Arcade Expression Crash Course with COVID-19 data

Written for ArcGIS Pro 2.5

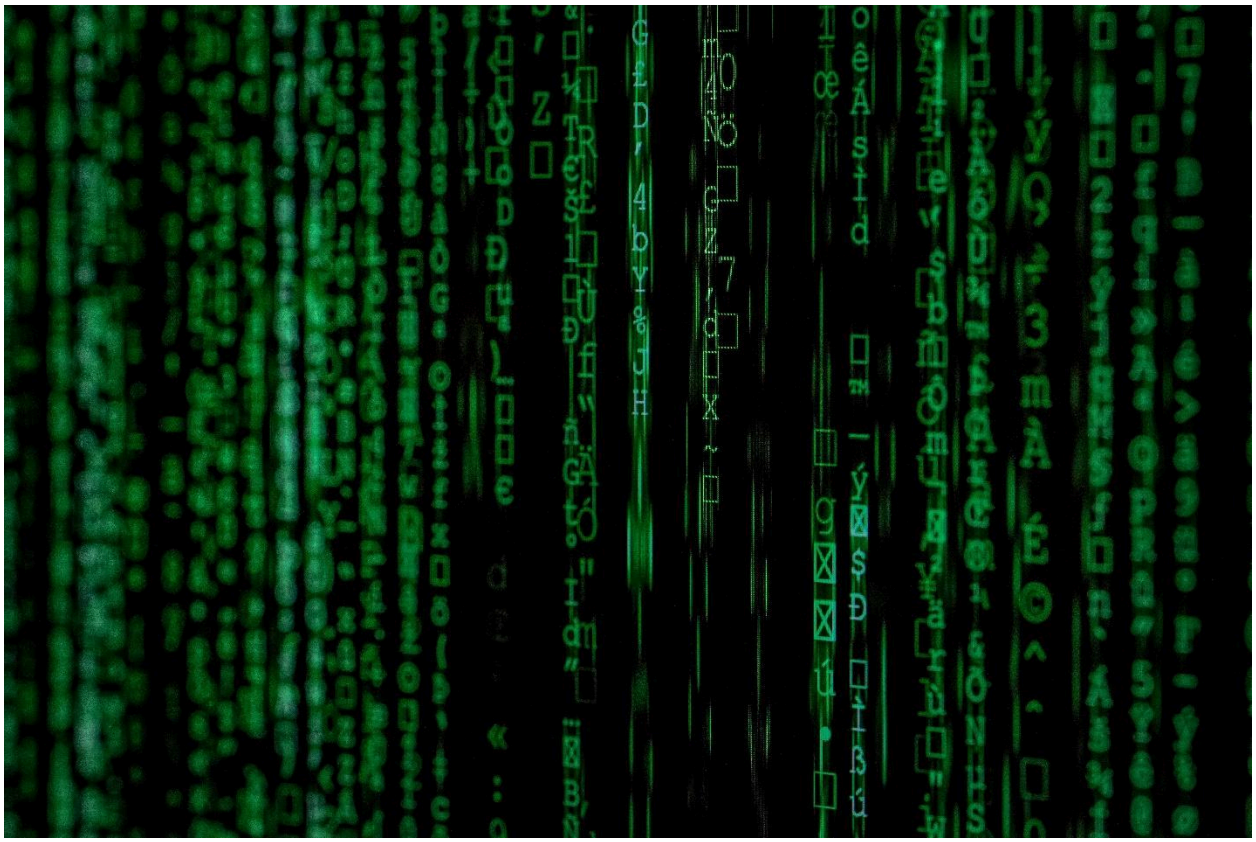


Photo by Markus Spiske on Unsplash

Written by:

Wing Cheung

Palomar College

Arcade Expression Crash Course with COVID-19 data

Written for ArcGIS Pro 2.5

Wing Cheung, Palomar College

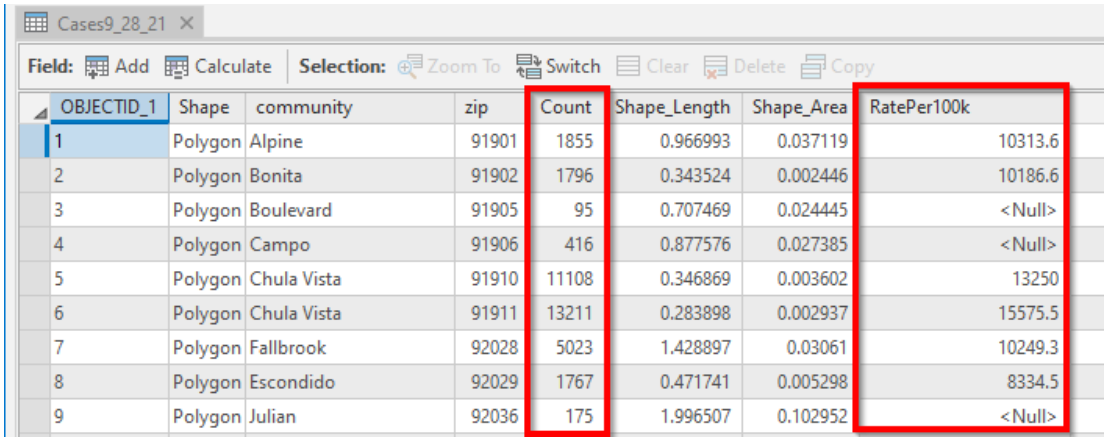
Contents

Download and unzip the required data at https://bit.ly/arcadeex

Part 1: Configuring pop-ups and labels in ArcGIS Pro with Arcade expressions

1. Open ArcGIS Pro. Start a new project and new map. Add the **Cases9_28_21** feature class from the CasesTutorial.gdb geodatabase to your map. This feature class was prepared by the County of San Diego's Health and Human Services Agency, and it shows a summary of COVID-19 cases by zip code of residence in the county as of 9/28/2021.

2. Open the attribute table for the Cases9_28_21 feature class, you should see 8 fields present. In particular, the two fields we will focus on in this exercise are **[Count]** and **[RatePer100k]**, they respectively tell us the number of COVID cases at each zip code and the case rate (i.e. $\frac{number\ of\ COVID\ cases}{population\ at\ zip\ code} * 100,000$) at each San Diego County zip code on 9/28/2021. Don't be alarmed by the <Null> values for some of the zip codes in the RatePer100k field, this is because "rates are not calculated for counts under 5 cases or populations under 10,000" according to source of this data (i.e. County of San Diego).


Cases9_28_21

Field: Add Calculate Selection: Zoom To Switch Clear Delete Copy

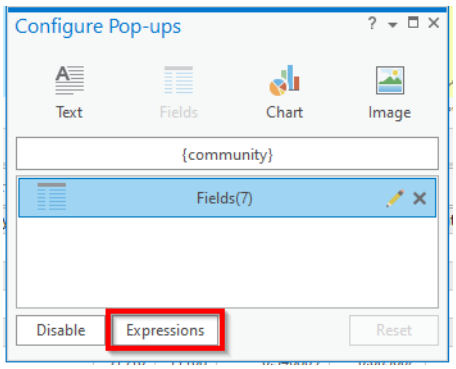
OBJECTID_1	Shape	community	zip	Count	Shape_Length	Shape_Area	RatePer100k
1	Polygon	Alpine	91901	1855	0.966993	0.037119	10313.6
2	Polygon	Bonita	91902	1796	0.343524	0.002446	10186.6
3	Polygon	Boulevard	91905	95	0.707469	0.024445	<Null>
4	Polygon	Campo	91906	416	0.877576	0.027385	<Null>
5	Polygon	Chula Vista	91910	11108	0.346869	0.003602	13250
6	Polygon	Chula Vista	91911	13211	0.283898	0.002937	15575.5
7	Polygon	Fallbrook	92028	5023	1.428897	0.03061	10249.3
8	Polygon	Escondido	92029	1767	0.471741	0.005298	8334.5
9	Polygon	Julian	92036	175	1.996507	0.102952	<Null>

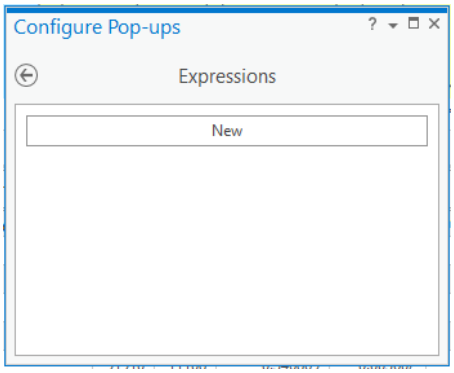
3. Click on any of the polygons in the map. You should see a pop-up window with almost all of the fields that you previously saw in the attribute table (except the Shape field) presented in a slightly different format. Close the attribute table and the pop-up window.

Part 1.1 Creating a new field with Arcade expression

4. In the Contents pane, right click on Cases9_28_21, and select **Configure Pop-ups**.

   Click the **Expression** icon at the bottom of the window, and click **New** in the Expressions window.



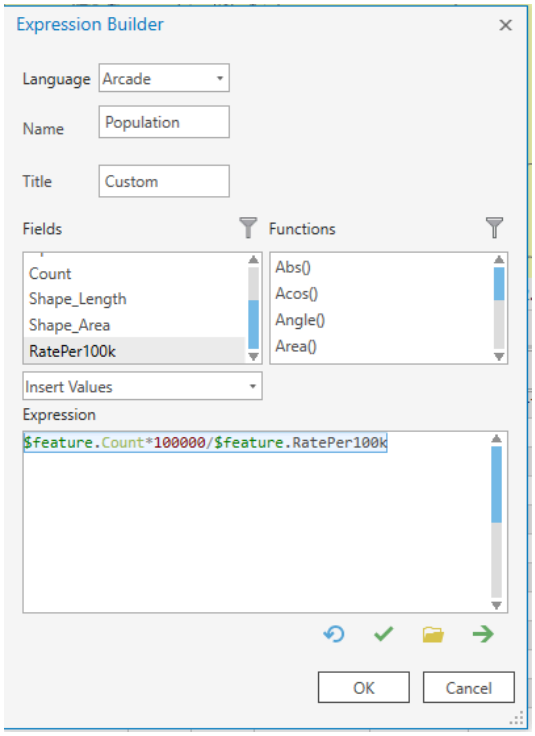


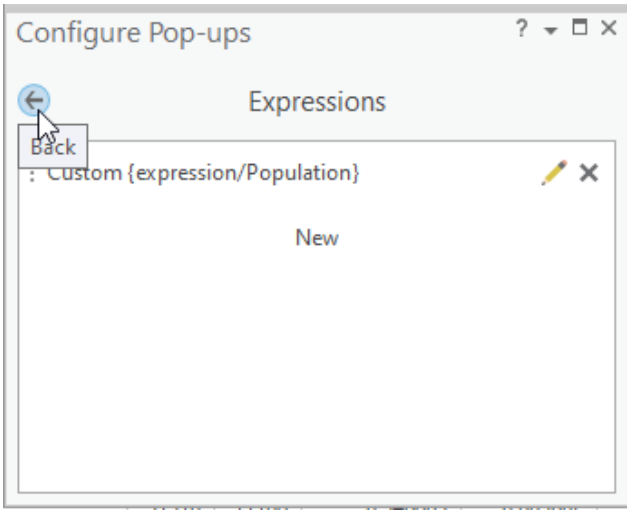
5. We will create an expression to work backward, and calculate on-the-fly the approximate population at each zip code based on the information in the **[Count]** and **[RatePer100k]** fields.

   In the Expression Builder window, change the **Name** to **Population.** And for **Expression**, enter the following:

```
$feature.Count*100000/$feature.RatePer100k
```

   Click **OK**. You should now see the Population expression in the Expressions window. Click the **back arrow** to return to the Configure Pop-ups window.



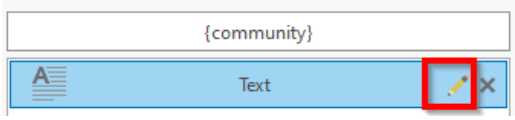


Part 1.2 Customizing pop-up messages with Arcade expression

6. In the Configure Pop-ups window, click on the **Text** icon 

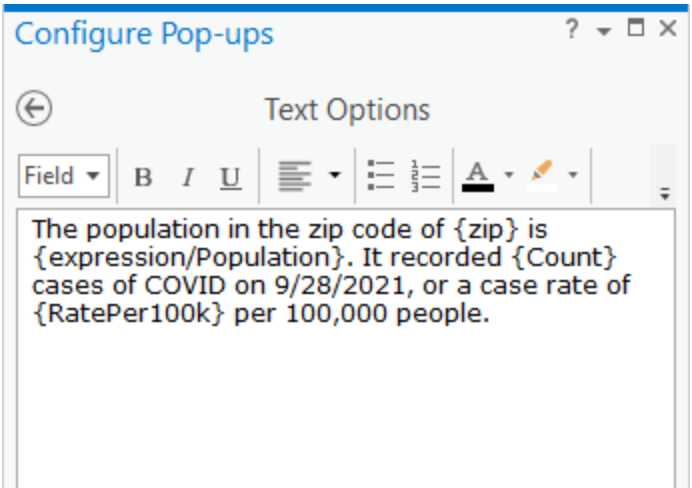
to customize the message that will be shown in the pop-up window.

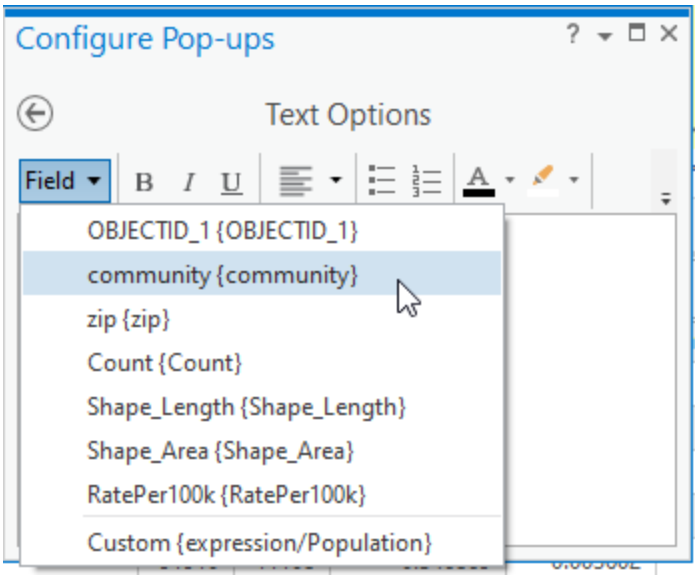
7. Now click the **pencil** icon to the right of the word Text to customize the message.



8. In the Text Options window, enter the following text message. Be sure to take a look at the screenshots below to enhance your understanding of the directions.

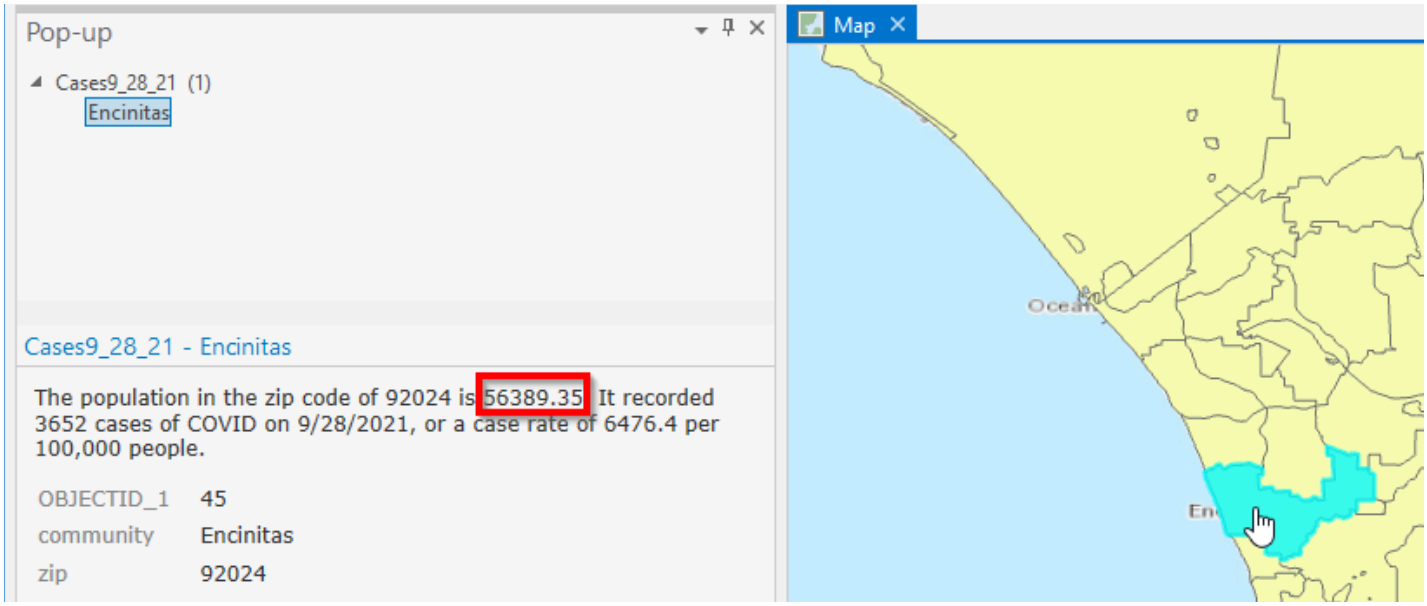
   The population in the zip code of *(select **zip** from the Field dropdown)* is *(select **Custom** from the Field dropdown)*. It recorded *(select **Count** from the Field dropdown)* cases of COVID on 9/28/2021, for a case rate of *(select **RatePer100k** from the Field dropdown)* per 100,000 people.



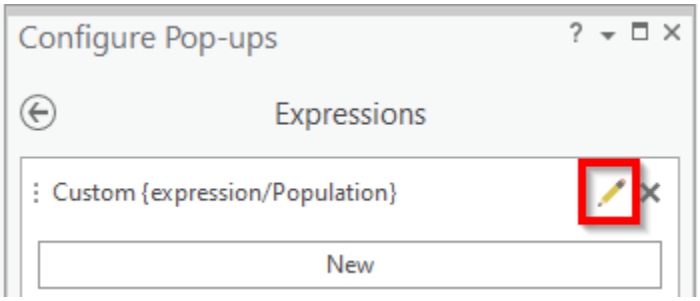


Click the **back arrow** after entering the message. Do not close the Configure Pop-ups window.

9. Click on any feature in the map to see its pop-up with your newly customized message. While the message may be displaying properly, the population has some decimal/fractional values that should be rounded. We will correct this in the next step.



10. Go back to the Configure Pop-ups window, and click on **Expression** as you did in step 4.
11. In the Expressions window, click on the pencil icon next to the Custom expression that we created earlier.

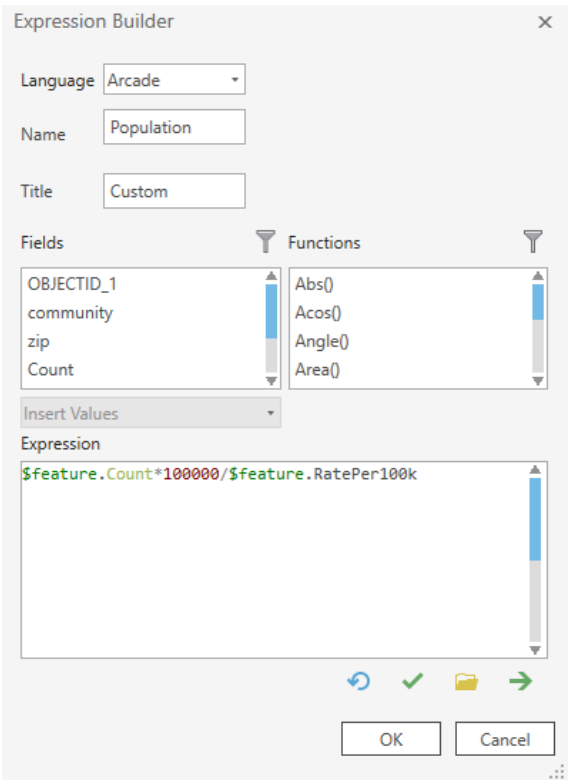


12. In the Expression Builder window, within the Expression section, we will use the Ceil function to round our population value. Enclose the existing expression with the Ceil function using parentheses as shown below. The new expression should read:

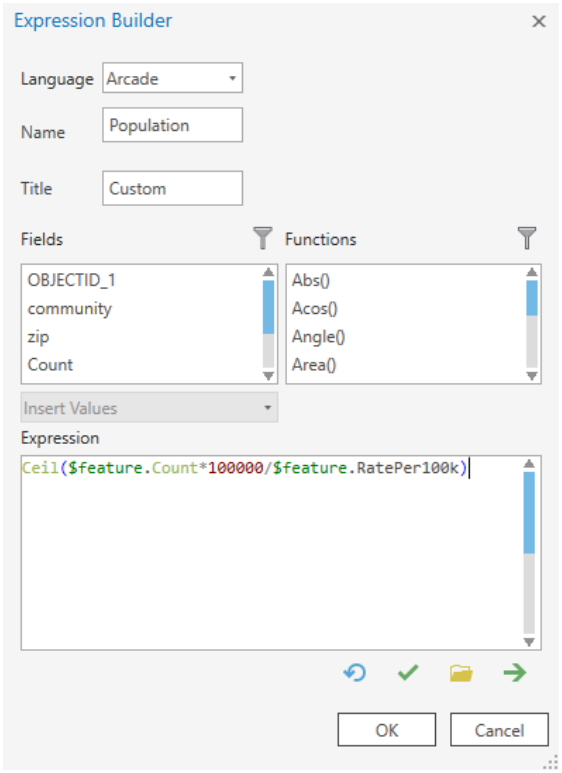
    ```
    Ceil($feature.Count*100000/$feature.RatePer100k)
    ```

    Click **OK**. To learn more about other functions in Arcade, see https://developers.arcgis.com/arcade/function-reference/

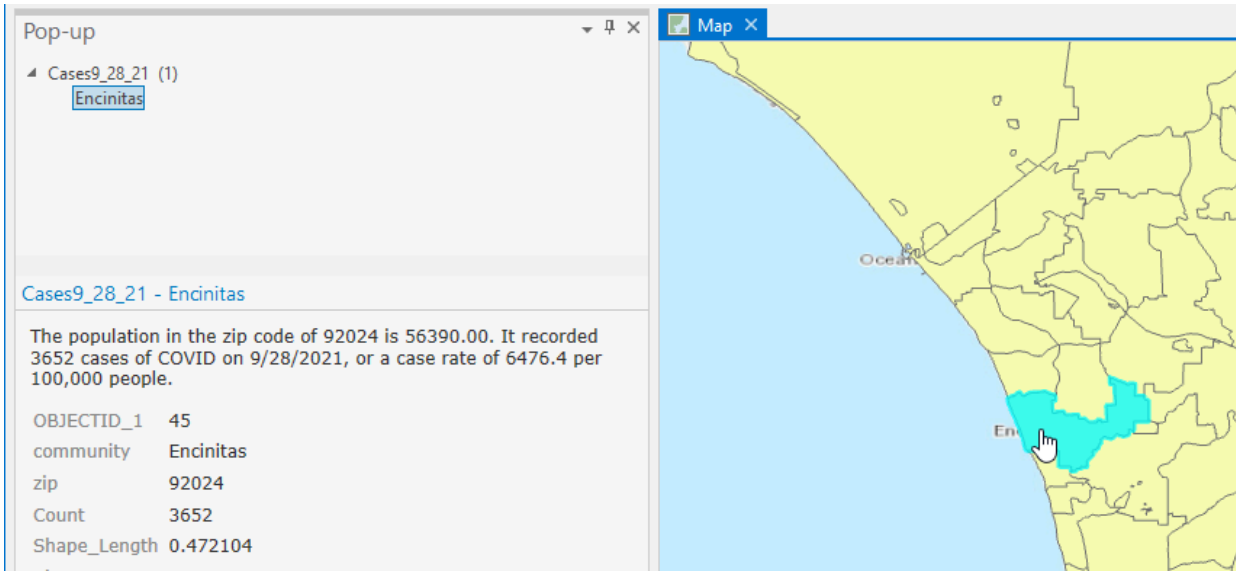
Before:



After:



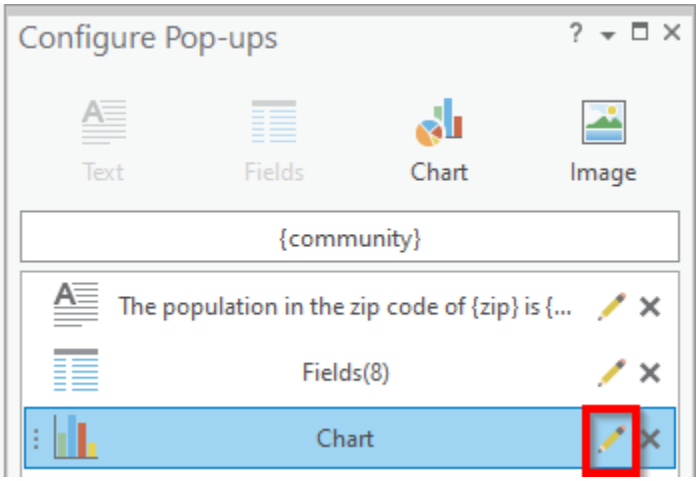
13. Click the **back arrow** in the Configure Pop-ups window. Click on any of the features in your map to see the new customized message. Notice that the population value is now rounded to the nearest whole number.



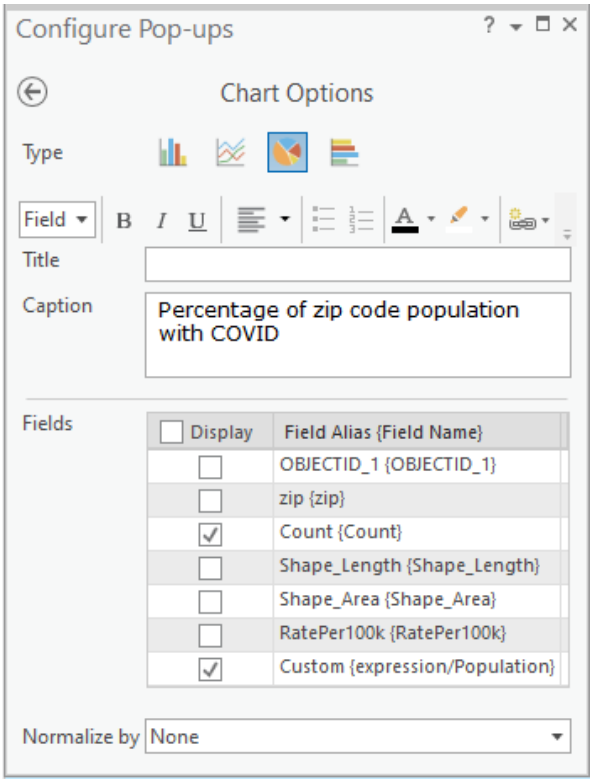
Part 1.3 Adding charts and customizing pop-up contents

14. We will now add a pie chart in the pop-up window to show the percentage of each zip code's population with COVID and without COVID. Click the **Chart** icon 

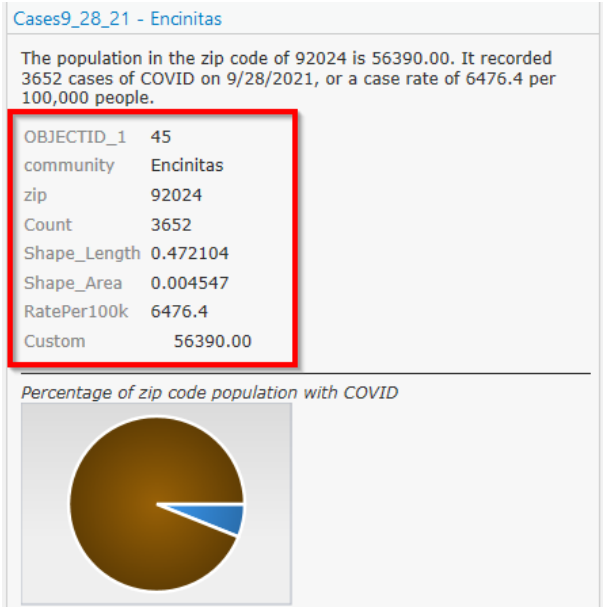
in the Configure Pop-ups window, and click the **pencil** icon next to the word Chart.



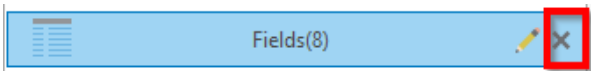
15. In the Chart Options window, click the **pie chart** icon  . Delete the word Chart in the Title. Change the Caption to **Percentage of zip code population with COVID as of 9/28/21**. Select **Count** and **Custom** in the Fields section, then click the **back** arrow.



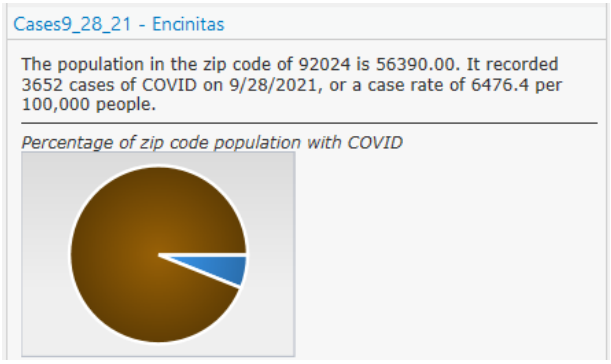
16. Click on any of the features in your map to see the new pop-up window with your customized message and the pie chart. Notice that there are some attribute values in the pop-up window that now seem a little redundant, since we already covered them in our customized message.



17. In the Configure Pop-ups window, click on the **X** icon to the right of the word Field(8) to hide those attribute values in your pop-ups.

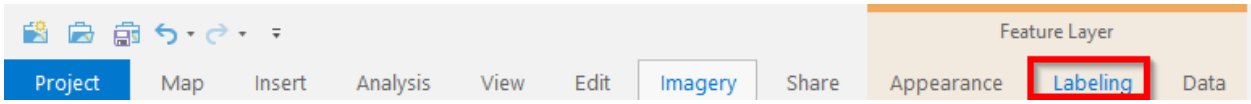


18. Click on any of the features in your map, you should now see the new pop-up window with your customized message and the pie chart without the redundant attribute information. Close the Configure Pop-ups window as well as any pop-up window.



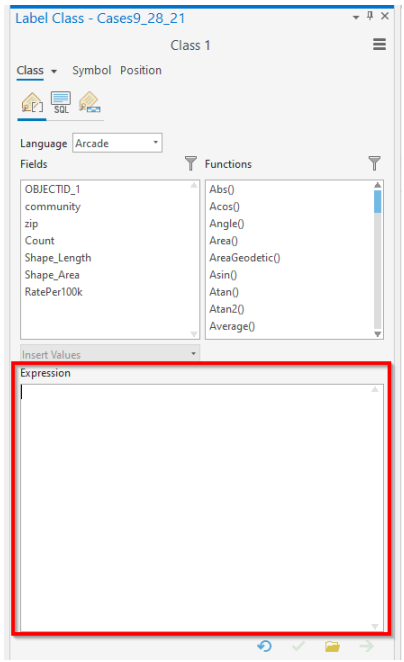
Part 1.4 Customizing labels with Arcade expression

19. To create a custom label that shows the name and zip code information for each polygon feature in the map, select the Cases9_28_21 layer in the Contents pane, and click the **Labeling** tab in the main menu.



20. Within the Labeling tab, click **Label** to turn on the labels for your features. And then click the **Expression** Icon to open the Expression pane.

21. Delete the existing values in the Expression area in the middle of the Label Class pane.



22. We will first create a variable called Name, where we will use the replace function to replace the community name of 'San Diego' (found in the attribute table) with 'City of San Diego' in our labels. This allows us to clarify that these areas are parts of the City of San Diego, rather than parts of the County of San Diego's unincorporated areas. In the now empty Expression area, enter the following:

```
var Name = Replace($feature.community, 'San Diego', 'City of San Diego')
```

Tip: Keep in mind that field names are shown as $feature.xxxxx in our expression, where xxxxx is the name of the field in the attribute table. You can insert those field names into your expression by double clicking on the field name in the Fields area in the Label Class pane.

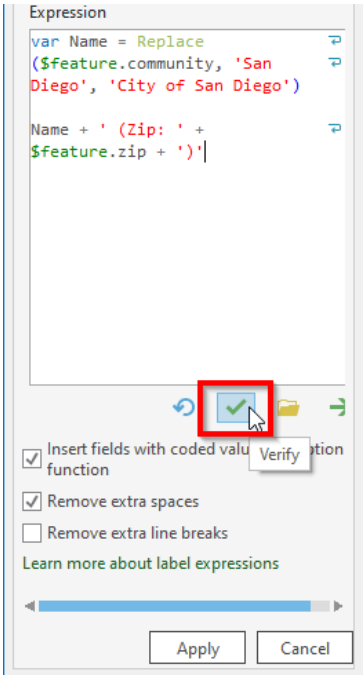
23. In the next line within the Expression area, we will format how the labels should appear in our map. Specifically, we want to display each polygon's zip code as well as the name of the community or city that it is in. Enter the following:

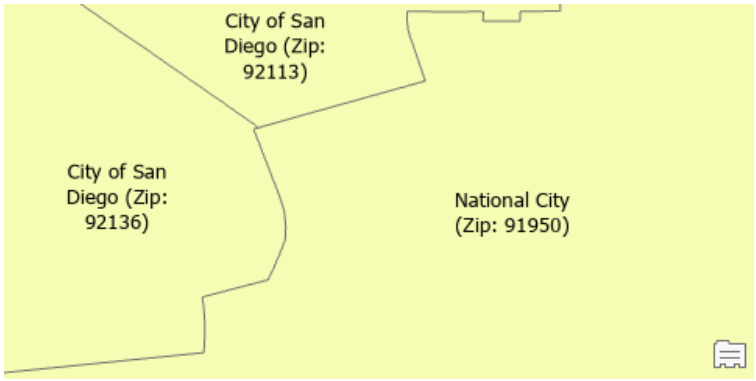
```
Name + ' (Zip: ' + $feature.zip + ')'
```

Tip: All of the text within the single quotes are custom text and characters (e.g. :, (), space) that you are adding to the label. Be sure to add in all of the characters (including spaces) within the single quotes in order to ensure that your new labels are formatted properly.

24. When you have finished entering the two lines of expressions in the Expression area, click the **green check**  to validate your expression, and click **Apply** to see the result in your map.

Tip: If you are not seeing any labels in your map, you need to turn labels on (see step 20). You may also need to zoom in and out on your map to see the labels refreshed.



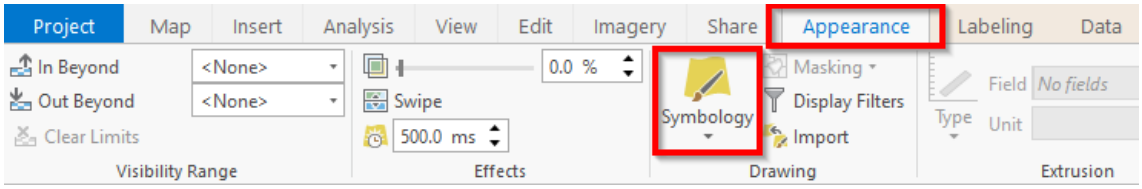


25. **Save** your ArcGIS Pro project and close the application.

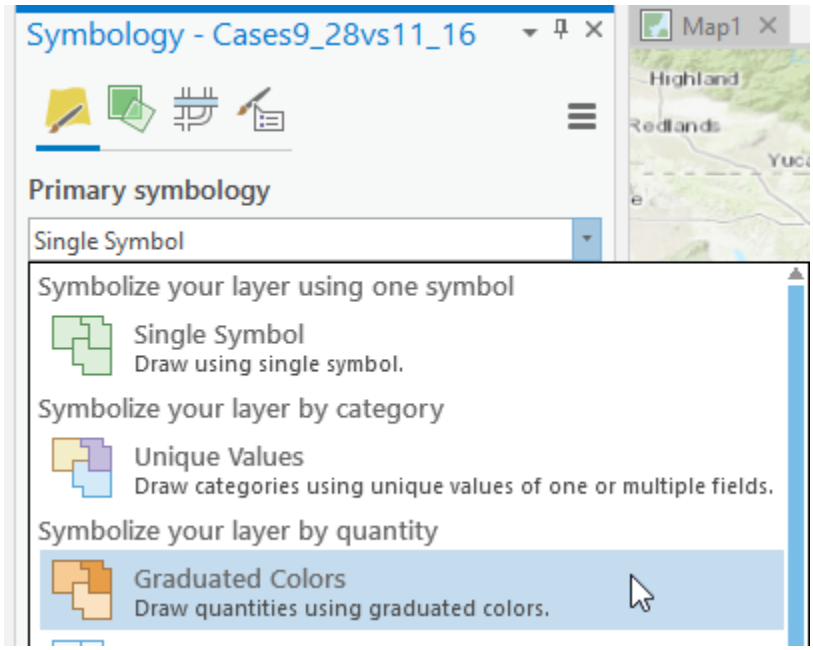
Part 2: Configuring symbology in ArcGIS Pro with Arcade expressions

1. Open ArcGIS Pro. Start a new project and new map. Add the **Cases9_28vs11_16** feature class from the CasesTutorial.gdb geodatabase to your map. The source of the data used to construct this feature class is the County of San Diego's Health and Human Services Agency. The feature class compares the number of reported COVID-19 cases by zip code of residence between 9/28/2021 and 11/16/2021.

2. Select the Cases9_28vs11_16 layer in the Contents pane, click the **Appearance** tab in the main menu, and select **Symbology**.



3. In the Symbology window, select **Graduated Colors** under the Primary symbology dropdown.

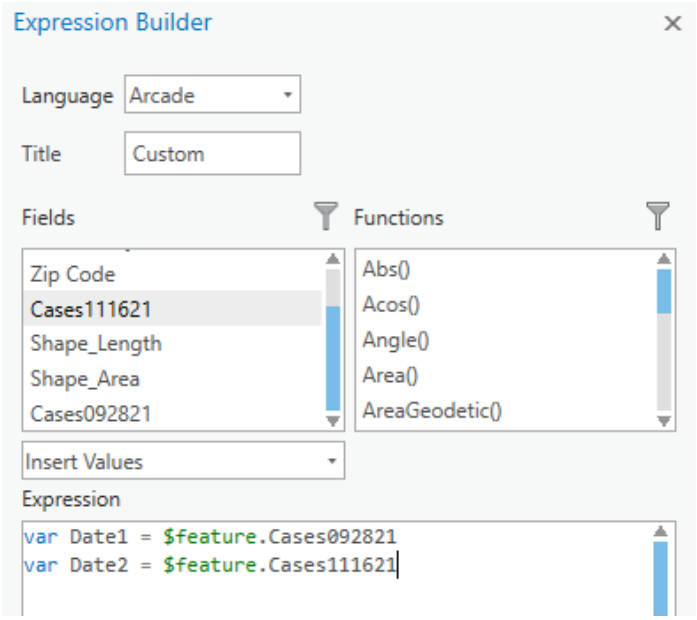


Part 2.1 Creating an expression to set symbology

4. Click the **Set an expression** icon  within the Symbology window.

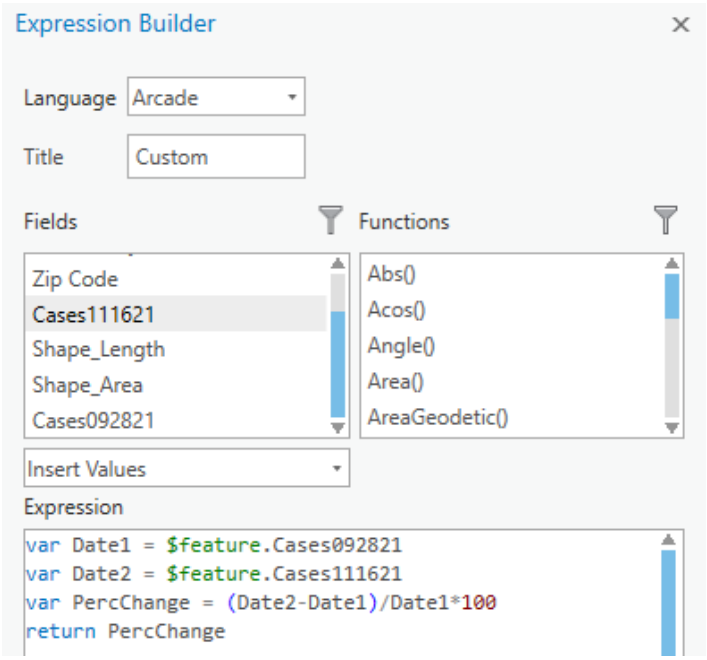
5. We will first create two variables to represent the COVID case count on 9-28-2021 (Date1) and 11-16-2021 (Date2). Within the Expression Builder window, in the Expression box, type in the following expressions:

```
var Date1 = $feature.Cases092821
var Date2 = $feature.Cases111621
```



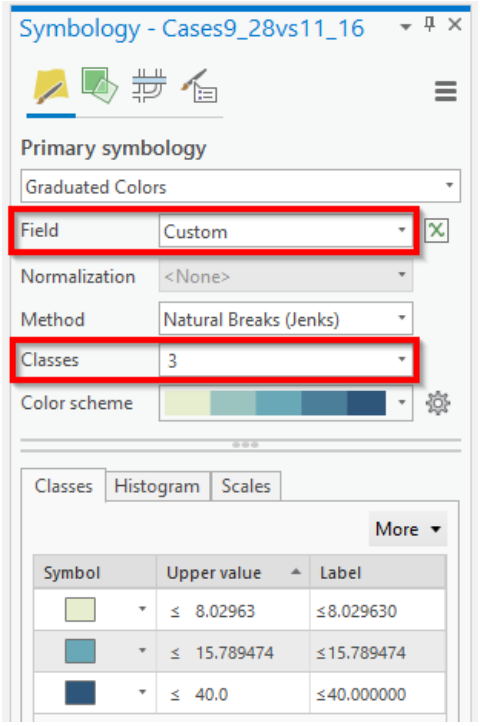
6. We will create an additional variable to represent the percent change (PercChange) in COVID cases between 9-28-2021 (Date1) and 11-16-2021 (Date2). Within the Expression Builder window, in the Expression box, type in the following expressions to calculate the percent change and print its value:

```
var PercChange = (Date2-Date1)/Date1*100
return PercChange
```

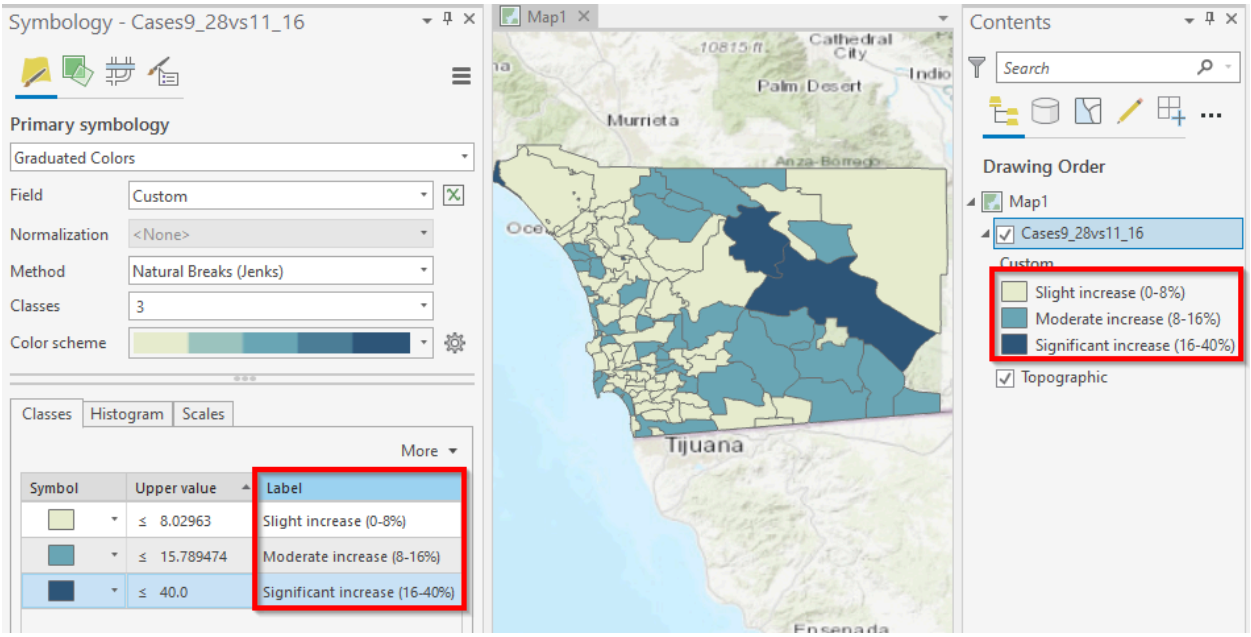


Click the **Verify** icon ✔ to check that your expressions have been entered properly. Click **OK**.

7. In the Symbology window, you should now see that our **Custom** expression is providing the values used to color our features in the map. Change the **number of classes** to **3.**

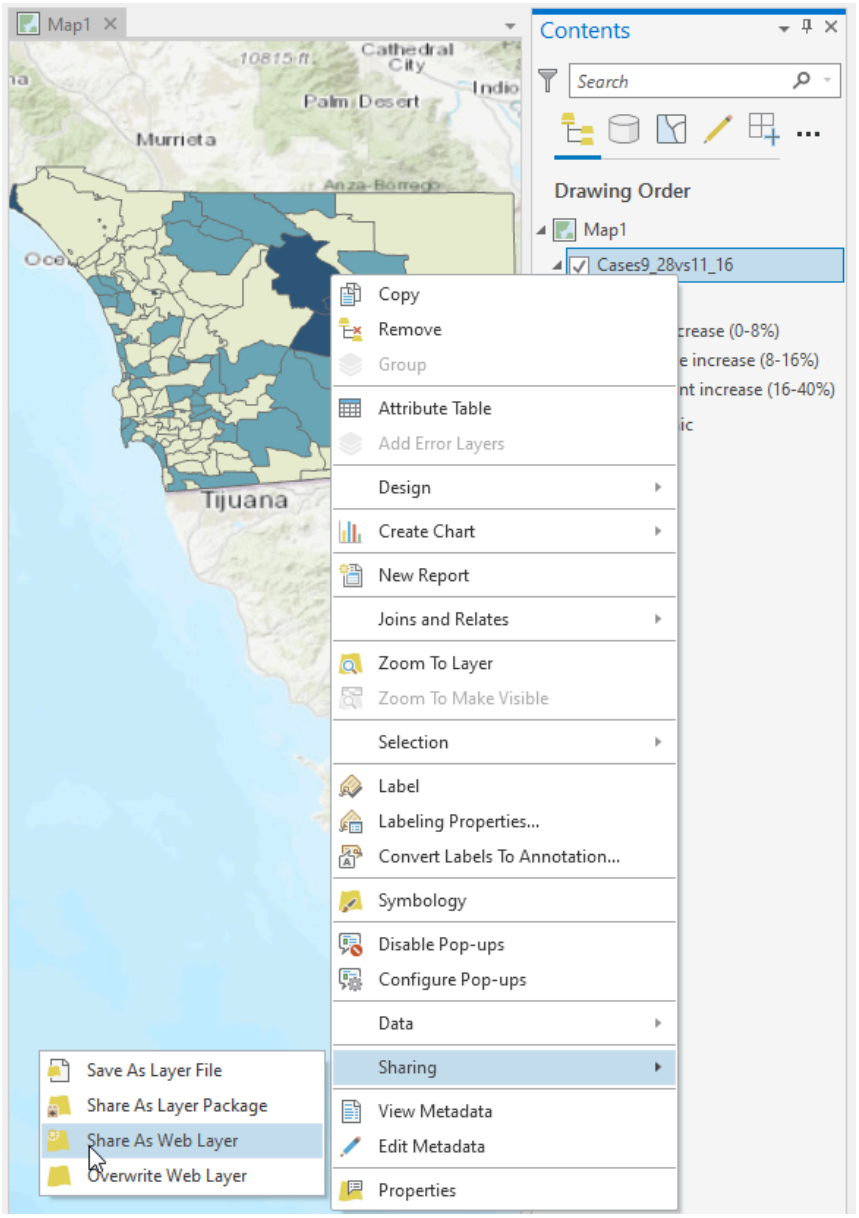


8. In the Symbology window, you will change the **labels** of the classes by single clicking on the existing values under the Label column. The class with the lowest percentage increase (i.e. ≤8.029630) will be relabeled "**Slight increase (0-8%)**", the class with the highest percentage increase (i.e. ≤40.000000) will be relabeled "**Significant increase (16-40%)**", and the remaining class will be relabeled "**Moderate increase (8-16%)**". You should now be able to see the updated class labels in the Symbology window as well as the Contents pane.



Part 2.2 Publishing a web layer that included an Arcade expression

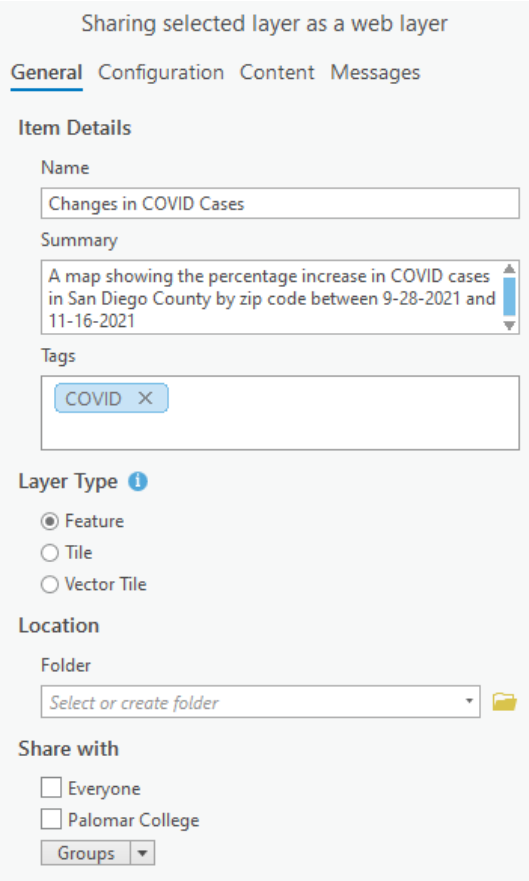
9. Your custom Arcade expressions, symbology, and labels will be honored when you publish your layer to ArcGIS Online. To do this, right click on the **Cases9_28vs11_16** layer in the Contents pane and select **Sharing**>**Share As Web Layer**.



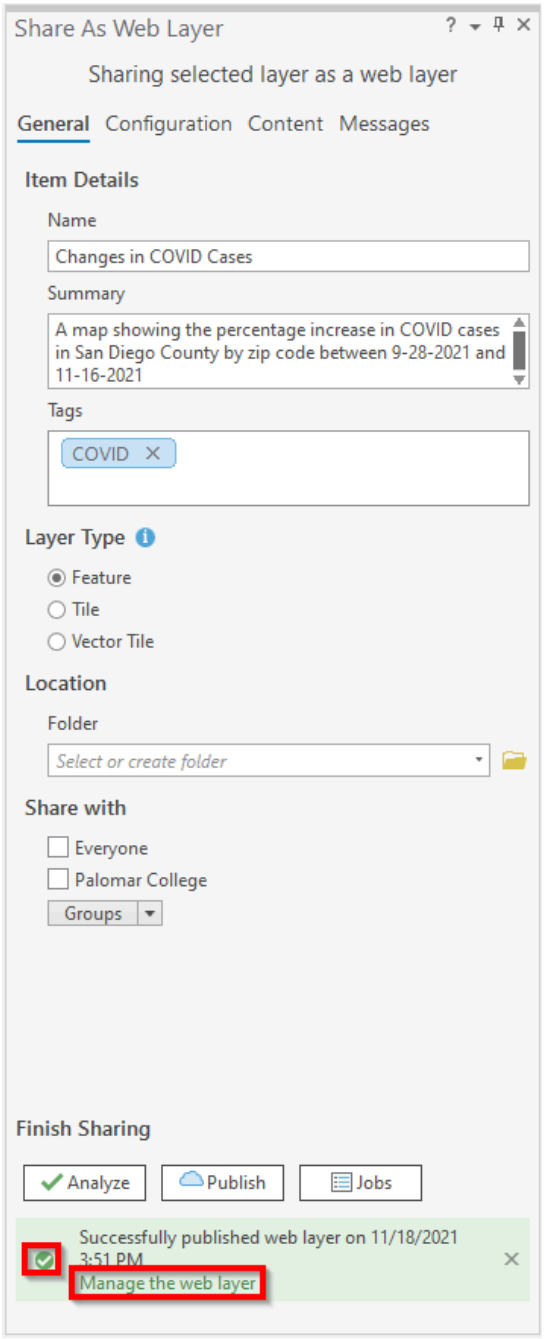
10. In the Share As Web Layer window, fill in Name, Summary, and Tags for the web layer as follow:

    Name: Changes in COVID Cases
    Summary: A map showing the percentage increase in COVID cases in San Diego County by zip code between 9-28-2021 and 11-16-2021
    Tags: COVID

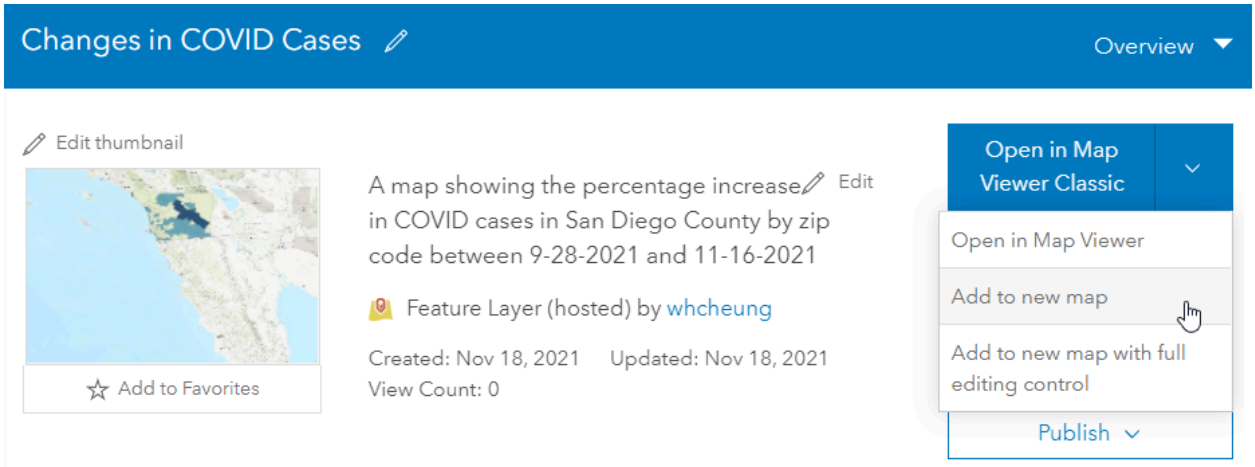
Click **Analyze**, ignore the warning about the Layer not having a feature template. Click **Publish** if no other warnings or errors are found.
Tip: If you see an error message saying 'A web layer name "Changes in COVID Cases" already exists…', click General in the Share As a Web Layer window, and add the last four digits of your phone number to the name of the layer to make the name unique.



11. When the sharing process is complete, you should see a green check at the bottom of the Share As Web Layer window, click **Mange the web layer** to view its details in your web browser.



12. In the layer's item details page in your web browser, select **Add to new map** in order to see the layer in a web map.



13. You should now see the web layer with the same symbology and custom class labels [i.e. Slight increase (0-8%), Moderate increase (8-16%), Significant increase (16-40%)] that we previously set in ArcGIS Pro. Remember that the symbology and custom class labels were made available by the custom Arcade expression that we wrote earlier in this exercise. Save your web map if you like and close your web browser. Save your ArcGIS Pro project and close the application.

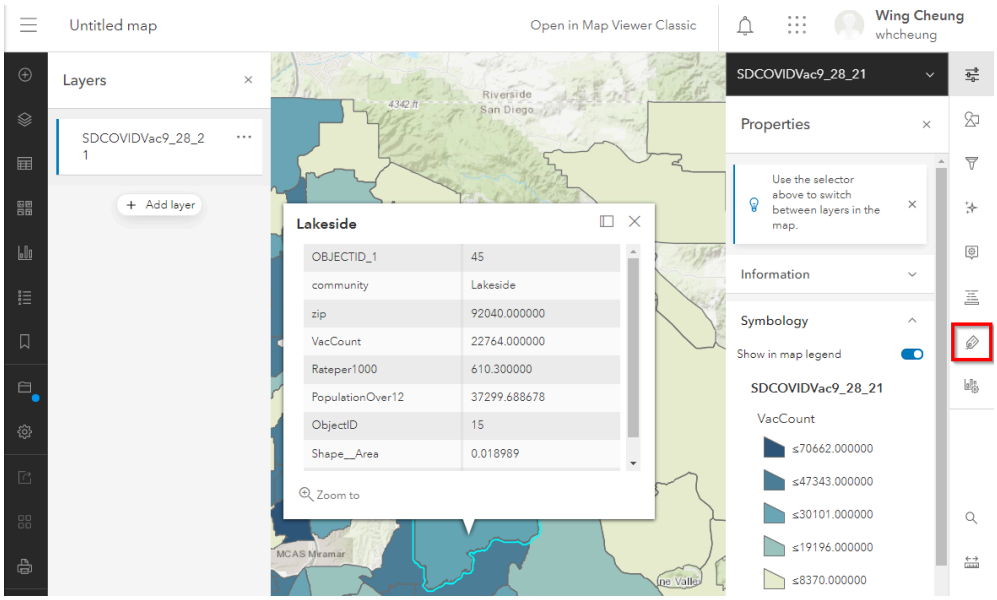
Part 3: Configuring pop-ups and labels in ArcGIS Online with Arcade expressions

1. Open a web browser, navigate to https://www.arcgis.com/, and **sign in**to your ArcGIS Online account.
2. Search for **SDCOVIDVac9_28_21 owner:whcheung**. Be sure to click on the name of **Feature Layer** result and not the Service Definition. This layer was prepared by the County of San Diego's Health and Human Services Agency. It shows the zip code of residence for San Diego County residents ages 12 years and older who received at least one dose of COVID-19 vaccine as of 9/28/2021.

3. In the item description/overview page, select **Open in Map Viewer**. You should now see a map of zip code polygons in the map viewer.
4. Click on any of the polygons in the map to see its attributes in a pop-up window. Of the attribute values in the pop-up window, we will use the information from the [**VacCount**] attribute (i.e. number of people over 12 years old that are vaccinated against COVID) and [**PopulationOver12**] attribute (i.e. number of people over 12 years old in the zip code) to create custom labels for each polygon feature in the following steps.

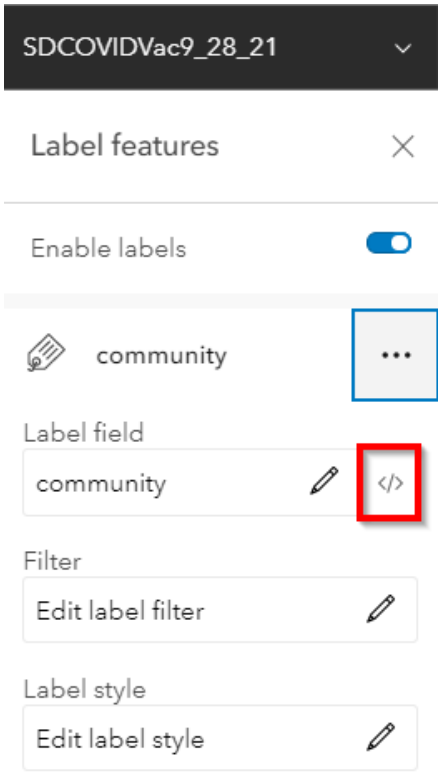
Part 3.1 Creating custom labels in ArcGIS Online

5. Click the **Labels** icon on the right side of the screen.



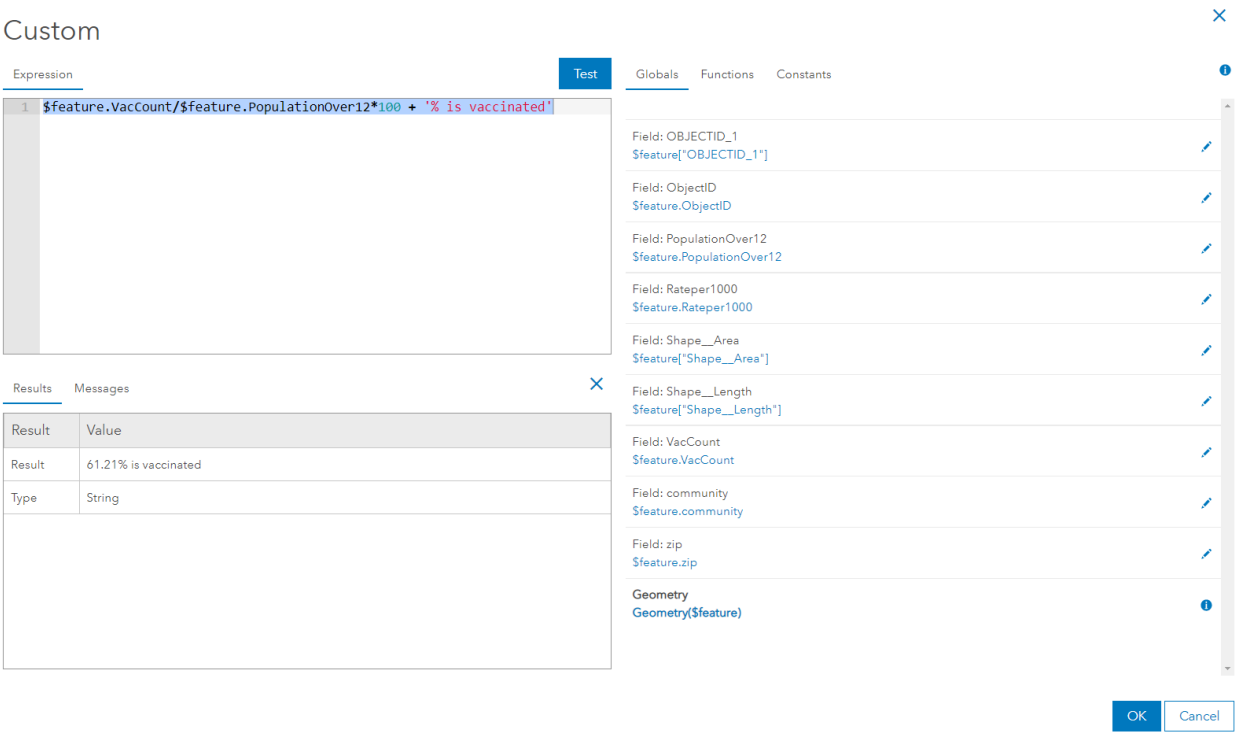
6. Ensure that Enable labels is turned on. Click **+ Add label class**.

7. Click the **Expression icon** </> under Label field to open the Expression window.



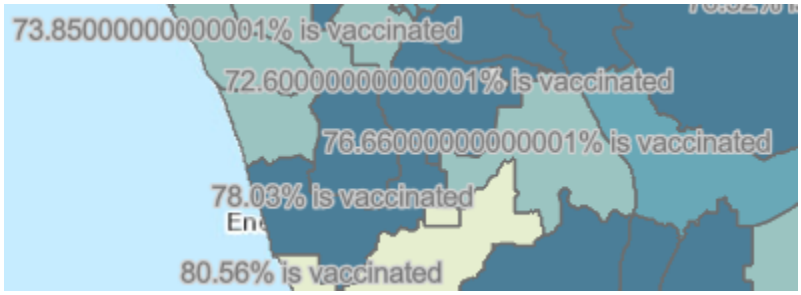
8. To create a custom label for each feature that shows the percentage of each zip code's population (over 12 years old) that is vaccinated against COVID, **delete** what is currently in the Expression box and enter the **following expression**. Click **Test** to ensure your expression is working. Click **OK**.

   $feature.VacCount/$feature.PopulationOver12*100 + '% is vaccinated'



Tip: You can always single click on any items starting with $feature… under the Global tab to add them to the expression.

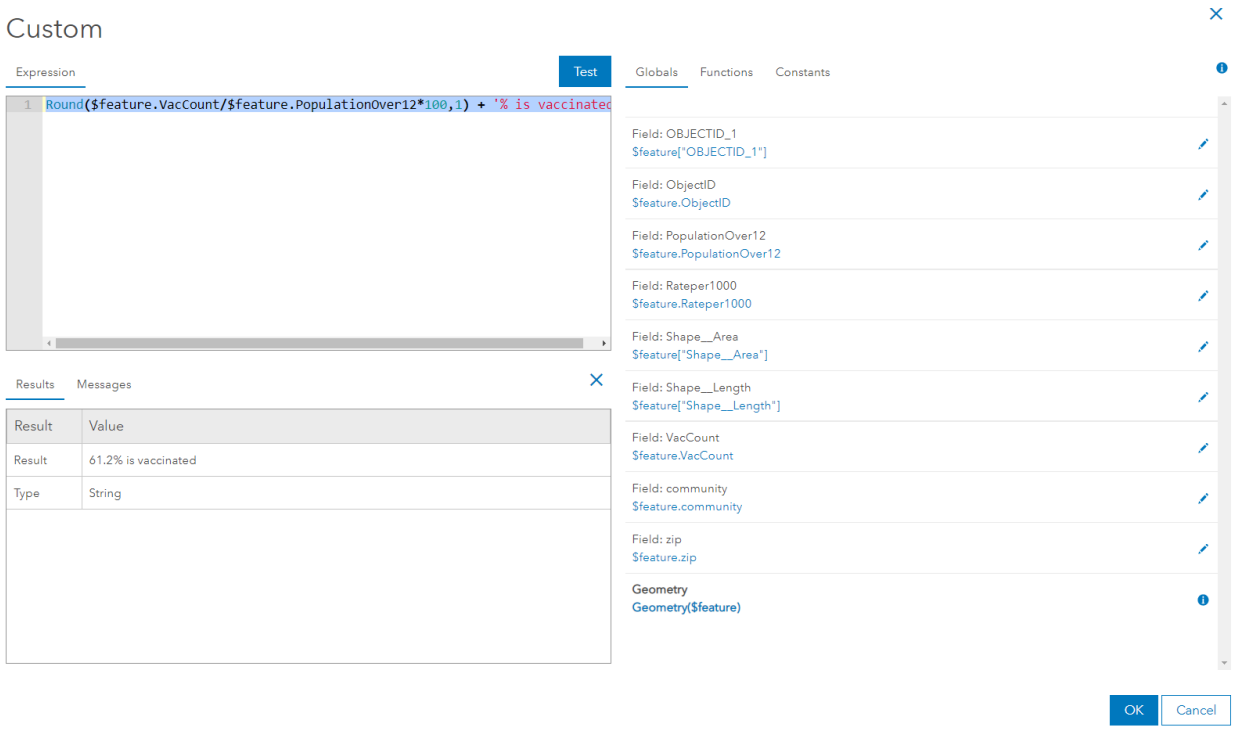
9. You should now see the new custom labels in your map. However, some of the labels are quite lengthy because the percentage value is not rounded (see graphic below). Click the **Expression icon** </> under Label field in the Label features pane to reopen the Expression window.



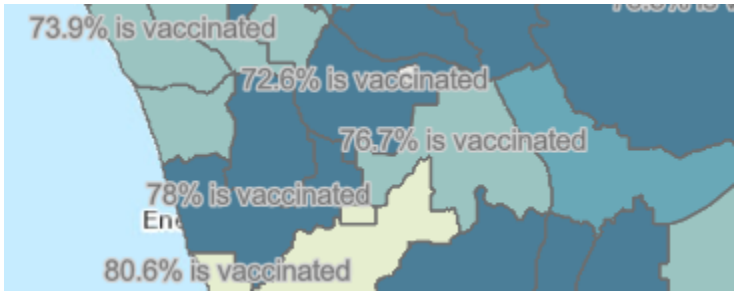
10. You should see our previous expression in the Expression box of the Expression window. We will use the Round() function to shorten our percentage values to include only one decimal place.

Modify the existing expression so it matches the following. Click **Test** to ensure your expression is working. Click **OK**.

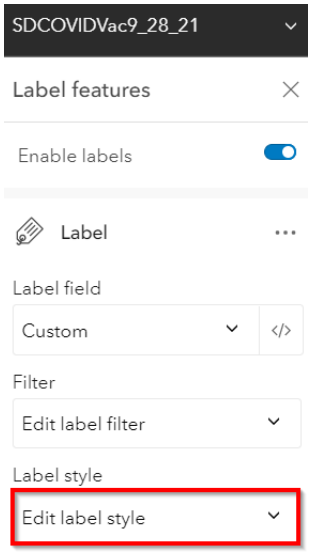
Round($feature.VacCount/$feature.PopulationOver12*100**,1)** + '% is vaccinated'



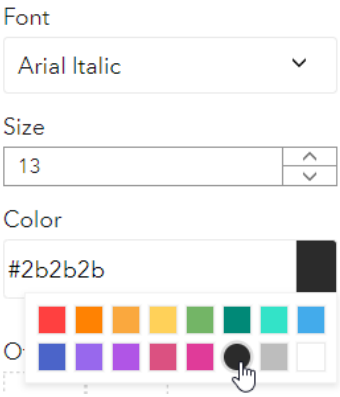
11. You should see that all of your labels in the map are now rounded to one decimal place.



12. To change the styling of your labels, click **Edit label style** under Label style in the Label features pane.

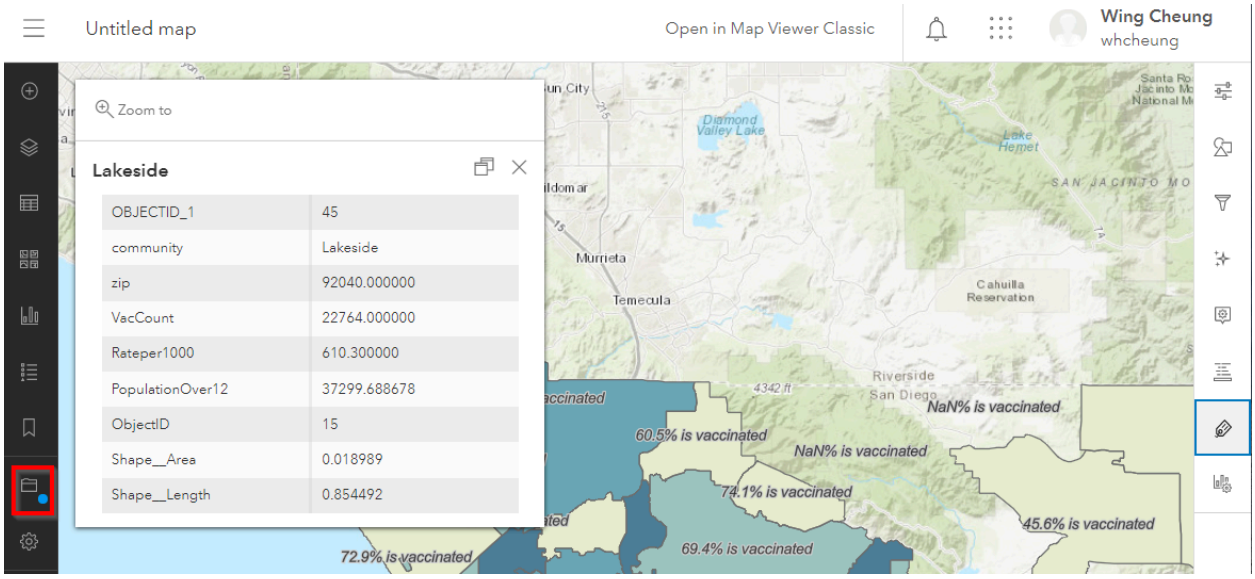


13. Within Label style, click the dropdown under Font to change it to **Arial Italic**. Click the color swatch to change the text color to **black**.



14. Close the Label features pane by clicking on **Labels** on the right side of the map viewer window. Your styling changes are automatically applied to the map.

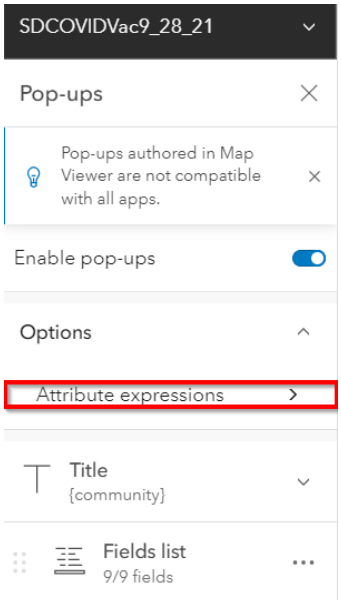
15. Save your map by clicking on the **Save and open icon** on the left side of the map viewer window, and select **Save as**. Fill out the Title, Tags, Summary for your map, and hit **Save map**.



Part 3.2 Configuring custom pop-ups in ArcGIS Online

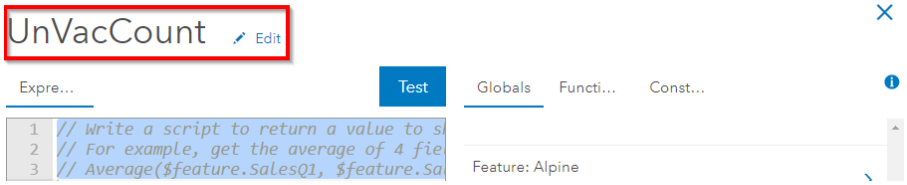
16. (Open the webmap that you worked with in the previous section if necessary.) Make sure that the **SDCOVIDVac9_28_21** layer name is selected in the Layers pane, and click on the **Configure pop-ups** icon on the right side of the map viewer.

17. Within the Pop-ups pane, make sure that **Enable pop-ups** is turned on, click **Attribute expressions**.



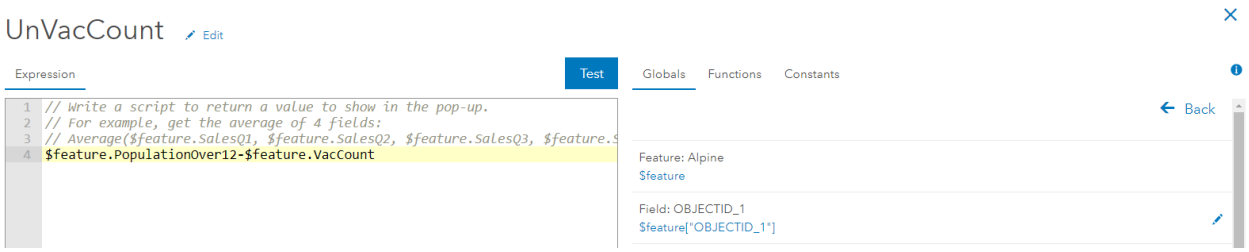
18. In the Attribute expressions pane, click **+ Add expression**. In the following steps, we will create an expression that allows us to calculate the number of unvaccinated people in each zip code based on the information that is currently in the [VacCount] and [PopulationOver12] fields.

19. In the Expression window, click the **edit icon** next to the word Custom, and change the title of the expression to **UnVacCount**. Click **Save**.

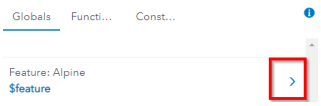


20. Since the number of unvaccinated people is calculated by subtracting the number of vaccinated people [VacCount] from the number of people that are 12 years old and older in each zip code [PopulationOver12], **enter the following expression** in the Expression box. Click **Test** to ensure that your expression is working.

$feature.PopulationOver12 - $feature.VacCount

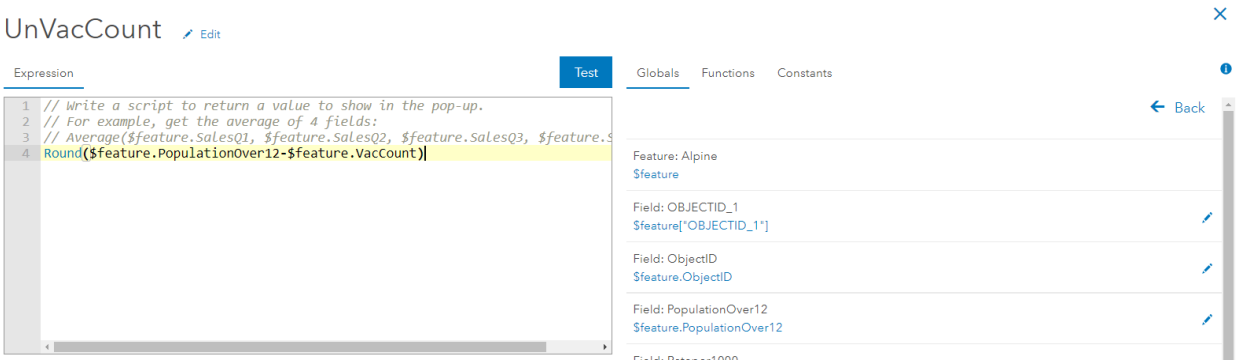


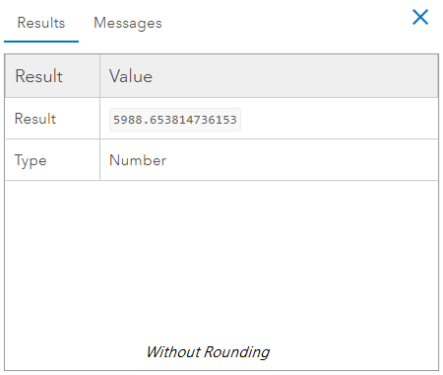
Tip: You can expand the $feature item under the Global tab to see the various fields, then single click on the field you want to add to your expression.

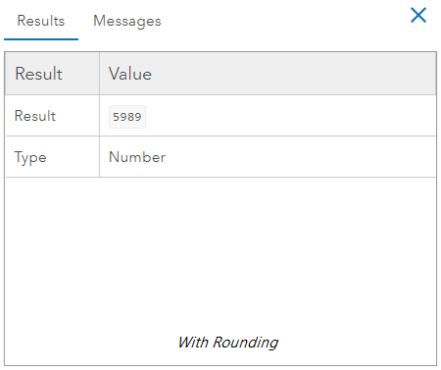


21. You should see in the Test area that the current expression will return answers with a number of decimal places. We will modify the expression using the Round() function as we have done in the previous section, so we are returning only whole numbers as our answers. Modify the current expression as shown below. Click **Test** to ensure that your expression is working, and click **OK**.

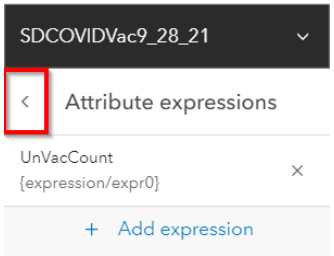
Round($feature.PopulationOver12 - $feature.VacCount**)**



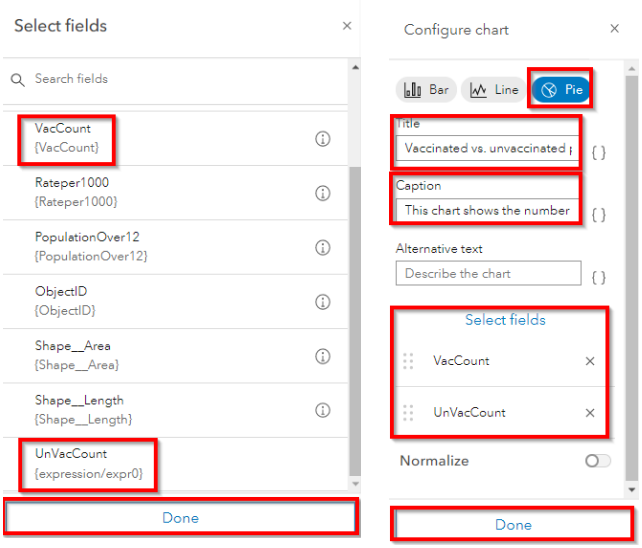




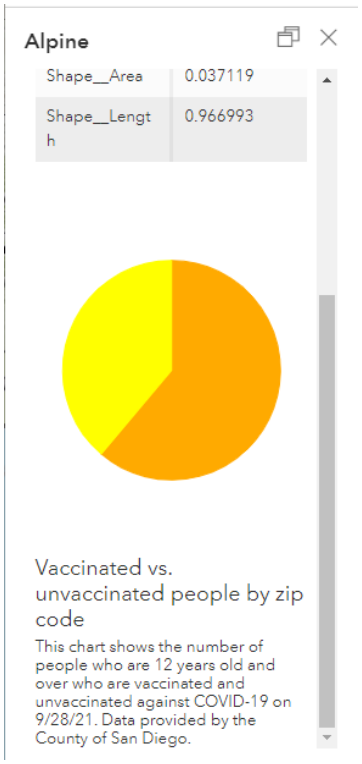
22. You should now see the new expression that you just created [UnVacCount] in the Attribute expressions pane. Click the **back arrow** to return to the Pop-ups pane.



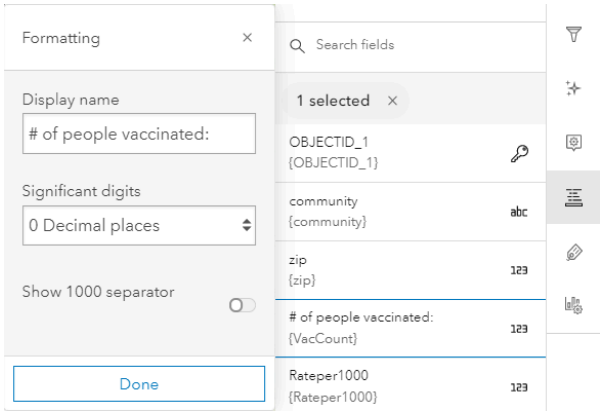
23. Within the Pop-ups pane, click **+ Add content**, and select **Chart**.  In the following steps, we will be adding into our pop-ups a pie chart showing the proportion of vaccinated vs. unvaccinated people in each zip code.
24. In the Configure charts window, select **Pie**.
    - For the Title, enter **Vaccinated vs. unvaccinated people by zip code**
    - For the Caption, enter **This chart shows the number of people who are 12 years old and over who are vaccinated and unvaccinated against COVID-19 on 9/28/21. Data provided by the County of San Diego.**
    - Click **Select fields**, and in the Select field window, pick **[VacCount]** and **[UnVacCount]** from the list. Click **Done**.
    - Click **Done** in the Configure charts window.



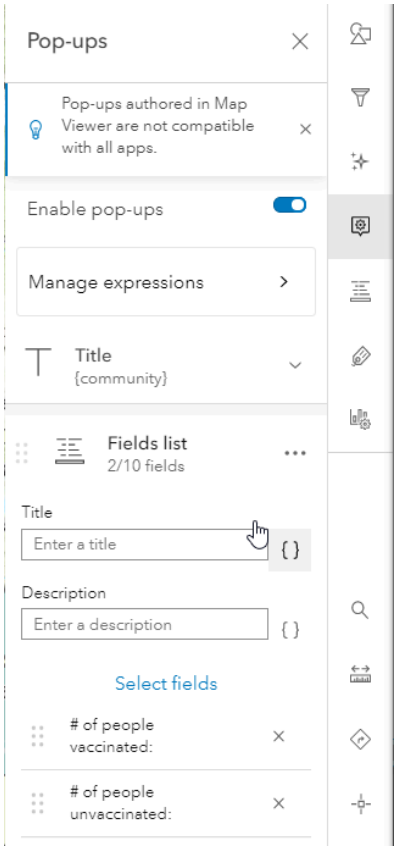
25. Back in the map viewer, click on any of the zip code polygons. You should now see the pie chart and its title and caption at the bottom of the pop-up.



26. To make the names of the fields in the pop-up more user-friendly, click the **Configure fields** icon on the right side of the window.
27. In the Fields pane, click **VacCount**, and change its Display name to **# of people vaccinated:** . Change the Significant digits to **0 decimal places**, then click **Done**.



28. In the Fields pane, click **UnVacCount**, and change its Display name to **# of people unvaccinated:** . Change the Significant digits to **0 decimal places**, then click **Done**.
29. However, there is still quite a bit of unnecessary information in the pop-up window. To specify which field(s) should show up in the pop-up, click the **Configure pop-ups icon** to reopen the pane.
30. Within the Pop-ups pane, click **Fields list**, click the **X** to hide all of the fields EXCEPT **# of people vaccinated:** and **# of people unvaccinated:** .

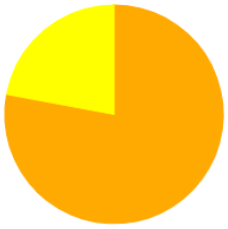


Tip: If you are not seeing # of people vaccinated and/or # of people unvaccinated in your list, make sure you have completed step 26-29 in this section.

31. Back in the map viewer, click on any of the zip code polygons. You should now see the pie chart and only two fields in the pop-up window. Save your map by clicking on the **Save and open icon**  on the left side of the map viewer window, and select **Save**.

Bonsall

# of people vaccinated:	3100
# of people unvaccinated:	877



Vaccinated vs. unvaccinated people by zip code

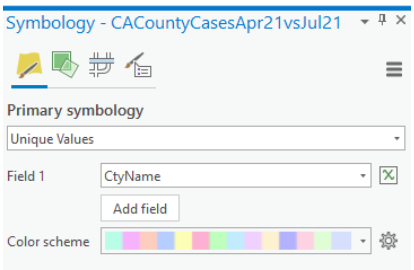
This chart shows the number of people who are 12 years old and over who are vaccinated and unvaccinated against COVID-19 on 9/28/21. Data provided by the County of San Diego.

Part 4: Using Logic in Arcade

1. Open ArcGIS Pro. Start a new project and new map. Add the **CACountyCasesApr21vsJul21** feature class from the CasesTutorial.gdb geodatabase to your map. This feature class contains information on COVID cases and deaths in various California counties on 4/30/21 and 7/31/21 as reported by the California Department of Public Health (https://data.ca.gov/dataset/covid-19-time-series-metrics-by-county-and-state1). Those two dates were specifically chosen because they roughly coincide with when the Delta variant was first detected in California (April 2021), and when it became the dominant COVID variant (July 2021) according to the California Department of Public Health (https://www.cdph.ca.gov/Programs/CID/DCDC/Pages/COVID-19/COVID-Variants.aspx).

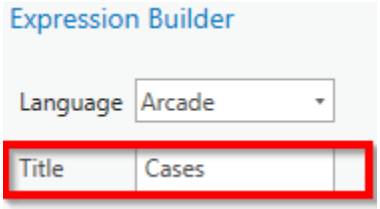
2. Select the CACountyCasesApr21vsJul21 layer in the Contents pane, click the **Appearance** tab in the main menu, and select **Symbology**.

3. In the Symbology pane, select **Unique Values** under Primary symbology. And select **CtyName** for Field. Click the **Add all values** icon . We are now coloring the county polygon features in the map by their names.



Part 4.1 Using the IF logic to set symbology

4. Within the Symbology pane, click the **Set an expression** icon .
5. In the Expression Builder window, Change the Title to **Cases**.

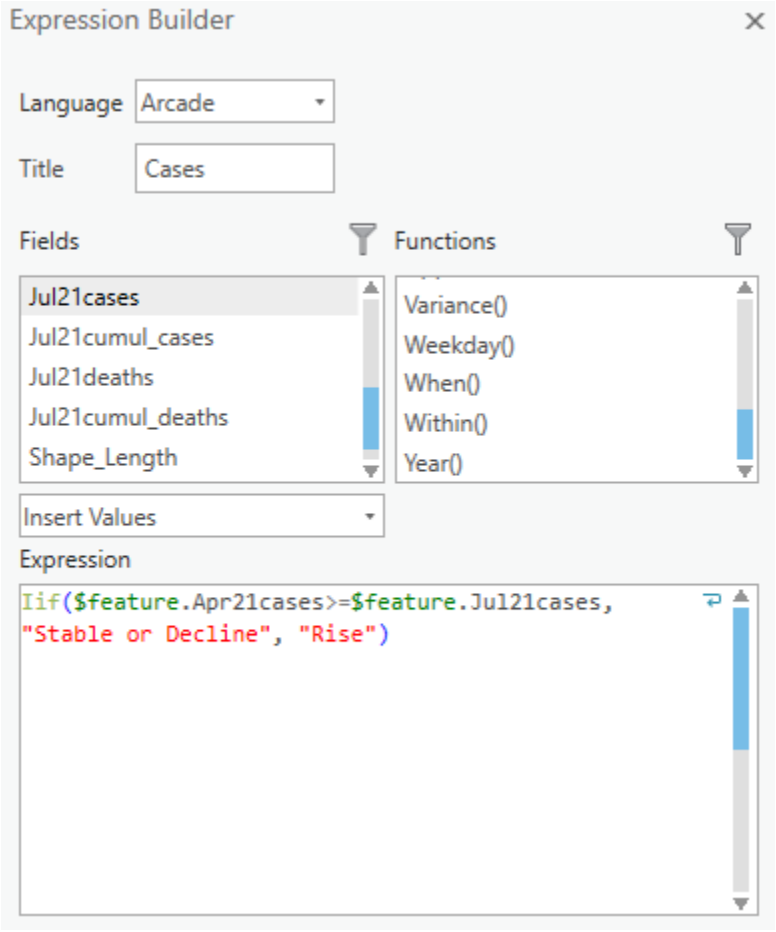


6. In the Expression Builder window, clear the existing contents in the Expression area, and replace it with the following expression. Be sure to use a capital I for the first letter in Iif, it is case sensitive.

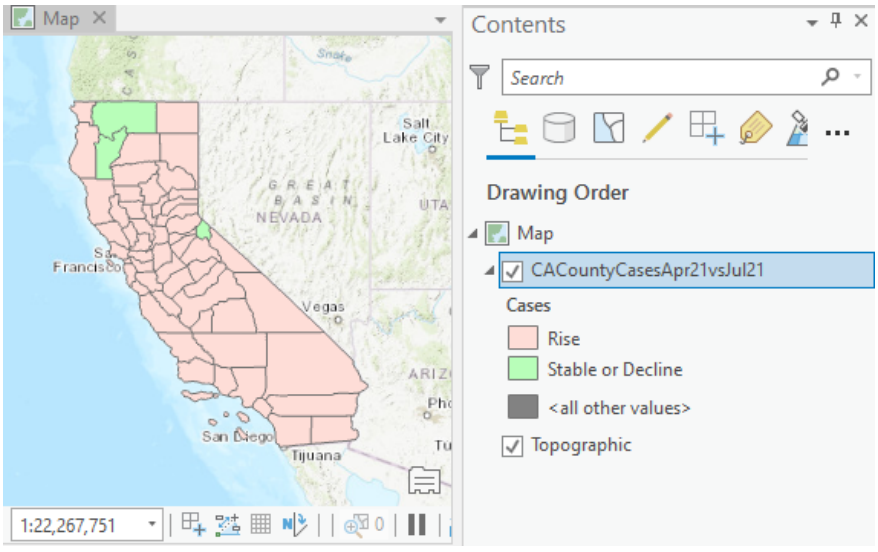
```
Iif($feature.Apr21cases>=$feature.Jul21cases, "Stable or Decline", "Rise")
```

In this If/Then/Else expression, the format is Iif(Condition, what happens when the condition is true, what happens when the condition is false). Specifically, we are seeing if the cases

reported on 4/30/21 (Apr21cases) is greater than or equal to the cases reported on 7/31/21 (Jul21cases), counties that satisfy this condition will have a label of “Stable or Decline”, all other counties will have a label of “Rise”.

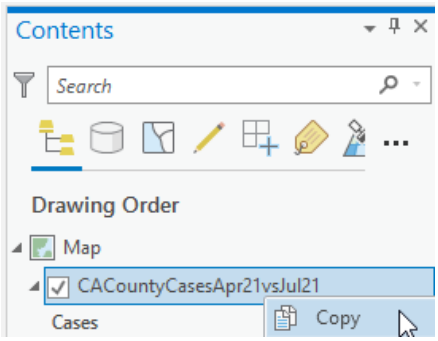


7. Click the **Verify** icon to check your expression. Click **OK**. You should now see the map updated with the symbology (i.e. Stable or Decline, Rise) that we specified in the expression.

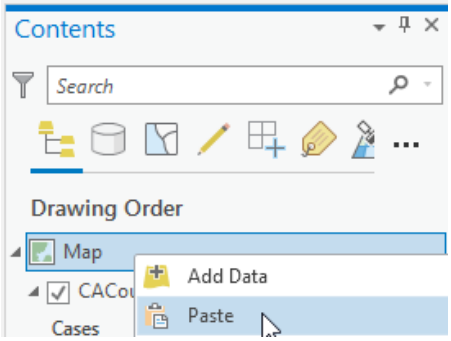


Part 4.2 Using the WHEN logic to set symbology

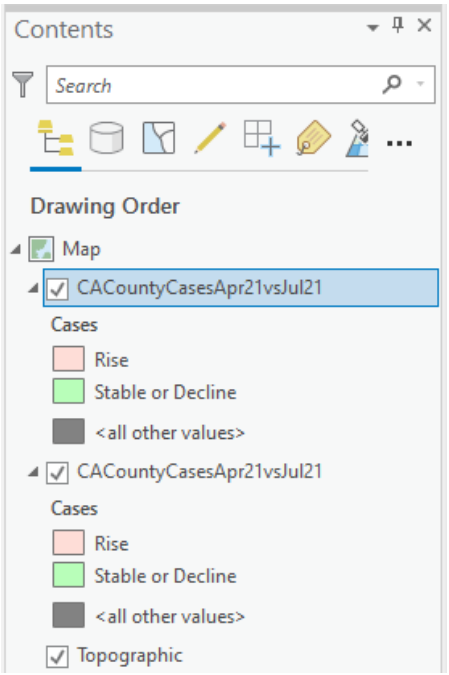
8. In the Contents pane, right click on **CACountyCasesApr21vsJul21** and select **Copy**.



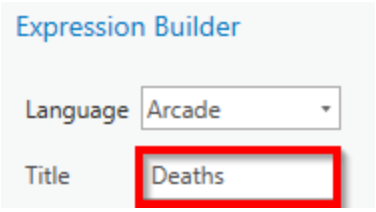
9. Right click on **Map** and select **Paste** to create a copy of our layer in the map.



10. You should now see two CACountyCasesApr21vsJul21 layers in the Contents pane. Select either one of them, click the **Appearance** tab in the main menu, and select **Symbology**.



11. Within the Symbology pane, click the **Set an expression** icon .
12. In the Expression Builder window, Change the Title to **Deaths**.

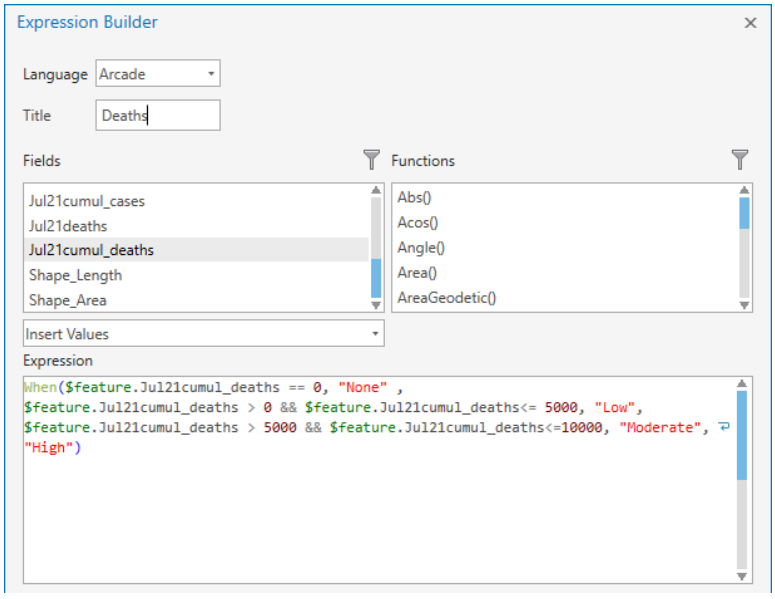


13. In the Expression Builder window, clear the existing contents in the Expression area, and replace it with the following expression.

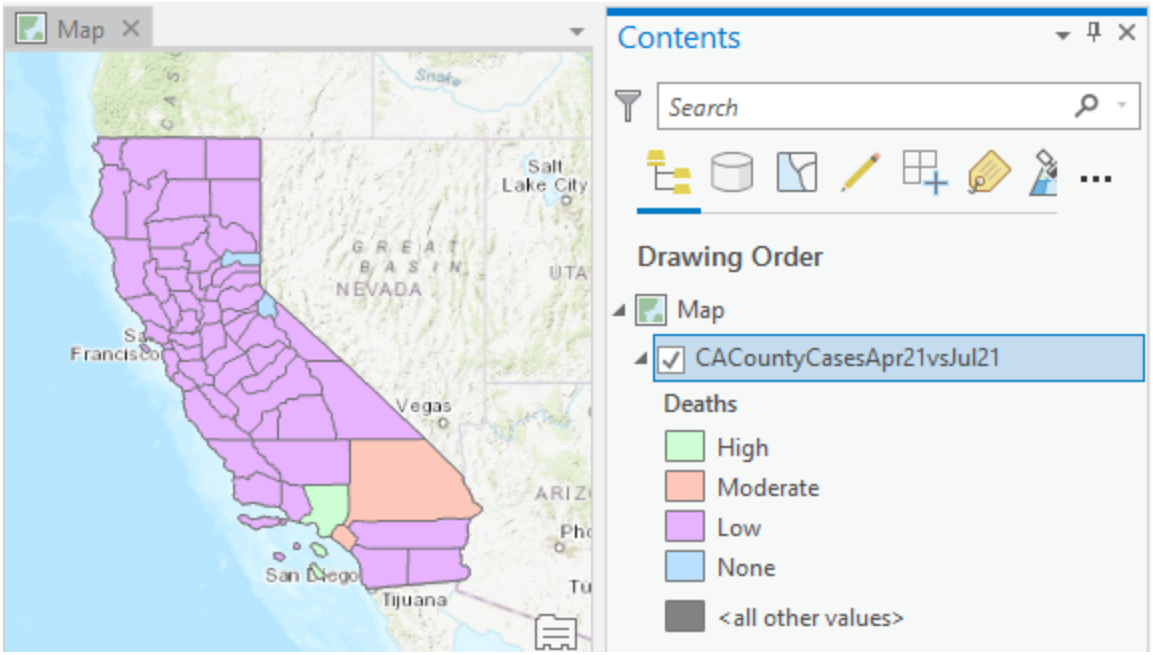
```
When($feature.Jul21cumul_deaths == 0, "None" , $feature.Jul21cumul_deaths > 0
&& $feature.Jul21cumul_deaths<= 5000, "Low", $feature.Jul21cumul_deaths > 5000
&& $feature.Jul21cumul_deaths<=10000, "Moderate", "High")
```

Unlike the If/Then/Else expression, the When expression allows us to specify multiple conditions and what should occur when each of those conditions are met. Specifically, in the expression above, there are four conditions.

(1) When the cumulative death reported on 7/31/21 (Jul21cumul_deaths) is zero, assign the value of None.
(2) When the cumulative death reported on 7/31/21 (Jul21cumul_deaths) is greater than zero and less than or equal to 5000, assign the value of Low.
(3) When the cumulative death reported on 7/31/21 (Jul21cumul_deaths) is greater than 5000 and less than or equal to 10000, assign the value of Moderate.
(4) All other counties will be assigned the value of High.



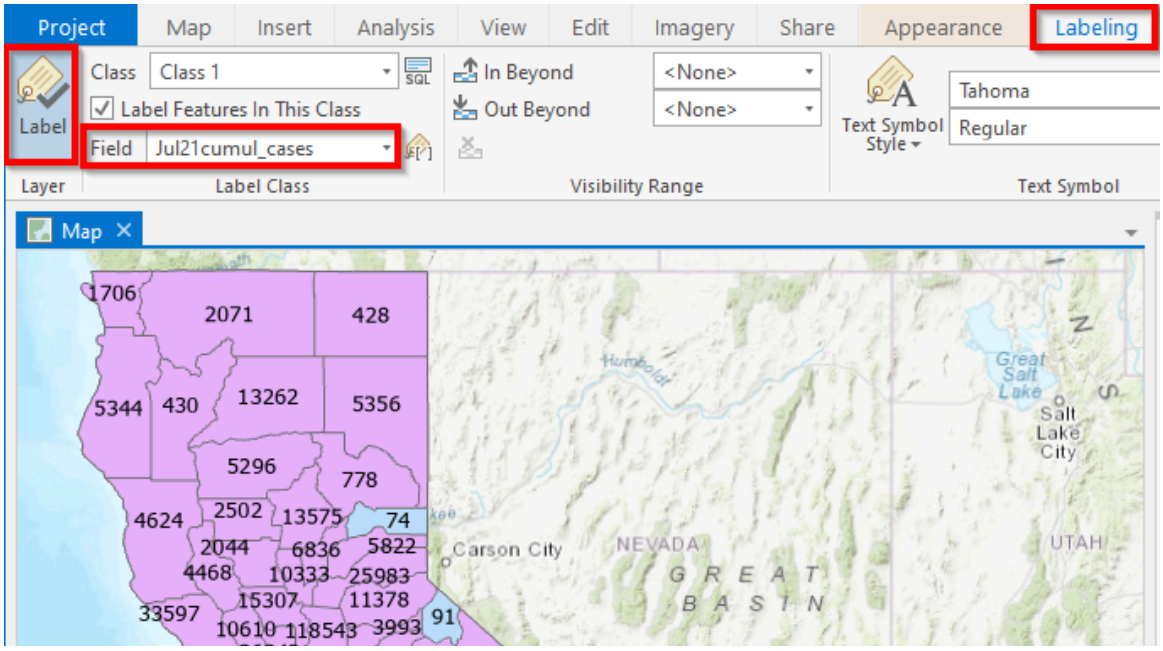
14. Click the **Verify** icon ✓ to check your expression. Click **OK**. You should now see the map updated with the symbology (i.e. High, Moderate, Low, None) that we specified in the expression.



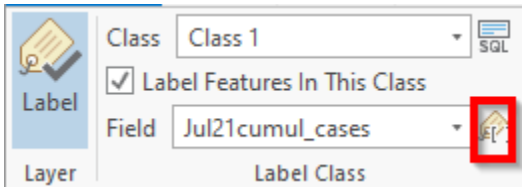
Part 4.3 Using the WHEN logic to customize feature labels

15. We will now label our county features by the cumulative cases reported on 7/31/21 (Jul21cumul_cases), and use arcade expression to customize the labels by adding in thousand separators (i.e. comma) in the appropriate places. In the Contents pane, select either of the CACountyCasesApr21vsJul21 layers, click the **Labeling** tab in the main menu, click the **Label** icon to turn on labeling, and select **Jul21cumul_cases** for Field.

    You should now see each county labeled by the cumulative cases it reported as of 7/31/21. Note that none of the labels currently have thousand separators.



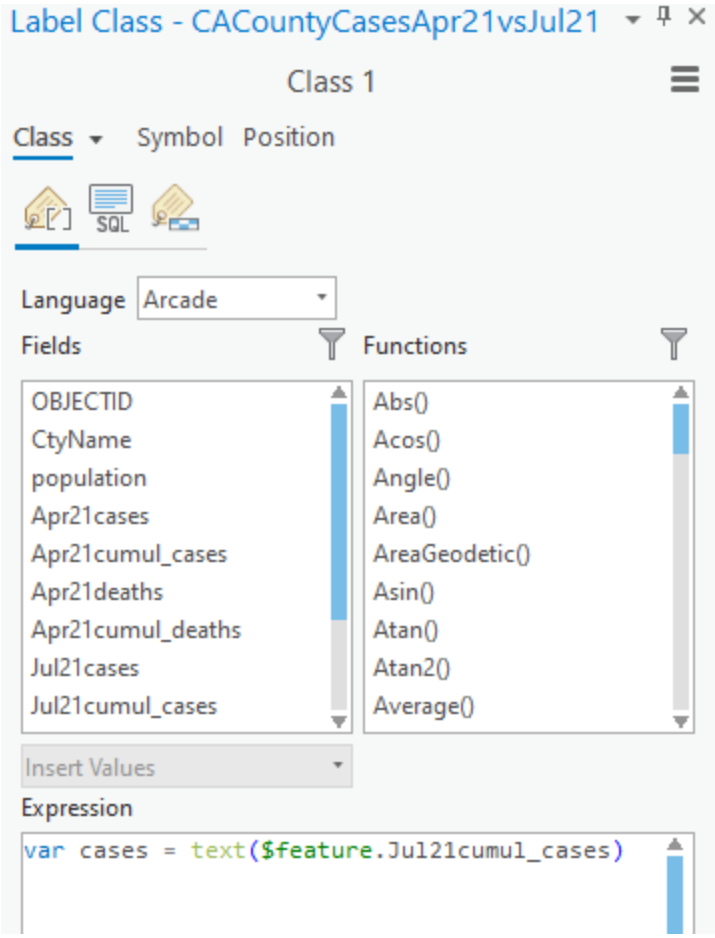
16. Click the **Expression** icon next to the Field dropdown to open the Label Class pane.



17. Clear the existing contents in the Expression area, and add in the following expression to:
(1) create a new variable called cases based on the value in the Jul21cumul_cases attribute,
(2) convert the values in the Jul21cumul_cases attribute into the text format

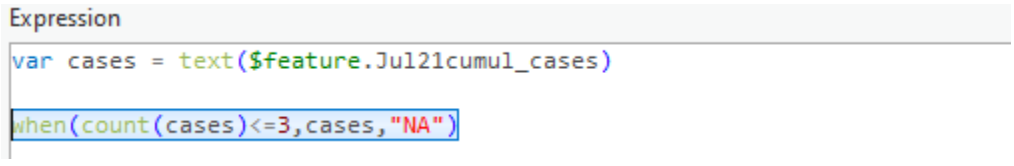
```
var cases = text($feature.Jul21cumul_cases)
```

Click the **Verify** icon to check your expression and click Apply. Do not be alarmed if the labels disappeared in your map for now.



18. In the Expression area, add in the following expression below our initial expression from the previous step to:
(1) get a character count of each of our labels using the Count function,
(2) use the value in the cases variable if the label character count is less than three character (i.e. less than 1000 cases, hence no thousand separators is needed),
(3) temporarily assign the label of "NA" to all counties with labels that have more than 3 characters.

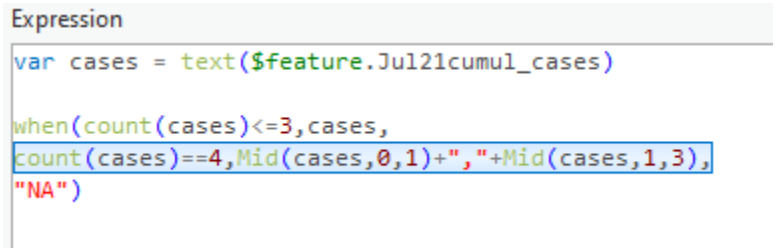
```
when(count(cases)<=3,cases,"NA")
```



Click the **Verify** icon ✔ to check your expression and click **Apply**.

19. Expand our expression from the previous step by identifying those counties with labels that have exactly 4 characters. Once those labels are found, we will:
(1) use the Mid function to pull the first character/digit out (starting with character #0 [i.e. the first digit], and take 1 character),
(2) add in the thousand separator comma,
(3) use the Mid function to put the remaining 3 characters back into the label (starting with character #1 [i.e. the second digit], and take 3 characters).

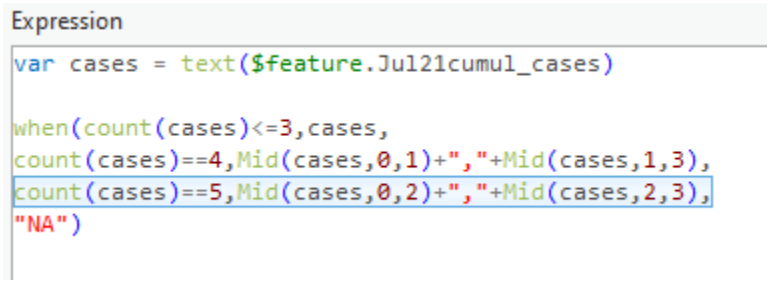
```
count(cases)==4,Mid(cases,0,1)+","+Mid(cases,1,3),
```



Click the **Verify** icon ✔ to check your expression and click **Apply**.

20. Expand our expression from the previous step by identifying those counties with labels that have exactly 5 characters. Once those labels are found, we will:
(1) use the Mid function to pull the first two characters/digits out (starting with character #0, and take 2 characters),
(2) add in the thousand separator comma,
(3) use the Mid function to put the remaining 3 characters back into the label (starting with character #2 [i.e. the third digit], and take 3 characters).

```
count(cases)==5,Mid(cases,0,2)+","+Mid(cases,2,3),
```



Click the **Verify** icon ✔ to check your expression and click **Apply**.

21. Expand our expression from the previous step by identifying those counties with labels that have exactly 6 characters. Once those labels are found, we will:
(1) use the Mid function to pull the first three characters/digits out (starting with character #0, and take 3 characters),

(2) add in the thousand separator comma,
(3) use the Mid function to put the remaining 3 characters back into the label (starting with character #3 [i.e. the third digit], and take 3 characters).

```
count(cases)==6,Mid(cases,0,3)+","+Mid(cases,3,3),
```

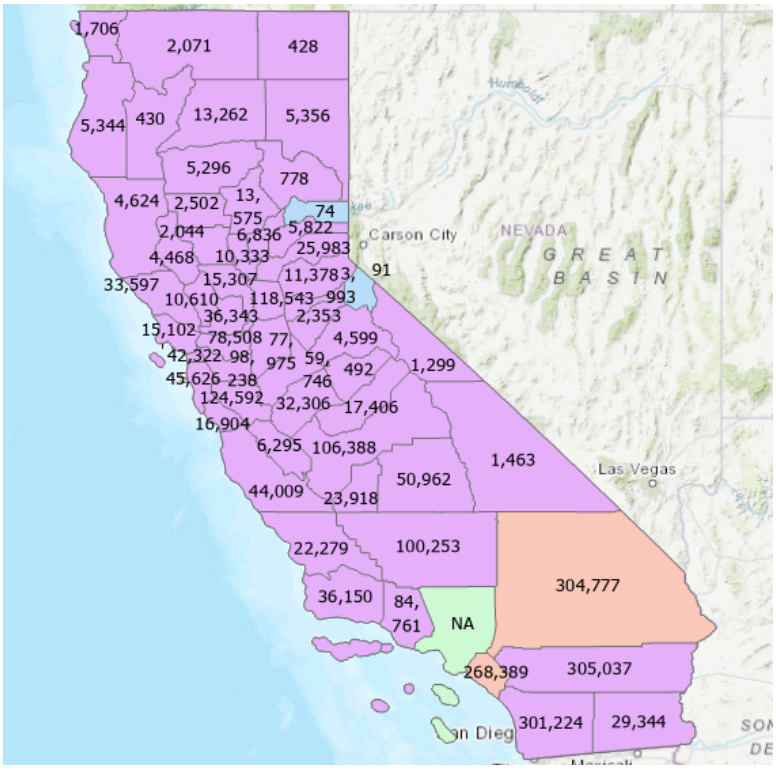
Expression

```
var cases = text($feature.Jul21cumul_cases)

when(count(cases)<=3,cases,
count(cases)==4,Mid(cases,0,1)+","+Mid(cases,1,3),
count(cases)==5,Mid(cases,0,2)+","+Mid(cases,2,3),
count(cases)==6,Mid(cases,0,3)+","+Mid(cases,3,3),
"NA")
```

Click the **Verify** icon ✔ to check your expression and click **Apply**.

22. You should now see that most of the features in the maps now have labels with the thousand separators, with the exception of Los Angeles County (green in the graphic below), which still has the label of NA.



23. Sorting the attribute table of the layer by the Jul21cumul_cases attribute, we can see that the case count for most California counties is 6 characters in length or less, except Los Angeles County, which has a case count of 1,268,294 with a length of 7 characters.

CtyName	population	Apr21cases	Apr21cumul_cases	Apr21deaths	Apr21cumul_deaths	Jul21cases	Jul21cumul_cases
Los Angeles County	10257557	275	1197555	4	24264	2311	1268294
Riverside County	2468145	67	289963	1	4486	637	305037
San Bernardino County	2217398	72	291249	2	5140	510	304777
San Diego County	3370418	150	277538	1	3746	772	301224
Orange County	3228519	69	253716	0	5150	627	268389
Santa Clara County	1967585	55	117200	0	1723	224	124592

24. Continue expanding our expression from the previous steps by identifying those counties with labels that have exactly 7 characters. Once those labels are found, we will:
(1) use the Mid function to pull the first characters out (starting with character #0, and take 1 character),
(2) add in the thousand separator comma,
(3) use the Mid function to pull the next three characters (starting with character #1, and take 3 characters),
(4) add in the thousand separator comma,
(5) use the Mid function to put the remaining 3 characters back into the label (starting with character #4, and take 3 characters).

```
count(cases)==7,Mid(cases,0,1)+","+Mid(cases,1,3)+","+Mid(cases,4,3),
```

Expression

```
var cases = text($feature.Jul21cumul_cases)

when(count(cases)<=3,cases,
count(cases)==4,Mid(cases,0,1)+","+Mid(cases,1,3),
count(cases)==5,Mid(cases,0,2)+","+Mid(cases,2,3),
count(cases)==6,Mid(cases,0,3)+","+Mid(cases,3,3),
count(cases)==7,Mid(cases,0,1)+","+Mid(cases,1,3)+","+Mid(cases,4,3),
"NA")
```

Click the **Verify** icon ✔ to check your expression and click **Apply**

.

25. You should now see that all of the features in the maps now have labels with the thousand separators.

Part 4.4 Creating your own functions

26. We can turn the lines of codes that we wrote in the previous section into a function, so it can be reused to properly format (i.e. add in the thousand separators) the values stored in other fields within the layer. A user defined function should be formatted and must consist of the three components as labeled and underlined below:

    function Name of Function (Name of Input) {Code that Specifies What to Do}

    Within the Expression section of the Label Class pane, add in the word **function** before your existing code.

```
Expression
function var cases = text($feature.Jul21cumul_cases)

when(count(cases)<=3,cases,
count(cases)==4,Mid(cases,0,1)+","+Mid(cases,1,3),
count(cases)==5,Mid(cases,0,2)+","+Mid(cases,2,3),
count(cases)==6,Mid(cases,0,3)+","+Mid(cases,3,3),
count(cases)==7,Mid(cases,0,1)+","+Mid(cases,1,3)+","+Mid(cases,4,3),
"NA")
```

27. Now we will add in the first component of the user defined function, which is the name that we want to call our function. We will call our function ValueFormat. Within the Expression section of the Label Class pane, add in the word **ValueFormat** after the word Function.

```
Expression
function ValueFormat var cases = text($feature.Jul21cumul_cases)

when(count(cases)<=3,cases,
count(cases)==4,Mid(cases,0,1)+","+Mid(cases,1,3),
count(cases)==5,Mid(cases,0,2)+","+Mid(cases,2,3),
count(cases)==6,Mid(cases,0,3)+","+Mid(cases,3,3),
count(cases)==7,Mid(cases,0,1)+","+Mid(cases,1,3)+","+Mid(cases,4,3),
"NA")
```

28. Now we will add in the second component of the user defined function, which is the name that we want to call our input. We will call our input UnformatCases. Within the Expression section of the Label Class pane, add in **(UnformatCases)** after the word ValueFormat.

```
Expression
function ValueFormat(UnformatCases) var cases = text
($feature.Jul21cumul_cases)

when(count(cases)<=3,cases,
count(cases)==4,Mid(cases,0,1)+","+Mid(cases,1,3),
count(cases)==5,Mid(cases,0,2)+","+Mid(cases,2,3),
count(cases)==6,Mid(cases,0,3)+","+Mid(cases,3,3),
count(cases)==7,Mid(cases,0,1)+","+Mid(cases,1,3)+","+Mid(cases,4,3),
"NA")
```

29. Now we will add in the third component of the user defined function, which entails enclosing our code in curvy brackets {}, and specifying where the input that we named in the previous step should go in our code. First, enclose all the code starting with the word var in the curvy brackets. Second, replace $feature.Jul21cumul_cases with the word **UnformatCases**, since that is where the input values will be found within our code.

```
Expression
function ValueFormat(UnformatCases) { var cases = text
($feature.Jul21cumul_cases)

when(count(cases)<=3,cases,
count(cases)==4,Mid(cases,0,1)+","+Mid(cases,1,3),
count(cases)==5,Mid(cases,0,2)+","+Mid(cases,2,3),
count(cases)==6,Mid(cases,0,3)+","+Mid(cases,3,3),
count(cases)==7,Mid(cases,0,1)+","+Mid(cases,1,3)+","+Mid(cases,4,3),
"NA") }
```

```
Expression
function ValueFormat(UnformatCases) { var cases = text(UnformatCases)

when(count(cases)<=3,cases,
count(cases)==4,Mid(cases,0,1)+","+Mid(cases,1,3),
count(cases)==5,Mid(cases,0,2)+","+Mid(cases,2,3),
count(cases)==6,Mid(cases,0,3)+","+Mid(cases,3,3),
count(cases)==7,Mid(cases,0,1)+","+Mid(cases,1,3)+","+Mid(cases,4,3),
"NA") }
```

Click the **Verify** icon ✔ to check your expression is valid.

30. To use our new ValueFormat function, within the Expression section of the Label Class pane, start a new line after the close curvy bracket } , and type in the following to use our new function (i.e. ValueFormat) to add thousand separators to the values found in the Apr21cumul_death attribute within our layer (be sure to pay attention to capitalization):

    Return ValueFormat($feature.Apr21cumul_deaths)

    Expression

    ```
    function ValueFormat(UnformatCases) { var cases = text(UnformatCases)

    when(count(cases)<=3,cases,
    count(cases)==4,Mid(cases,0,1)+","+Mid(cases,1,3),
    count(cases)==5,Mid(cases,0,2)+","+Mid(cases,2,3),
    count(cases)==6,Mid(cases,0,3)+","+Mid(cases,3,3),
    count(cases)==7,Mid(cases,0,1)+","+Mid(cases,1,3)+","+Mid(cases,4,3),
    "NA") }

    Return ValueFormat($feature.Apr21cumul_deaths)
    ```

    Click the **Verify** icon ✔ to check your expression and click **Apply**. You should now see the labels in your map are updated with properly formatted values showing the cumulative deaths reported on 4/30/21.

31. To reuse our function to label the map with properly formatted values showing the cumulative deaths reported on 7/31/21, change the input within the return statement to $feature.Jul21cumul_deaths.

    Expression

    ```
    function ValueFormat(UnformatCases) { var cases = text(UnformatCases)

    when(count(cases)<=3,cases,
    count(cases)==4,Mid(cases,0,1)+","+Mid(cases,1,3),
    count(cases)==5,Mid(cases,0,2)+","+Mid(cases,2,3),
    count(cases)==6,Mid(cases,0,3)+","+Mid(cases,3,3),
    count(cases)==7,Mid(cases,0,1)+","+Mid(cases,1,3)+","+Mid(cases,4,3),
    "NA") }

    Return ValueFormat($feature.Jul21cumul_deaths)
    ```

    Click the **Verify** icon ✔ to check your expression and click **Apply**.

Part 5: Decoding Values and Dates

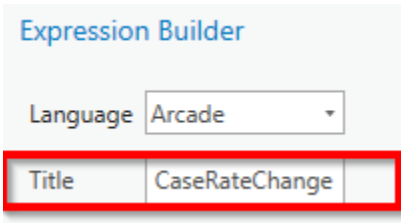
1. Open ArcGIS Pro. Start a new project and new map. Add the **CaseRate11_30vs12_26** feature class from the CasesTutorial.gdb geodatabase to your map. This feature class contains information on COVID case rate (a 7-day moving average of cases per 100,000 people) in various California counties on 11/30/21 and 12/26/21 as reported by the Centers for Disease Control and Prevention (CDC) (https://covid.cdc.gov/covid-data-tracker/#county-view?list_select_state=all_states&data-type=Cases&list_select_county=6073). Those two dates were specifically chosen because the United States first classified Omicron as a COVID variant of concern on 11/30/21, and it spread throughout the country at an unprecedented rate in the following weeks. Note that five California counties with missing data were intentionally deleted from the layer.

Part 5.1 Performing quick calculations to symbolize features

2. Select the CaseRate11_30vs12_26 layer in the Contents pane, click the **Appearance** tab in the main menu, and select **Symbology**.

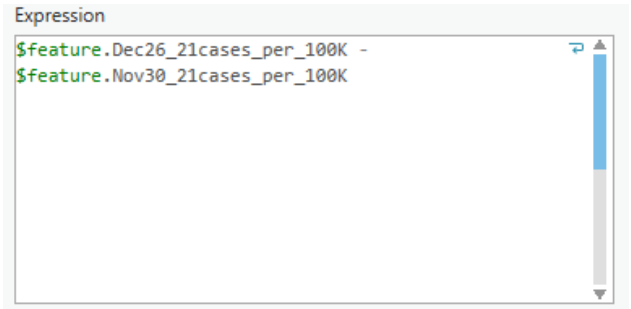
3. In the Symbology pane, select **Graduated Colors** under Primary symbology. Click the **Set an expression** icon  .

4. We will color each county based on the difference in COVID case rate between 11/30/21 and 12/26/21. In the Expression Builder window, Change the Title to **CaseRateChange**.

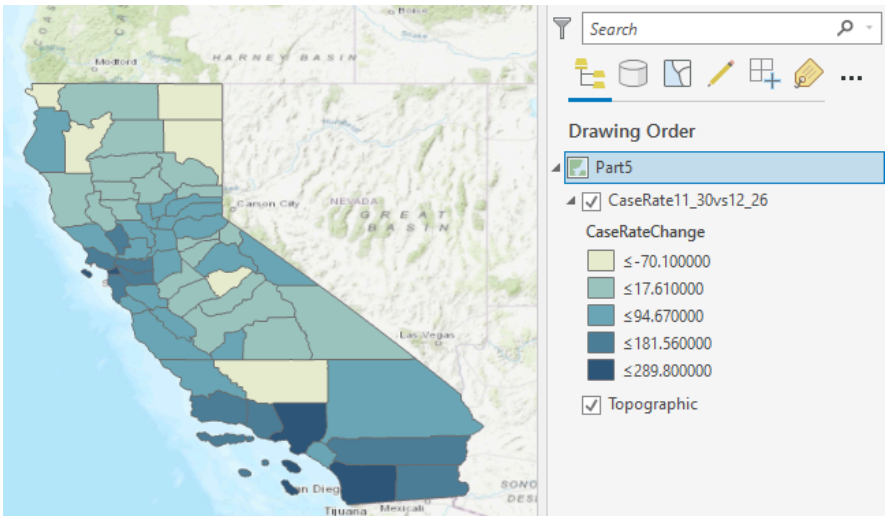


5. In the Expression Builder window, clear the contents in the Expression area, and replace it with the following expression to calculate the difference in case rate between the two dates.

```
$feature.Dec26_21cases_per_100K - $feature.Nov30_21cases_per_100K
```



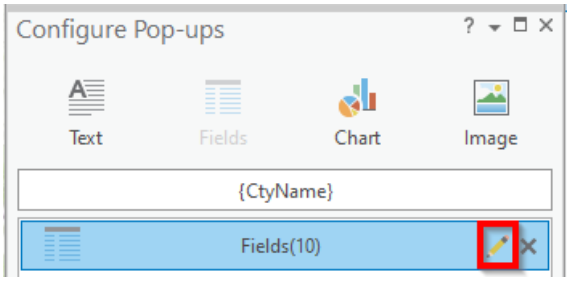
6. Click the **Verify** icon  to check your expression. Click **OK**. You should now see the map updated with different colors based on the changes in case rate between the two dates.



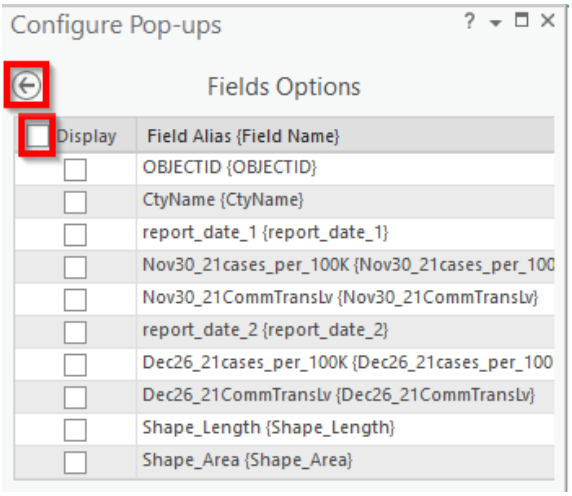
Part 5.2 Using date and decode functions to customize pop-ups

7. Right click the CaseRate11_30vs12_26 layer in the Contents pane, and select **Configure Pop-ups**.

8. In the Configure Pop-ups window, click the **edit** icon  next to Fields(10).



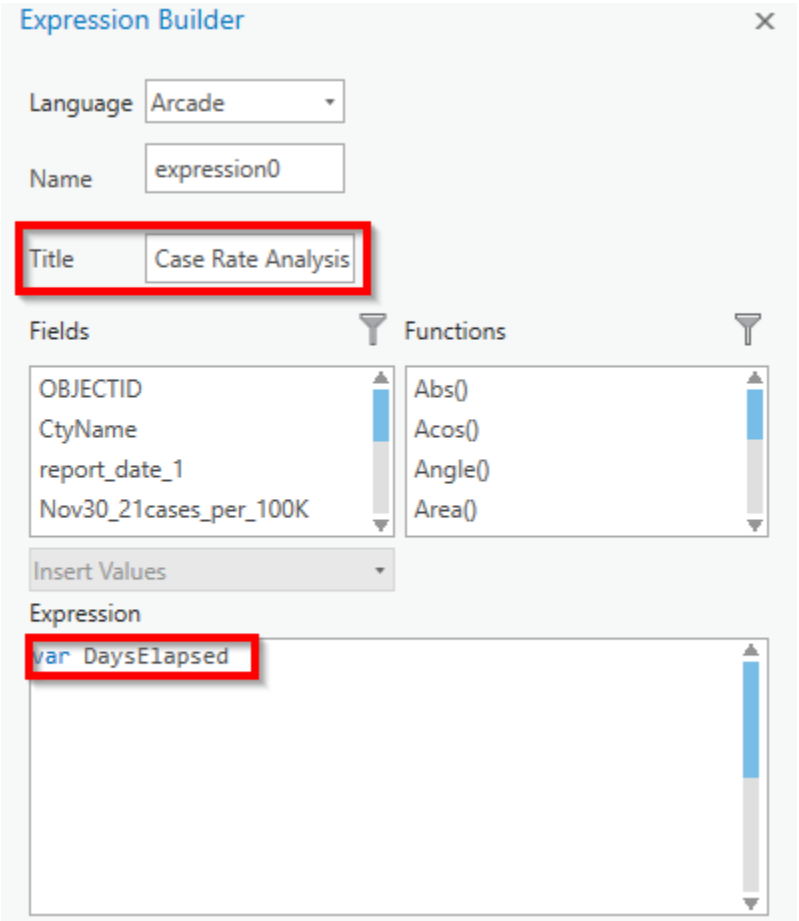
9. Uncheck the **box** to the left of word “Display” to exclude all the fields from our pop-up, and click the **back arrow** to continue.



10. In the Configure Pop-ups window, click **Expression**, then select **New** to create a new expression.

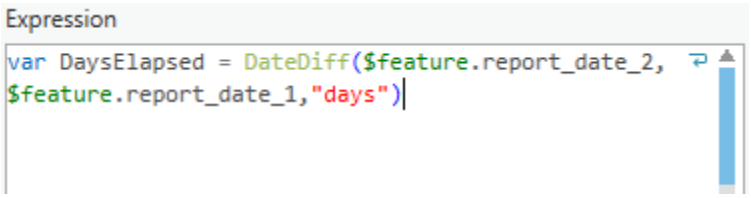
11. In the Expression Builder window, give the new expression the Title of **Case Rate Analysis**.

12. In the Expression area, we will first create a variable called **DaysElapsed**



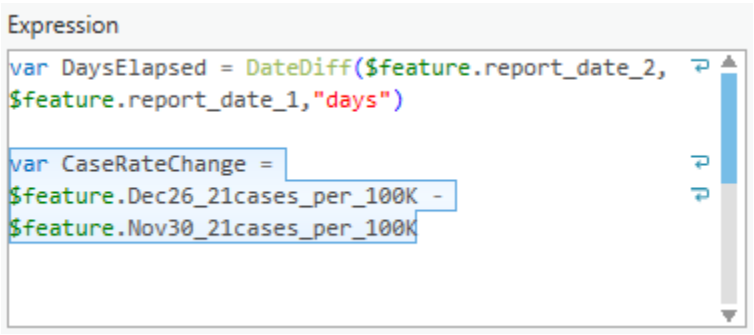
13. Since we are working with dates, we will use the **DateDiff** function in our expression to calculate the number of days that elapsed between 11/30/21 (i.e. report_date_1) and 12/26/21 (i.e. report_date_2). Define the DaysElapsed variable create in the previous step with the following expression:

    ```
    DateDiff($feature.report_date_2,$feature.report_date_1,"days")
    ```

14. Next, we will create a second variable called **CaseRateChange** which will store the changes in case rate between 11/30/21 and 12/26/21. Define the variable with the following expression:

    ```
    $feature.Dec26_21cases_per_100K - $feature.Nov30_21cases_per_100K
    ```

15. Since ArcGIS has a tendency to assume the date in the attribute table are in the UTC timezone and automatically changes it to the timezone it thinks you are in, we will first create a third variable called **Date1**, then define it using the **ToUTC** function in order to prevent ArcGIS from automatically changing our first date value (i.e. report_date_1). Create and define the Date1 variable with the following expression:

    ```
    var Date1 = ToUTC($feature.report_date_1)
    ```

Expression

```
var DaysElapsed = DateDiff($feature.report_date_2,
$feature.report_date_1,"days")

var CaseRateChange =
$feature.Dec26_21cases_per_100K -
$feature.Nov30_21cases_per_100K

var Date1 = ToUTC($feature.report_date_1)
```

16. Next, we will create a fourth variable called **DayofWeek1**, and use the **Weekday** function to spell out the day of the week that 11/30/21 (i.e. Date1) fell on. Create and define the DayofWeek1 variable with the following expression:

    var DayofWeek1 = Weekday(Date1)

Expression

```
var CaseRateChange =
$feature.Dec26_21cases_per_100K -
$feature.Nov30_21cases_per_100K

var Date1 = ToUTC($feature.report_date_1)

var DayofWeek1 = Weekday(Date1)
```

17. Since the Weekday function will only return integers corresponding to the day of the week (e.g. 0 = Sunday, … , 5 = Friday), we will create a fifth variable called **DayName1** and use the **Decode** function to translate the integer values returned by the Weekday function into days of the week. Note that we only used the Decode function to translate the values of 0 through 5 in our expression below, and all other values will be translated to Saturday.

    var DayName1 = Decode(DayofWeek1, 0,"Sunday", 1, "Monday", 2, "Tuesday", 3, "Wednesday", 4, "Thursday", 5, "Friday", "Saturday")

Expression

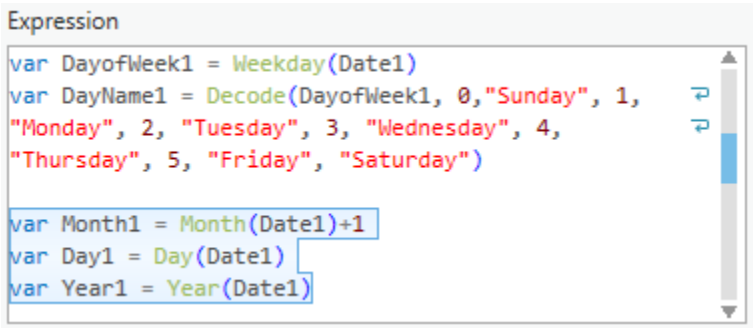
```
$feature.Nov30_21cases_per_100K

var Date1 = ToUTC($feature.report_date_1)

var DayofWeek1 = Weekday(Date1)
var DayName1 = Decode(DayofWeek1, 0,"Sunday", 1,
"Monday", 2, "Tuesday", 3, "Wednesday", 4,
"Thursday", 5, "Friday", "Saturday")
```

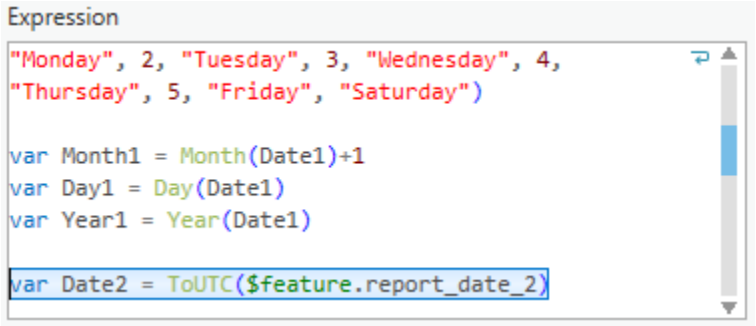
18. We will now create three additional variables (i.e. **Month1, Day1, Year1**) to respectively extract and store the Month, Day, and Year information from the report_date_1 field. Create and define the three new variables with the following expressions. Note that for the Month1 variable's expression, we must add one to the value returned by the Month function, otherwise that function will return values starting with 0 for January and 11 for December.

```
var Month1 = Month(Date1)+1
var Day1 = Day(Date1)
var Year1 = Year(Date1)
```



19. In the next few steps, we will repeat some of the previous steps to create variables that will store the information coming from the report_date_2 field. Create a variable called **Date2** and define it using the **ToUTC** function to prevent ArcGIS from automatically changing our second date value (i.e. report_date_2). Create and define the Date2 variable with the following expression:

```
var Date2 = ToUTC($feature.report_date_2)
```



20. Much like before, we will create a new variable called **DayofWeek2**, and use the **Weekday** function to spell out the day of the week that 12/26/21 (i.e. Date2) fell on. Create and define the variable with the following expression:

```
var DayofWeek2 = Weekday(Date2)
```

Expression

```
var Month1 = Month(Date1)+1
var Day1 = Day(Date1)
var Year1 = Year(Date1)

var Date2 = ToUTC($feature.report_date_2)

var DayofWeek2 = Weekday(Date2)
```

21. Create a variable called **DayName2** and use the **Decode** function to translate the integer values returned by the Weekday function.

```
var DayName2 = Decode(DayofWeek2, 0,"Sunday", 1, "Monday", 2, "Tuesday", 3, "Wednesday", 4, "Thursday", 5, "Friday", "Saturday")
```

Expression

```
var Date2 = ToUTC($feature.report_date_2)

var DayofWeek2 = Weekday(Date2)

var DayName2 = Decode(DayofWeek2, 0,"Sunday", 1,
"Monday", 2, "Tuesday", 3, "Wednesday", 4,
"Thursday", 5, "Friday", "Saturday")
```

22. Create three more variables (i.e. **Month2, Day2, Year2**) to respectively extract and store the Month, Day, and Year information from the report_date_2 field. Define the three new variables with the following expressions.

```
var Month2 = Month(Date2)+1
var Day2 = Day(Date2)
var Year2 = Year(Date2)
```

Expression

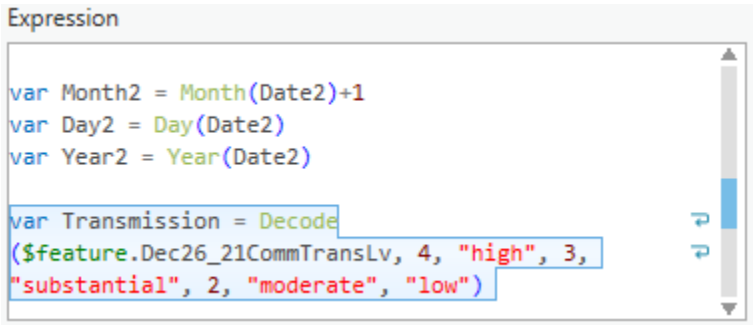
```
var DayName2 = Decode(DayofWeek2, 0,"Sunday", 1,
"Monday", 2, "Tuesday", 3, "Wednesday", 4,
"Thursday", 5, "Friday", "Saturday")

var Month2 = Month(Date2)+1
var Day2 = Day(Date2)
var Year2 = Year(Date2)
```

23. The Dec26_21CommTransLv field specifies different county's level of community transmission as classified by the Centers for Disease Control and Prevention. It ranges from a

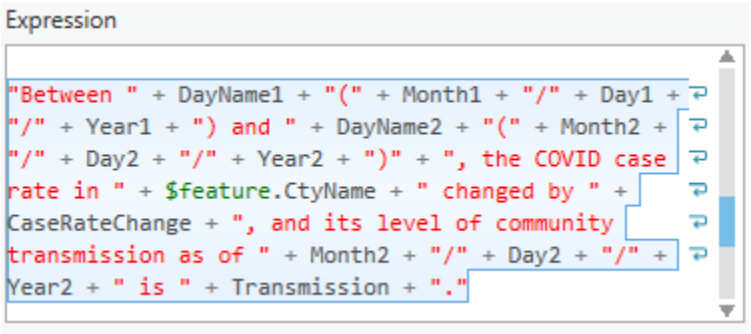
value of 4, which corresponds to a high level of transmission, to a value of 1, which corresponds to a low level of transmission. Create a variable called **Transmission** using the Decode function to translate the meaning of the integer values found in the Dec26_21CommTransLv field. Note that we only used the Decode function to translate the values of 4 through 2 in our expression below, and all other values will be defined as low.

```
var Transmission = Decode($feature.Dec26_21CommTransLv, 4, "high", 3, "substantial", 2, "moderate", "low")
```



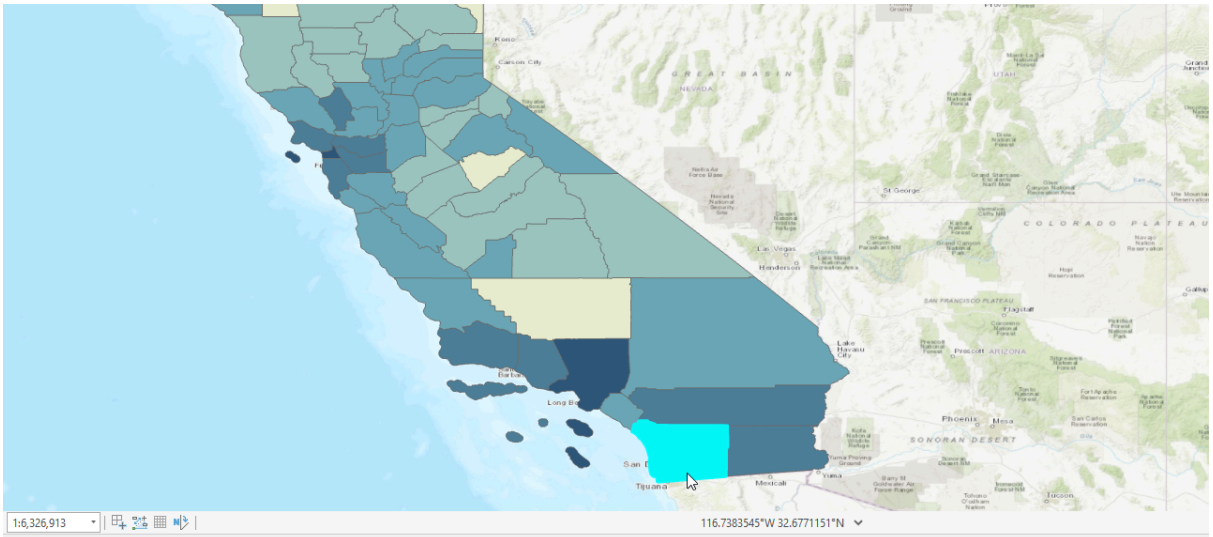
24. Finally, we will compile a sentence that make use of all the variables that we created in this section of the exercise.

```
"Between " + DayName1 + "(" + Month1 + "/" + Day1 + "/" + Year1 + ") and " + DayName2 + "(" + Month2 + "/" + Day2 + "/" + Year2 + ")" + ", the COVID case rate in " + $feature.CtyName + " changed by " + CaseRateChange + ", and its level of community transmission as of " + Month2 + "/" + Day2 + "/" + Year2 + " is " + Transmission + "."
```



Click the **Verify** icon ✔ to check your expression and click **OK**. Close the Configure Pop-ups window.

25. Click on any features in the CaseRate11_30vs12_26 layer within the map, you should now see the sentence that you configured in the previous step in the pop-up window.


1:6,326,913
116.7383545°W 32.6771151°N
Pop-up
CaseRate11_30vs12_26 (1)
San Diego County
CaseRate11_30vs12_26 - San Diego County
Case Rate Analysis Between Tuesday(11/30/2021) and Sunday(12/26/2021), the COVID case rate in San Diego County changed by 226.49, and its level of community transmission as of 12/26/2021 is high.

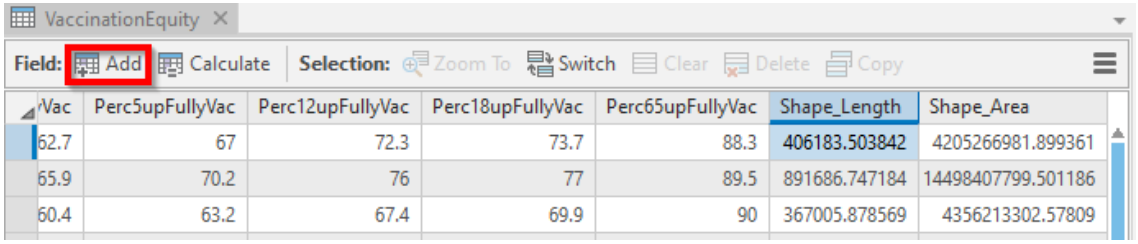
Part 6: Working with geometry in Arcade

1. Open ArcGIS Pro. Start a new project and new map. Add the **VaccinationEquity** feature class from the CasesTutorial.gdb geodatabase to your map. This feature class contains information on the social vulnerability index and COVID vaccination progress (as of 12/30/2021) in different California counties according to the Centers for Disease Control and Prevention (CDC) (https://covid.cdc.gov/covid-data-tracker/#vaccination-equity). Note that eight California counties with missing data were intentionally deleted from the layer.

Part 6.1 Calculating feature area

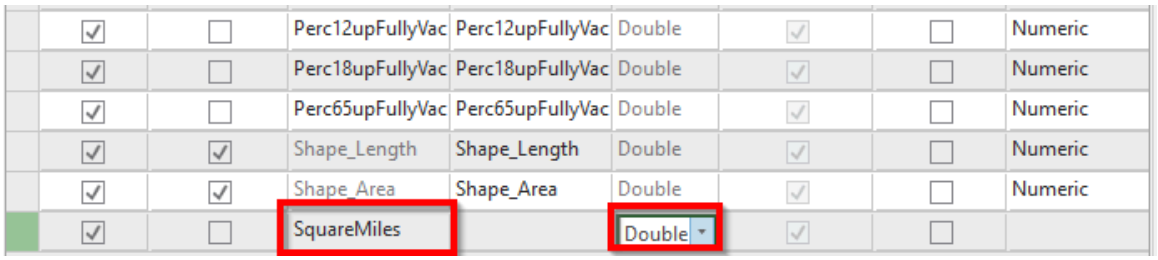
2. Right click the VaccinationEquity layer in the Contents pane, select **Attribute Table** to open the layer's attribute table.

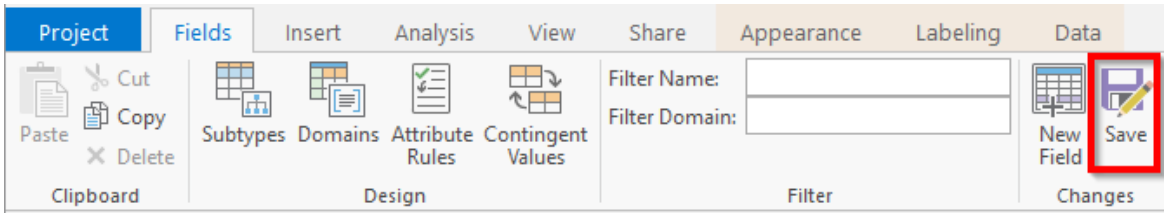
3. In the attribute table, you should see that there is a Shape_Area field that was automatically created by ArcGIS for the feature class. However, since we do not know the unit of measurement used for that field, we will create and calculate a new field to find out each feature's area in square miles. At the top of the attribute table window, click the **Add** icon next to the word Field.



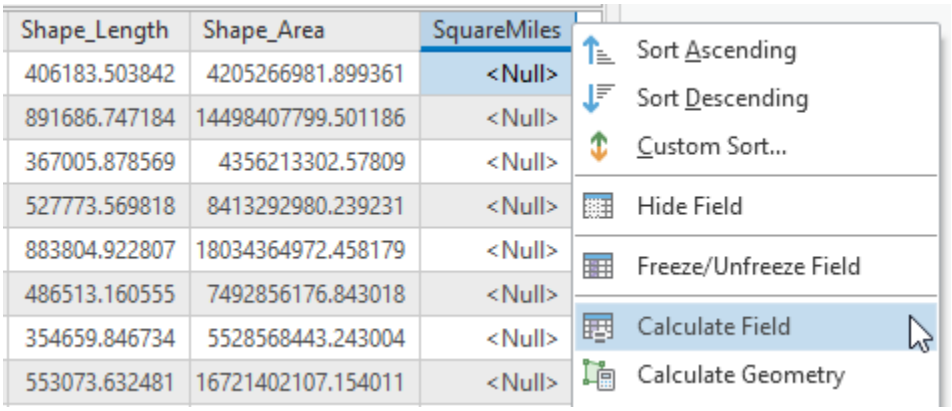
Vac	Perc5upFullyVac	Perc12upFullyVac	Perc18upFullyVac	Perc65upFullyVac	Shape_Length	Shape_Area
62.7	67	72.3	73.7	88.3	406183.503842	4205266981.899361
65.9	70.2	76	77	89.5	891686.747184	14498407799.501186
60.4	63.2	67.4	69.9	90	367005.878569	4356213302.57809

4. In the Fields window that opened, at the bottom of the table, give the new field the Field Name of **SquareMiles**, and the Data Type of **Double.** Then hit the **Save** icon in the Fields tab found in ArcGIS Pro's main menu. Close the Fields window.

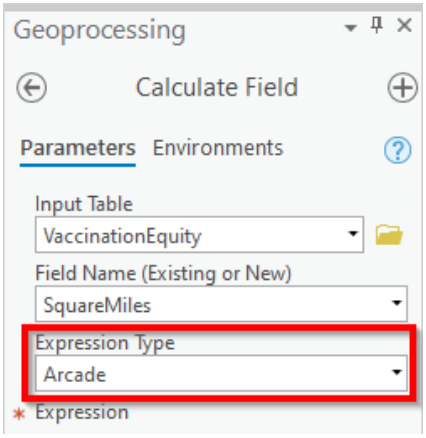




5. At the far-righthand side of the VaccinationEquity layer's attribute table, you should now see that a new field/column called SquareMiles has been created. Right click on the word **SquareMiles** in the attribute table and select **Calculate Field**.

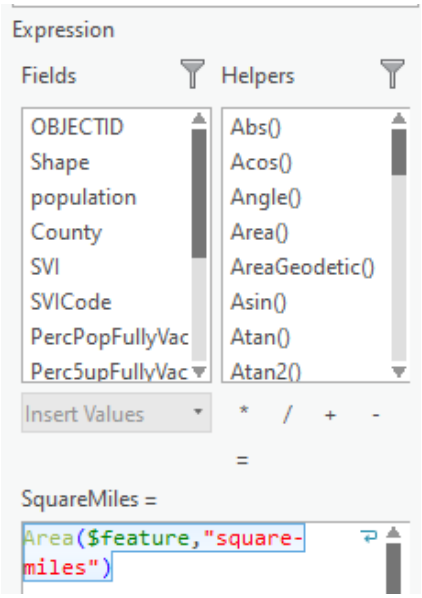


6. In the Calculate Field pane, change the Expression Type to **Arcade**.



7. In the empty area under the word "SquareMiles = ", type in the following expression:

```
Area($feature,"square-miles")
```

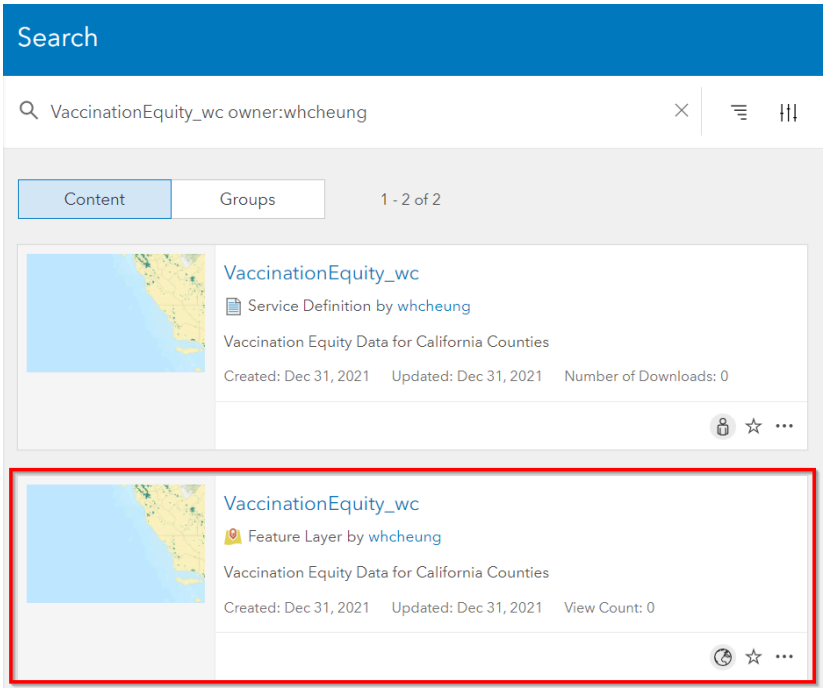


Click the **Verify** icon ✔ to check your expression and click **Run**. The area of each feature (in square miles) has now been calculated and stored in the SquareMiles column within the attribute table.

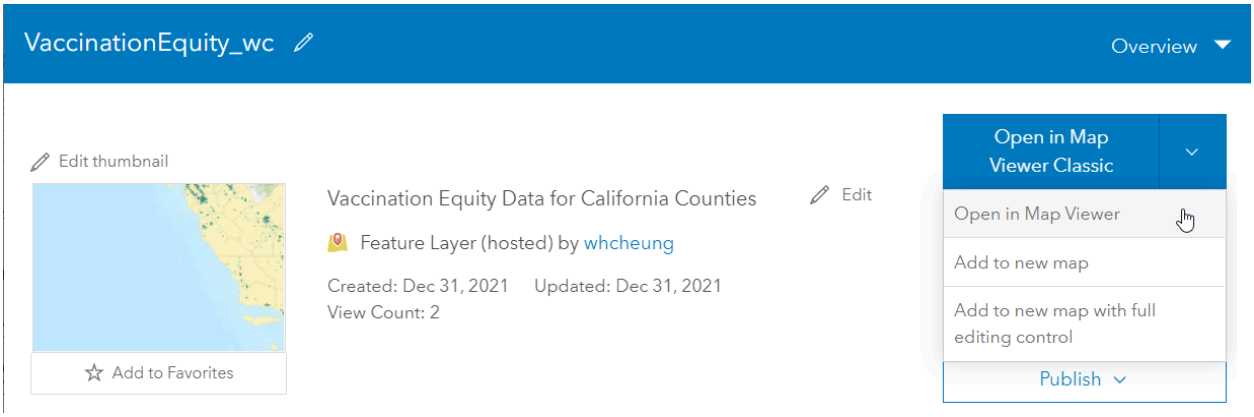
Part 6.2 Spatial analysis in ArcGIS Online with arcade expression

8. Open a web browser, navigate to https://www.arcgis.com/, and **sign in**to your ArcGIS Online account.

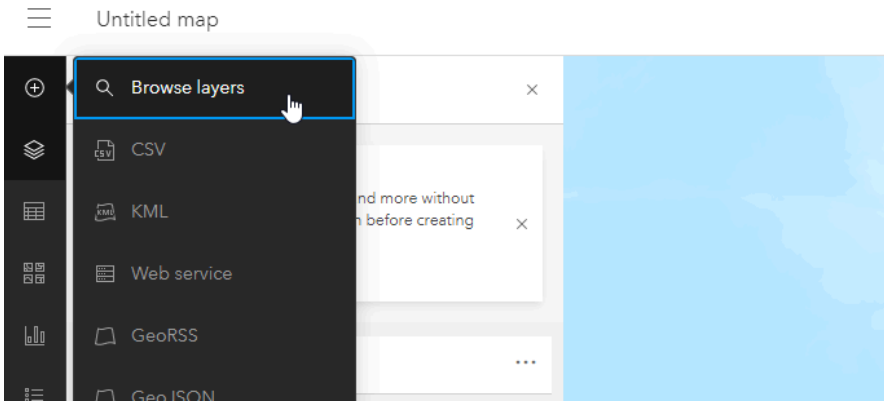
9. Search for **VaccinationEquity_wc owner:whcheung**. Be sure to click on the name of the **Feature Layer** and not the Service Definition.



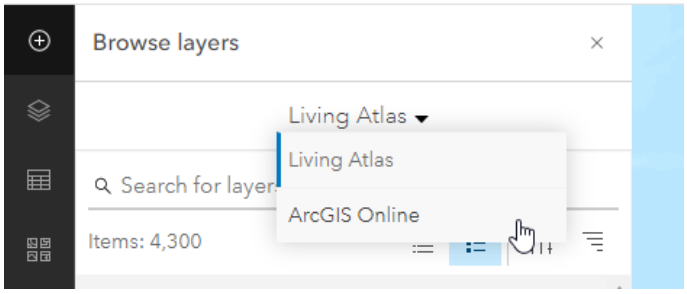
10. In the item description/overview page, select **Open in Map Viewer**. You should now see a map of California counties, much like the layer you worked with in the previous section.



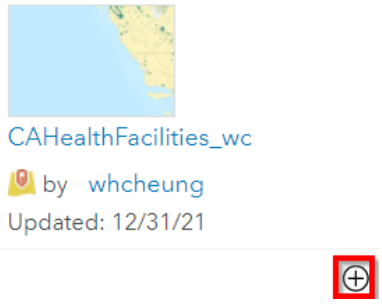
11. Within the map viewer, click **Add>Browse layers**.



12. In the Browse layers pane, change the dropdown to **ArcGIS Online**.

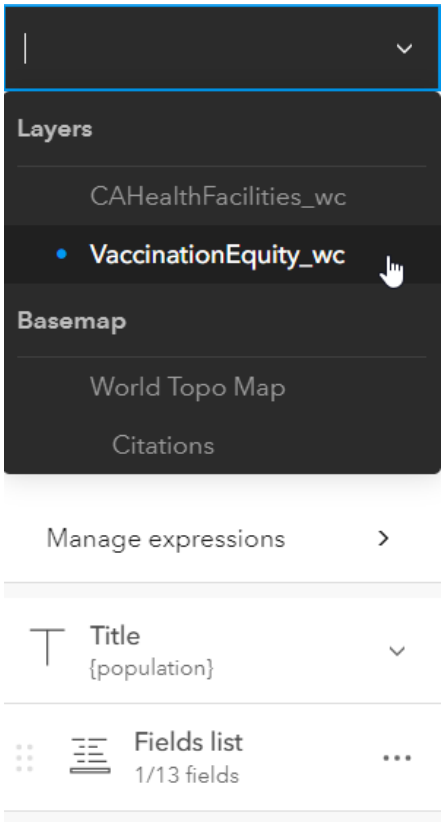


13. Search for **CAHealthFacilities_wc owner:whcheung**. Click the **Add** icon ⊕ in the lower right corner of the search result to add the layer to the map. In the following step, we will use arcade expression to help us find the number of hospitals within each California county using the data provided by the California Health and Human Services Agency.

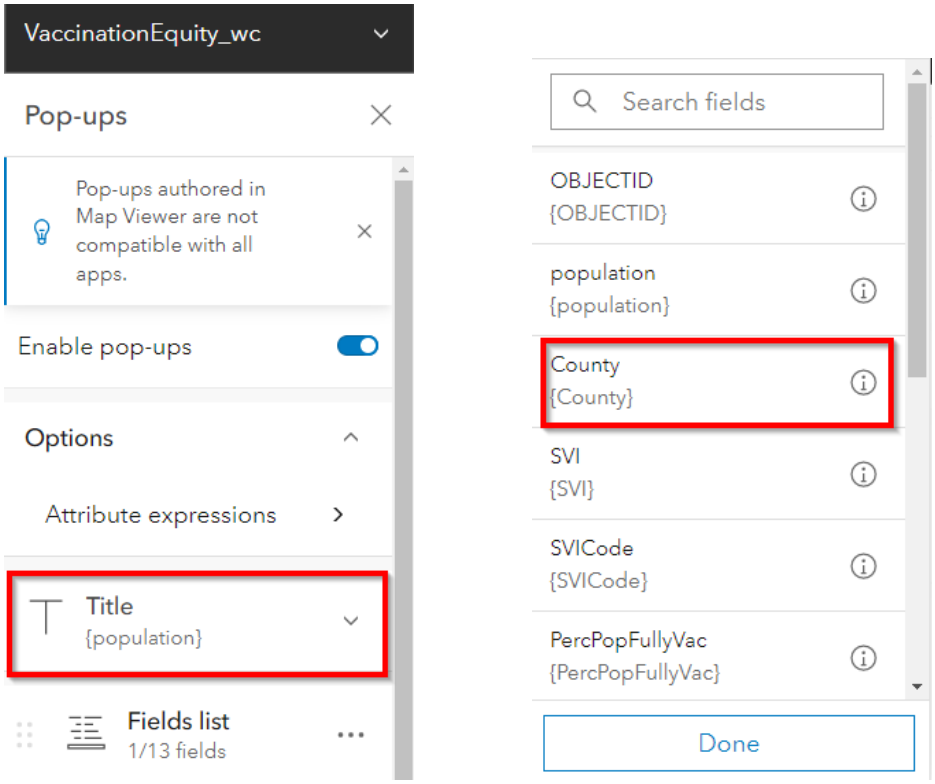


14. Click the **Configure Pop-ups** icon  on the right side of the map viewer window.

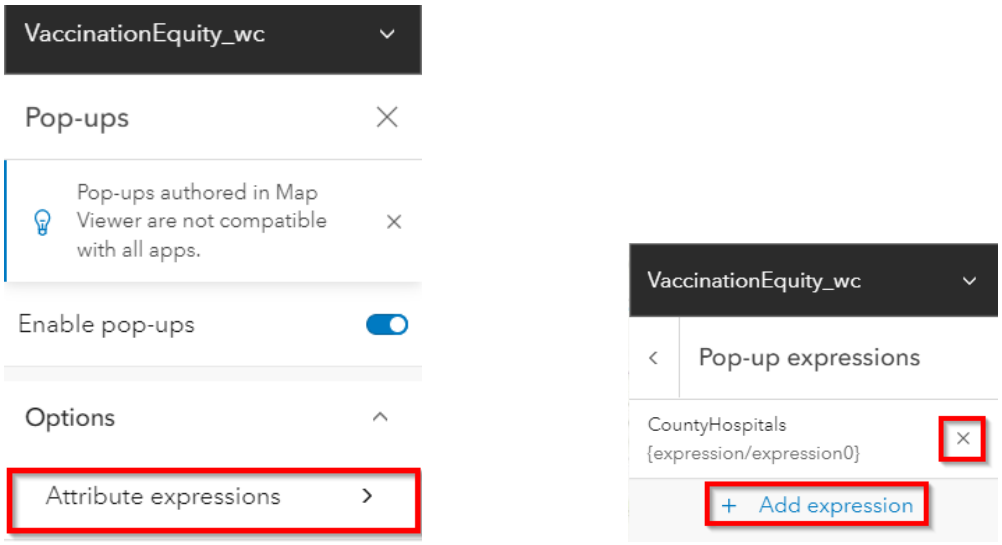
15. At the top of the Pop-ups pane, use the dropdown and be sure to select the name of the **VaccinationEquity_wc** layer.



16. In the Pop-ups pane, click **Title**, **delete** the existing value of {population}. Click the **curly brackets** icon { } , and select **County** as the attribute we want to display for the title of our pop-ups.

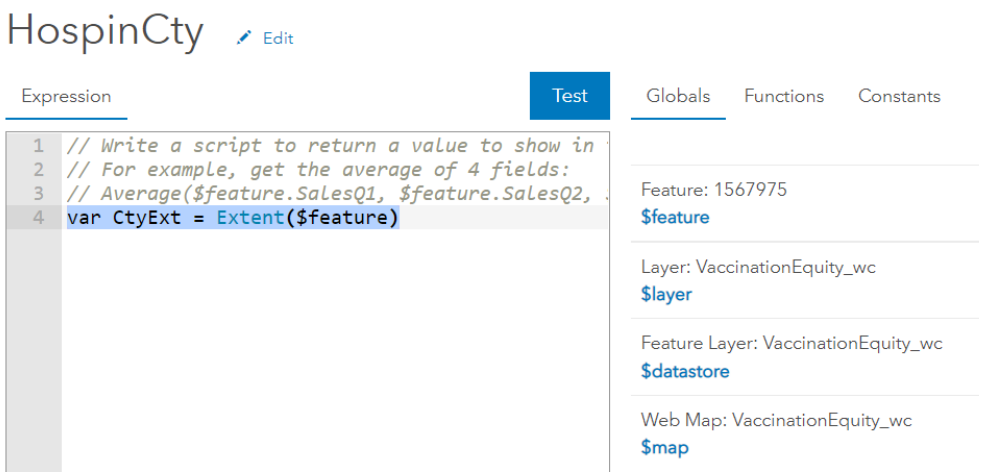


17. In the Pop-ups pane, click **Attribute expressions**, click **X** to delete existing expressions, and click **+ Add expression**.



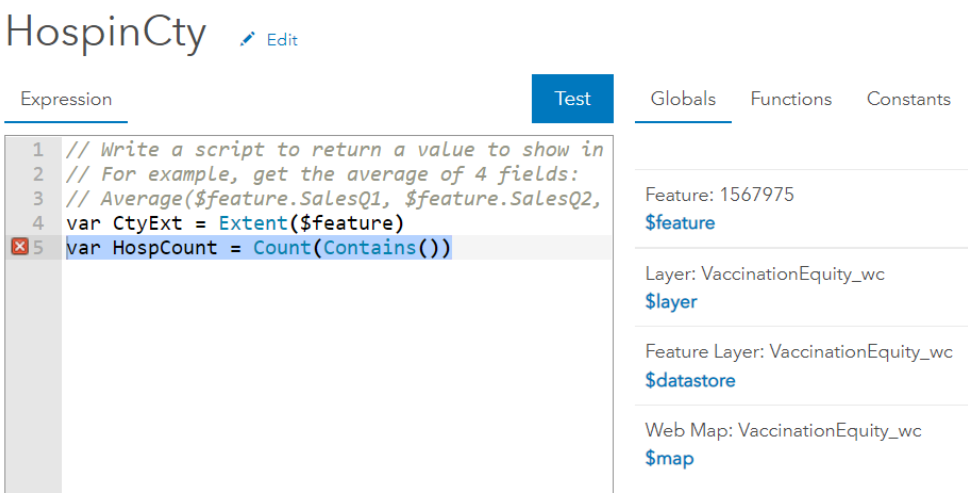
18. In the new Expression window, click **Edit** by the title, and replace the word "Custom" with **HospinCty**, and hit **Save**.
19. We will create a variable called **CtyExt** and use the **Extent()** function to establish the extent of each county. Within the Expression window, in the Expression box, enter the following expression:

    var CtyExt = Extent($feature)



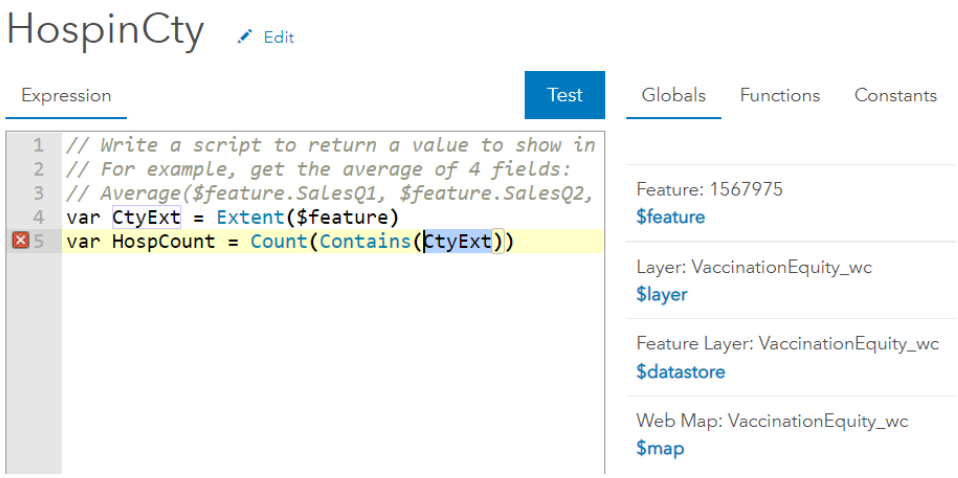
20. Start a new line of code in the Expression box, we will create a variable called **HospCount** and use the **Count()** and **Contains()** functions to count how many hospitals are contained within each county's extent. Enter the following expression:

    var HospCount = Count(Contains())

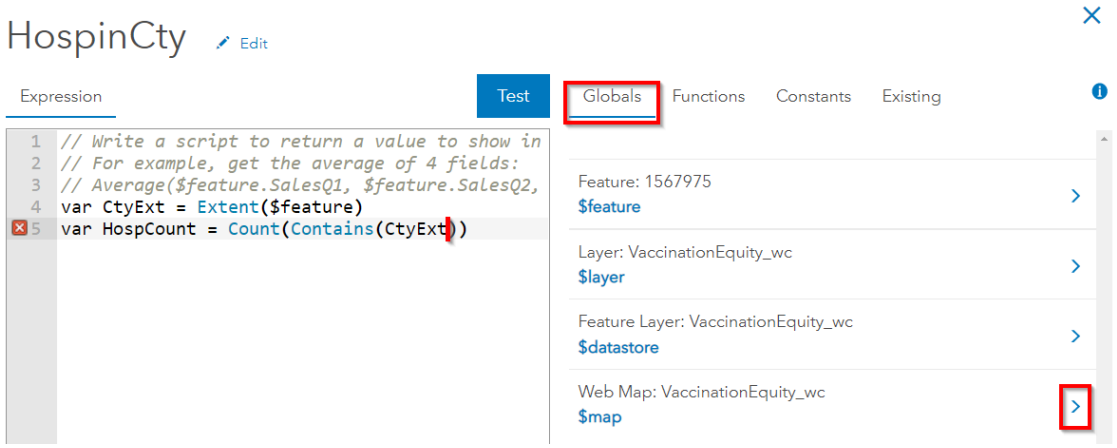


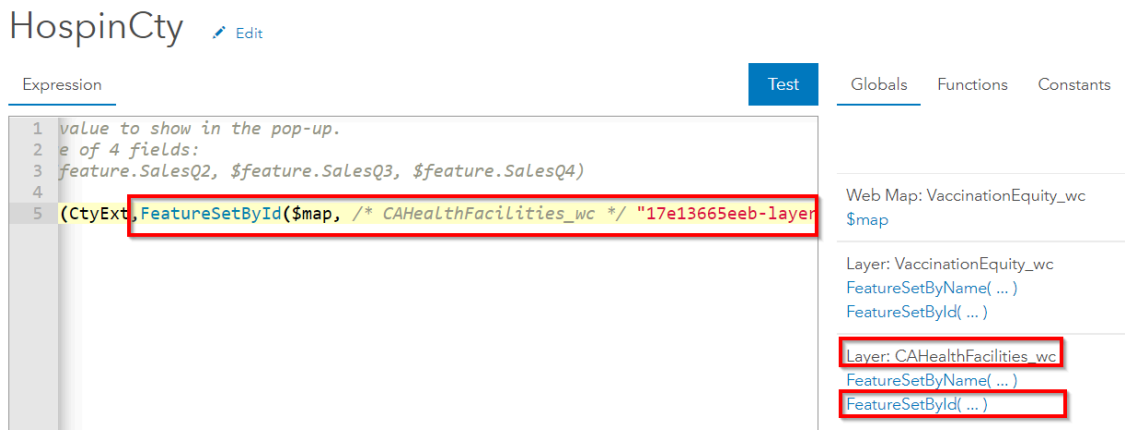
21. We now need to populate the parameters within the functions in the expression from the previous step. Firstly, within the parentheses of the Contains function, we need to type in the name of the variable that tells the expression the extent of each county (i.e. **CtyExt**).

    var HospCount = Count(Contains(**CtyExt**)))



22. Secondly, we need to specify the layer that contains our healthcare facilities feature. After the word CtyExt in the expression, add in a **comma**, then select the **>** sign to the right of the word $map under the Globals tab. Now select **FeatureSetById(…)** under the CAHealthFacilities_wc layer name.

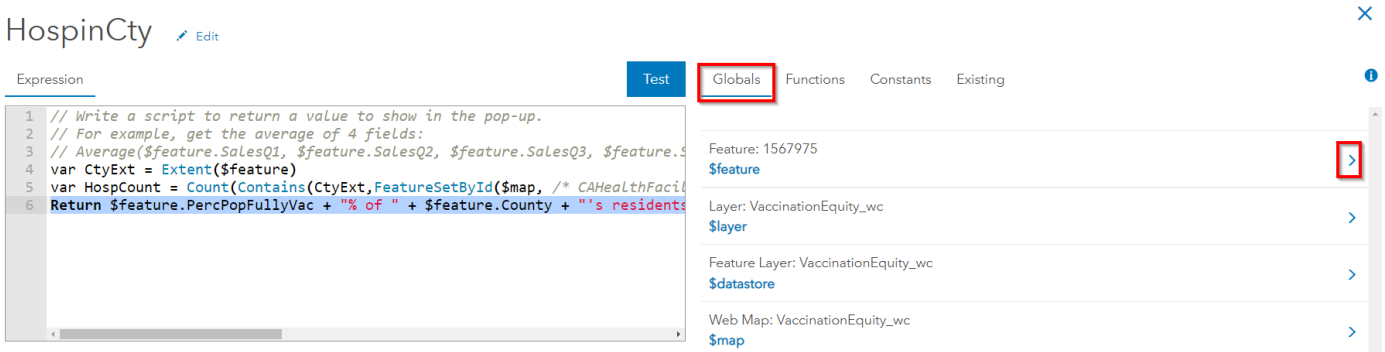


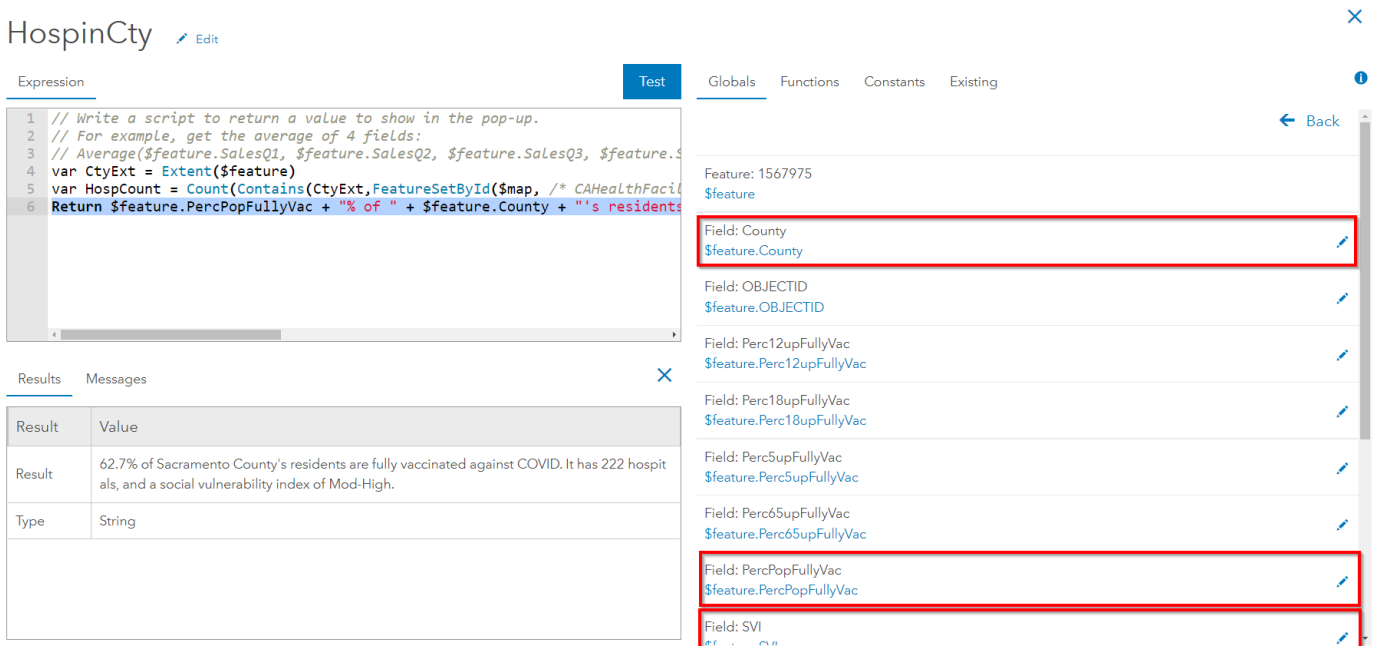


23. Start a new line of code in the Expression box by hitting the **Enter** key on your keyboard. We will add a return statement to embed our variables and attributes in the pop-up message. Enter the expression below.

Tip: You can click on the names of the layer's attributes (e.g. $feature.PercPopFullyVac, $feature.County, $feature.SVI) to insert them into the expression rather than manually typing them in. Do this by looking under the global tab, select the ***Back*** *icon if it is present* ← Back *, then select the* ***>*** *sign to the right of the word* ***$feature*** *under the Globals tab. You should now see the list of the layer's attribute names.*

Return $feature.PercPopFullyVac + "% of " + $feature.County + "'s residents are fully vaccinated against COVID. It has " + HospCount + " hospitals, and a social vulnerability index of " + $feature.SVI + "."

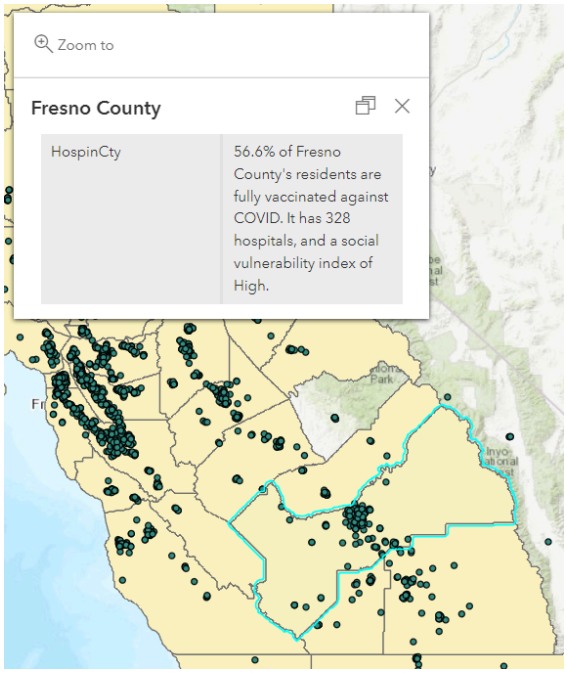




Click **Test** to ensure that your expression is working, and click **OK**.

32. Back in the map viewer, click on any of the county polygons. You should now see the newly configured message in the pop-up.

Tip: if your pop-up fails to update with the newly configured message, click the ***Configure pop-up*** *icon*  *on the right side of the map to open the Pop-ups pane. Expand the Fields list section of the pane, under Select fields, click the* ***X*** *next to HospinCty to delete it. Now click on* ***Select fields****, and reselect* ***HospinCty****, and hit* ***Done****. Close the Pop-ups pane.*

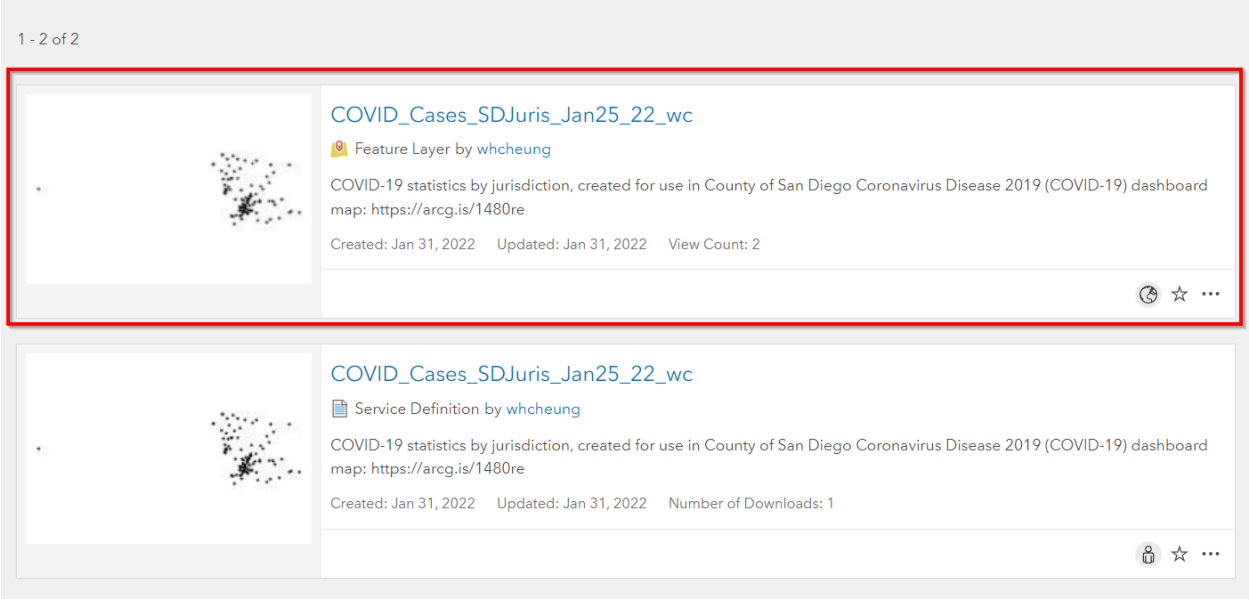


33. Save your map by clicking on the **Save and open** icon  on the left side of the map viewer window, and select **Save as**.

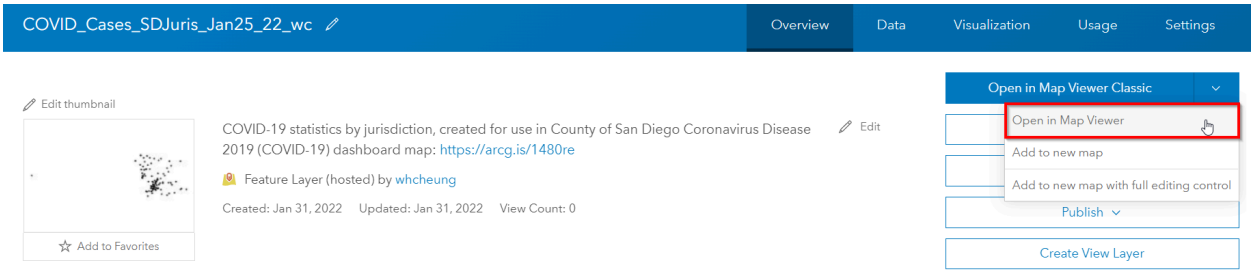
Part 7: Working with data dictionaries in ArcGIS Online

1. Open a web browser, navigate to https://www.arcgis.com/, and **sign in**to your ArcGIS Online account. In this exercise, we will work with a point layer that shows not only the location of various incorporated cities and unincorporated communities within San Diego County, but also the number of confirmed COVID cases recorded at those jurisdictions as of January 25, 2022.

2. Search for **COVID_Cases_SDJuris_Jan25_22_wc owner:whcheung**. Be sure to click on the name of **Feature Layer** result and not the Service Definition.

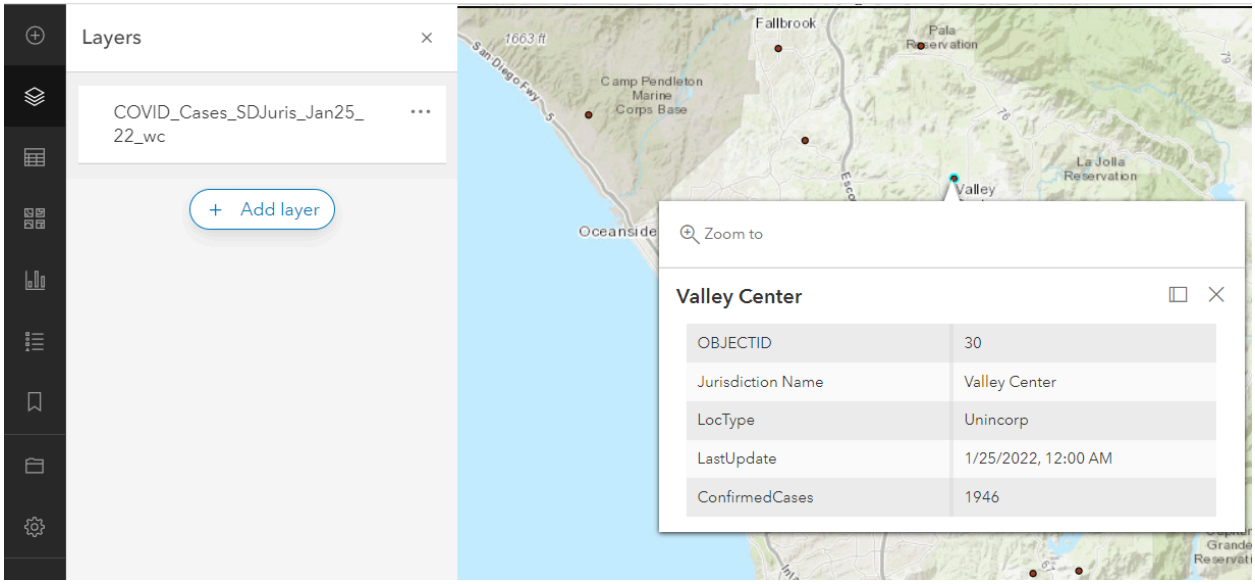


3. In the item description/overview page, select **Open in Map Viewer**. You should now see a map of points representing some of the various cities and unincorporated communities found in San Diego County.



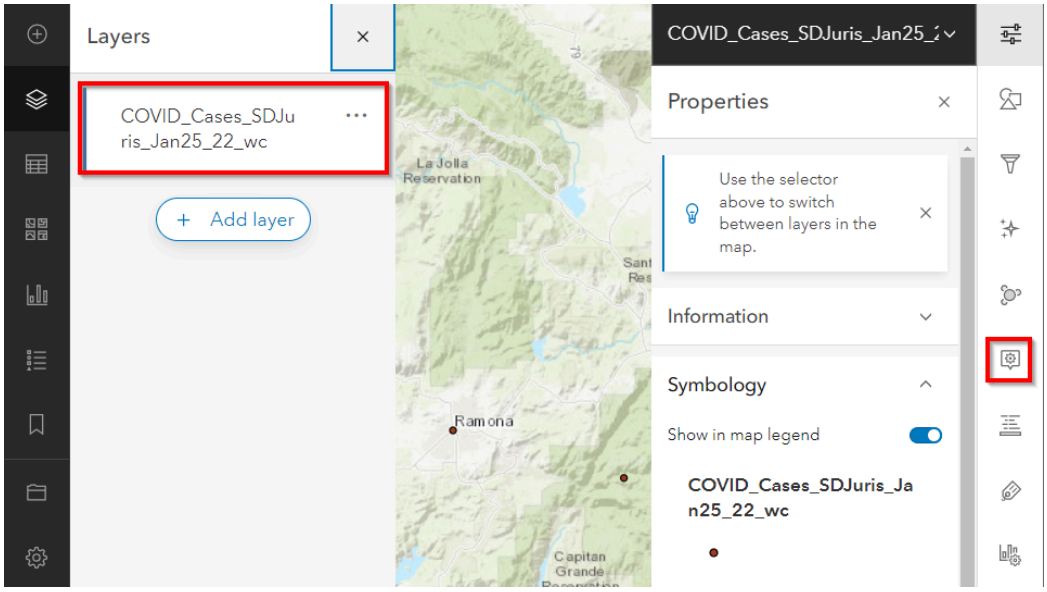
4. Once the layer is in your map viewer, click on any of the points in the map to view its pop-up window. In the next sections, we will use the data dictionary to translate the codes (i.e. Incorp, Unincorp) in the LocType field into complete descriptions (i.e. incorporated city,

unincorporated community), as well as include other attribute values in our new pop-up message.

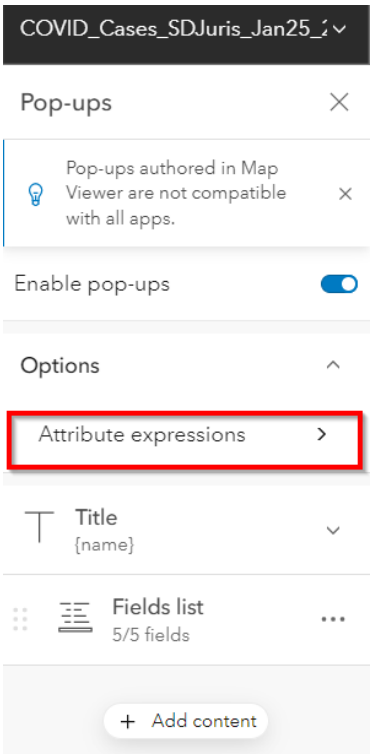


Part 7.1 Setting up the data dictionary and custom expression

1. Make sure that the **COVID_Cases_SDJuris_Jan25_22_wc** layer name is selected in the Layers pane, and click on the **Configure pop-ups** icon on the right side of the map viewer.



2. In the Pop-ups pane, click **Attribute expressions**, and select **+ Add expression**.



3. **Delete** the existing content that is in the Expression box.

4. We will first create a new variable called **LocCodes** to store our data dictionary. The data dictionary will be used to translate the abbreviations in the LocType field into full descriptions. Within the Expression window, enter the following:

```
var LocCodes = {
  "incorp":"incorporated city",
  "unincorp":"unincorporated community"
}
```

5. Next, in a new line within the Expression window, enter the following expression to create a variable called **Loc**, which will hold the values from the LocType field within our layer.

```
var Loc = $feature.loctype
```

6. We also need to create a new variable called **LocDesc** to store the full descriptions that have been translated from the abbreviations in the LocType field (which are now stored in the Loc variable) using the LocCodes dictionary. Enter the following expression into a new a line within the Expression window.

   var LocDesc = LocCodes[Loc]

7. We can also use the sum function calculate the sum of confirmed cases (this information is stored in the ConfirmedCases field within our layer). Enter the following expression into a new line within the Expression window to create a **SumCases** variable that will store the results.

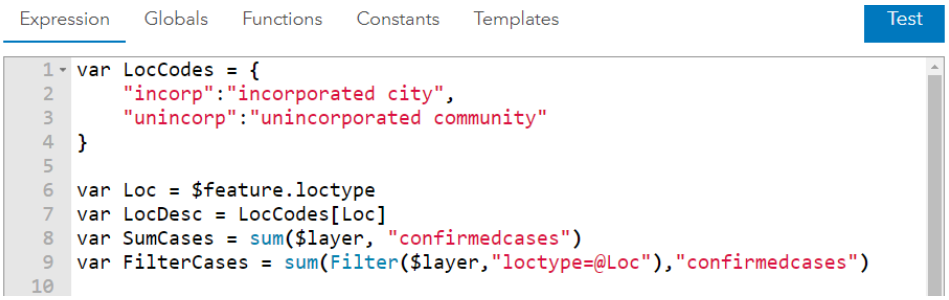
   var SumCases = sum($layer, "confirmedcases")

8. We will now expand on the previous line of code by using the filter function, so we can calculate the sum of confirmed cases for each type of jurisdiction (i.e. incorporated cities vs. unincorporated communities) instead of just getting the sum of cases for all of the jurisdictions. We will create a new variable called **FilterCases** by first **making a copy** of the line of code associated with the SumCases variable, and then **add in the following elements** as shown in bold below:

   var FilterCases = sum**(Filter**($layer**,"loctype=@Loc")**,"confirmedcases")

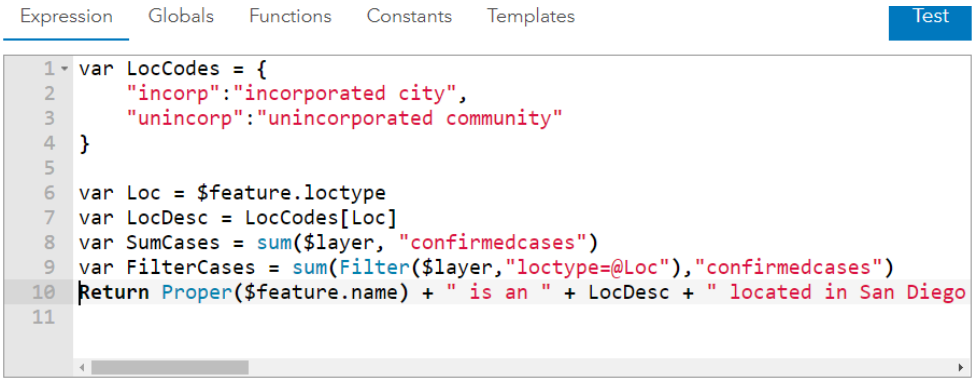
   Tip: In this expression, we are using a SQL (structured query language) expression, and substituting in an Arcade variable by using the @ character. Specifically, the @Loc term enables

us to only filter out those features that have the same loctype value as our currently selected feature and include them in our sum calculation.



9. Finally, we will use the Return function to create our new pop-up message. Enter the following expression in the Expression window, note the use of various functions that we previously covered in this class.

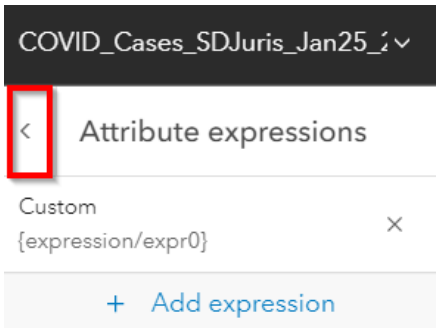
Return Proper($feature.name) + " is an " + LocDesc + " located in San Diego County. It recorded " + $feature.confirmedcases + " COVID cases as of Jan, 25, 2022. This represents " + Round($feature.confirmedcases/FilterCases*100,2) + "% of the total cases recorded in all of the " + LocDesc + " jursidiction, and " + Round($feature.confirmedcases/SumCases*100,2) + "% of the total cases recorded for the entire county."



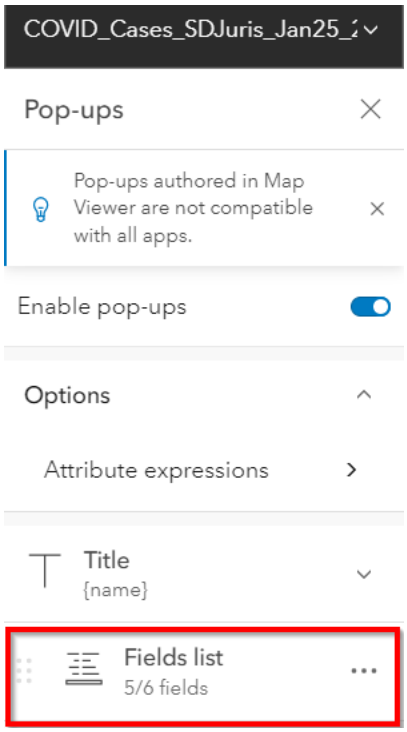
Click **Test** to ensure that your expression is working, and click **OK**.

Part 7.2 Updating pop-up with custom expression

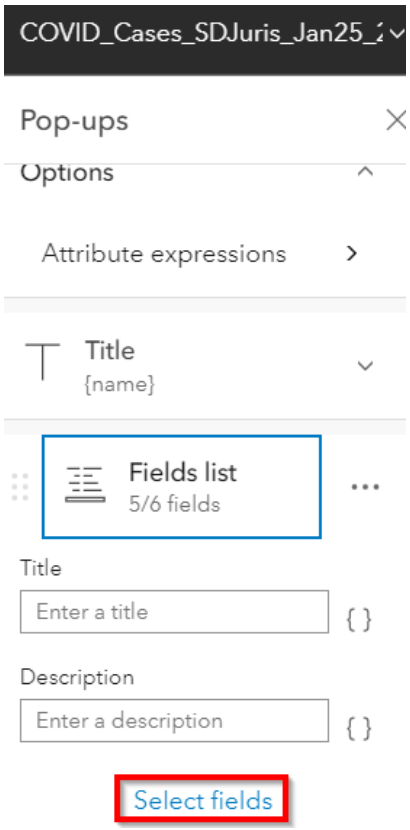
10. Finally, we need to add the custom expression we just created into the pop-up. In the Attribute expressions pane, click the **back arrow**.



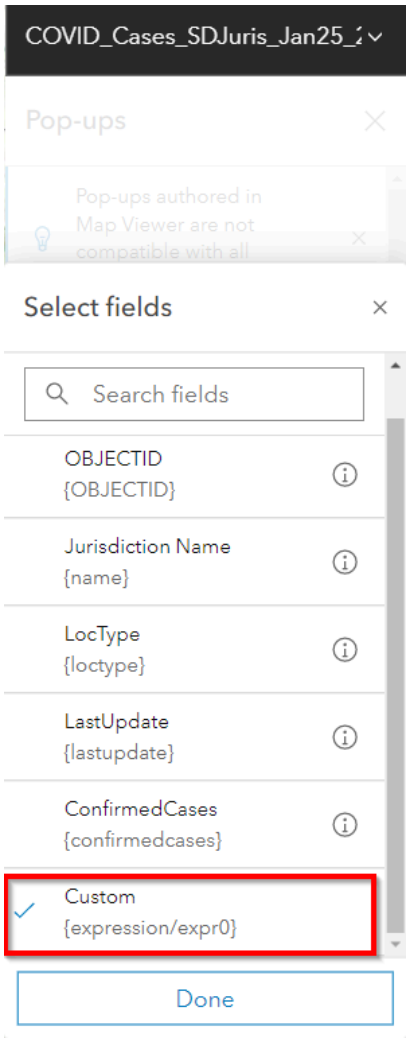
11. Within the pop-ups pane, select **Fields list.**



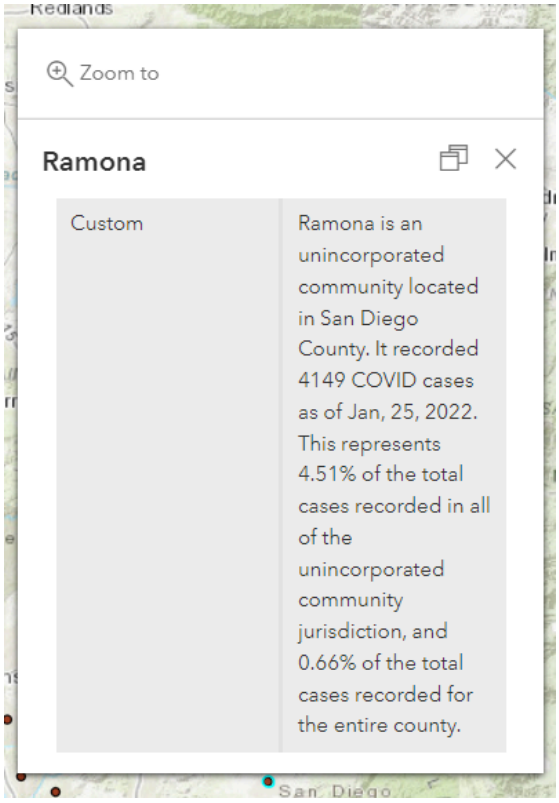
12. With the Fields list section expanded, click **Select fields**.



13. Within the Select fields dialog, check the **Custom** expression field that we just created, and **uncheck** all the other fields. Click **Done**.



14. Click on any features within your map. You should now see the updated pop-up with the paragraph that is driven by our custom expression.



15. Save your map by clicking on the **Save and open** icon  on the left side of the map viewer window, and select **Save as**. Fill out the Title, Tags, Summary for your map, and hit **Save map**.