties. You will not use these maps for the final proportional labels, but they can help you understand the underlying pattern of the data.
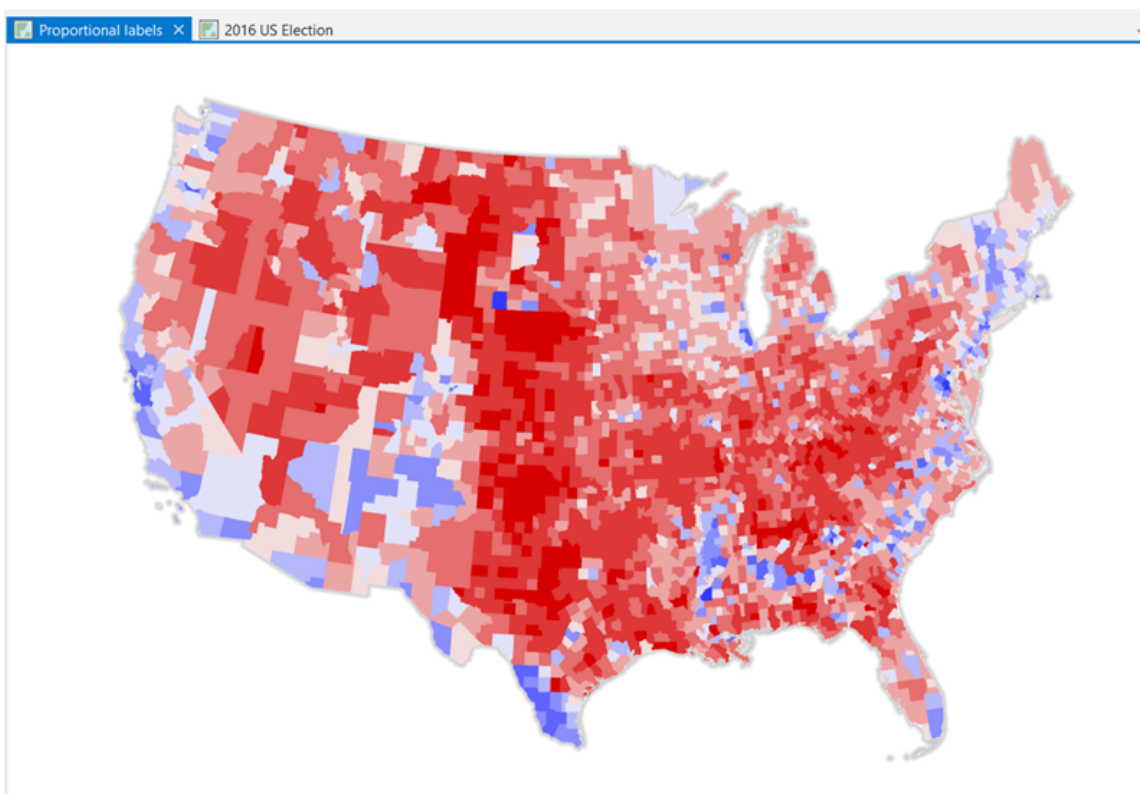
Note: If you would like to check your work as you go through this exercise, result layers are provided in the 2016 US Election map.

(d) Save your project with a name such as **LabelsAsSymbols_<your first and last name>**.aprx in the folder on your computer where you are saving your work.

Step 3: Explore the data

For this exercise, you will work with United States election data from 2016.

---

a If necessary, for the Proportional Labels map, open the Contents pane.

Hint: On the ribbon, click the View tab, and then in the Windows group, click Contents.

b In the Contents pane, open the attribute table for either the Diverging Hue or Blended Hue layers (both layers point to the same data).

Hint: In the Contents pane, right-click the layer name and choose Attribute Table.

OBJECTID	Shape	NAME	STATE_NAME	STATE_FIPS	CNTY_FIPS	FIPS	DEM_votes	REP_votes	DEM_percent	REP_percent	Total_votes
1	Polygon	Autauga	Alabama	01	001	01001	5908	18110	23.956855	73.435789	24661
2	Polygon	Baldwin	Alabama	01	003	01003	18409	72780	19.56531	77.351472	94090
3	Polygon	Barbour	Alabama	01	005	01005	4848	5431	46.66025	52.271415	10390
4	Polygon	Bibb	Alabama	01	007	01007	1874	6733	21.422039	76.966164	8748
5	Polygon	Blount	Alabama	01	009	01009	2150	22808	8.469902	89.851875	25384
6	Polygon	Bullock	Alabama	01	011	01011	3530	1139	75.090406	24.228887	4701
7	Polygon	Butler	Alabama	01	013	01013	3716	4891	42.786413	56.315486	8685
8	Polygon	Calhoun	Alabama	01	015	01015	13197	32803	27.855876	69.239699	47376
9	Polygon	Chambers	Alabama	01	017	01017	5763	7803	41.827551	56.633764	13778

The layer attribute table contains several fields, some of which are described in the following table:

Field name	Description
OBJECTID	Alphanumeric code used to identify individual constituencies
NAME	County name
STATE_NAME	State name
STATE_FIPS	Administrative boundary ID
CNTY_FIPS	Administrative boundary ID
FIPS	Administrative boundary ID
DEM_votes	Number of votes cast for the Democratic party
REP_votes	Number of votes cast for the Republican party
DEM_percent	Percentage share of votes cast for Democratic party
REP_percent	Percentage share of votes cast for Republican party
Total_votes	Total number of votes cast
Vote_difference	Difference between votes for the two candidates
Winner	Surname of the winning candidate
Win_percent	Percentage share of vote for winning candidate
Continuous_percent	Percentage share recoded so Republicans are positive values and Democrats are negative values; used to create the two maps in the Contents pane
Win_votes	Number of votes for the winning candidate
Win_code	Numeric code that can be used for definition queries

You can examine the values in the different fields to familiarize yourself with the data that you will be working with.
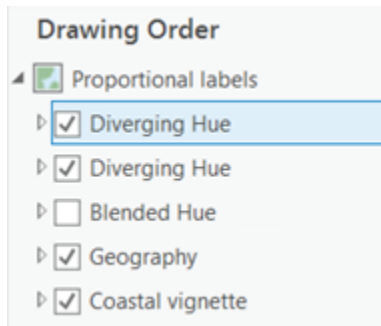
c When you are finished, close the attribute table.

Step 4: Prepare the map

First, in the Contents pane, you will make a copy of one of the mapped layers, which will form the basis of the new map layer.
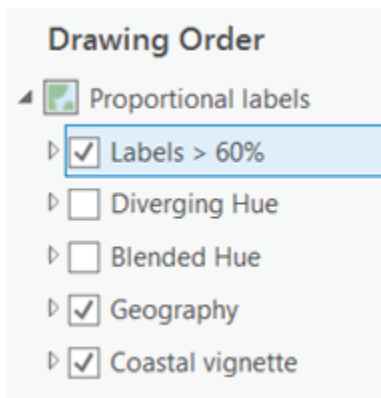
a In the Contents pane, right-click the Diverging Hue layer and choose Copy.

**b** At the top of the Contents pane, right-click the Proportional Labels map item and choose Paste.

Drawing Order
- ▲ Proportional labels
  - ▷ ☑ Diverging Hue
  - ▷ ☑ Diverging Hue
  - ▷ ☐ Blended Hue
  - ▷ ☑ Geography
  - ▷ ☑ Coastal vignette

The copied layer is added at the top of the Contents pane, and you now have two Diverging Hue layers.

c Turn off the lower Diverging Hue layer.

d Rename the new layer **Labels > 60%**.

Drawing Order
- ▲ Proportional labels
  - ▷ ☑ Labels > 60%
  - ▷ ☐ Diverging Hue
  - ▷ ☐ Blended Hue
  - ▷ ☑ Geography
  - ▷ ☑ Coastal vignette

Because you will be adding many labels and do not want a conflicting background, you will create a completely neutral background for the map by symbolizing the layer with a single symbol.

e In the Contents pane, click the Labels > 60% layer to select it, if necessary.

f From the Feature Layer contextual tab, on the Appearance tab, in the Drawing group, click the Symbology down arrow and choose Single Symbol.

g In the Symbology pane, next to Symbol, click the symbol.
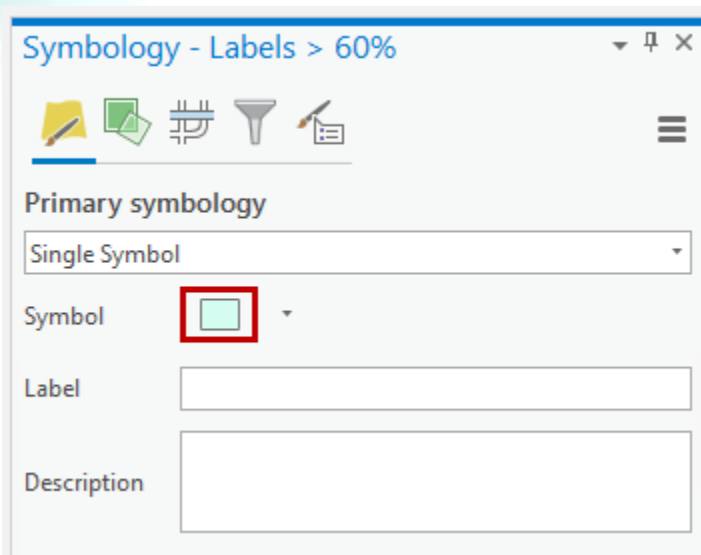
The Format Polygon Symbol pane opens.

(h) In the Format Symbol Polygon pane, click the Properties tab.

(i) In the Appearance section, change Color to No Color, the Outline Color to No Color, and the Outline Width to 0 pt.
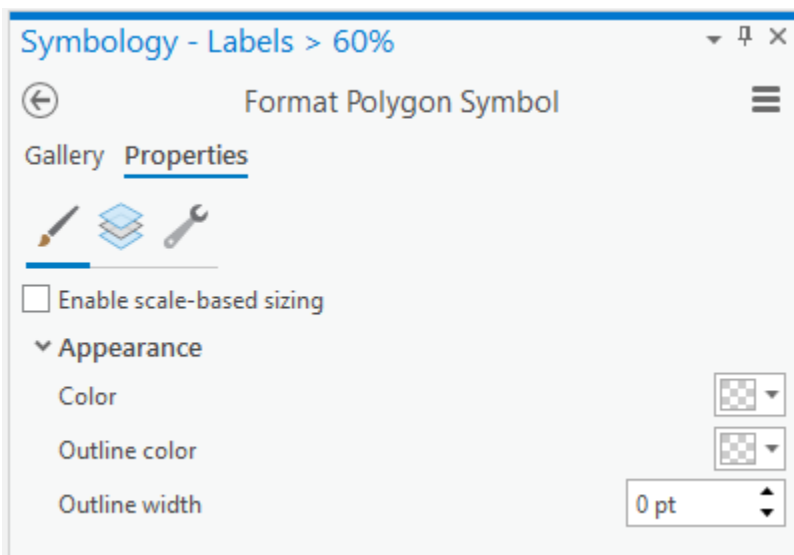
(j) Click Apply, and then close the Symbology pane.

(k) If necessary, in the Contents pane, turn off the Diverging Hue layer.

You now have a completely blank canvas on which to make your labeled map, with just the Geography and Coastal_vignette layers visible below the transparent Labels > 60% layer.

Note: If you are zoomed in, you will see the state outlines. For context, you may want to zoom out so that you see the entire continental United States.

Step 5: Use a query expression to filter labels on a map

Now you can begin to add labels to your map. Then, you will use a query expression to filter the labels to display only those that meet certain criteria. This is referred to as a definition query.

(a) In the Contents pane, with the Labels > 60% layer selected, click the Labeling tab.
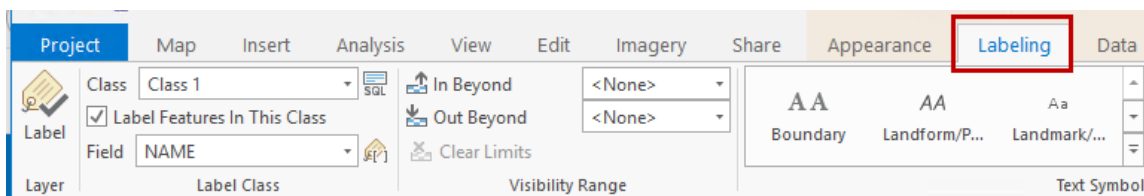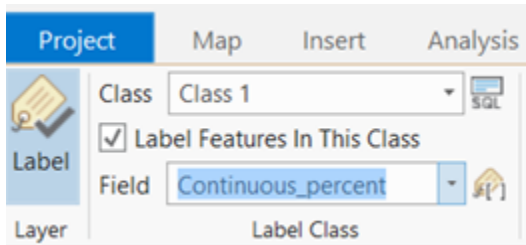

This tab provides access to the core labeling properties of your layer.

(b) In the Label Class group, for Field, choose Continuous_percent.

Hint: Refer to the table in Step 3b for layer attribute table field descriptions.

---

c From the Labeling tab, in the Layer group, click Label to activate dynamic labeling on the layer.

If you wait a short while (you can see the spinning Drawing icon in the bottom-right corner of the map window that tells you ArcGIS Pro is busy drawing), your labels appear, although they are currently far from impressive.
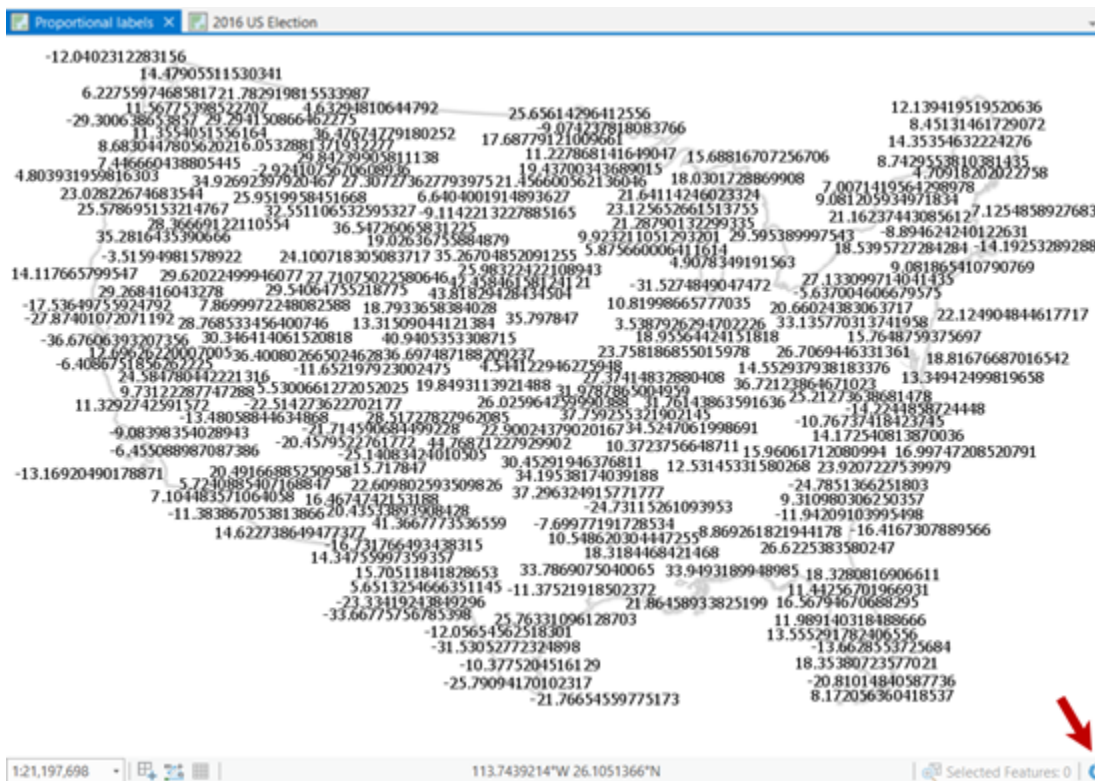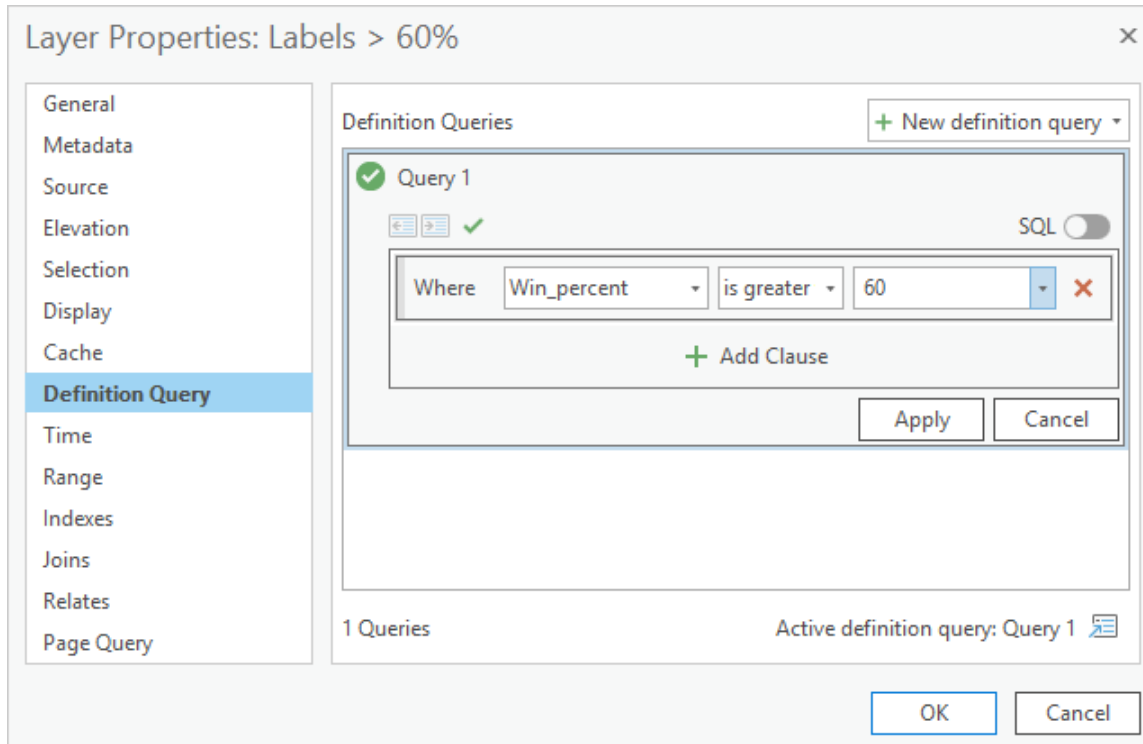

Because the drawing of many labels takes time, you can apply a definition query to work on only a subset of labels.

d In the Contents pane, right-click the Labels > 60% layer and choose Properties.

e In the Layer Properties dialog box, click the Definition Query tab, and then click New Definition Query.

f Create the following query expression: `Where Win_percent Is Greater Than Or Equal To` **60**.

Note: Delete any extra numbers to the right of the decimal place so that you are left with only the integer 60.

g Click Apply, and then click OK.

Your map will now show labels only for areas that meet the criteria.

You have just used an expression to apply a filter to how the map behaves. This technique is a simple form of specifying an expression using the dialog box.

You can also change the color of your labels and adjust the number of significant figures shown in the Labeling pane, using Arcade Expressions to modify the properties of the labels. This ability introduces you to the idea that you can build complex symbology based on different aspects of the data that cannot be implemented through the default renderers. It provides the possibility of effectively modifying symbology in any way you want over and above the out-of-the-box options.

Step 6: Modify the properties of labels using Arcade Expressions

There are several plain text files in the folder where you extracted your exercise files. You will use the contents of these text files to help you build some Arcade Expressions to modify the properties of the labels in your map.

(a) In a text editor, such as Notepad (Windows) or TextEdit (Mac), open the Expressions1.txt file from the location on your computer where you saved the exercise data files.

Take a deep breath … because the file contains code.

Arcade Expressions is a lightweight expression language. Like other expression languages, it can perform mathematical calculations and evaluate logical statements. It was designed

specifically for creating custom visualizations and labeling expressions in the ArcGIS platform, and it provides a powerful way to customize your maps.

This course is not about how to code, but coding can be useful in cartography to extend out-of-the-box functionality. One of the best ways to learn is to look at some code and follow along.

The following code is listed in the Expressions1.txt file:

```
function FindText(continuousValue){
if (continuousValue >= 0){
    return Text(50 + continuousValue, '#');}
else {
    return Text(50 - continuousValue, '#');}
}

"<CLR " + ">" + FindText($feature.Continuous_percent) + "</CLR>"
```

Remember that the Continuous_percent field in the attribute table of the data that you are using to generate labels had both positive and negative values. The column includes the percentage that Democrats won, with a negative sign in front of the value if Democrats won the county. The column includes the percentage Republicans won, with no sign in front if Republicans won the county.

The Continuous_percent field, therefore, holds the winning percentage for each county and provides a way to determine whether it was Republican or Democratic.

You will use this field to make your labels and show the winning percentage by county across the map.

However, you do not want to display the negative sign on the finished map. Using logic and calculation, the code writes a short function to return values that are all in the positive range.

The first part of the code states that if the value in the field is assessed as positive based on the calculation `50 + continuousValue` (the Continuous_percent value), return the value and return only the numeric values before the decimal point by using a single #.

If you specify '#.#', a single decimal place would be returned and so on. The next part of the code states that if this calculation is false, the value must be negative.

The calculation `50 - continuousValue` (the Continuous_percent value) checks for negative values and returns just the integer portion of the value.

The final line of code draws the labels.

b In the Contents pane, ensure that the Layers > 60% layer is selected.

(c) From the Labeling tab, in the Label Class group, next to the Field drop-down list, click the Expression button 🏠.

The Label Class pane opens. You can dock or undock it as desired.

Hint: At the top of the pane, click the Options down arrow and choose Dock.

(d) In the Label Class pane, from the Class tab, ensure that Language is set to Arcade.

(e) In your text editor, copy the code from the Expressions1.txt file.

(f) In the Label Class pane, in the Expression field, paste the code, overwriting any existing content.

```
Label Class - Labels > 60%                          ▼ 🗗 ✕

                        Class 1                           ≡

Class ▾   Symbol  Position

   🏠    📄    🏷
        SQL

Language  Arcade              ▼

Fields                    ▼    Functions              ▼

  OBJECTID               ▲       Abs()                ▲
  NAME                           Acos()
  STATE_NAME                     Angle()
  STATE_FIPS                     Area()
  CNTY_FIPS                      AreaGeodetic()
  FIPS                           Asin()
  DEM_votes                      Atan()
  REP_votes              ▼       Atan2()              ▼

Insert Values                 ▼

Expression
function FindText(continuousValue){    ▲
if (continuousValue >= 0){
    return Text(50 + continuousValue, '#');}
else {
    return Text(50 - continuousValue, '#');}
}

"<CLR " + ">" + FindText              ↩
($feature.Continuous_percent) + "</CLR>"



                                      ▼

            🔄    ✓    📁    →
```

g Underneath the Expression field, click the Verify button ✓ .

Clicking this button verifies that the syntax of the code is acceptable. The expression should return as valid.

h Click Apply.

After a few seconds, integer labels replace the previous labels, and you are on your way to making a useful map.
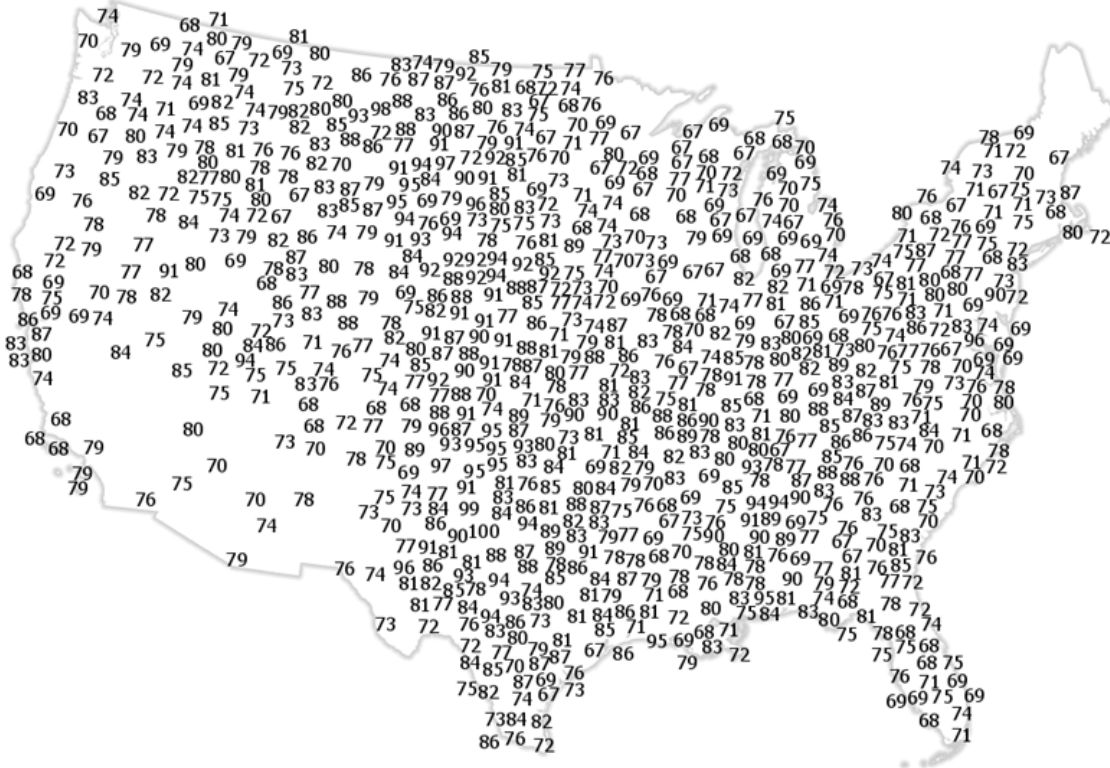
ⓘ Save your project.

Step 7: Add color to labels

You have now used a piece of code to handle some of the rendering of the labels, but you need a way to distinguish Democratic-winning counties from Republican-winning counties. You will add a function to the code to make this distinction using color.

ⓐ In your text editor, open the Expressions2.txt file.

The following code is listed in the Expressions2.txt file (new code is shown here in a darker color):

```
function FindDEMColor(votePercentage)
{
if (votePercentage >= 50){
    return "red = '18' green = '15' blue='224'";}
else{
    return "red = '0' green = '0' blue='0'";}
}

function FindGOPColor(votePercentage)
{
if (votePercentage >= 50){
    return "red = '237' green = '28' blue='36'";}
else{
    return "red = '0' green = '0' blue='0'";}
}

function FindColor(continuousValue){
if (continuousValue >= 0){
    return FindGOPColor(50 + continuousValue);}
else {
    return FindDEMColor(50 - continuousValue);}
}

function FindText(continuousValue){
if (continuousValue >= 0){
    return Text(50 + continuousValue, '#');}
else {
    return Text(50 - continuousValue, '#');}
}

"<CLR " + FindColor($feature.Continuous_percent) + ">" +
FindText($feature.Continuous_percent) + "</CLR>"
```

This version of the code extends the previous one. It includes the function that assesses whether the value in the Continuous_percent field is positive or negative, but there are additional lines of code and three additional functions.

- FindDEMColor specifies a mix of colors using the RGB scheme for values of data that are eventually returned as Democratic, blue.
- FindGOPColor specifies a mix of colors using the RGB scheme for values of data that are eventually returned as Republican, red.
- FindColor uses the same logic as FindText in the original code to apply the FindGOPColor function to positive values and the FindDEMColor function to negative values.

An additional command is inserted in the last line to draw the text using the specified colors.

b In the Label Class pane, click in the Expression field, press Ctrl + A to select all the code, and then press Delete to delete it.

c In your text editor, copy the code from the Expressions2.txt file.

d In the Label Class pane, paste the code into the Expression field.

e Validate the syntax of the code.

Hint: Click the Verify button ✔ *under the Expression field.*

The expression should return as valid.

f Click Apply.

The same labels appear, but they are red or blue depending on whether the Democratic or Republican candidate won the county.
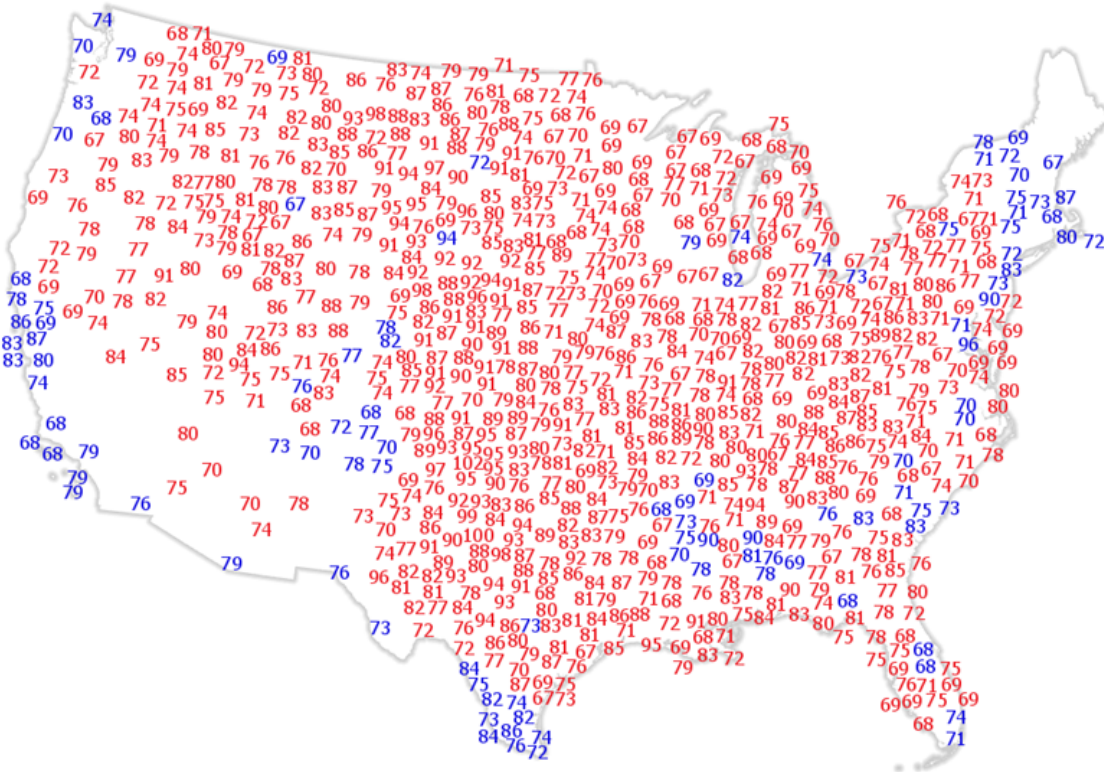
g Save your project.

Step 8: Vary label color

Strength of vote is also an important variable. The original diverging hue choropleth map used different shades of red or blue to emphasize the strength of voting for each party in the county. You can also use Arcade Expressions to define the shades of red or blue for the labels. Rather than specify a single blue or red for the label color, you can specify multiple colors to be used when criteria in the expression are met.

a. In your text editor, open the Expressions3.txt file.

The following code is listed in the Expressions3.txt file (new code is shown here in a darker color):

```
function FindDEMColor(votePercentage)
{
if (votePercentage < 50){
    return "red = '185' green = '82' blue='159'";}
if (votePercentage >= 50 && votePercentage < 55){
    return "red = '178' green = '82' blue='159'";}
if (votePercentage >= 55 && votePercentage < 60){
    return "red = '169' green = '83' blue='160'";}
if (votePercentage >= 60 && votePercentage < 65){
    return "red = '161' green = '83' blue='160'";}
if (votePercentage >= 65 && votePercentage < 70){
    return "red = '151' green = '83' blue='161'";}
if (votePercentage >= 70 && votePercentage < 75){
    return "red = '140' green = '83' blue='161'";}
if (votePercentage >= 75 && votePercentage < 80){
    return "red = '131' green = '83' blue='162'";}
if (votePercentage >= 80 && votePercentage < 85){
    return "red = '117' green = '83' blue='162'";}
if (votePercentage >= 85 && votePercentage < 90){
    return "red = '100' green = '83' blue='163'";}
if (votePercentage >= 90 && votePercentage < 95){
    return "red = '88' green = '83' blue='163'";}
if (votePercentage >= 95){
    return "red = '18' green = '15' blue='224'";}
else{
    return "red = '0' green = '0' blue='0'";}
}

function FindGOPColor(votePercentage)
{
if (votePercentage < 50){
    return "red = '185' green = '82' blue='159'";}
if (votePercentage >= 50 && votePercentage < 55){
    return "red = '191' green = '79' blue='146'";}
if (votePercentage >= 55 && votePercentage < 60){
    return "red = '196' green = '76' blue='136'";}
if (votePercentage >= 60 && votePercentage < 65){
    return "red = '200' green = '72' blue='126'";}
if (votePercentage >= 65 && votePercentage < 70){
    return "red = '205' green = '69' blue='115'";}
if (votePercentage >= 70 && votePercentage < 75){
    return "red = '209' green = '65' blue='105'";}
if (votePercentage >= 75 && votePercentage < 80){
    return "red = '214' green = '61' blue='94'";}
if (votePercentage >= 80 && votePercentage < 85){
    return "red = '220' green = '54' blue='80'";}
if (votePercentage >= 85 && votePercentage < 90){
    return "red = '222' green = '51' blue='73'";}
if (votePercentage >= 90 && votePercentage < 95){
    return "red = '225' green = '48' blue='67'";}
if (votePercentage >= 95){
    return "red = '237' green = '28' blue='36'";}
else{
    return "red = '0' green = '0' blue='0'";}
}

function FindColor(continuousValue){
if (continuousValue >= 0){
    return FindGOPColor(50 + continuousValue);}
else {
    return FindDEMColor(50 - continuousValue);}
}

function FindText(continuousValue){
if (continuousValue >= 0){
    return Text(50 + continuousValue, '#');}
else {
    return Text(50 - continuousValue, '#');}
}

"<CLR " + FindColor($feature.Continuous_percent) + ">" +
FindText($feature.Continuous_percent) + "</CLR>"
```

The difference between the code in Expressions2.txt and this code is that `FindDEMColor` and `FindGOPColor` have expanded if-then-else statements to specify different colors for values that fall within different bands. These expanded statements effectively apply a choropleth-style shading scheme to the labels.

b In the Label Class pane, in the Expression field, delete the existing code.

Hint: Press Ctrl + A to select all the code, and then press Delete.

c In your text editor, copy the code from the Expressions3.txt file.

d In the Label Class pane, paste the code into the Expression field and validate the syntax of the code.

Like the other expressions, this expression should be valid.

e Click Apply.
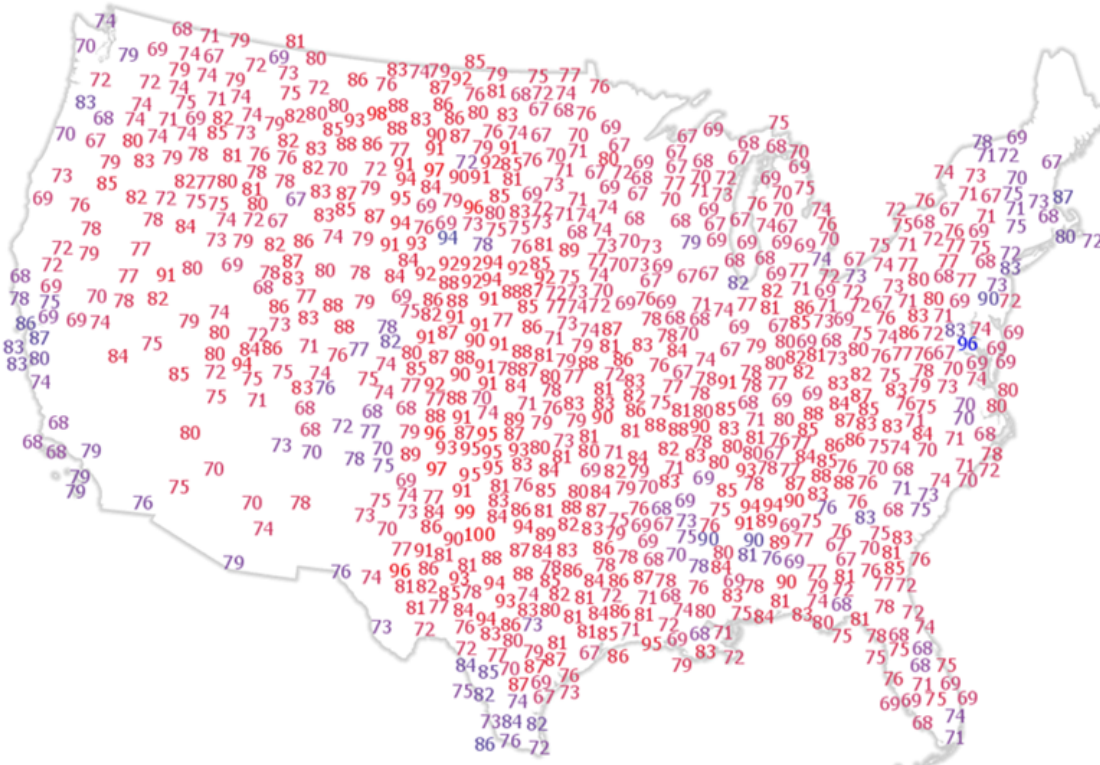
Your labels should now have a different value of blue or red to emphasize the strength of vote using a blended hue color scheme, where light colors represent low data values and dark colors represent high data values. For now, the labels appear only for counties with values

above 60% (because the definition query that you originally set up still applies) but not for those in the mid-range.

Next, you will apply some transparency to the labels to give some more emphasis than others.

f In your text editor, open the Expressions4.txt file.

The following code is listed in the Expressions4.txt file (new code is shown here in a darker color):

```
function FindDEMColor(votePercentage){
var baseColor = "red = '0' green = '0' blue='255'"
if (votePercentage < 50){
    return baseColor + " alpha='0'";}
if (votePercentage >= 50 && votePercentage < 55){
    return baseColor + " alpha='10'";}
if (votePercentage >= 55 && votePercentage < 60){
    return baseColor + " alpha='20'";}
if (votePercentage >= 60 && votePercentage < 65){
    return baseColor + " alpha='30'";}
if (votePercentage >= 65 && votePercentage < 70){
    return baseColor + " alpha='40'";}
if (votePercentage >= 70 && votePercentage < 75){
    return baseColor + " alpha='50'";}
if (votePercentage >= 75 && votePercentage < 80){
    return baseColor + " alpha='60'";}
if (votePercentage >= 80 && votePercentage < 85){
    return baseColor + " alpha='70'";}
if (votePercentage >= 85 && votePercentage < 90){
    return baseColor + " alpha='80'";}
if (votePercentage >= 90 && votePercentage < 95){
    return baseColor + " alpha='90'";}
if (votePercentage >= 95 && votePercentage < 100){
    return baseColor + " alpha='100'";}
else{
    return baseColor + " alpha='0'";}
}
function FindGOPColor(votePercentage){
var baseColor = "red = '255' green = '0' blue='0'"
if (votePercentage < 50){
    return baseColor + " alpha='0'";}
if (votePercentage >= 50 && votePercentage < 55){
    return baseColor + " alpha='10'";}
if (votePercentage >= 55 && votePercentage < 60){
    return baseColor + " alpha='20'";}
if (votePercentage >= 60 && votePercentage < 65){
    return baseColor + " alpha='30'";}
if (votePercentage >= 65 && votePercentage < 70){
    return baseColor + " alpha='40'";}
if (votePercentage >= 70 && votePercentage < 75){
    return baseColor + " alpha='50'";}
if (votePercentage >= 75 && votePercentage < 80){
    return baseColor + " alpha='60'";}
if (votePercentage >= 80 && votePercentage < 85){
    return baseColor + " alpha='70'";}
if (votePercentage >= 85 && votePercentage < 90){
    return baseColor + " alpha='80'";}
if (votePercentage >= 90 && votePercentage < 95){
    return baseColor + " alpha='90'";}
if (votePercentage >= 95 && votePercentage < 100){
    return baseColor + " alpha='100'";}
else{
    return baseColor + " alpha='0'";}
}

function FindColor(continuousValue){
if (continuousValue >= 0){
    return FindGOPColor(50 + continuousValue);}
else {
    return FindDEMColor(50 - continuousValue);}
}

function FindText(continuousValue){
if (continuousValue >= 0){
    return Text(50 + continuousValue, '#');}
else {
    return Text(50 - continuousValue, '#');}
}

"<CLR " + FindColor($feature.Continuous_percent) + ">" +
FindText($feature.Continuous_percent) + "</CLR>"
```
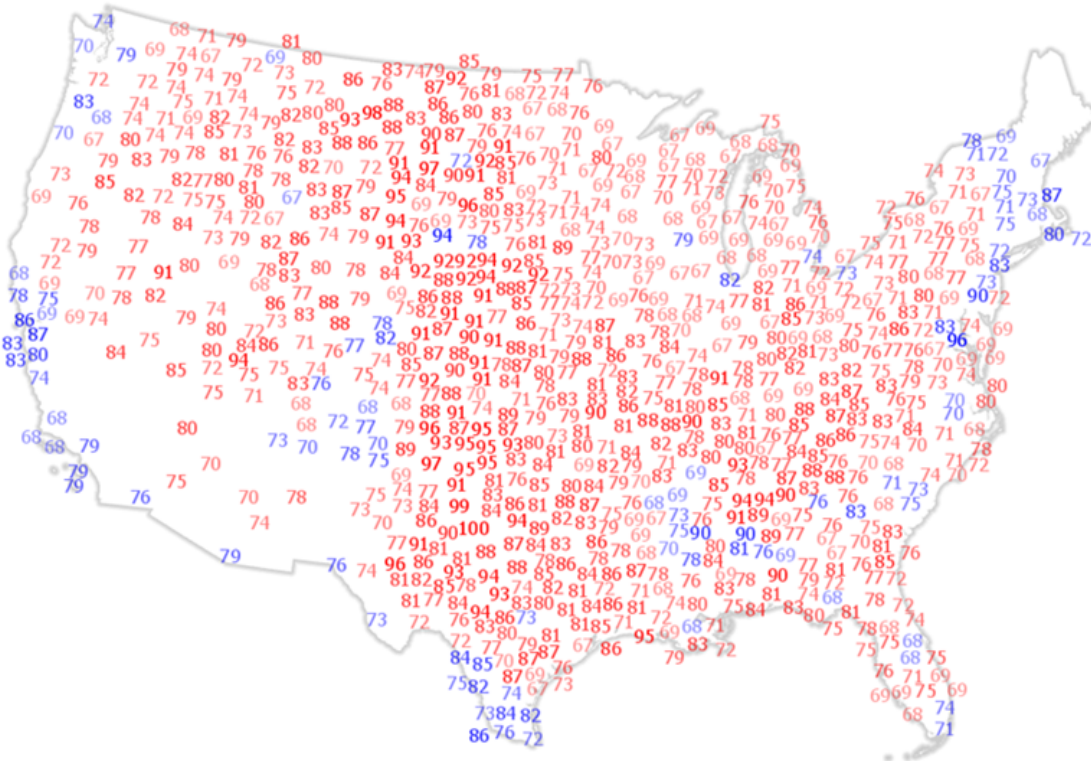
g In the Label Class pane, delete any existing code in the Expression field.

h In your text editor, copy the code from the Expressions4.txt file.

i In the Label Class pane, paste the code into the Expression field and validate the syntax of the code.

Like the other expressions, this expression should be valid.

j Click Apply.

The code is virtually the same except for how color is specified. In Expressions3.txt, a different set of RGB values was specified for each label class. A variable, `baseColor`, is used to set a solid red and a solid blue within each function. In this updated code, the colors are additionally modified by varying the alpha channel for each label class. Varying the alpha channel controls the amount of transparency applied to each color, which has the effect of emphasizing and de-emphasizing different label colors. Each symbol has the same red or blue, but the lower values have increased transparency applied, blending them into the background. This expression applies the same technique as the value-by-alpha choropleth mapping technique that you created in the Section 3 exercise *Thematic Mapping*, except applied to labels instead of the map itself.

k Save your project.

Step 9: Vary label size

Finally, you will change the font size using Arcade Expressions to use proportional symbol scaling to modify the labels.

a In your text editor, open the Expressions5.txt file.

The following code is listed in the Expressions5.txt file (new code is shown here in a darker color):

```
function FindDEMColor(votePercentage){
var baseColor = "red = '0' green = '0' blue='255'"
if (votePercentage < 50){
    return baseColor + " alpha='0'";}
if (votePercentage >= 50 && votePercentage < 55){
    return baseColor + " alpha='10'";}
if (votePercentage >= 55 && votePercentage < 60){
    return baseColor + " alpha='20'";}
if (votePercentage >= 60 && votePercentage < 65){
    return baseColor + " alpha='30'";}
if (votePercentage >= 65 && votePercentage < 70){
    return baseColor + " alpha='40'";}
if (votePercentage >= 70 && votePercentage < 75){
    return baseColor + " alpha='50'";}
if (votePercentage >= 75 && votePercentage < 80){
    return baseColor + " alpha='60'";}
if (votePercentage >= 80 && votePercentage < 85){
    return baseColor + " alpha='70'";}
if (votePercentage >= 85 && votePercentage < 90){
    return baseColor + " alpha='80'";}
if (votePercentage >= 90 && votePercentage < 95){
    return baseColor + " alpha='90'";}
if (votePercentage >= 95){
    return baseColor + " alpha='100'";}
else{
    return baseColor + " alpha='0'";}
}

function FindGOPColor(votePercentage){
var baseColor = "red = '255' green = '0' blue='0'"
if (votePercentage < 50){
    return baseColor + " alpha='0'";}
if (votePercentage >= 50 && votePercentage < 55){
    return baseColor + " alpha='10'";}
if (votePercentage >= 55 && votePercentage < 60){
    return baseColor + " alpha='20'";}
if (votePercentage >= 60 && votePercentage < 65){
    return baseColor + " alpha='30'";}
if (votePercentage >= 65 && votePercentage < 70){
    return baseColor + " alpha='40'";}
if (votePercentage >= 70 && votePercentage < 75){
    return baseColor + " alpha='50'";}
if (votePercentage >= 75 && votePercentage < 80){
    return baseColor + " alpha='60'";}
if (votePercentage >= 80 && votePercentage < 85){
    return baseColor + " alpha='70'";}
if (votePercentage >= 85 && votePercentage < 90){
    return baseColor + " alpha='80'";}
if (votePercentage >= 90 && votePercentage < 95){
    return baseColor + " alpha='90'";}
if (votePercentage >= 95){
    return baseColor + " alpha='100'";}
else{
    return baseColor + " alpha='0'";}
}

function FindSize(totalVotes)
{
if (totalVotes < 16187){
    return "7";}
if (totalVotes >= 16187 && totalVotes < 43656.999999){
    return "9";}
if (totalVotes >= 43657 && totalVotes < 85764.999999){
    return "10";}
if (totalVotes >= 85765 && totalVotes < 143827.999999){
    return "11";}
if (totalVotes >= 143828 && totalVotes < 219676.999999){
    return "12";}
if (totalVotes >= 219677 && totalVotes < 325219.999999){
    return "13";}
if (totalVotes >= 325220 && totalVotes < 464183){
    return "15";}
if (totalVotes >= 464183 && totalVotes < 716852.999999){
    return "17";}
if (totalVotes >= 716853 && totalVotes < 1185477.999999){
    return "20";}
if (totalVotes >= 1185478 && totalVotes < 2330793){
    return "24";}
}

function FindColor(continuousValue){
if (continuousValue >= 0){
    return FindGOPColor(50 + continuousValue);}
else {
    return FindDEMColor(50 - continuousValue);}
}

function FindText(continuousValue){
if (continuousValue >= 0){
    return Text(50 + continuousValue, '#');}
else {
    return Text(50 - continuousValue, '#');}
}

"<FNT size = '" + FindSize($feature.Total_votes) + "'>" + "<CLR " +
FindColor($feature.Continuous_percent) + ">" +
FindText($feature.Continuous_percent) + "</CLR></FNT>"
```

This version of the code has an added function that applies different font sizes to different labels depending on how they are specified in the `FindSize` function. The 10 specified classes return font sizes between 7 and 24 points.

(b) In the Label Class pane, in the Expression field, select the existing code and delete it.

(c) In your text editor, copy the code from the Expressions5.txt file.

(d) In the Label Class pane, paste the code into the Expression field, validate the syntax of the code, and click Apply.
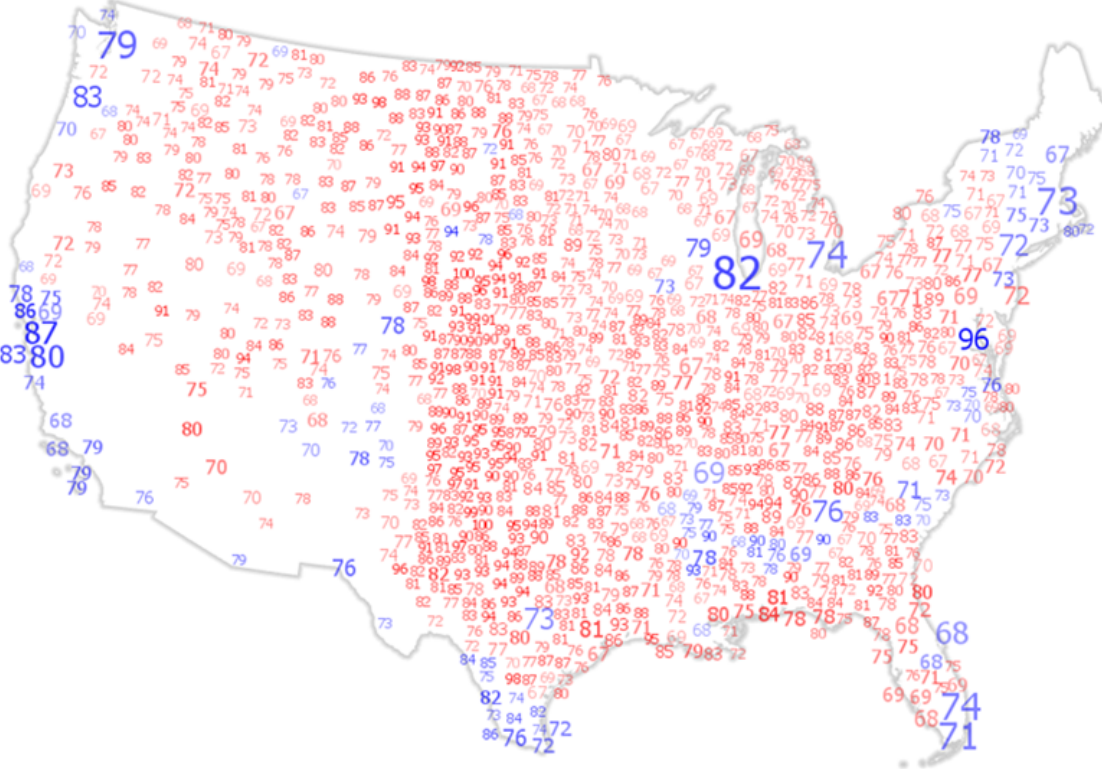
Your labels should now take on a different value of blue or red to emphasize the strength of vote using a diverging hue color scheme. They should also now differ in size based on the total number of votes cast for the candidate.

(e) Close the Label Class pane and save your project.

You have now created a multivariate proportional symbol map using labels. Hue represents the political party. Value represents the strength of the vote. Size represents actual numbers of votes. A lot of information is encoded into the symbol, and the symbol itself is literal—placing the data on the map.

Step 10: Manage visibility of labels based on scale

You have mapped labels for counties with values above 60 percent. Part of the reason for this was to assist with the speed of drawing, but there is another important cartographic reason. At small map scales, it will be impossible to show all labels on the same map because there are more than 3,000 counties and associated labels.

There are many approaches to reduce clutter, such as allowing the automated label placement engine in ArcGIS Pro (Maplex) to declutter through selective positioning. However, Maplex will omit labels due to their proximity with other nearby labels. Such a computational approach ignores the relative importance of a label cartographically. You would rather see more important labels consistently appearing at smaller scales and the rest progressively appearing as you zoom in to larger scales.

You will achieve that by using multiple layers of data, definition queries, and visibility range settings.

a In the Contents pane, ensure that the Labels > 60% layer is selected and click the Appearance tab.

b In the Visibility Range group, in the In Beyond field, type **1:6,058,572** and press Enter to set the scale.

This scale sets a value roughly midway between two of the standard web map zoom scales (which you used in the Section 3 exercise *Thematic Mapping*). It ensures that when the map is published as a web map, the map will be visible as a user settles on a particular zoom scale. It will turn off at the point when another scale-based rendering becomes visible.

c In the Out Beyond field, type **1:14,791,436** and press Enter to set the scale.
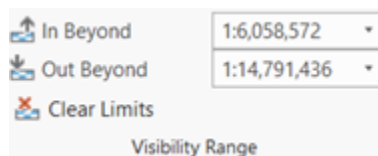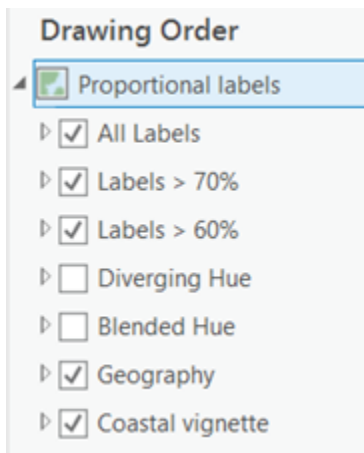

This layer will now appear only between those two zoom scales. You can zoom in and out of the map beyond the two scale extremities to see how the layer visibility changes.

Note: It may take a moment for the labels to draw.

You will now create another layer to manage visibility at even smaller scales.

d In the Contents pane, right-click the Labels > 60% layer and choose Copy.

---

e At the top of the Contents pane, right-click the Proportional Labels map item and choose Paste.

f Rename the top layer **Labels > 70%**.

g Paste another copy of the layer to the Proportional Labels map item and rename it **All Labels**.

Drawing Order

⊿ ▣ Proportional labels
  ▷ ☑ All Labels
  ▷ ☑ Labels > 70%
  ▷ ☑ Labels > 60%
  ▷ ☐ Diverging Hue
  ▷ ☐ Blended Hue
  ▷ ☑ Geography
  ▷ ☑ Coastal vignette

h In the Contents pane, select the Labels > 70% layer and click the Appearance tab, if necessary.
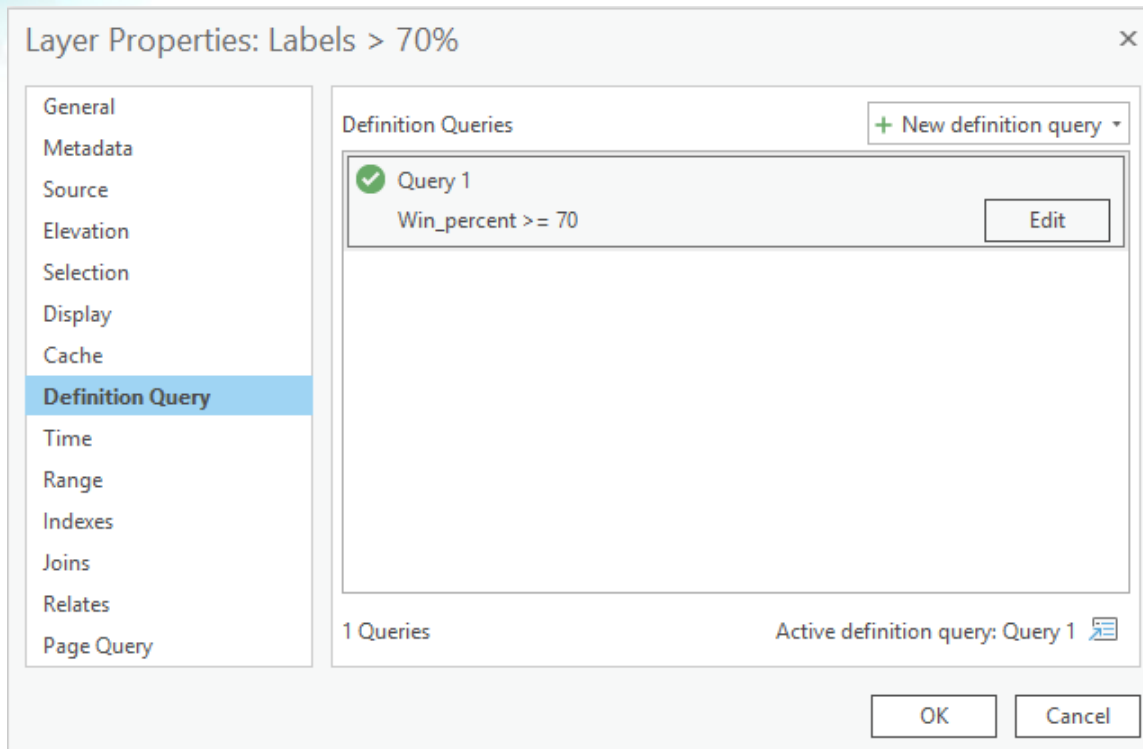
Again, you will set a zoom scale to manage visibility.

i In the Visibility Range group, for In Beyond, type **1:14,791,437** and press Enter to set the scale.

j For Out Beyond, from the drop-down list, choose <None>.

k In the Contents pane, right-click the Labels > 70% layer and choose Properties.

Next, you will edit the definition query to define which labels are visible.

l In the Layer Properties dialog box, click the Definition Query tab, if necessary, and then on the existing query, click Edit.

m Modify the existing query to the following query expression: `Where Win_percent Is Greater Than Or Equal To` **70**.

Note: If necessary, delete any extra numbers to the right of the decimal place so that you are left with only the integer 70.
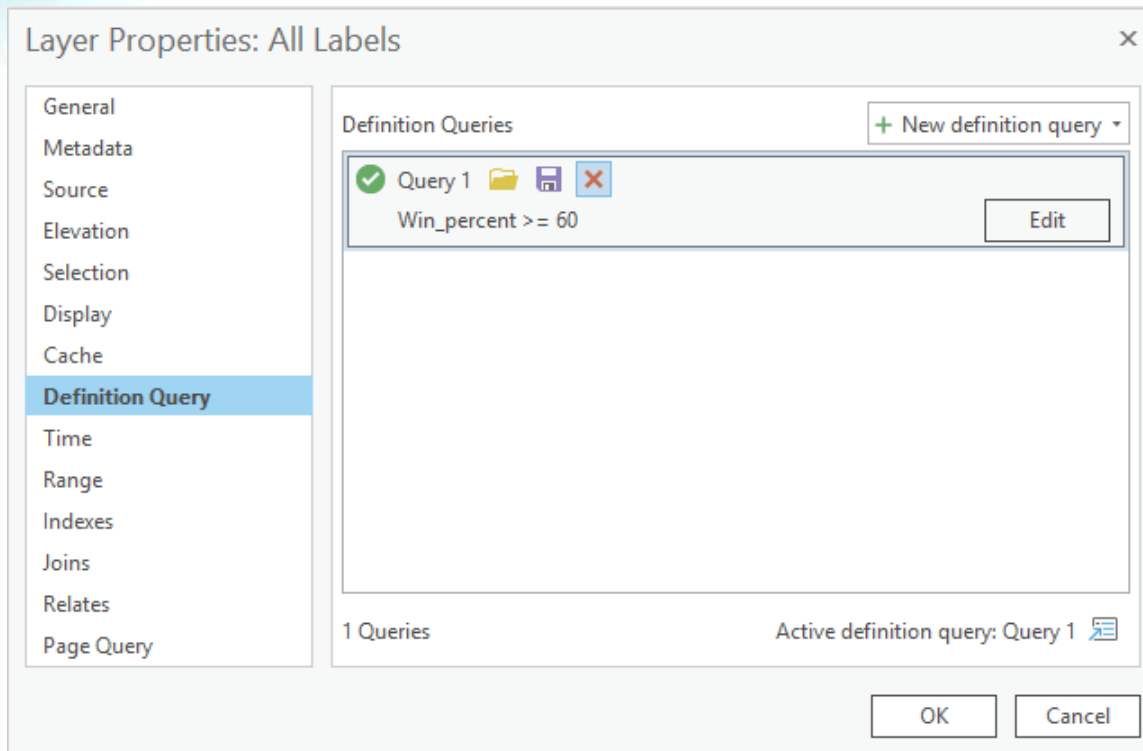
n Click Apply.

Layer Properties: Labels > 70%   ✕

General	Definition Queries
Metadata	
Source	
Elevation	
Selection	
Display	
Cache	
Definition Query	
Time	
Range	
Indexes	
Joins	
Relates	
Page Query	

+ New definition query ▾

✓ Query 1
   Win_percent >= 70   Edit

1 Queries   Active definition query: Query 1

                                                        OK   Cancel

**o** Click OK to close the Layer Properties dialog box.

The Labels > 70% layer now draws only when the scale is 1:14,791,437 or smaller and draws labels only for values of the percentage share of votes greater than 70%. Now you will set visibility for the All Labels layer.

p In the Contents pane, select the All Labels layer and click the Appearance tab, if necessary.

q In the Visibility Range group, for In Beyond, choose <None>.

r For Out Beyond, type **1:6,058,573** and press Enter to set the scale.

s In the Contents pane, double-click the All Labels layer to open the Layer Properties dialog box.

t From the Definition Query tab, point to the existing query and click the red X to remove the existing `Win_percent >= 60` query.

Layer Properties: All Labels

General
Metadata
Source
Elevation
Selection
Display
Cache
Definition Query
Time
Range
Indexes
Joins
Relates
Page Query

Definition Queries   + New definition query ▾

✓ Query 1 📁 💾 ✕
   Win_percent >= 60   Edit

1 Queries   Active definition query: Query 1

                                                    OK   Cancel

(u) If the Remove Definition Query dialog box appears, click Yes to verify that you want to delete the definition query from the layer.

(v) Click OK.

The All Labels layer now draws only when the scale is 1:6,058,573 or larger. Because the map is at a scale sufficient enough to allow enough space for all labels, all labels are drawn.
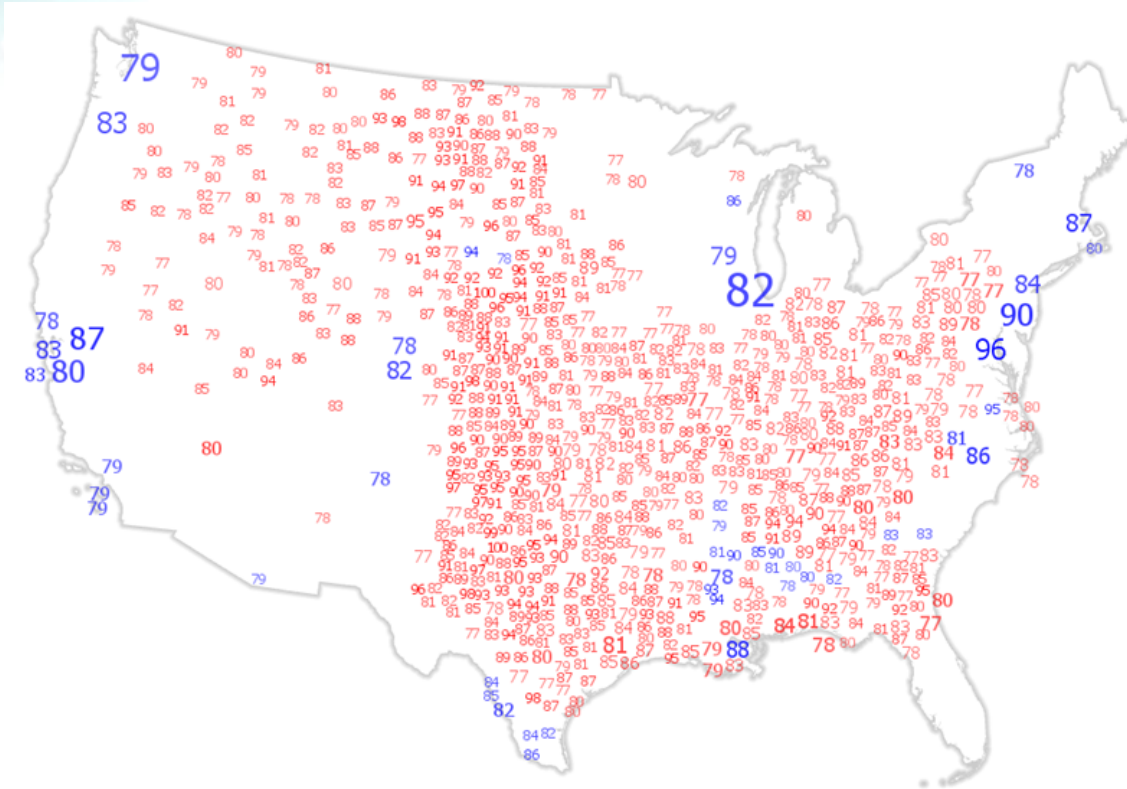
You are using visibility ranges to change the amount of detail on the map and to make the data legible at different scales. You can use this approach whenever you need to have some control over the scales at which information appears on the map.

(w) Zoom in and out of the map to see how the layers turn on and off at different times.

(x) When you are finished, save your project.

Step 11: Publish a tile layer

From the many ways to publish a map, you may prefer to make a layout and print it. You can use any of the techniques that you learned in the Section 1 exercise *Make a Map* to create a layout.

In this step, you will explore a way to publish the map to your ArcGIS Online account as a web map using the technique that you may have tried as a stretch goal in the Section 3 exercise *Thematic Mapping*.

When you have a lot of complex detail in the map, a very efficient way to publish is to create cached raster tiles. This method works well for this kind of map, which has a lot of complex

symbology set up using Arcade Expressions. You are effectively going to take a picture of your map and publish the resulting tiles. It will draw far more rapidly as a web map than alternative options.

a From the Share tab, in the Share As group, click the Web Layer button (not the down arrow) to open the Share As Web Layer pane.
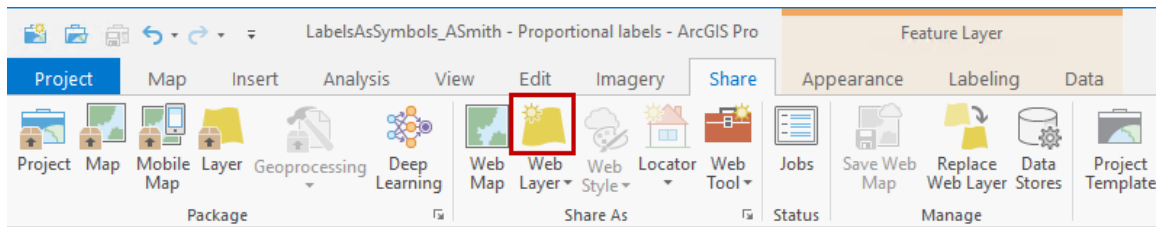
b In the Share As Web Layer pane, from the General tab, confirm that the Name field is set to Proportional Labels.

c Add your first and last name to the end of the file name so that it is unique.

Note: Because all students in this class are creating a web layer for this exercise, the name of your web layer must be unique to avoid overwriting other people's work. Make a note of the name that you typed because you will need it to find the layer in ArcGIS Online.

Next, you will add a summary item description and tags to help you and others find this layer, either in your organization or the general public, depending on how you want it shared. Tags also help group different types of items together, such as maps, map layers, or applications, so they can easily be searched.

d For Summary, type **Cached tile layer for Section 4: Labels as Symbols**.

e For Tags, type **2016 US election** and press Enter.

f If you would like, add additional tags, pressing Enter after each one.

g For Layer Type, select Tile.

h For Location, click the down arrow and choose your Username.cart folder.

i For Share With, check the Everyone and Cartography. boxes, if necessary.

Share As Web Layer   ? ▾ � ✕

Sharing Proportional labels As A Web Layer

General Configuration Content

Item Details

Name

Proportional Labels Your First and Last Name

Summary

Cached tile layer for Section 4: Labels as Symbols.

Tags

2016 US Election ✕

Layer Type ⓘ

○ Feature
◉ Tile
○ Vector Tile

Location

Folder

Your ArcGIS Username.cart ▾ 📁

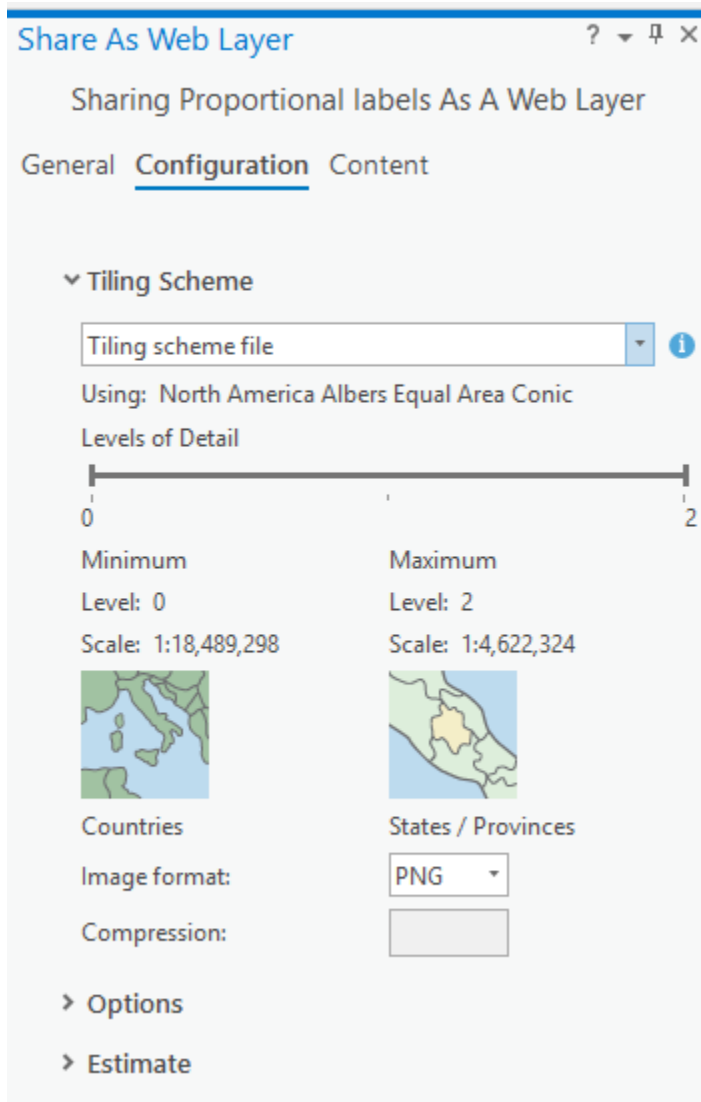**Share with**

☑ Everyone
☑ Cartography.
Groups ▾

ⓙ At the top of the Share As Web Layer pane, click the Configuration tab.

Next, you will import a custom tiling scheme XML file that contains your preferred scales and settings.
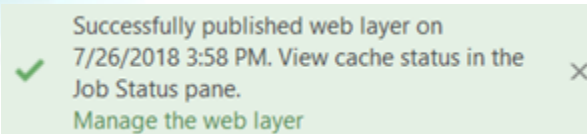
ⓚ For Tiling Scheme, choose Tiling Scheme File.

l   In the Load Tiling Scheme dialog box, browse to the location where you saved the exercise data files.

m   Select the Proportional_Text_Tiling_Scheme.xml file and click OK.

Share As Web Layer   ? ▾ ⊼ ✕

Sharing Proportional labels As A Web Layer

General **Configuration** Content

⌄ Tiling Scheme

Tiling scheme file   ▾ ⓘ

Using: North America Albers Equal Area Conic

Levels of Detail

0   2

Minimum	Maximum
Level: 0	Level: 2
Scale: 1:18,489,298	Scale: 1:4,622,324

Countries   States / Provinces

Image format: PNG ▾

Compression:

❯ Options

❯ Estimate

n   In the Share As Web Layer pane, leave all other settings as the default, and then click Publish.

The process may take a few minutes. When the layer is successfully published, you will see a green message at the bottom of the Share As Web Layer pane.

Successfully published web layer on
7/26/2018 3:58 PM. View cache status in the
Job Status pane.
Manage the web layer

This process publishes a cached raster map at three scales (in the Albers Equal Area Conic projection) to your ArcGIS Online account. We do not want to use Web Mercator because it is inappropriate for a thematic map of a large area like the United States (where equal area is important for interpretation). These projection and scale settings are specified in the XML file that ArcGIS Pro uses to publish.

In the Section 3 exercise *Thematic Mapping* stretch goal, you may have made one of these XML files yourself, but here, it is provided for you. If you wanted to modify the parameters, you could go back to the Section 3 exercise *Thematic Mapping* and recall how you made one of these XML specification files.

Step 12: Add a tile layer to a web map

Next, you will sign in to your ArcGIS Online account and add the tile layer to a map.

ⓐ Open a new private or incognito web browser and go to www.arcgis.com.

ⓑ Sign in to ArcGIS Online using your provided course ArcGIS credentials (user name ending with _cart), as explained in the Section 1 exercise *Make a Map*.

Hint: In ArcGIS Pro, you can find the ArcGIS user name that you are currently using by clicking on your name in the top-right corner of the ArcGIS Pro window. You will use the same user name to sign in to ArcGIS Online.

ⓒ If necessary, at the top of the page, click Home to go to the ArcGIS Online organization home page for this course.

ⓓ At the top of the organization home page, click Map to open a new map.
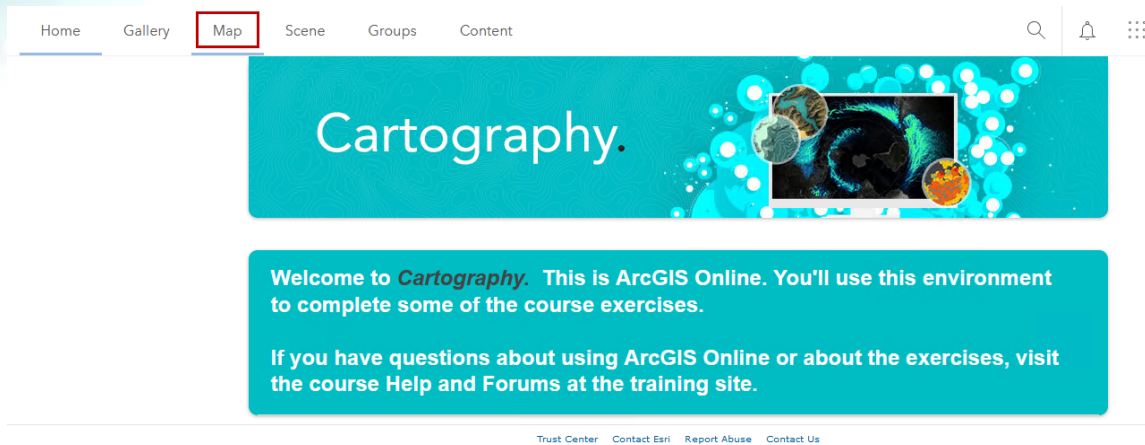
e Click Add and choose Search For Layers.
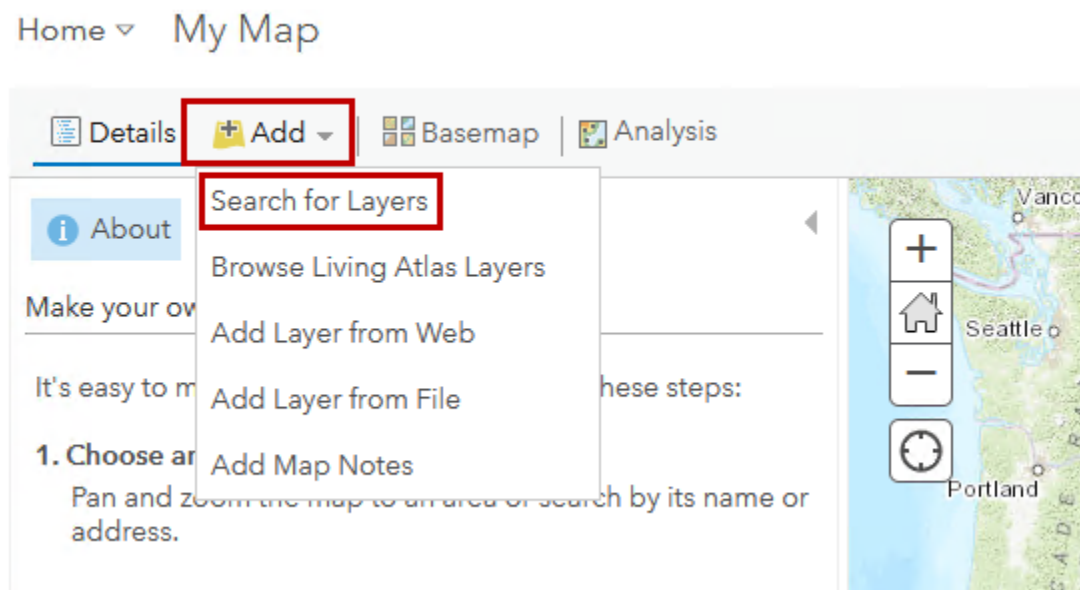
f In the Search For Layers pane, type the name of your tile layer (for example, **Proportional labels_<your first and last name>**) and press Enter to search for your newly created cached tiles layer.

You should see one result in the search result list.

Note: Avoid adding the layer to the map because it is in a different projection.

g Click the layer name, and then click Use As Basemap.

You will save your web map in ArcGIS Online before exiting.

h At the top of the ArcGIS Online map window, click Save and choose Save As.

i In the Save Map dialog box, give your web map a title, tags, and a summary.

Save Map

Title:	My Election Basemap
Tags:	election ×	basemap ×	MOOC ×
	Add tag(s)
Summary:	A basemap for the Cartography. MOOC Sec4
Save in folder:	JohnSmith_cart

SAVE MAP CANCEL

j Click Save Map.

From here, you can create pop-ups and publish your web map in a web map app (see the stretch goals).

k When you are finished, sign out of ArcGIS Online.

Hint: In the upper-right corner of the ArcGIS Online window, click your name and choose Sign Out.

l Close the incognito web browser window.

m Return to ArcGIS Pro and save your project.

n If you are continuing to the stretch goals, leave ArcGIS Pro open; otherwise, exit ArcGIS Pro.

Conclusion

You have made a Proportional Labels map using some code built with Arcade Expressions. This technique introduces you to a powerful way to modify symbols and labels beyond the defaults and the options in the various symbology and labeling panes.

Although this course is not about teaching you how to code, learning a little about how to create expressions can greatly expand your ability to modify the rendering of your map data. It allows you to go beyond the defaults of what is offered out-of-the-box.

Finally, you have published your map as a web map. If you follow the stretch goals, you will also make an app in ArcGIS Online that includes pop-ups to give people access to meaningful results.

Stretch goals

One of the primary purposes of cartography is to effectively communicate information to a wider audience. The most beautiful map ever created is not useful if no one ever sees it. One way that you can share your maps is using ArcGIS Online and ArcGIS Web AppBuilder to create a web app that anyone can see and interact with. If you would like to continue to explore how to use the map you created to communicate more effectively using pop-ups and share your map as an app, you can complete the following high-level tasks:

1. Create pop-up information in ArcGIS Pro (https://bit.ly/36Es4Cm) that provides details about the election results when you click the map.
2. Publish your web map as an app using Web AppBuilder (https://bit.ly/2GRoqdy).

   - Create a cleaner version of your web map suitable for sharing with others; remember that one of the key ideas behind cartography is to omit detail to clarify the message.
   - See how minimal you can make the web map app's layout.
   - Because the labels are actually on the map, consider adding a small information box to explain what it is that people are seeing.

Hint: Review Section 3 Exercise 4: Thematic Mapping for some details.

Use the Lesson Forum to post your questions and observations. Be sure to include **#stretch** hashtag in the posting title. Remember, stretch goals are community supported and a great opportunity to work together to learn together.

If you did some experimenting on your own beyond the steps of an exercise, we would love to hear about and see what you did! If you completed a map based on the exercise steps, that is awesome! However, there is no need to share it in the Forum.