

DATA SOCIETY:

Introduction to Tableau

Part 11



Recap: functions in Tableau

- Number Functions
- String Functions
- Date Functions
- Type Conversion
- Logical Functions
- Aggregate Functions
- Table Calculation Functions
- User Functions
- Spatial Functions
- Additional Functions

The screenshot displays three separate instances of Tableau's Table Calculation dialog box, each showing a different calculation and its corresponding Tableau code.

- Top Dialog:** The input field contains "years since independence". The resulting Tableau code is:

```
ZN(DATEDIFF("year", [Indep Year], TODAY(), "sunday"))
```
- Middle Dialog:** The input field contains "population rank". The resulting Tableau code is:

```
RANK(avg([Population (country.csv)]))
```

With a note below stating: "Results are computed along Table (across).".
- Bottom Dialog:** The input field contains "Pop Level". The resulting Tableau code is:

```
IF [Population (country.csv)] > 100000000  
THEN "high"  
ELSEIF [Population (country.csv)] > 50000000  
THEN "medium"  
ELSE "low"  
END
```

Each dialog box includes a status message at the bottom left ("The calculation is valid."), a dependency count ("1 Dependency"), and two buttons: "Apply" and "OK". To the right of the middle dialog is a sidebar titled "Table Calculation" containing a list of available functions: FIRST, INDEX, LAST, LOOKUP, PREVIOUS_VALUE, RANK, RANK_DENSE, RANK_MODIFIED, RANK_PERCENTILE, RANK_UNIQUE, RUNNING_AVG, RUNNING_COUNT, and RUNNING_MAX.

Warm-up: Tableau Public

- Spend a few minutes exploring the Cracking the Rubik's Cube dashboard from the Tableau Public Viz of the Day collection.
- How many different ways do you see the dashboard leverage **geospatial data?**



[Cracking the Rubik's Cube dashboard \(link\)](#)

Module completion checklist

Objective	Complete
Introduce the concept of geospatial visualization	
Implement geospatial visualization for given dataset	
Identify and correct errors to facilitate proper joining with geospatial data	
Introduce the concept of Dashboards	
Combine previously created visualizations into Dashboard	

Why is mapping useful?

- Spatial data is a prevalent data type, and can be especially useful with financial and economic data.
- Visualizing spatial data can uncover interaction between economic variables and geographic locations.
- Can you think of any such interactions with data you work with?

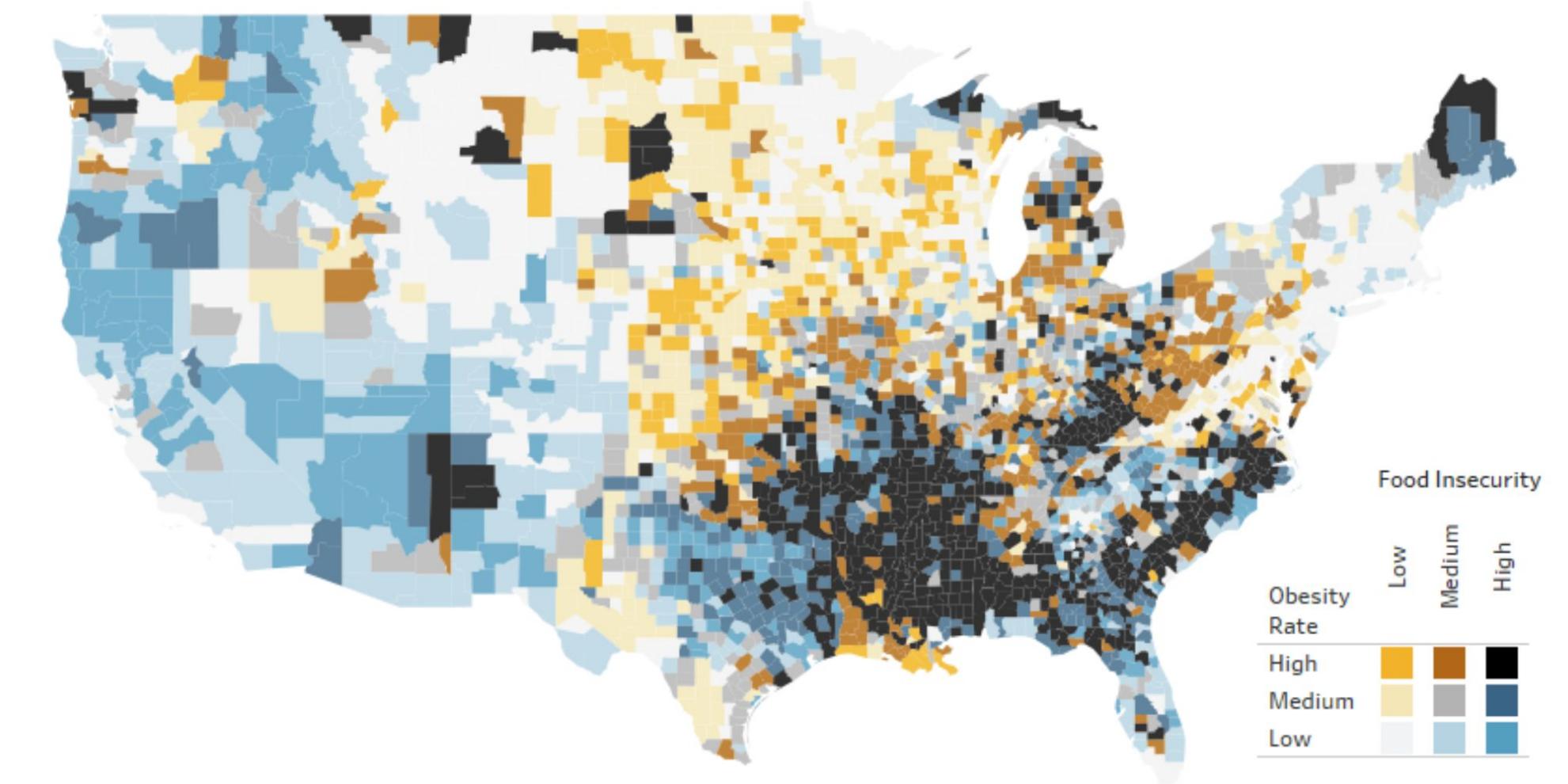


Mapping in Tableau

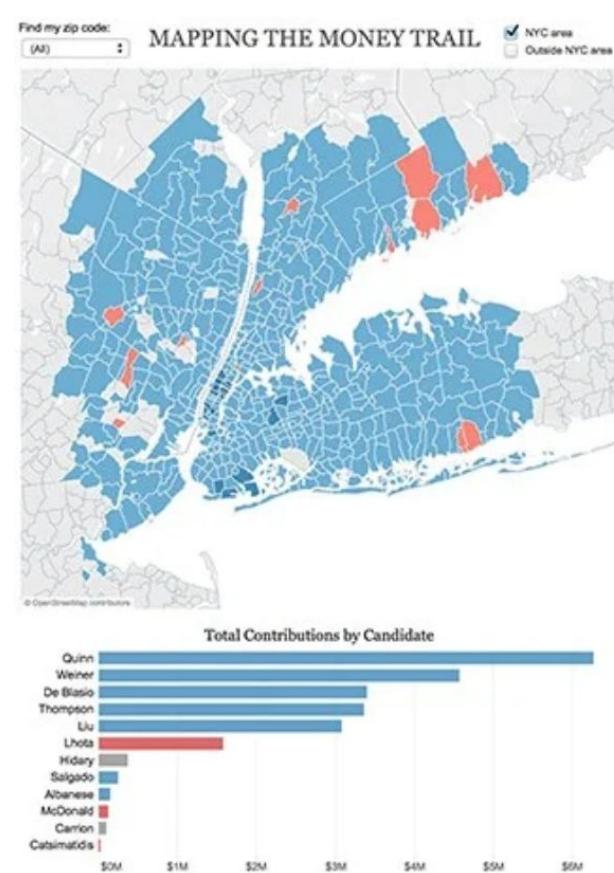
- **Geospatial visualization** is one of the most compelling uses for Tableau.
- Tableau makes mapping convenient by offering ready-made map types and plugins.
- Custom mapping capacity
 - Makes mapping custom regions possible.
 - Allows custom backgrounds and coordinates.
 - You could map spots on the sun or annotate regions of the brain.
- Tableau mapping can be extended through connections to more advanced tools, like Mapbox.

Geospatial visualization

- Some examples of where one would use maps include:
 - Obesity rates plotted across the US.
 - Crime events in Washington DC.
 - Campaign contributions by district in NYC.

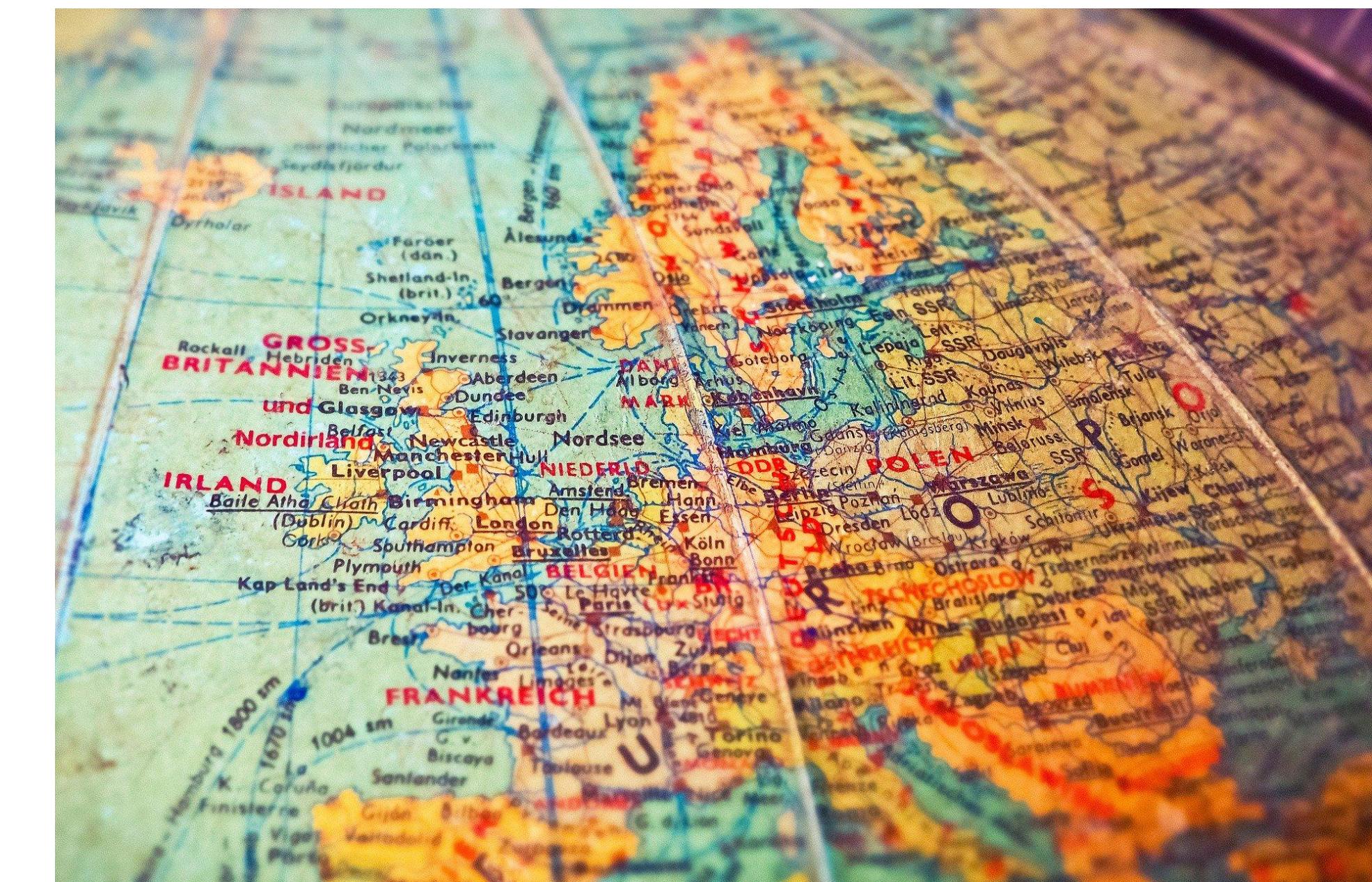


- You can see more examples of Tableau maps ([link](#)).



Automatic coordinate generation

- Tableau automatically recognizes data that naturally has coordinates, like countries, cities, counties, etc.
- For these data it **automatically generates longitude and latitude coordinates** as measures.



Reminder: save your work!

- In the next few modules, we will be creating more elaborate visualizations.
- We will see a lot of different insights from the data as we learn more in Tableau.
- **Make sure to save all your classwork** (including Exercises), because we will be putting it all together at the end of the unit to create a story.

Module completion checklist

Objective	Complete
Introduce the concept of geospatial visualization	✓
Implement geospatial visualization for given dataset	
Identify and correct errors to facilitate proper joining with geospatial data	
Introduce the concept of Dashboards	
Combine previously created visualizations into Dashboard	

Generating automatic coordinates

- Let's implement Tableau's mapping function using our world dataset.
- Notice that “**Name(country.csv)1**” in the country table has been assigned coordinates.
 - The globe indicates geospatial data.
- Note that other geospatial dimensions such as “**City**” were not recognized.
 - We will fix this later in the class.

The screenshot shows the Tableau Data pane with the following structure:

- Data** tab selected.
- Dimensions** section:
 - city.csv**: District, ID, Name (highlighted with a blue bar).
 - country.csv**: Code, Code2, Continent, Government Form, Head Of State, Indep Year, Local Name, Name (highlighted with a blue bar), Region1.
 - country.csv1**: add5 indep year, Calculation1 + 5.
- Measures** section:
 - # regional population, -# years since independence.
 - # Number of Records.
 - Latitude (generated), Longitude (generated) (highlighted with a blue bar).

A purple arrow points from the "Name (country.csv)1" dimension in the Dimensions section to a callout bubble labeled "Name (country.csv)1". Another purple arrow points from the "Latitude (generated)" and "Longitude (generated)" measures in the Measures section to a callout bubble labeled "Latitude (generated)" and "Longitude (generated)".

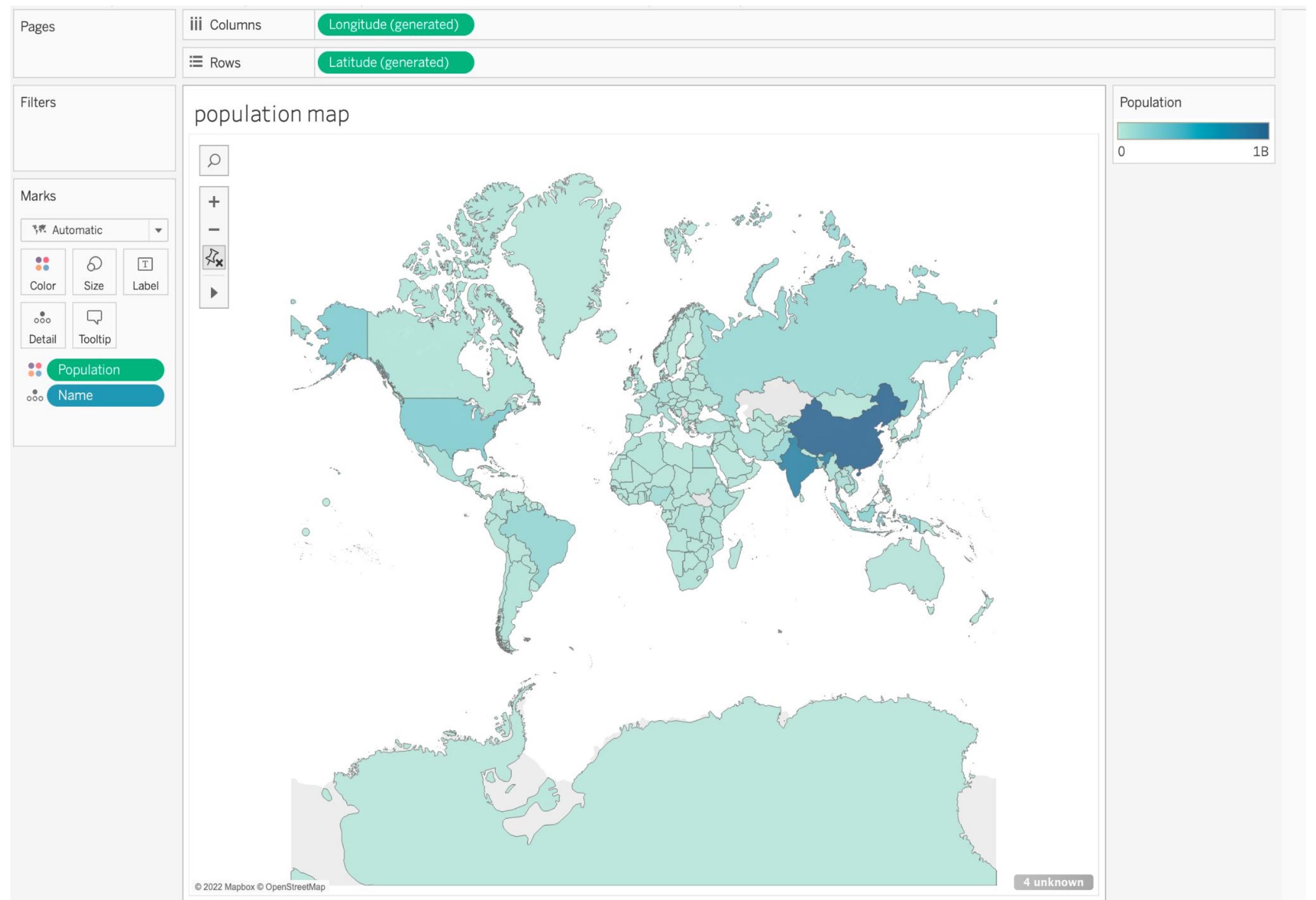
Manually converting to geospatial data

- Note that Tableau didn't recognize some of our geospatial data, like for “**Name(country.csv)**”
- To fix this, use the dropdown from the pill and tell Tableau which type of geospatial coordinates we have.
- In this case we have to tell Tableau that this “**Name**” is country data.

The screenshot shows the Tableau Data Source interface. On the left, the 'Tables' pane lists three CSV files: 'city.csv', 'country.csv', and 'countrylanguage.csv'. The 'country.csv' table is expanded, showing fields like 'Code', 'Code2', 'Continent', etc., with 'Name' highlighted. A purple arrow points from the 'Name' field in the 'country.csv' table to the context menu on the right. The context menu for 'Name' includes options like 'Add to Sheet', 'Duplicate', 'Rename', 'Hide', 'Aliases...', 'Create', 'Transform', 'Convert to Measure', 'Change Data Type', 'Geographic Role' (which is selected), 'Default Properties', 'Group by', and a dropdown menu with items such as 'None', 'Airport', 'Area Code (U.S.)', 'CBSA/MSA (U.S.)', 'City', 'Congressional District (U.S.)', and 'Country/Region'. The 'Drop field here' placeholder is visible on the right side of the interface.

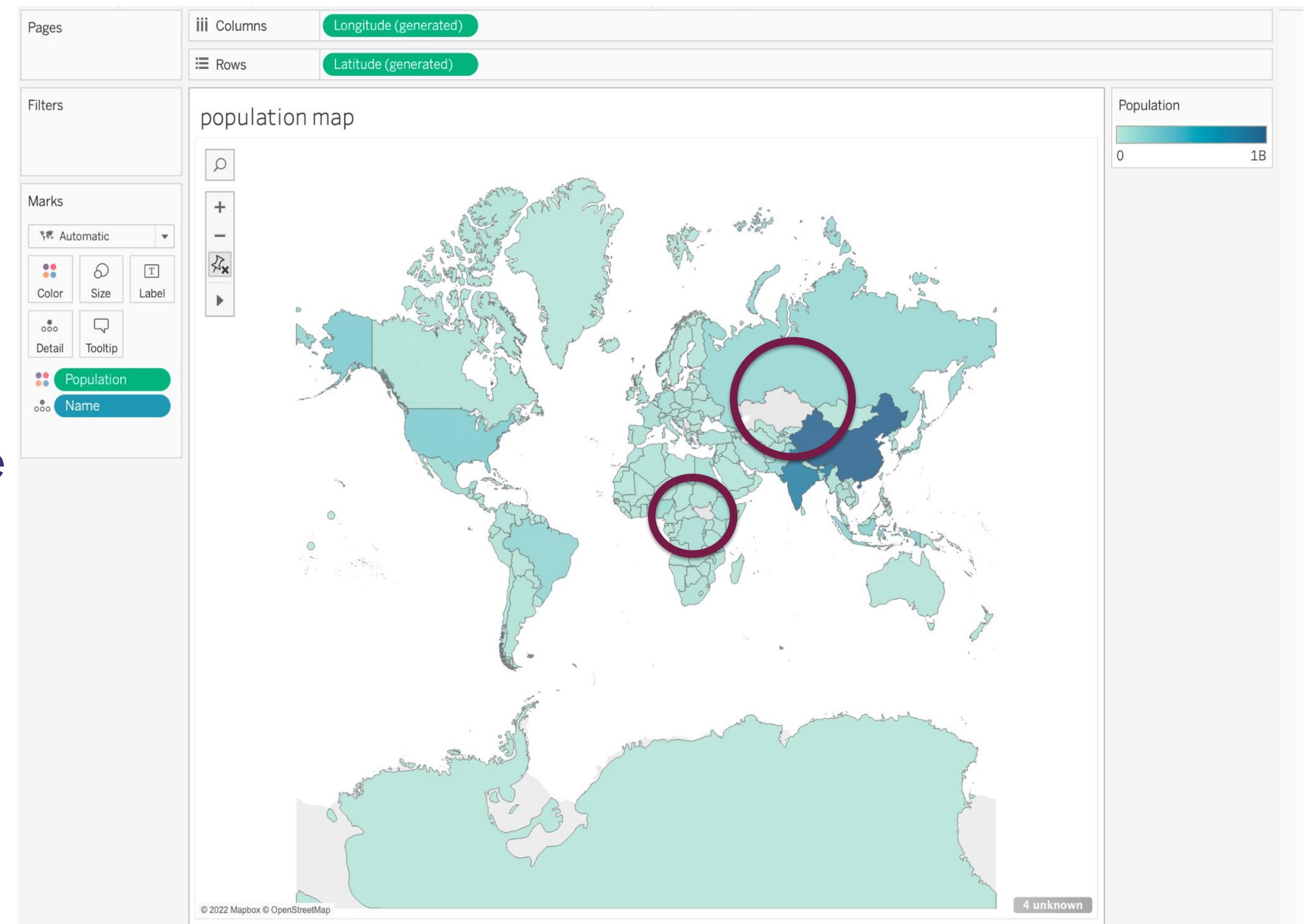
World map of population

- Together, we will walk through the process of generating a population map.
- Drag:
 - **Longitude** to **columns**.
 - **Latitude** to **rows**.
 - **Population** to the **marks color** panel.
 - **Country name** to the **detail** panel.
- Tableau automatically generates a filled map where we have data.



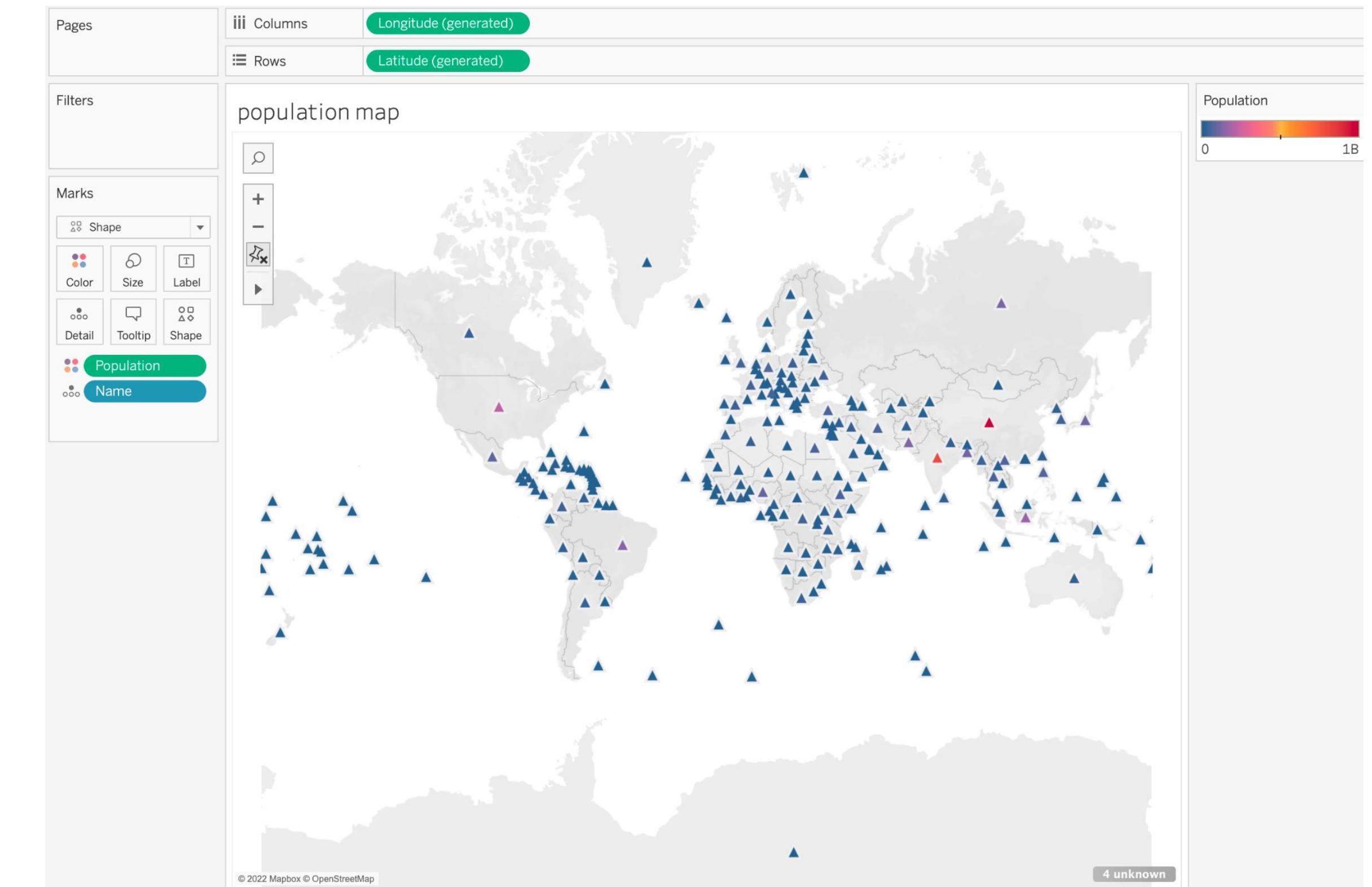
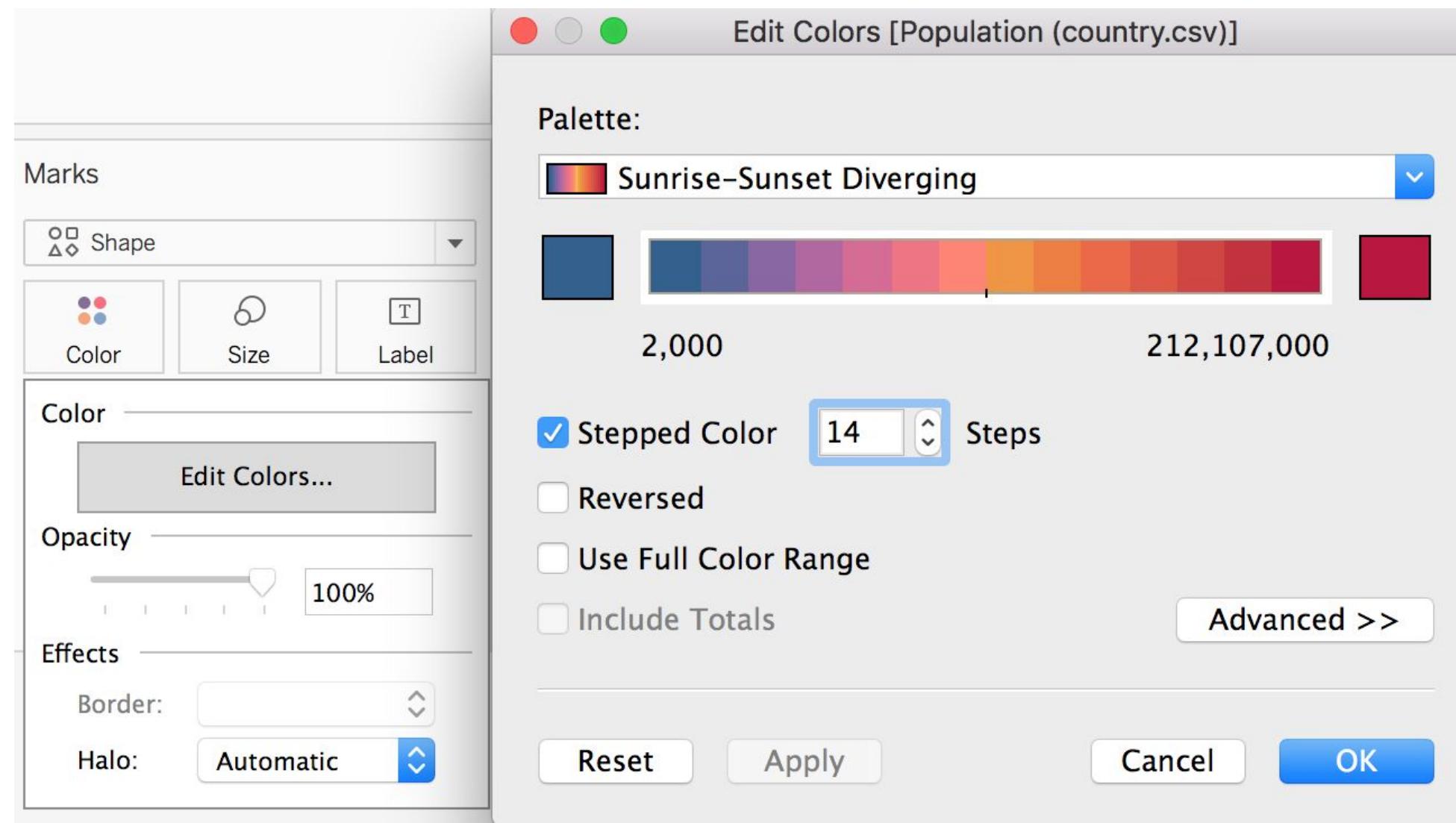
A few gray on that map...

- Remember how we found that there was missing population data way back in our first Tableau session?
- From this view, we can also see the missing data.



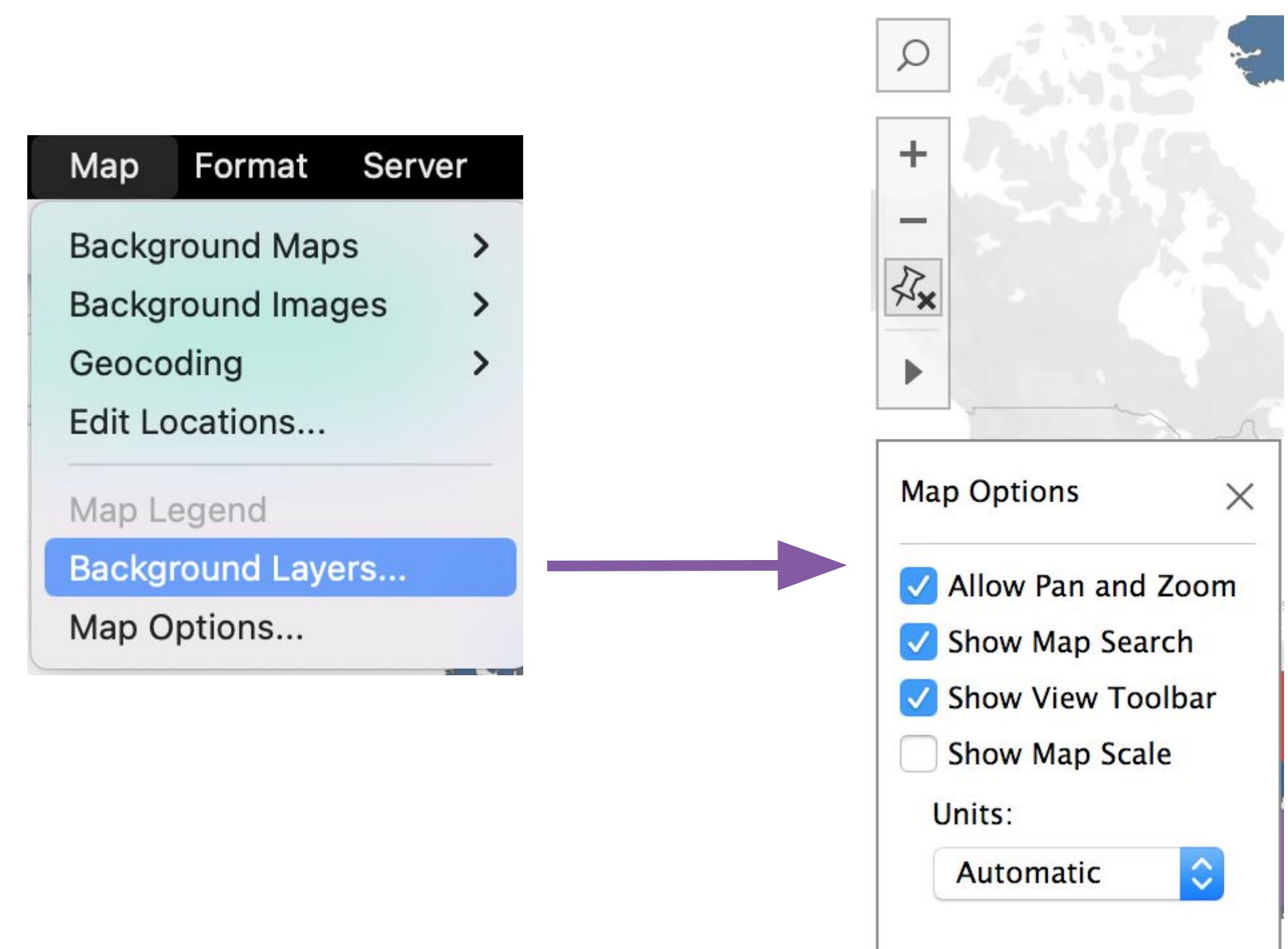
Symbols and filled maps

- It is easy to change the look of the map view with the marks dropdowns.
- Edit colors on the marks tab to give a more divergent range of colors so that we can distinguish less populous countries.



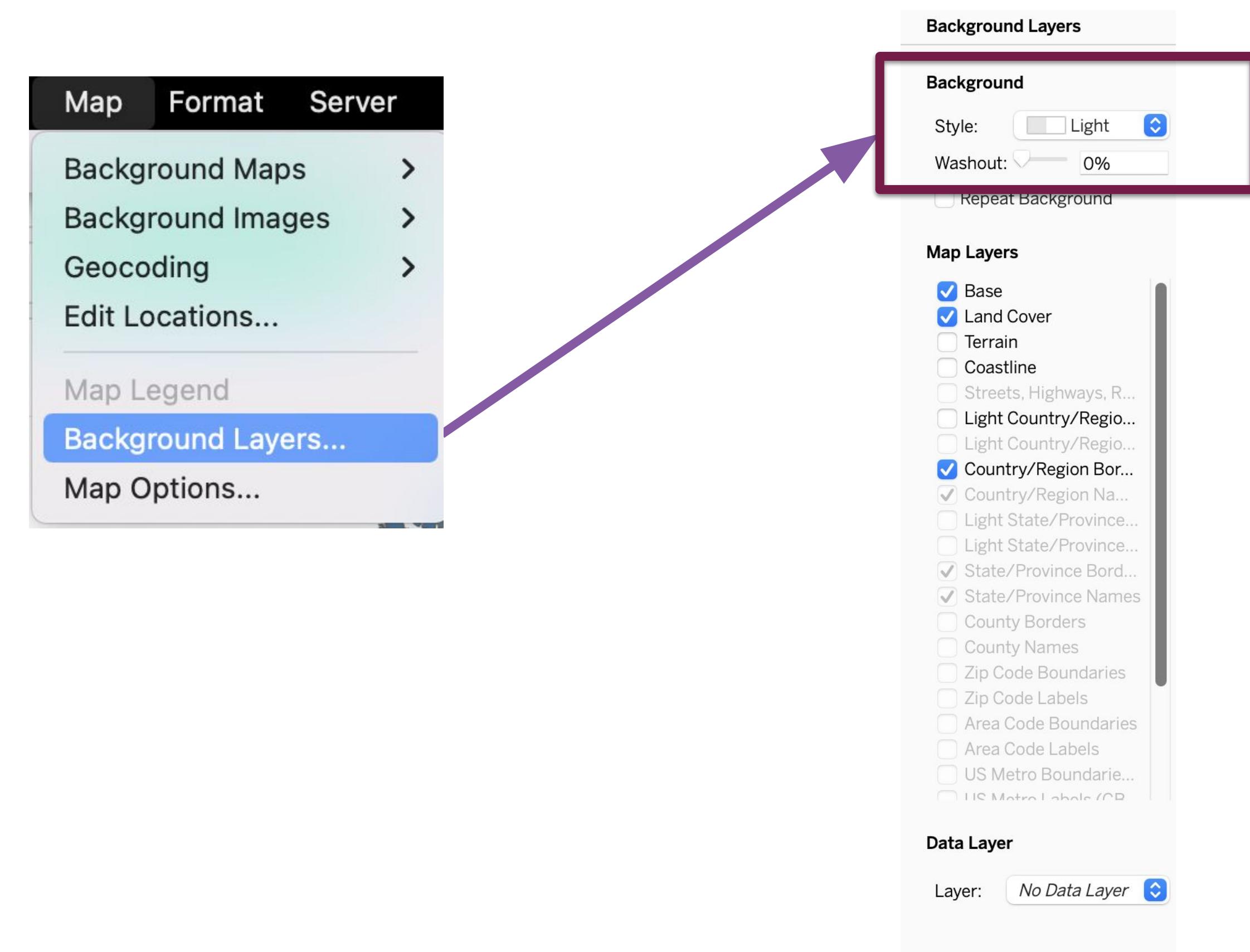
Map control options

- Tableau offers map control options like pan, zoom, search, etc.
- You have control of these options in the map menu in the top bar.
- Notice that pan and zoom are automatically given to you in the view.



Background layers options: background

- The options in the **Background Layers** submenu let you change the way the map looks.
- It can also be used to overlay different maps over each other.
 - For example, one could look at rainfall map layer over a crop yield map.
- For now, try toggling the **Background** style from the dropdown menu.

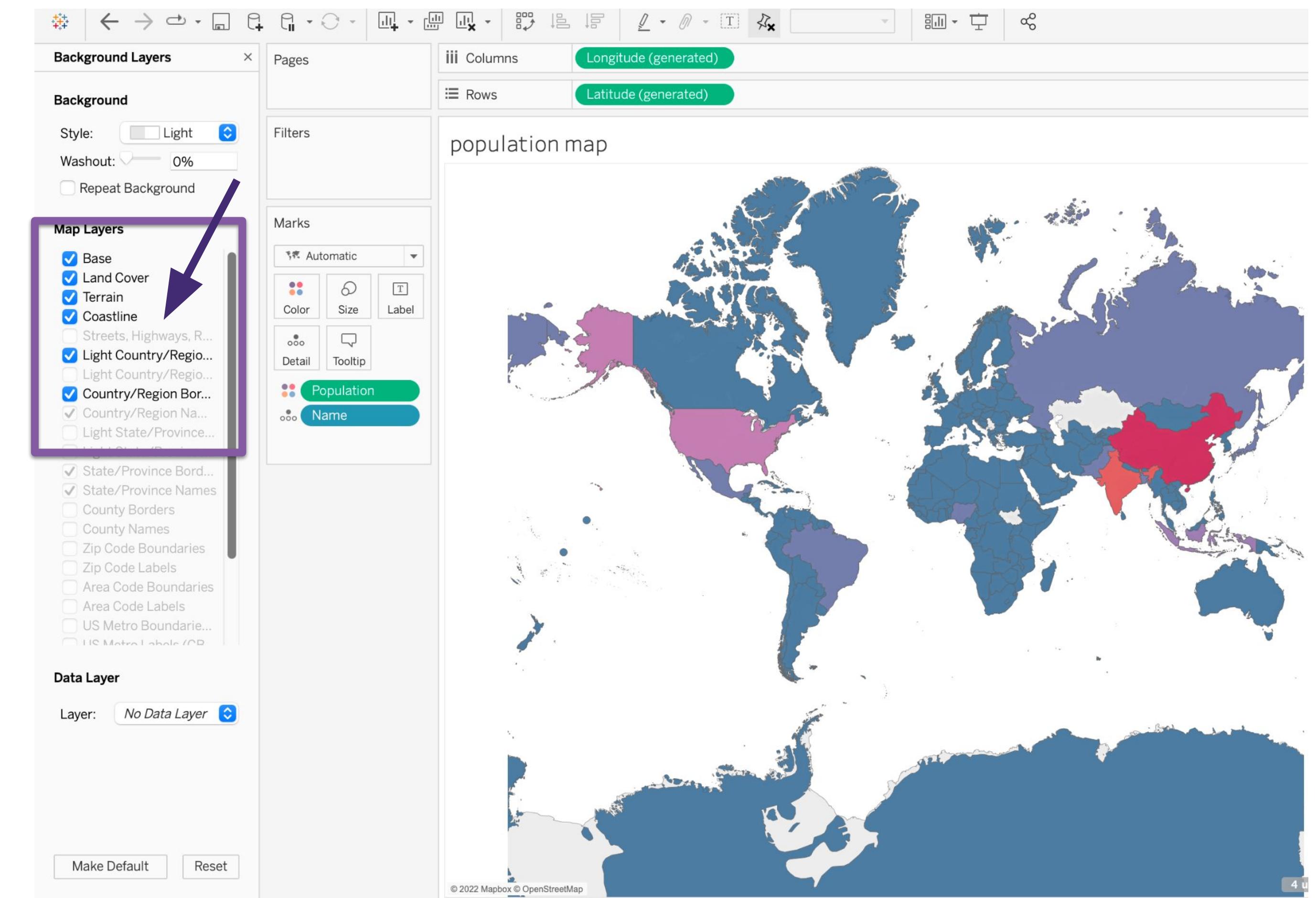


A few possible backgrounds



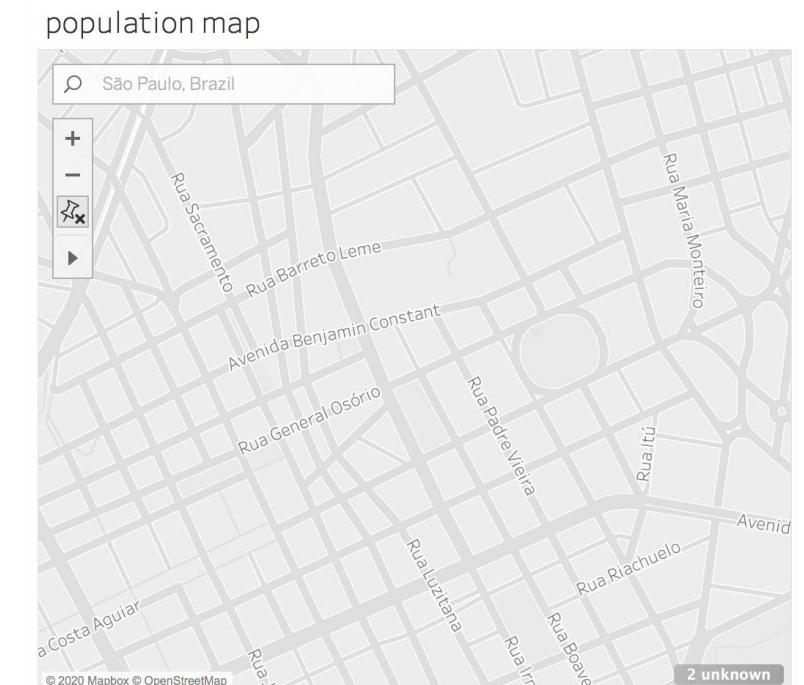
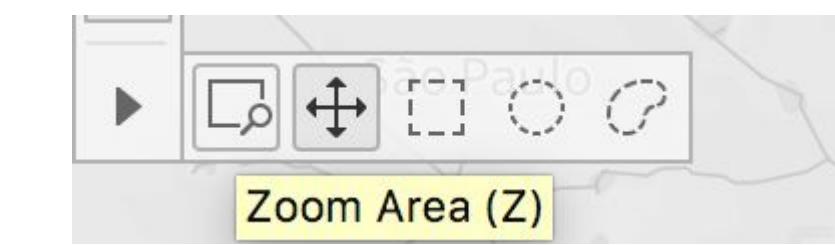
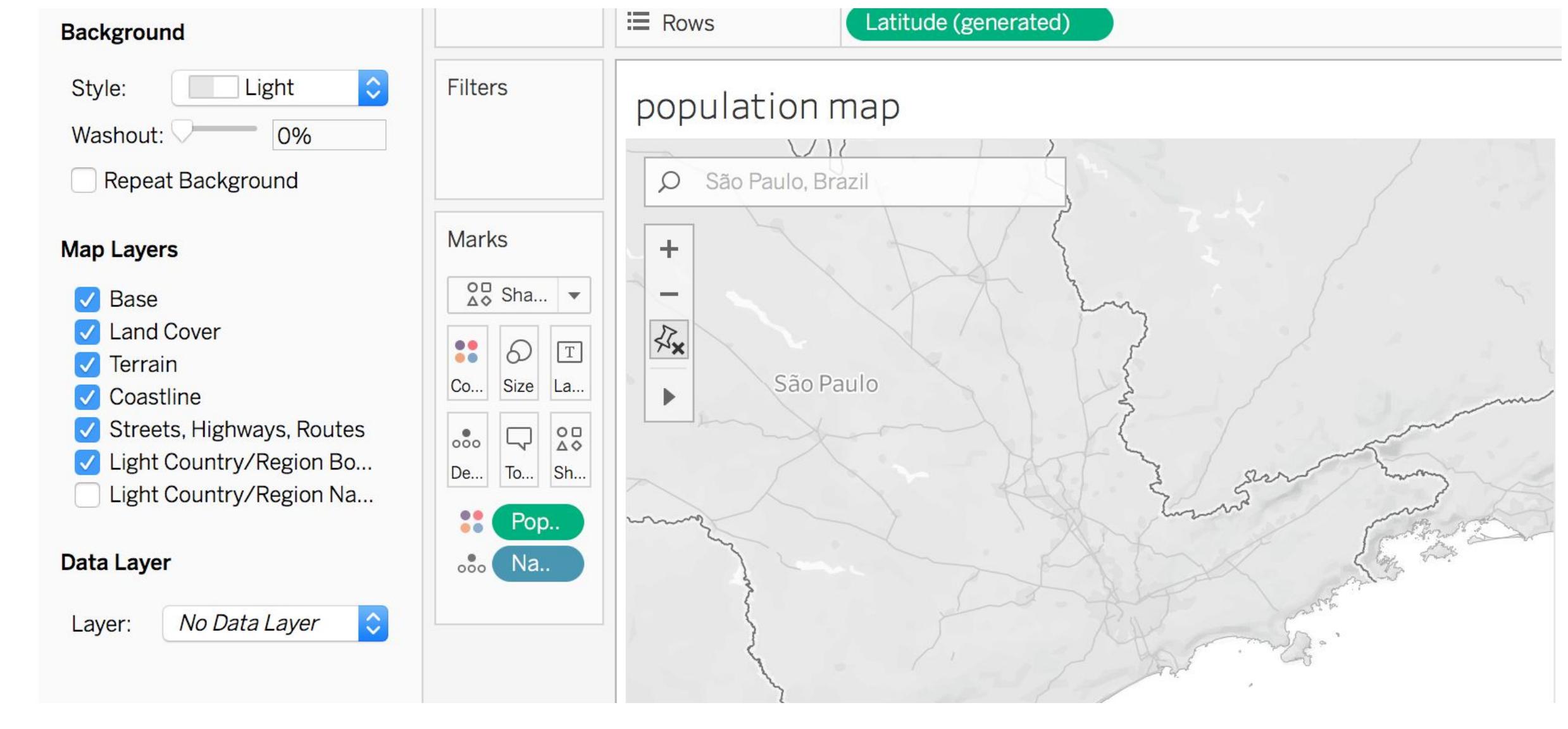
Map layer options: layer checkboxes

- The map layers checkboxes allow finer control.
- Note: some options are not allowed at this resolution.
 - Streets and Highways need to be at a higher resolution.



Map layer options: local features

- Zoom in to enable local features.
- Since Brazil has a high population on this map let's go to the city level for São Paulo, Brazil.
 - Note : It is easier to see street details **without** the population color fill.
- Zoom in further to an area that looks interesting, and we can get street level information.

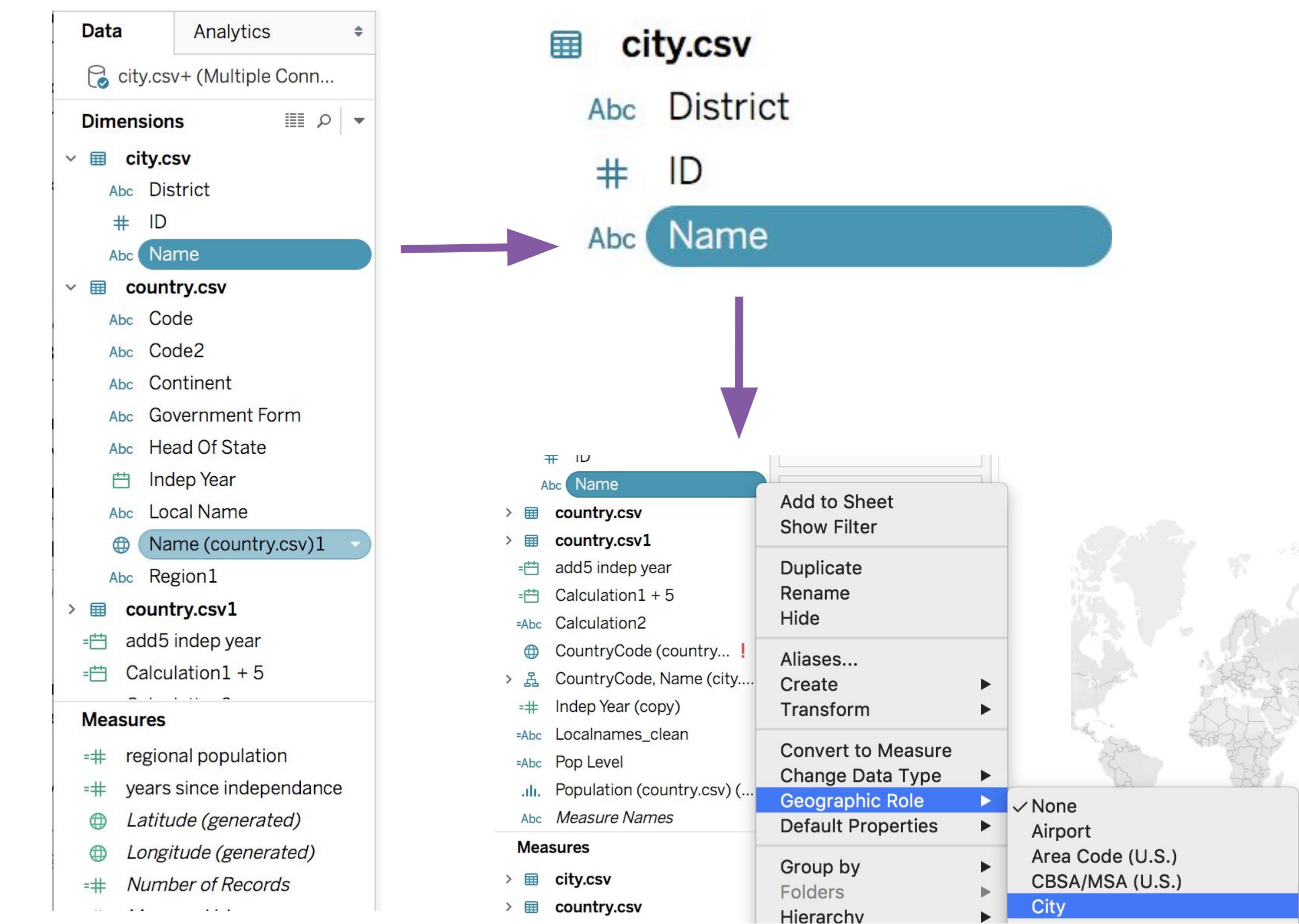


Module completion checklist

Objective	Complete
Introduce the concept of geospatial visualization	✓
Implement geospatial visualization for given dataset	✓
Identify and correct errors to facilitate proper joining with geospatial data	
Introduce the concept of Dashboards	
Combine previously created visualizations into Dashboard	

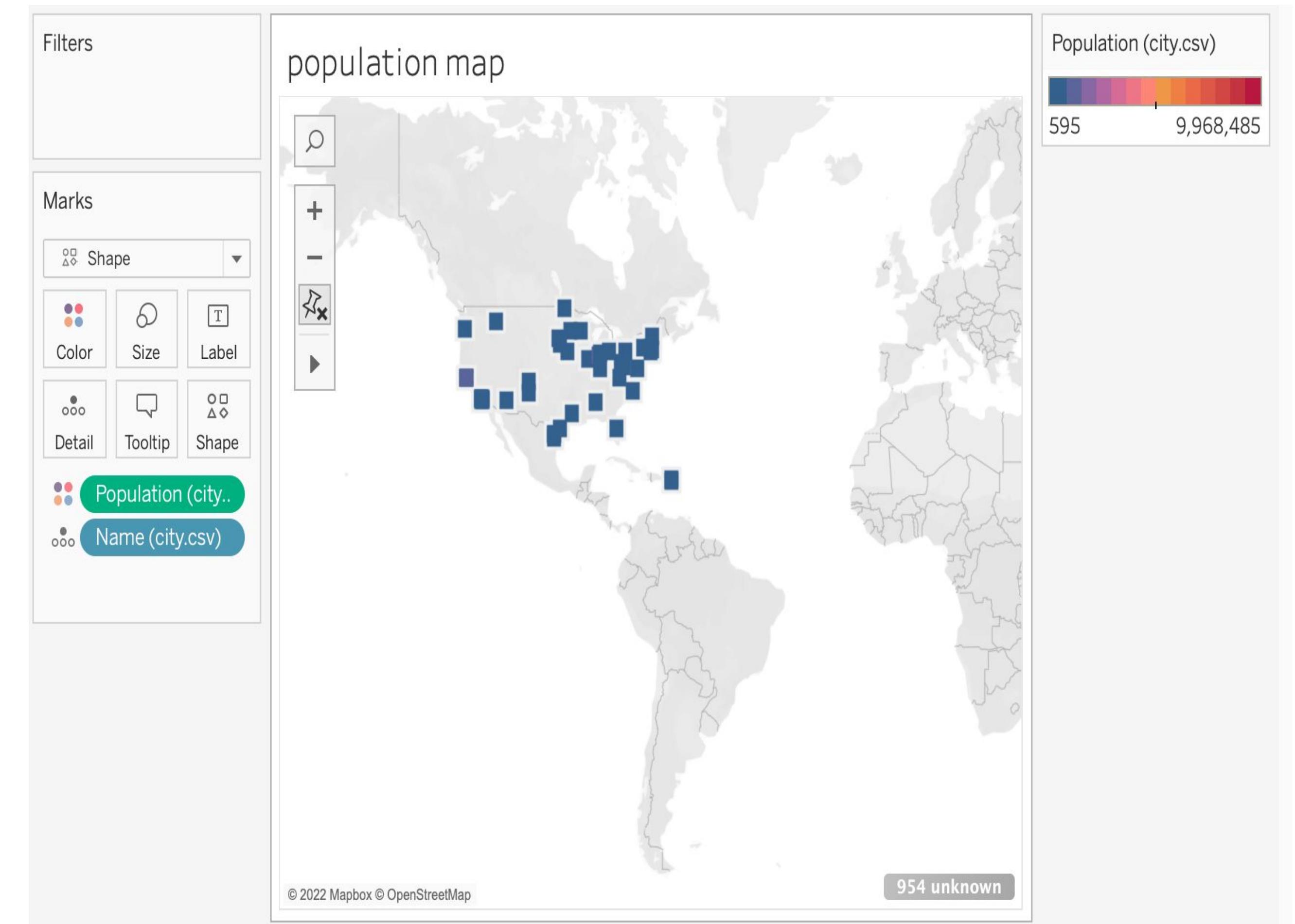
Manually converting to geospatial data

- Remember: Tableau didn't recognize some of our geospatial data, like for “**City**.”
 - We can tell because “**City**” doesn’t have any globe icons.
- To fix this, use the dropdown from the pill and tell Tableau which type of geospatial coordinates we have.
- In this case we have to tell Tableau that this is “**City**” data.



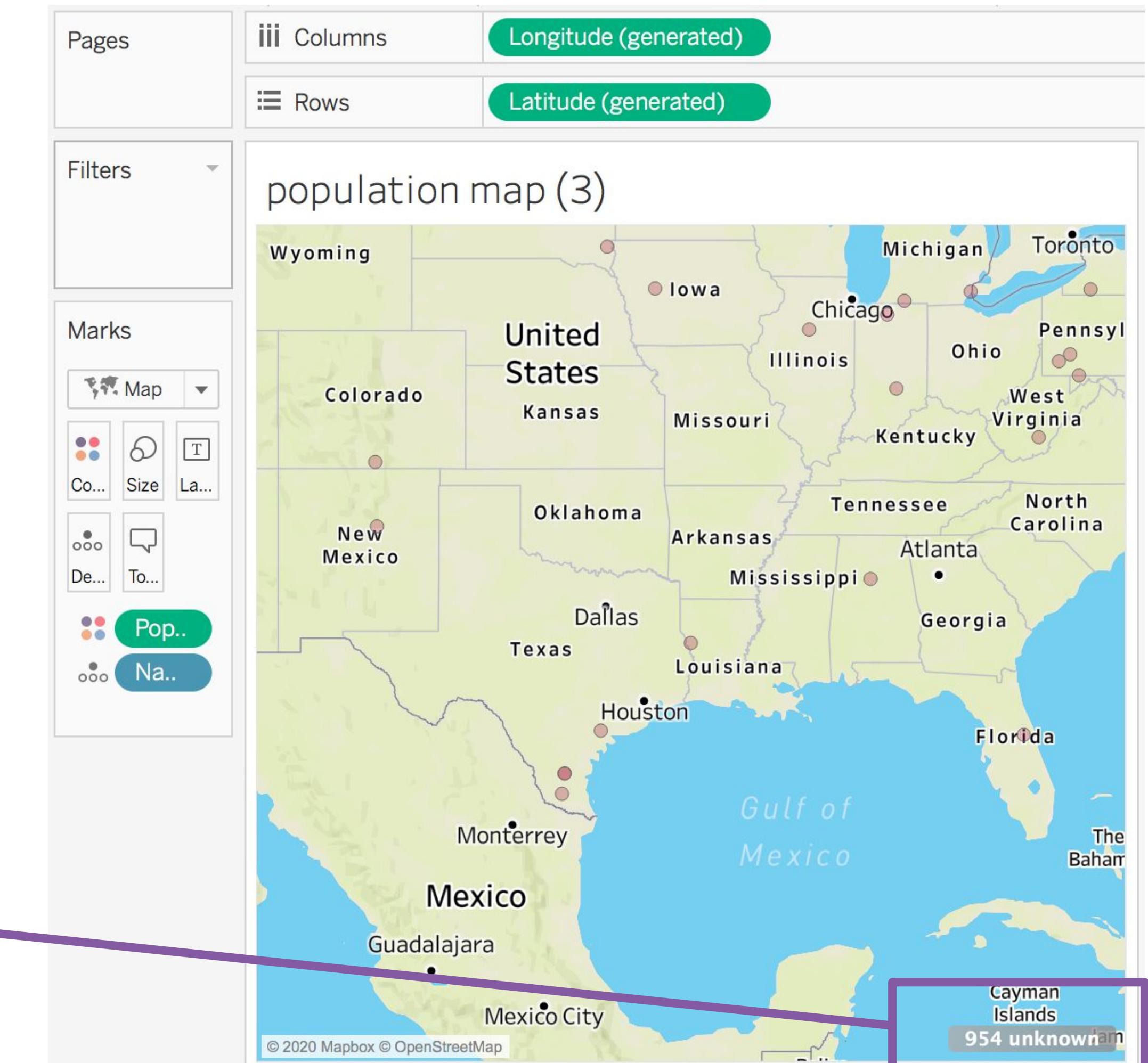
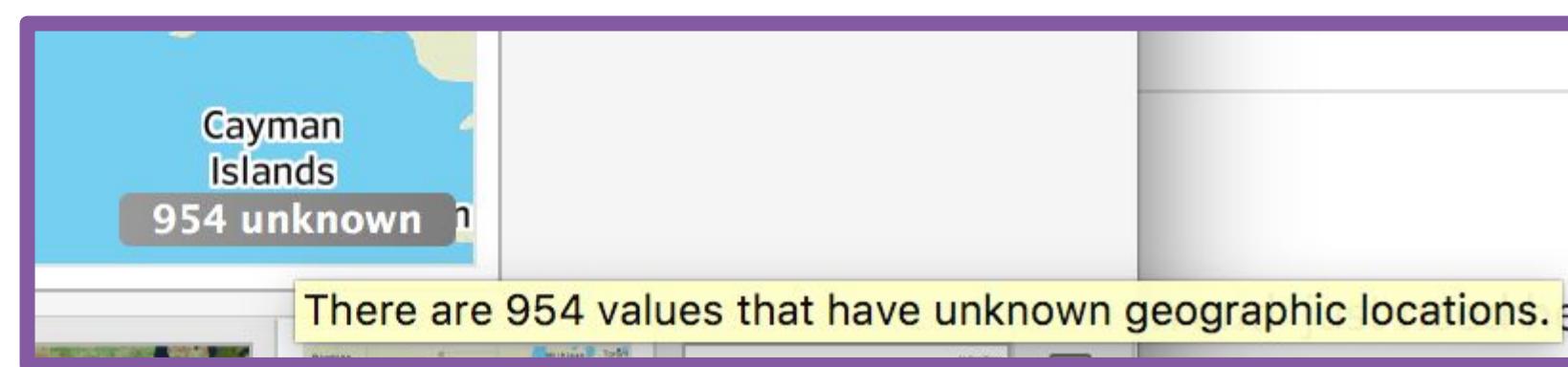
Plot city populations

- Now that we have city level coordinates, replace country data with city data in the Marks palette.
- What happened when we made that change?
- How would you troubleshoot this issue?



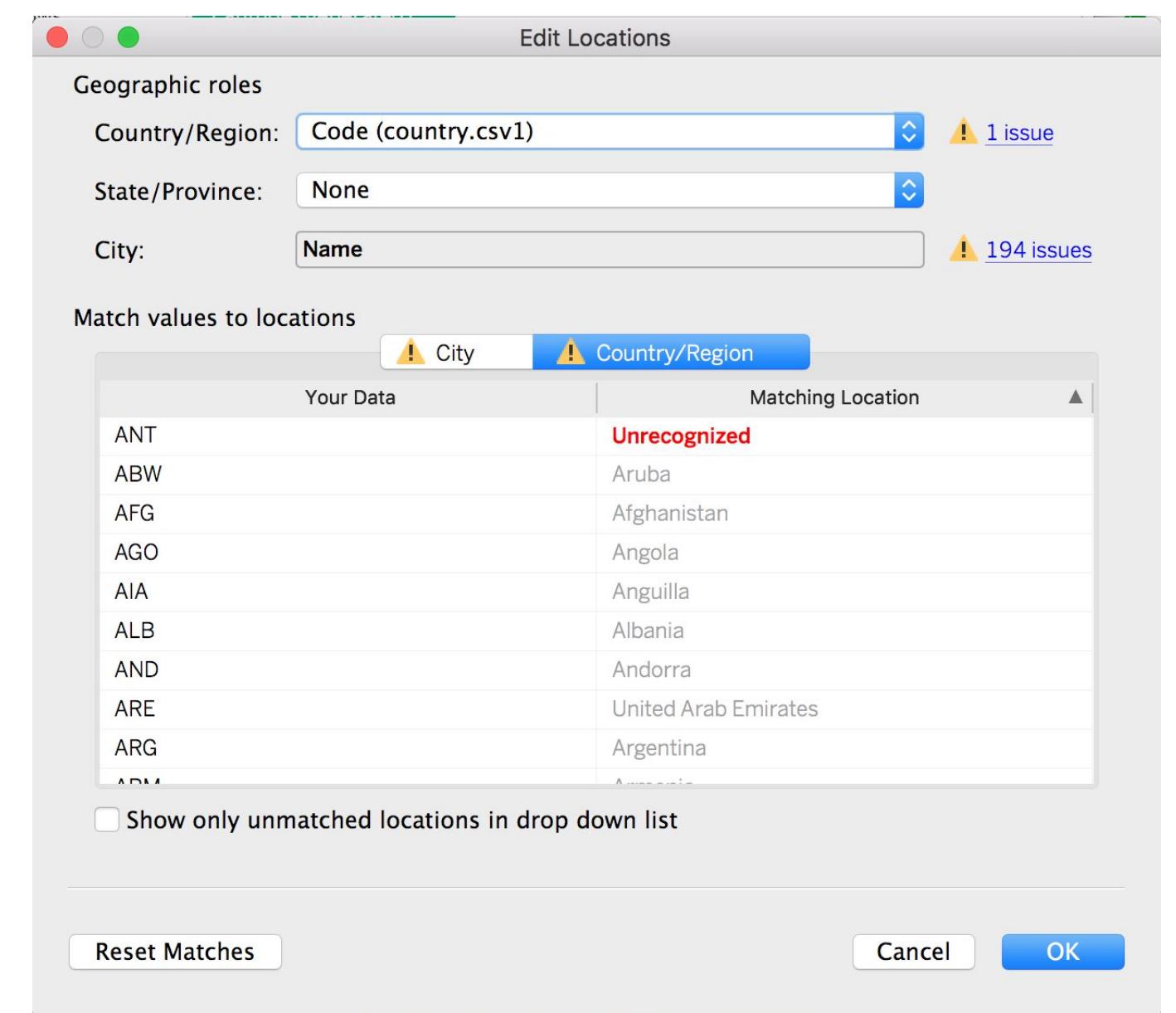
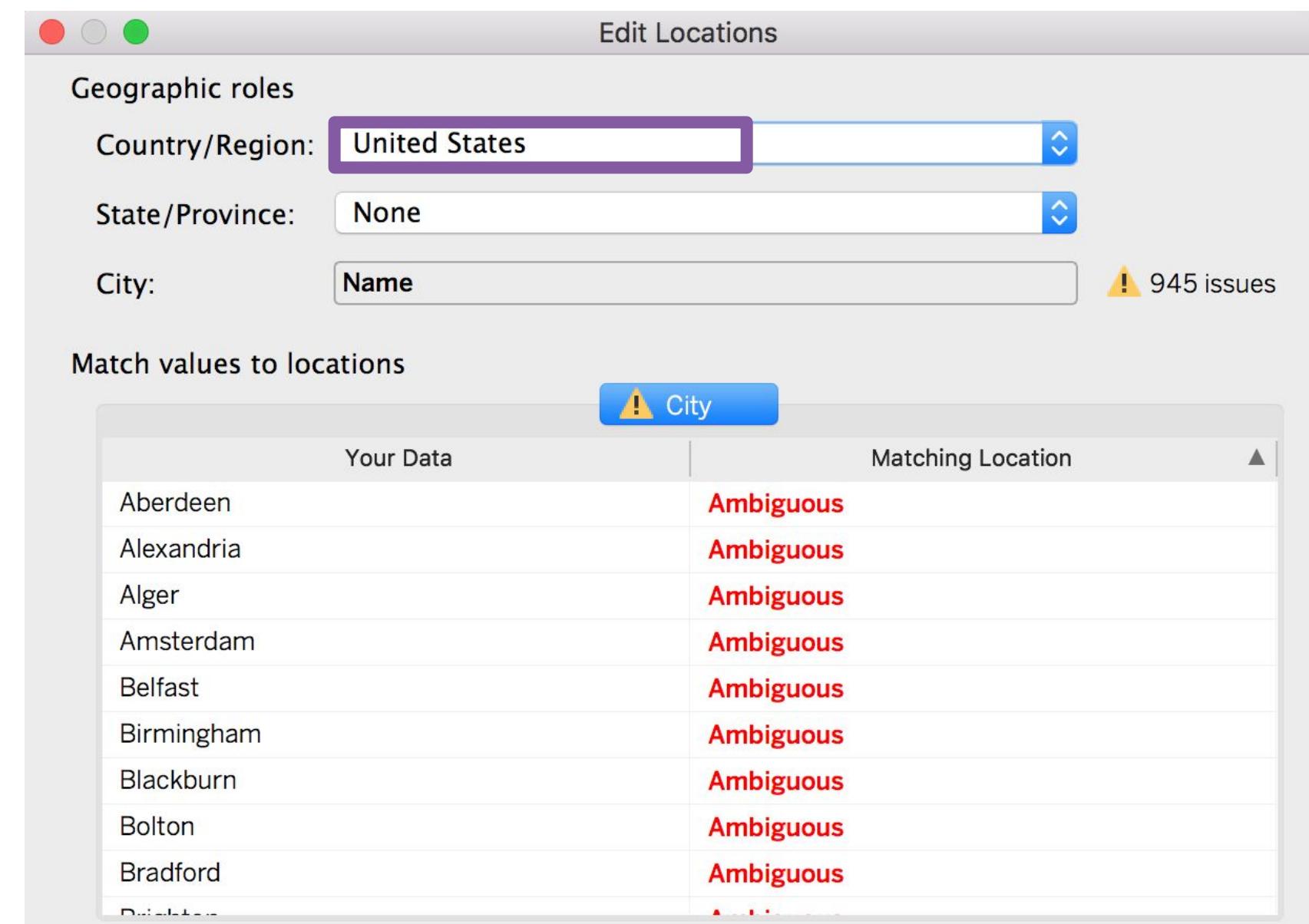
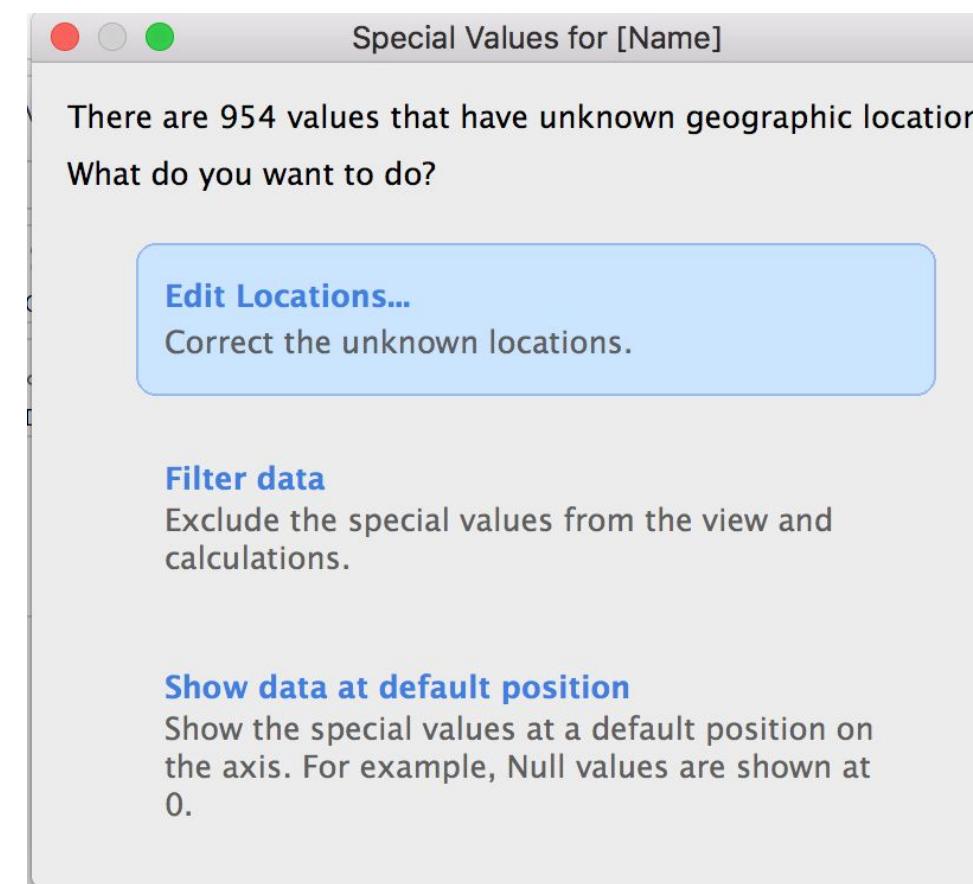
Missing cities

- We can see that Tableau has caught **954 cities** that it cannot assign a location to.
 - Click on the “**unknown**” button on the lower right to troubleshoot and edit the locations.
 - We could suppress the error message by right clicking it, but in this case that is a bad idea.



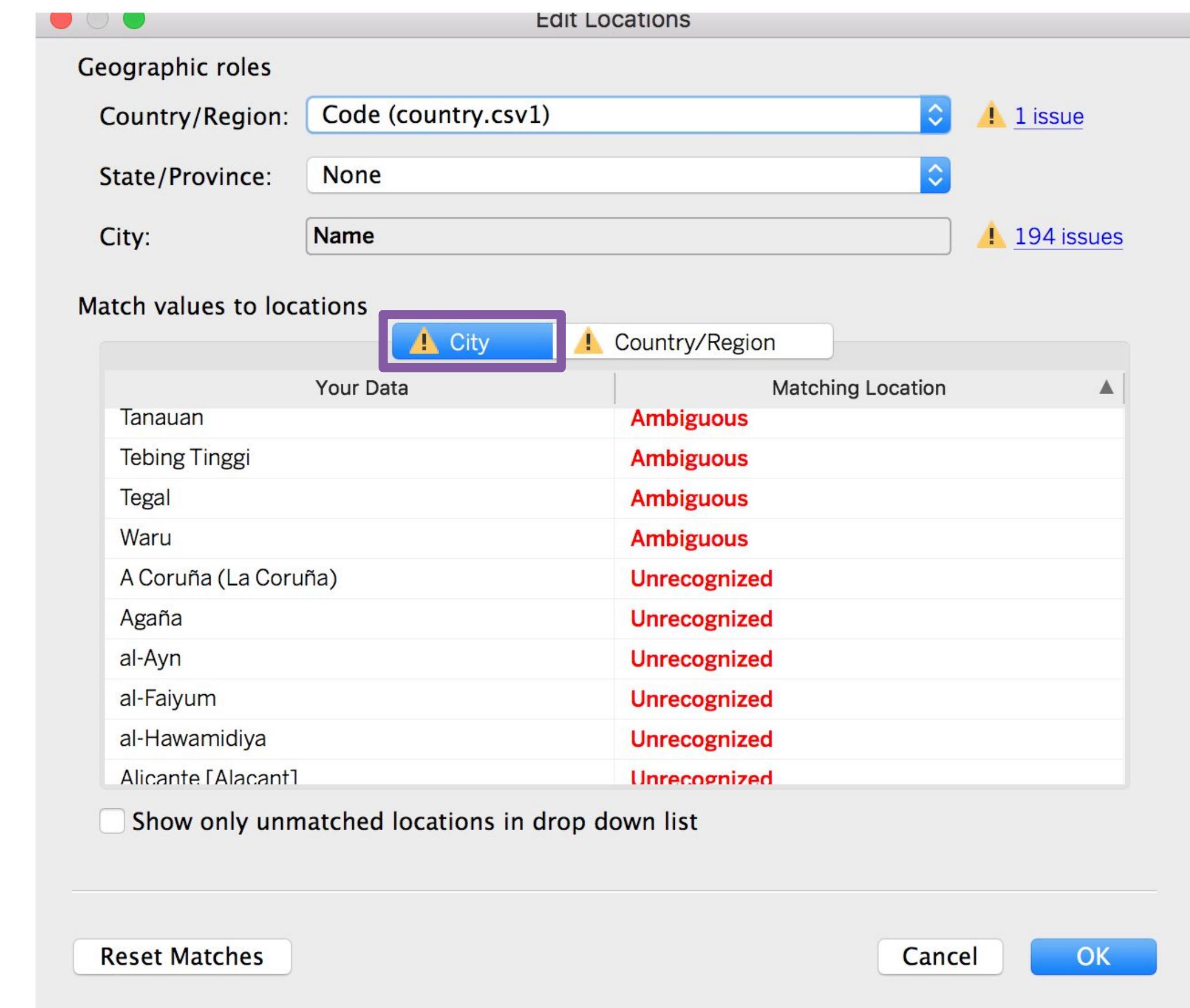
Country assignment issue

- When we click on “**Edit Locations...**,” we can see that the country was limited to the United States.
- We want to associate all of these cities with their **Country Code**, which fixes **757** issues.



Explore the remaining issues

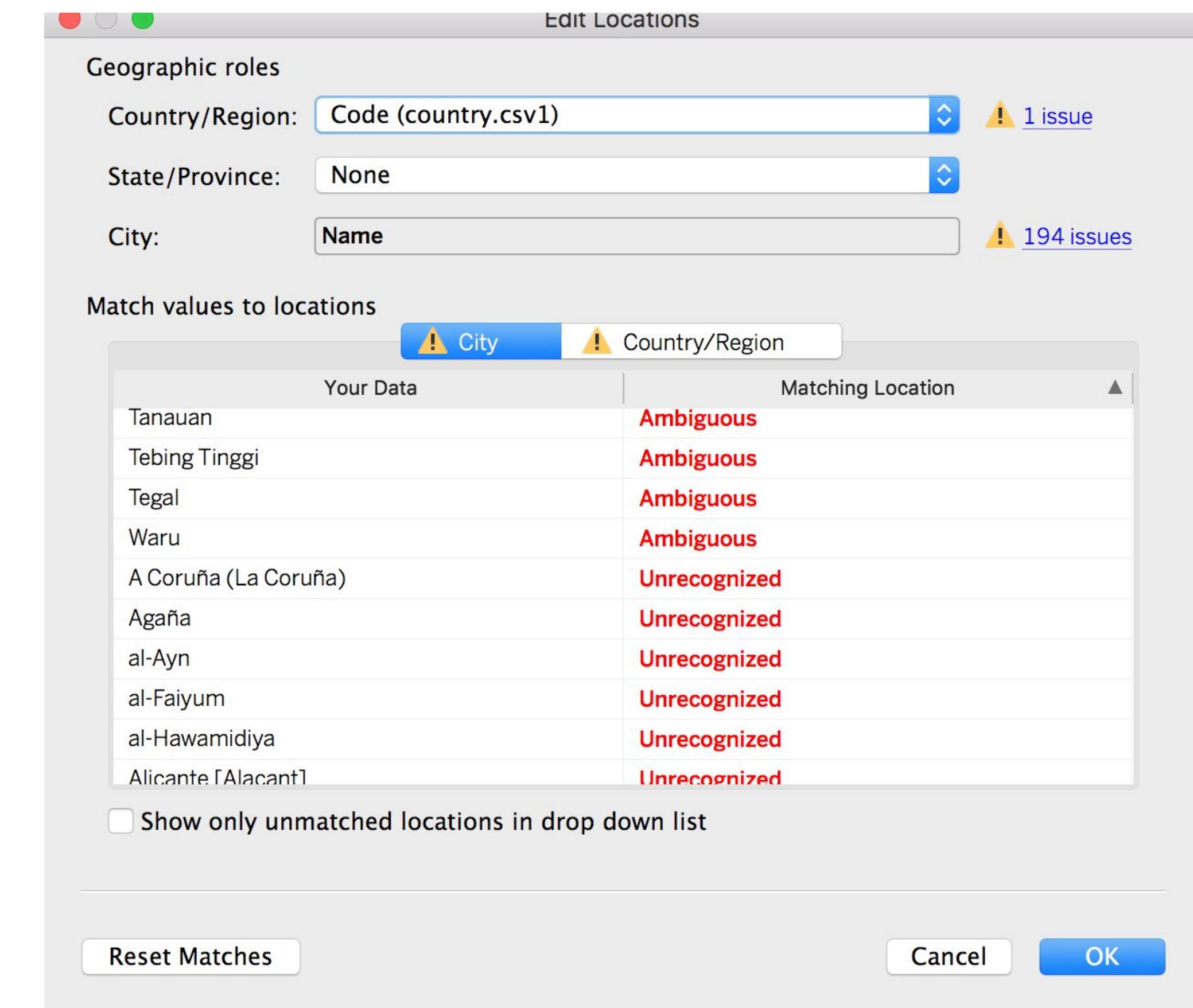
- Look at the cities that are missing.
- We see two types of issues:
 - “**Ambiguous**”: some cities or counties have the same name but could exist in multiple places.
 - “**Unrecognized**”: if there is a location or abbreviation that Tableau does not recognize.



Fix ambiguous issues, if possible

- Adding another geographic field, like **State/Province**, defines the correct location for that data.
- If there is a hierarchy in the data pane, Tableau will automatically use the appropriate levels of the hierarchy to solve location ambiguities.

Note: We do not have state/province data, so we will have to leave this out.



Fix unrecognized issues, if possible

- This will take some research.
- Many cities may include special characters, or may be entered as an alternate / local spelling.
 - **Go back to the dataframe** and find out if there is any other information.
 - **Search Tableau's database** for alternative spellings.
 - Verify your hypothesis by **cross-checking with Google**.
 - Alternatively, **enter longitude and latitude coordinates manually** if you have them.

View Data: city.csv+ (Multiple Connections)

Localnames_clean	Name	Name (country.csv)1
Al-Imarat_al-'Arabiya_al-Muttaahida	al-Ayn al-Faiyum	United Arab Emirates Egypt

El Ain (disambiguation)

From Wikipedia, the free encyclopedia

El Ain, or a variant thereof, may refer to:

- Al Ain, a city in Abu Dhabi, United Arab Emirates
- Al-Ayn, Oman, an archaeological site in Oman
- El Ain, Tunisia, a town in Sfax Governorate, Tunisia
- El Ain, Ash Shamal, Lebanon, a town in Batroun District, North Governorate, Lebanon
- El Ain, Beqaa, Lebanon, a town in Baalbek District, Beqaa Governorate, Lebanon
- El Ain, Jabal Lubnan, Lebanon (North), a town in Keserwan District, Mount Lebanon Governorate, Lebanon
- El Ain, Jabal Lubnan, Lebanon (South), a town in Baabda District, Mount Lebanon Governorate, Lebanon

A view over Green Mubazzarah in Al-Ain, at the base of Jebel Hafeet (Mount Hafeet)

Nickname(s): مَدِينَةُ الْحَرِيقَةِ^[1]
The Garden City^[2] (of Abu Dhabi,^[3] the UAE^[4] or the Gulf)^{[5][6]}

Tebing Tinggi
Tegal
Waru
A Coruña (La Coruña)
Agaña
al-Ayn
al-Faiyum
al-Hawamidiva

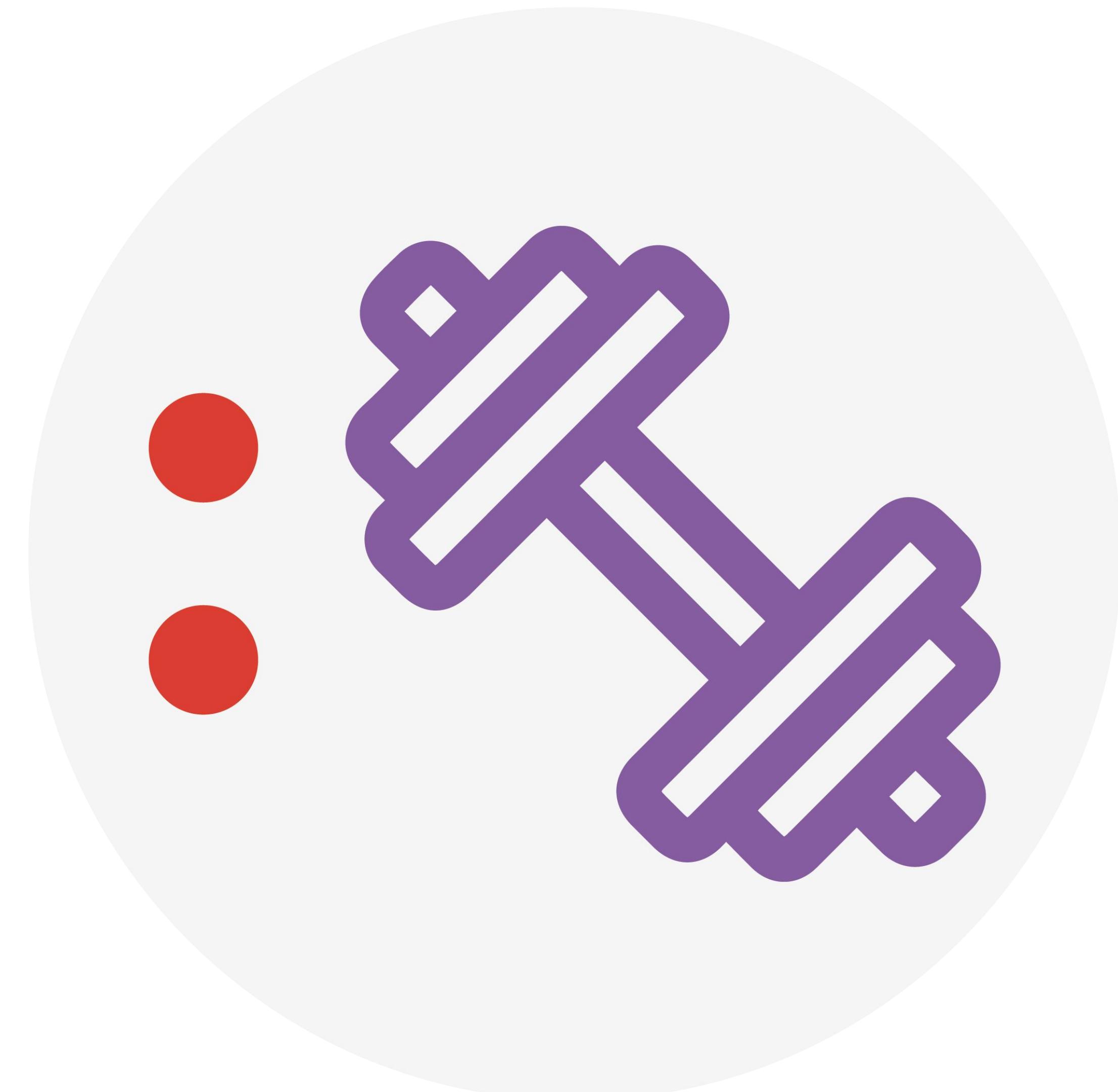
Alagon del Río
Alahanpanjang
Alabsa
Al Ain
Alaincourt
Alaior
Alairac

Unrecognized
Unrecognized

Knowledge check 11



Exercise 11



Module completion checklist

Objective	Complete
Introduce the concept of geospatial visualization	✓
Implement geospatial visualization for given dataset	✓
Identify and correct errors to facilitate proper joining with geospatial data	✓
Introduce the concept of Dashboards	
Combine previously created visualizations into Dashboard	

: End of Part 11

