DATA SOCIETY:

Introduction to Tableau

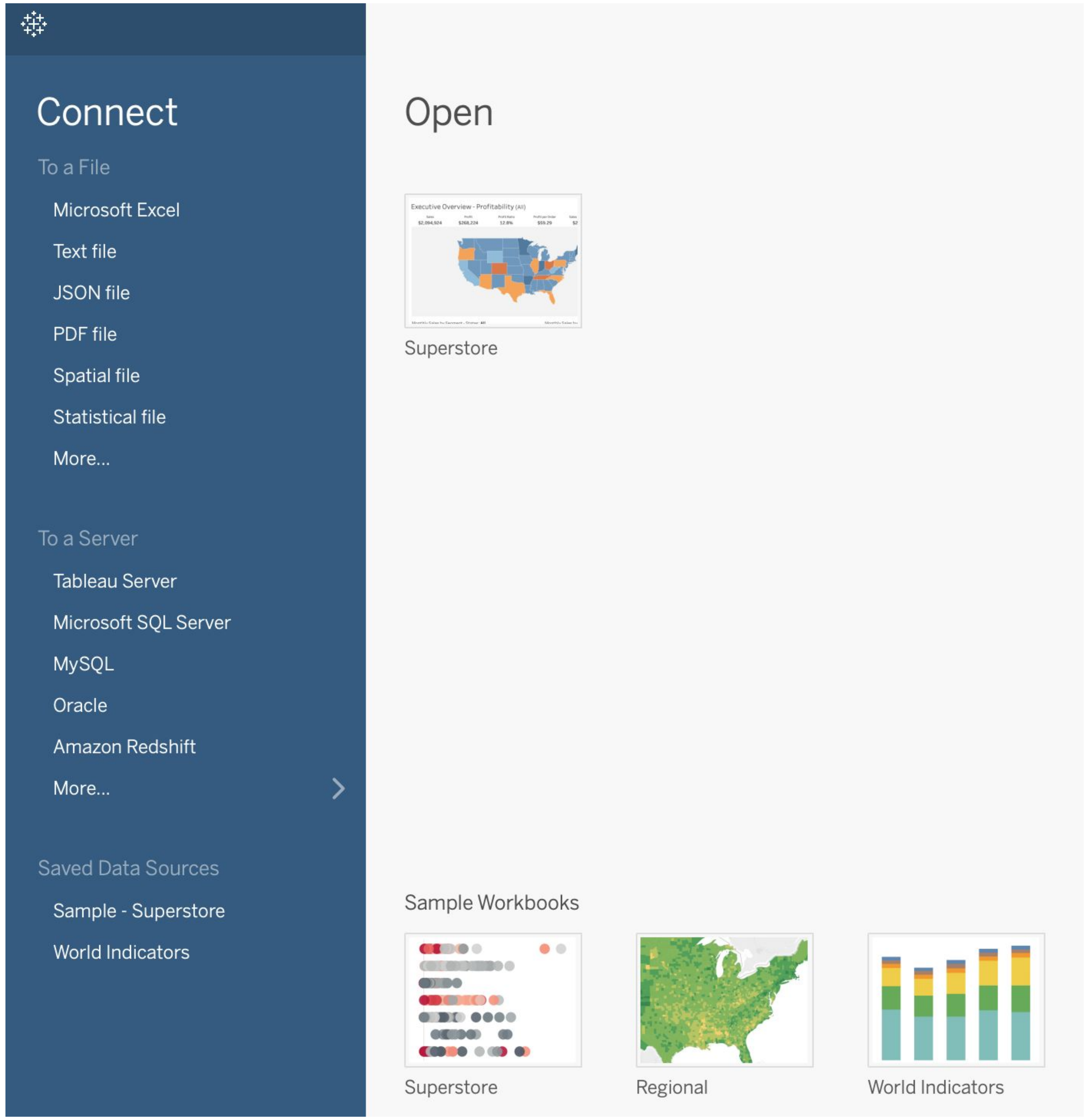
Part 3

Recap: importing data

- Import data with the **Connect** panel.

- Supports multiple formats such as:
  - Microsoft Excel (.xlsx).
  - Text (.txt, .csv).
  - JSON (.json).
  - PDF (.pdf).
  - R data format (.RData).

- Supports Database Connections such as:
  - MySQL.
  - Oracle.
  - Redshift.



Connect

To a File

Microsoft Excel

Text file

JSON file

PDF file

Spatial file

Statistical file

More...

To a Server

Tableau Server

Microsoft SQL Server

MySQL

Oracle

Amazon Redshift

More...

Saved Data Sources

Sample - Superstore

World Indicators

Open

Superstore

Sample Workbooks

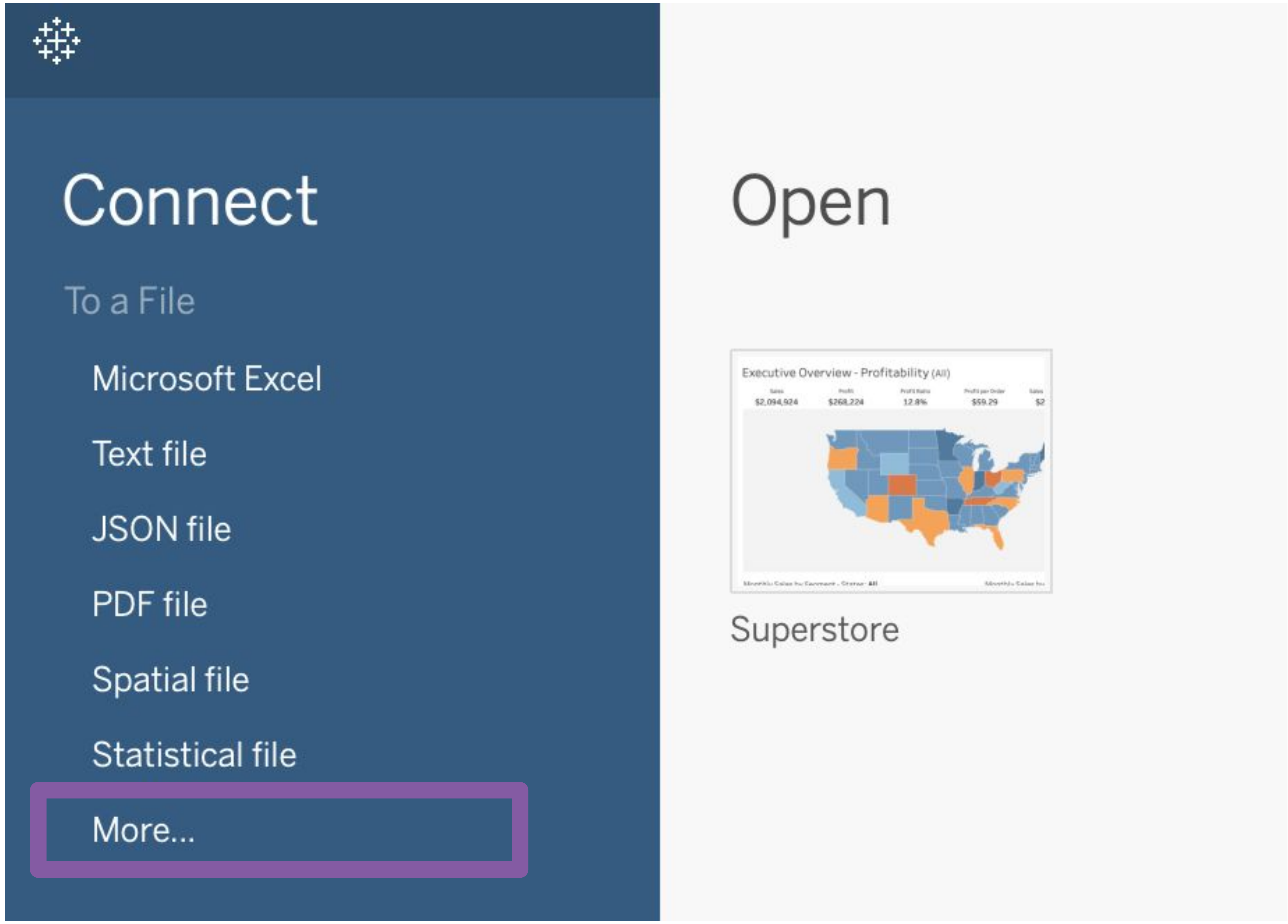
Superstore

Regional

World Indicators

Import World Data : CSV

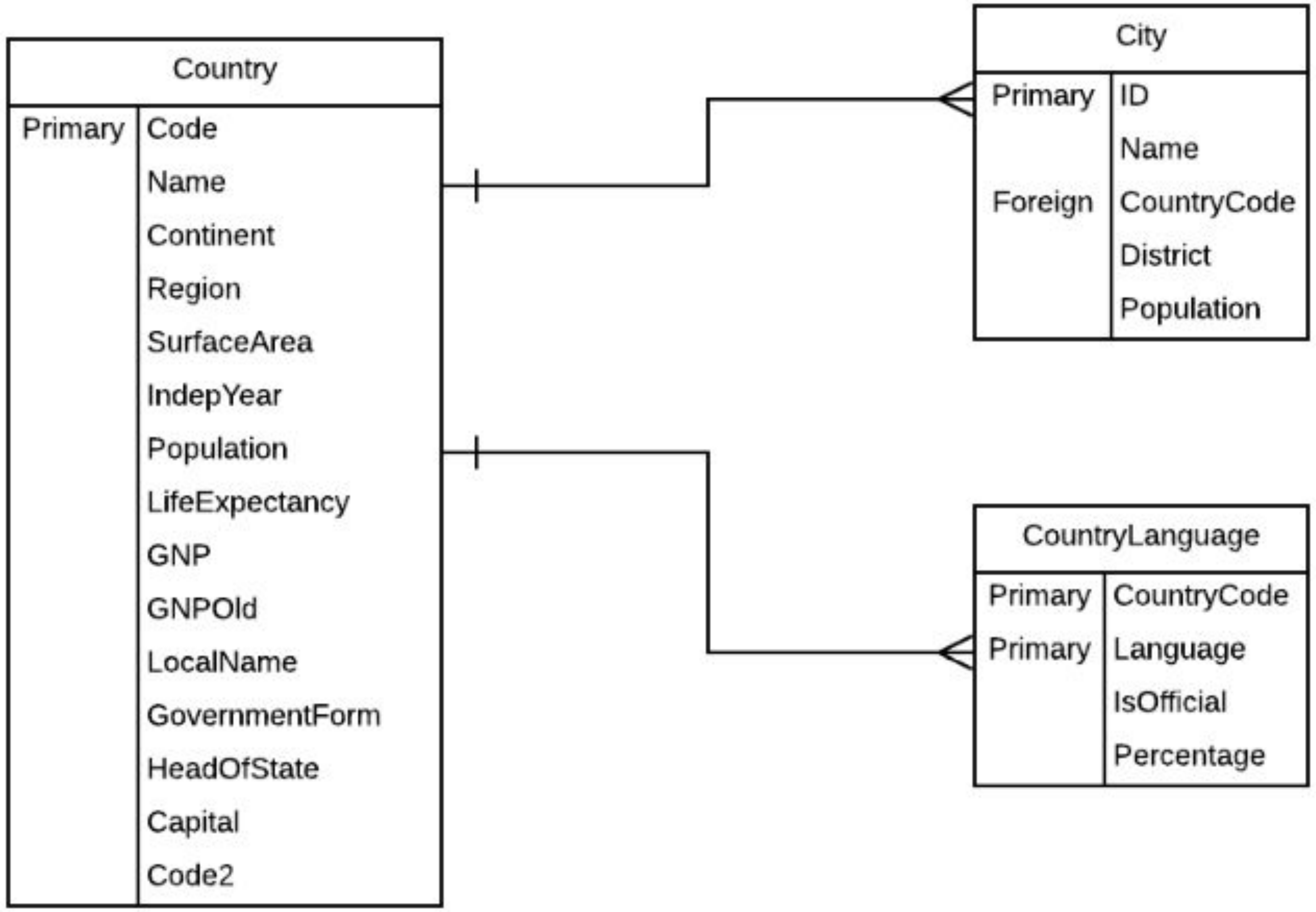
- Let's import some pieces of the world dataset today and see what sort of insights we can reveal.
- Click the **"More…"** item to browse your local CSV files.



Recap: World Database

- For now, import the following three CSV files:
  - **country.csv**
  - **city.csv**
  - **countrylanguage.csv**
- We'll use the other CSV files during our Exercises.

World Database ERD



Country	
Primary	Code
	Name
	Continent
	Region
	SurfaceArea
	IndepYear
	Population
	LifeExpectancy
	GNP
	GNPOld
	LocalName
	GovernmentForm
	HeadOfState
	Capital
	Code2

City	
Primary	ID
	Name
Foreign	CountryCode
	District
	Population

CountryLanguage	
Primary	CountryCode
Primary	Language
	IsOfficial
	Percentage

Recap: joining datasets using Relationships

- It is often necessary to combine data from multiple places - different tables or even data sources - to perform a desired analysis.

- Previously, Tableau utilized **joins** in order to merge and combine data in strategic ways.

- While it is still possible to perform traditional joins in Tableau, it is no longer the default option.

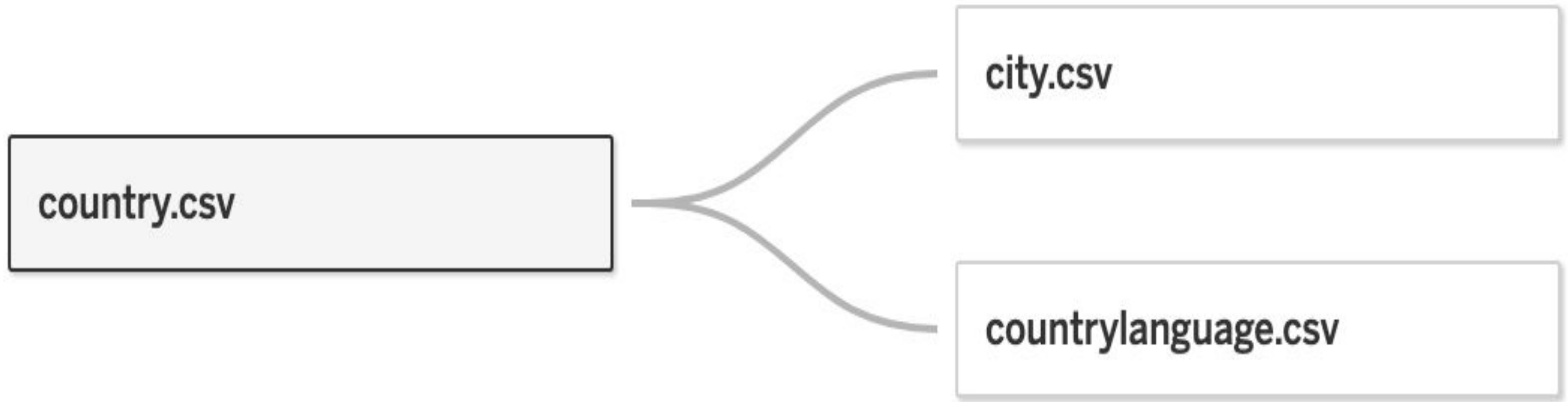
- Tableau has introduced a feature called **Relationships** to minimize the amount of data pre-processing required to visualize related datasets in multiple ways.

Recap: Relationships are contextual joins

- A single relationship will support all join types at the same time.

- For **Measures**, this means **all values are always retained**, even if they're unmatched nulls (not missing values).

- For **Dimensions**, relevant domains will be displayed **across tables**, and Tableau will display all values in the domain by default (even if there are no matching Measure values).

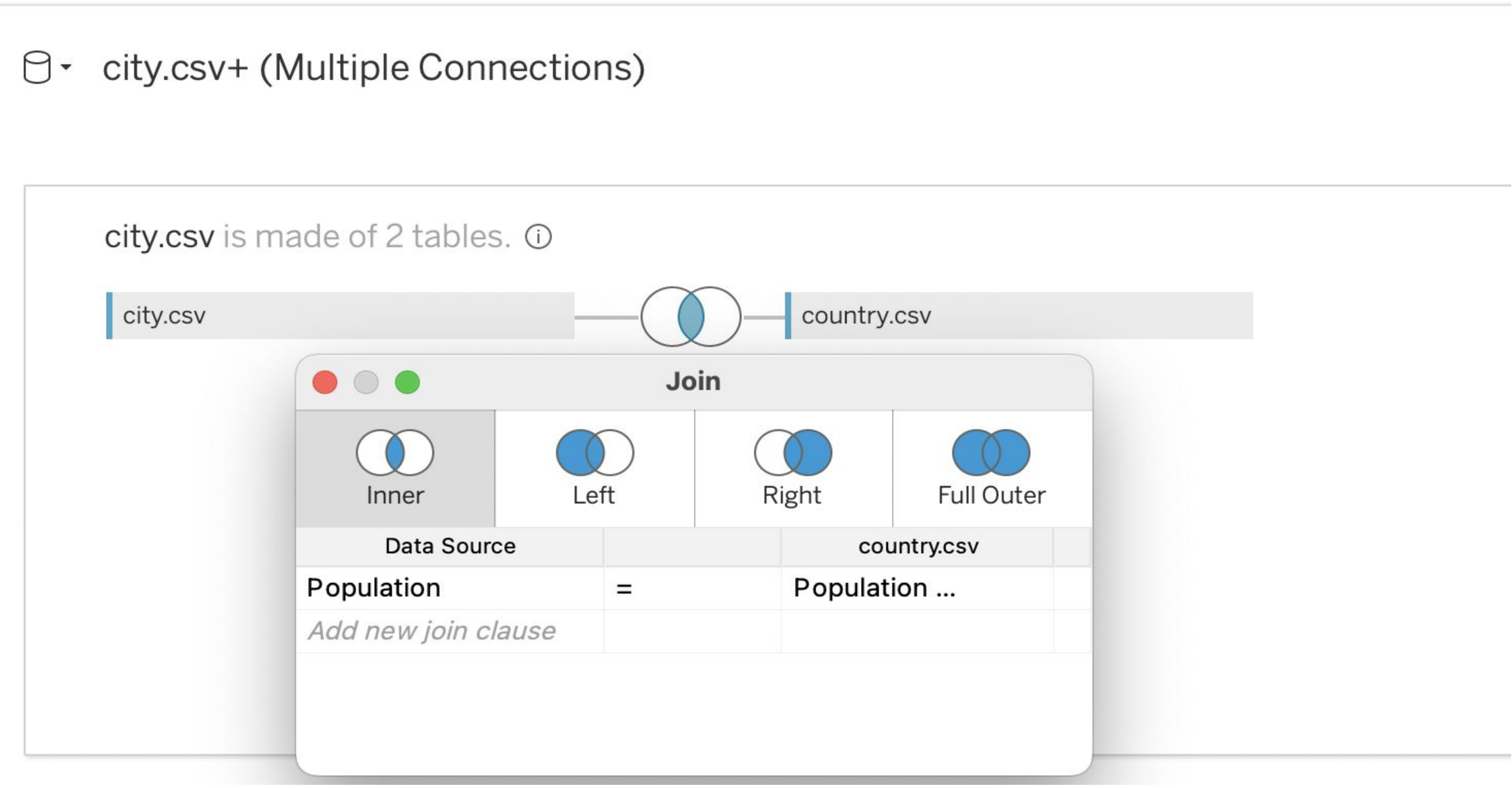


What dimension serves as the key linking city.csv and countrylanguage.csv to country.csv?

Recap: manually joining dataset using joins

- Previously, Tableau utilized **joins** in order to merge and combine data.

- It is still possible to join tables **manually** by specifying the precise variables the datasets will use as common keys, but this functionality is no longer the default.



city.csv+ (Multiple Connections)

city.csv is made of 2 tables.

Data Source		country.csv
Population	=	Population ...
Add new join clause		

Warm-up: finding a dataset

- Spend a few minutes looking through the **Kaggle** (link) or **Data.World** (link) repository.
    - *What kinds of data do you see that you might be interested in working with?*
    - *What kinds of data might you want to try to combine?*

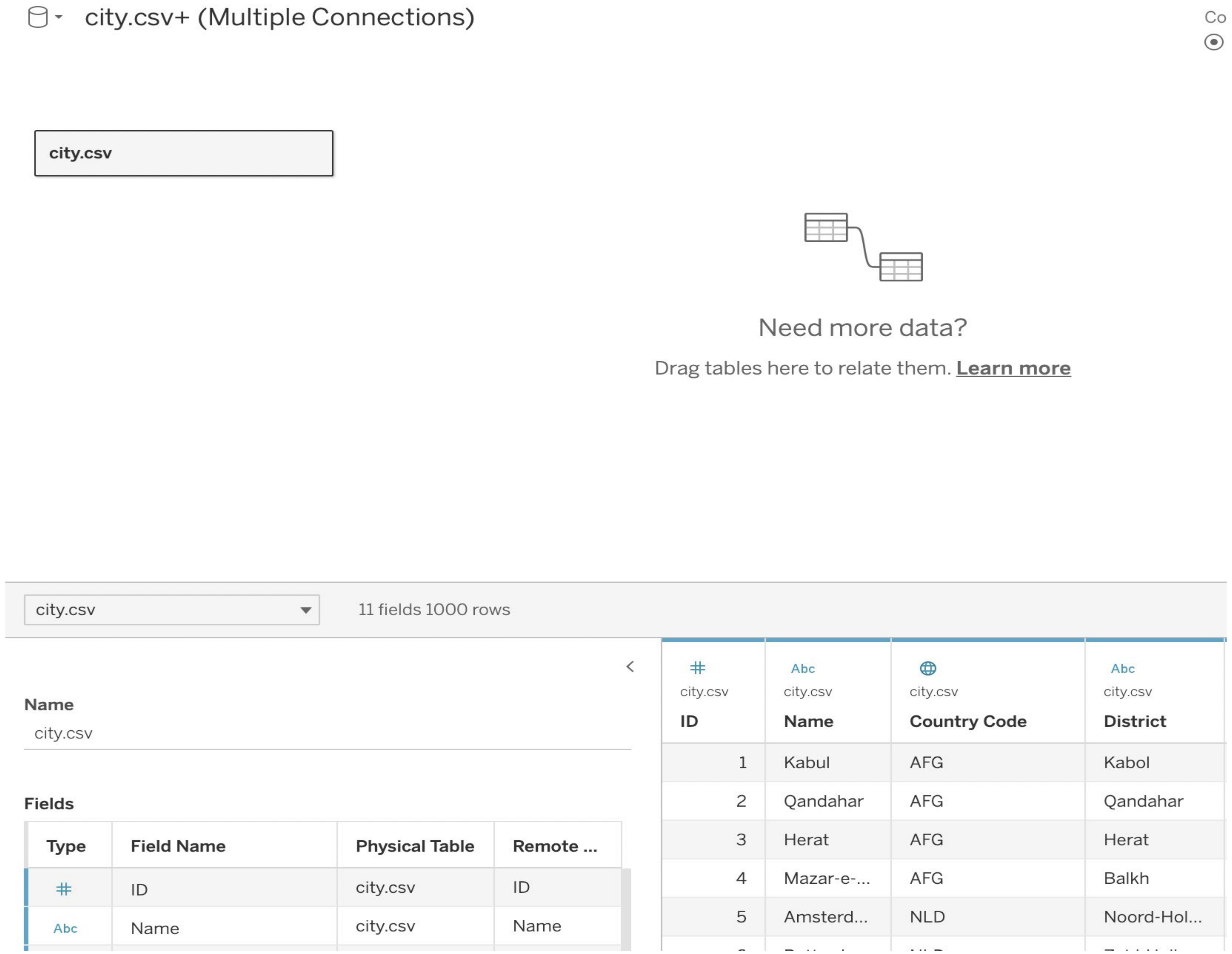
Module completion checklist

Objective	Complete
Manually join tables using joins	
Introduce the concept of aggregating, binning, and grouping	
Create and use bins and groups	
Explore filtering capabilities of Tableau	

Joining the tables using joins

Let's join the **city**, **country**, and **countrylanguage** tables using joins.

- **Step1:** Drag the city table into the canvas.



city.csv+ (Multiple Connections)

Need more data?

Drag tables here to relate them. Learn more

11 fields 1000 rows

Name
city.csv

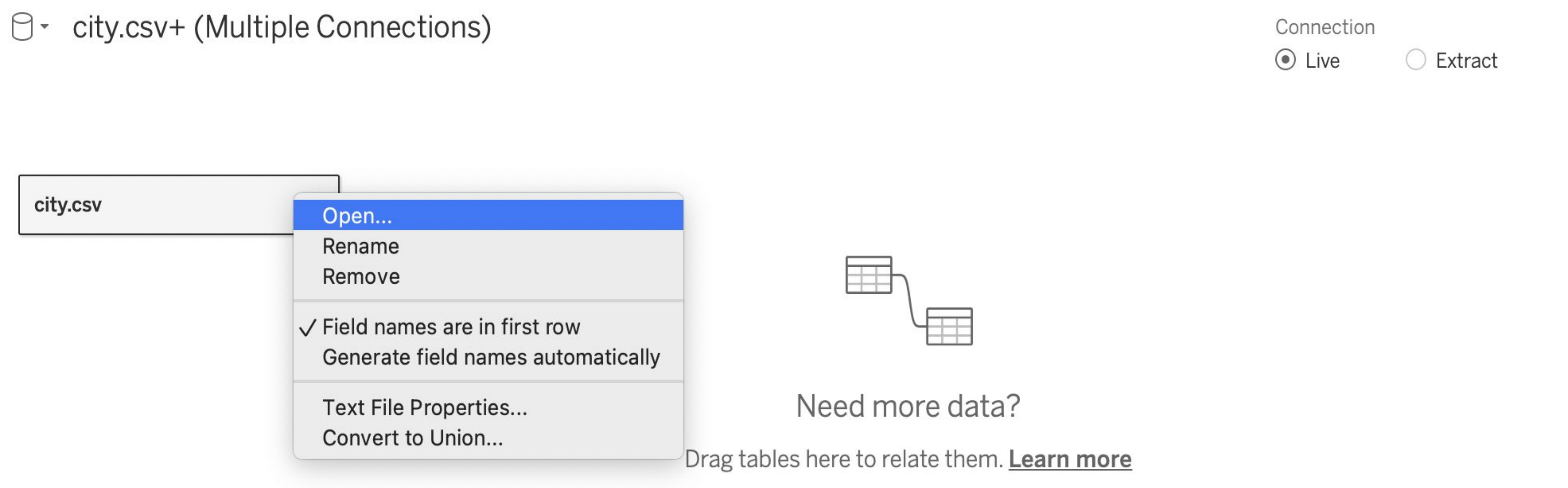
Fields

Type	Field Name	Physical Table	Remote ...
#	ID	city.csv	ID
Abc	Name	city.csv	Name

# city.csv ID	Abc city.csv Name	city.csv Country Code	Abc city.csv District
1	Kabul	AFG	Kabol
2	Qandahar	AFG	Qandahar
3	Herat	AFG	Herat
4	Mazar-e-...	AFG	Balkh
5	Amsterd...	NLD	Noord-Hol...

Joining the tables using joins, cont’d

- **Step 2:** Select **Open** from the menu or double-click the first table to open the join canvas (physical layer).



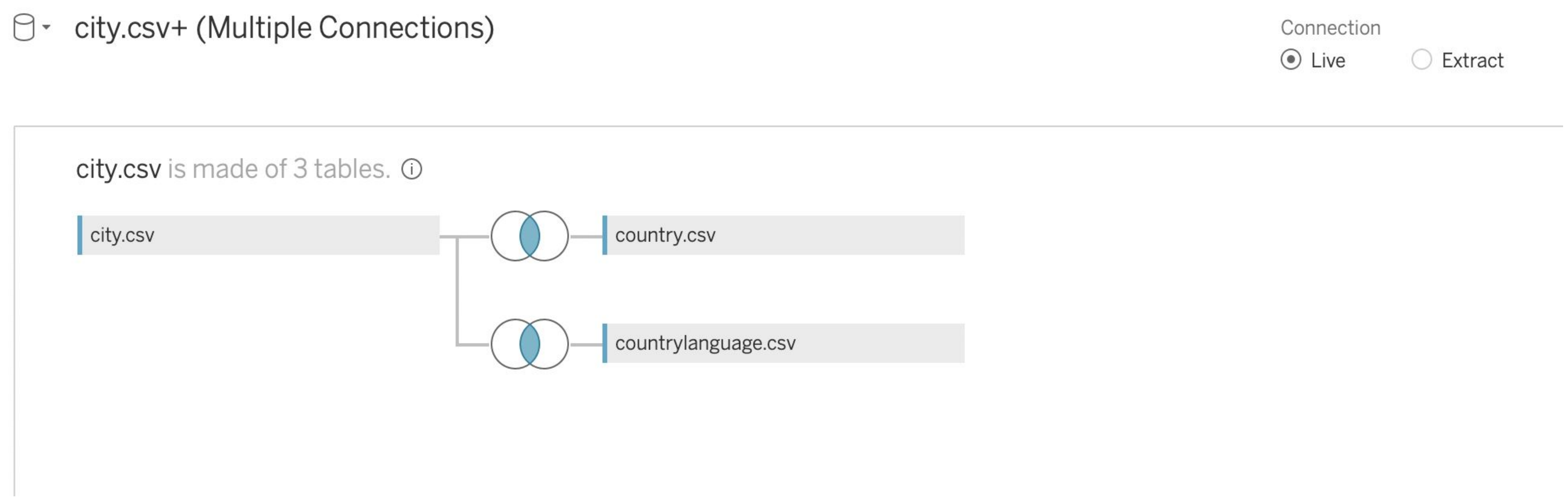
city.csv+ (Multiple Connections)

Connection
● Live ○ Extract

Need more data?

Drag tables here to relate them. Learn more

- **Step 3:** Double-click or drag **country** and **countrylanguage** table into the join canvas.



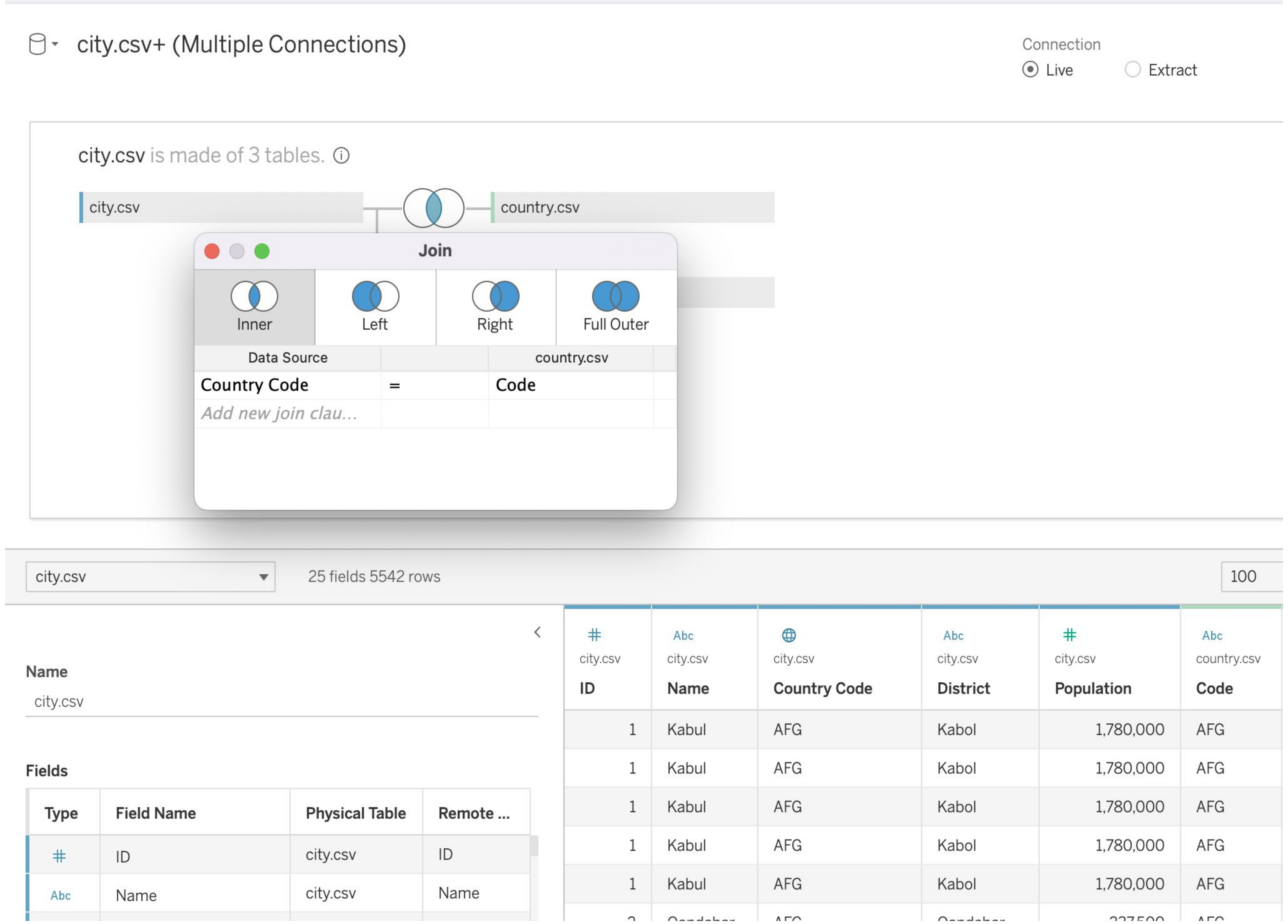
city.csv+ (Multiple Connections)

Connection
● Live ○ Extract

city.csv is made of 3 tables. ⓘ

Joining the tables using joins, cont’d

- **Step 4:** Click the **join icon** to configure the join.
  - In this use case, we’ll use **inner join** to join the tables.
  - The join clause is **country code**.



city.csv+ (Multiple Connections)

Connection ● Live ○ Extract

city.csv is made of 3 tables.

Data Source		country.csv
Country Code	=	Code
Add new join clau...		

city.csv 25 fields 5542 rows 100

Name
city.csv

Fields

Type	Field Name	Physical Table	Remote ...
#	ID	city.csv	ID
Abc	Name	city.csv	Name

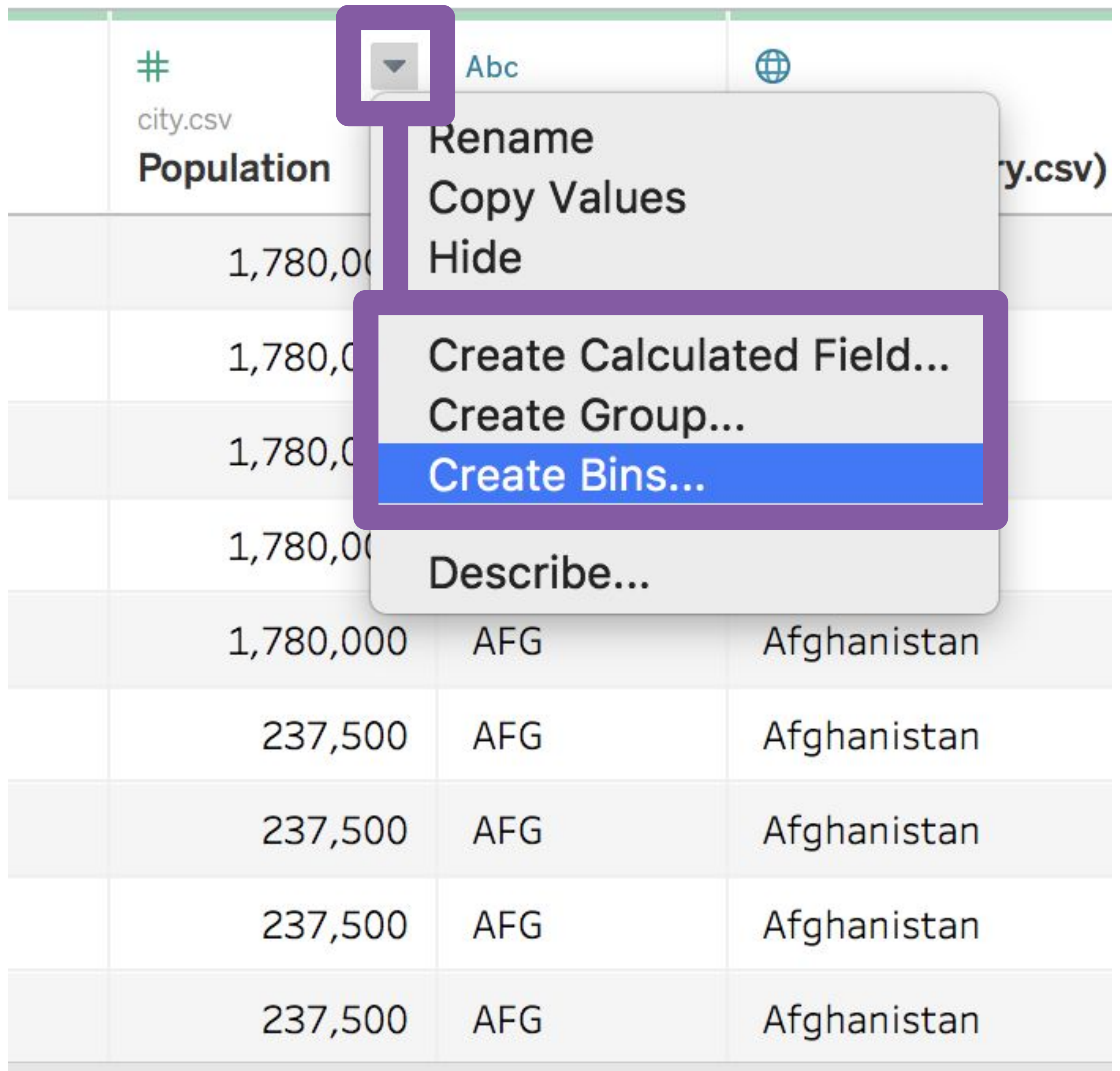
city.csv ID	city.csv Name	city.csv Country Code	city.csv District	city.csv Population	country.csv Code
1	Kabul	AFG	Kabol	1,780,000	AFG
1	Kabul	AFG	Kabol	1,780,000	AFG
1	Kabul	AFG	Kabol	1,780,000	AFG
1	Kabul	AFG	Kabol	1,780,000	AFG
1	Kabul	AFG	Kabol	1,780,000	AFG

Module completion checklist

Objective	Complete
Manually join tables using joins	✓
Introduce the concept of aggregating, binning, and grouping	
Create and use bins and groups	
Explore filtering capabilities of Tableau	

Aggregating, binning, and grouping

- Sometimes it makes sense to format a column into chunks.
  - Aggregating: Using a formula to calculate on some grouping of the data.
  - Binning: Sorting continuous data into bins by value.
  - Grouping: Using manual assignment to categorize data.
- Apply these with the dropdown menu to the right of each column.

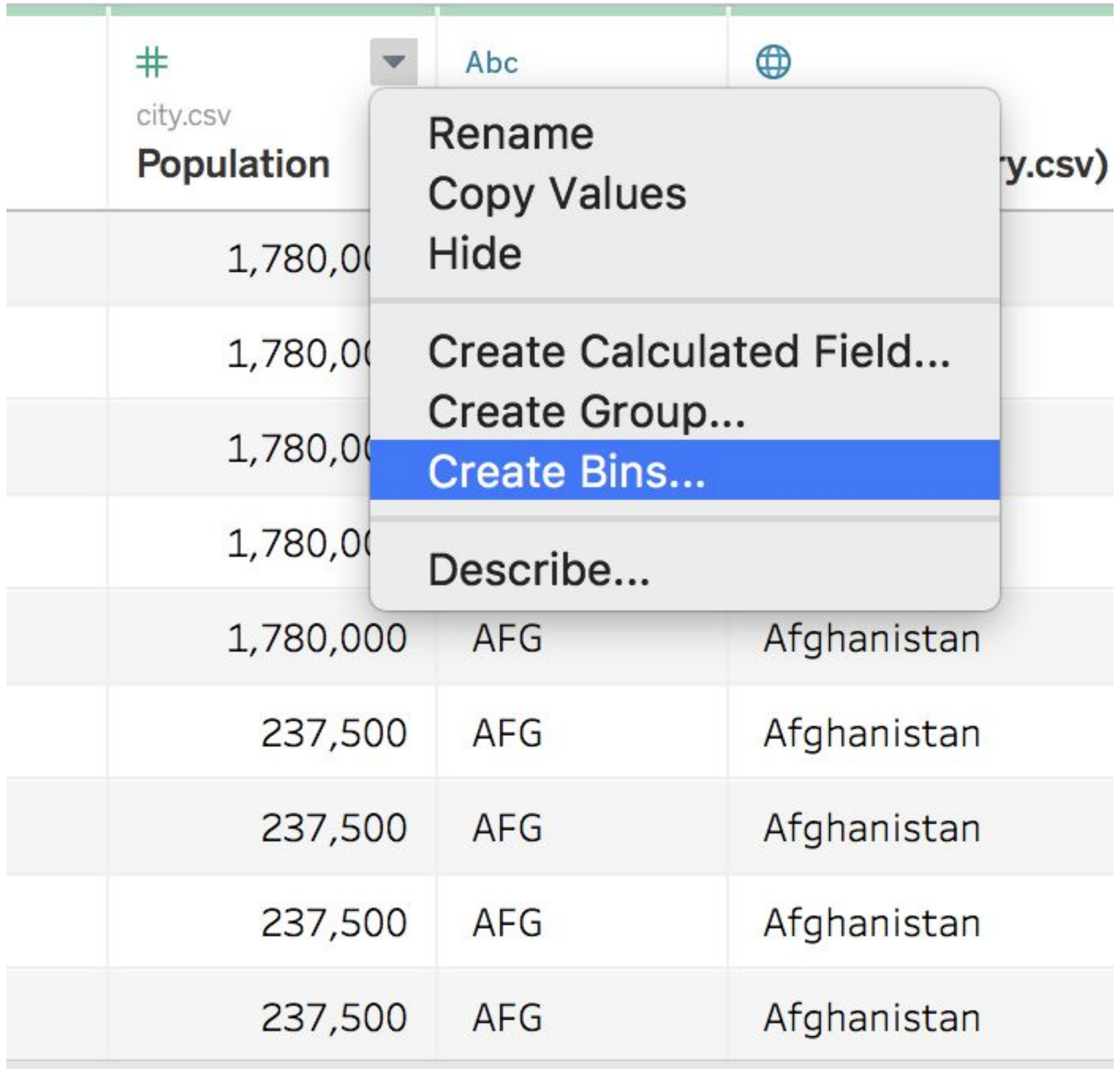


Population		y.csv)
1,780,0(		
1,780,(		
1,780,(		
1,780,0(		
1,780,000	AFG	Afghanistan
237,500	AFG	Afghanistan
237,500	AFG	Afghanistan
237,500	AFG	Afghanistan
237,500	AFG	Afghanistan

Where do these groups appear?

- Selecting any of these options will make new columns from the original column with some sort of summary of the column.

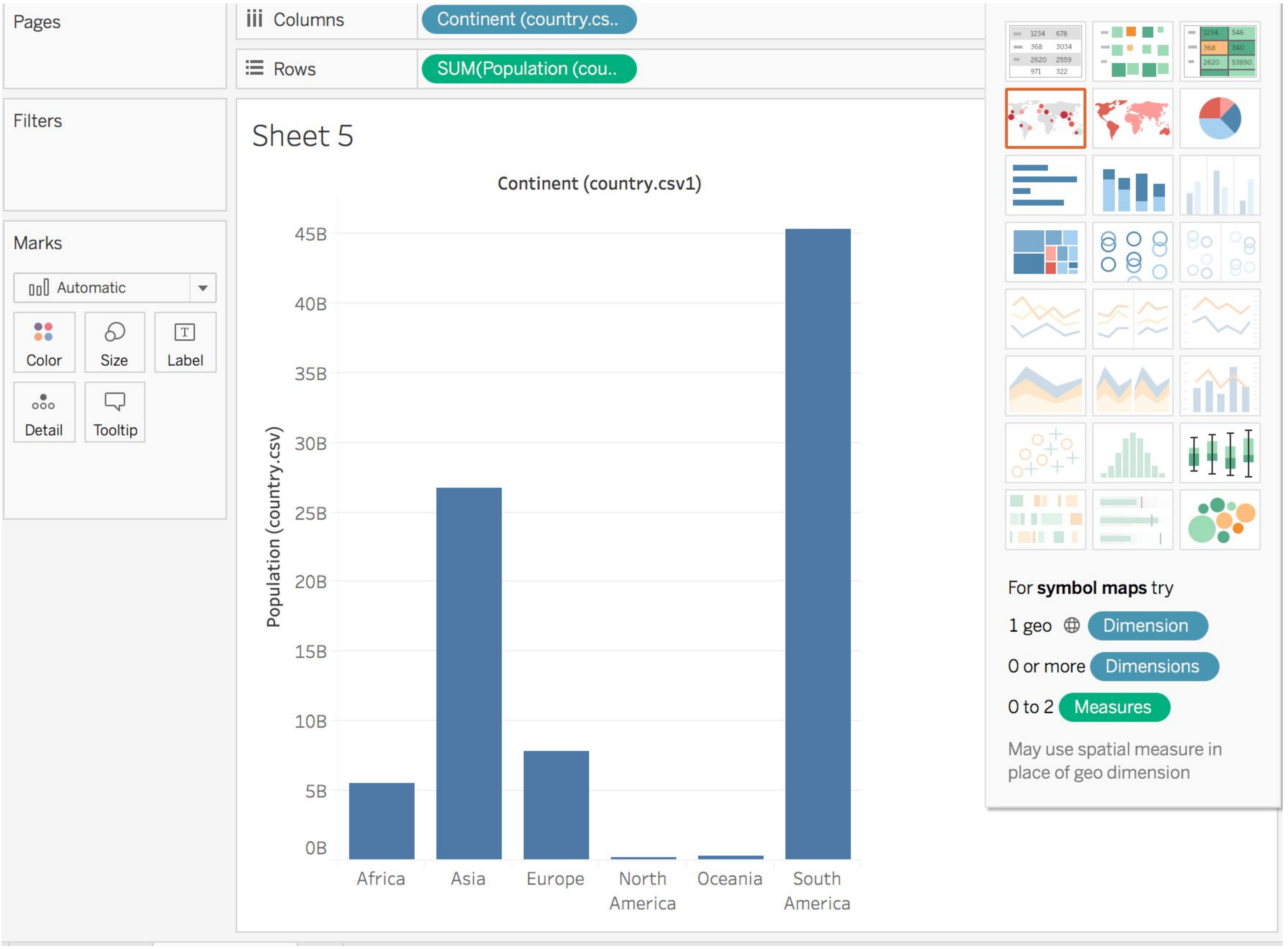
- Let's try this out on some of the world data.



# city.csv Population	Abc	y.csv)
1,780,0		
1,780,0		
1,780,0		
1,780,0		
1,780,000	AFG	Afghanistan
237,500	AFG	Afghanistan
237,500	AFG	Afghanistan
237,500	AFG	Afghanistan
237,500	AFG	Afghanistan

Data visualization: bar chart

- We will look at the **total population by continent.**



For **symbol maps** try

1 geo Dimension

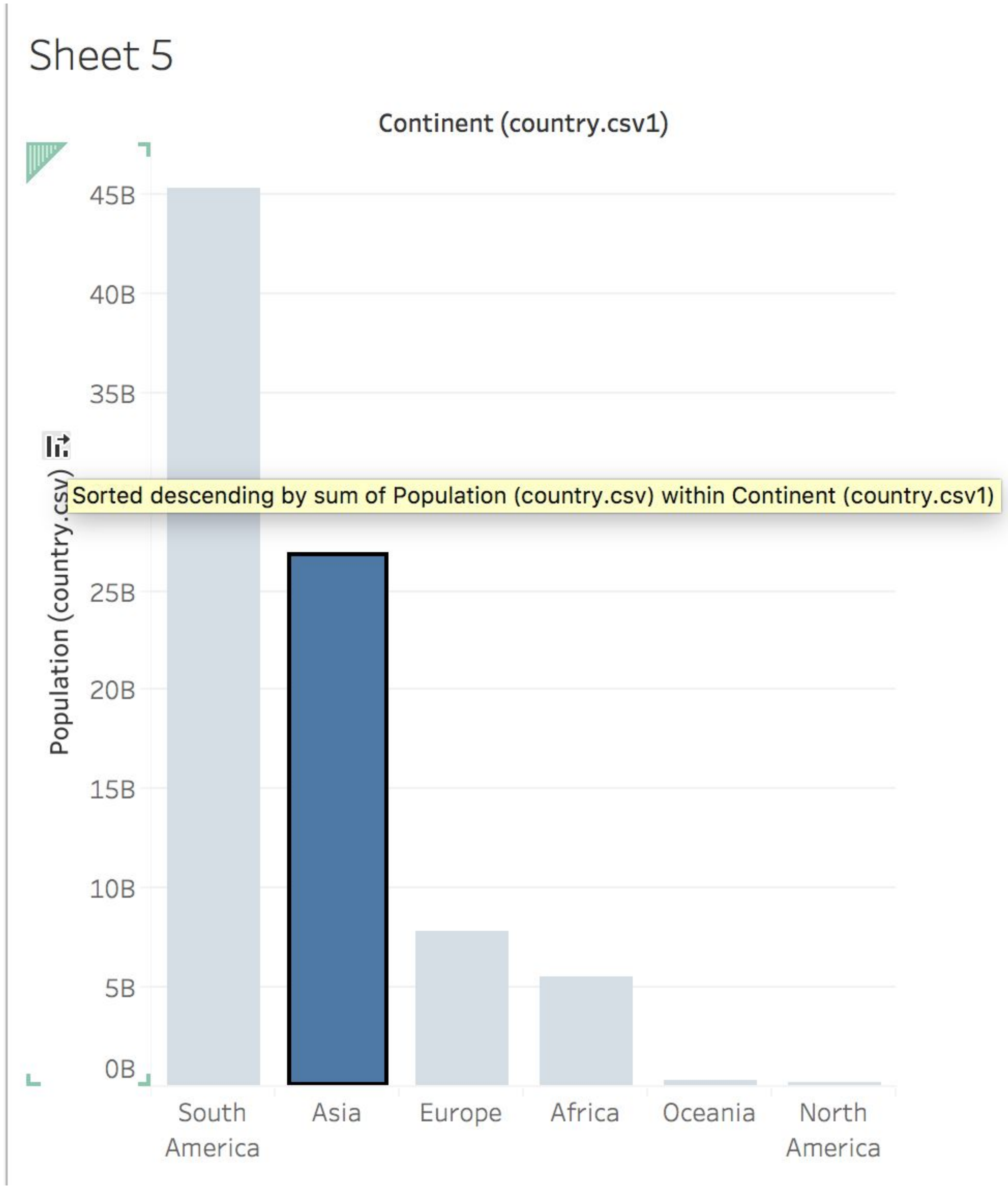
0 or more Dimensions

0 to 2 Measures

May use spatial measure in place of geo dimension

Sorting

- We can **sort by total population** by using this icon, located on the appropriate axis.



Sorted descending by sum of Population (country.csv) within Continent (country.csv1)

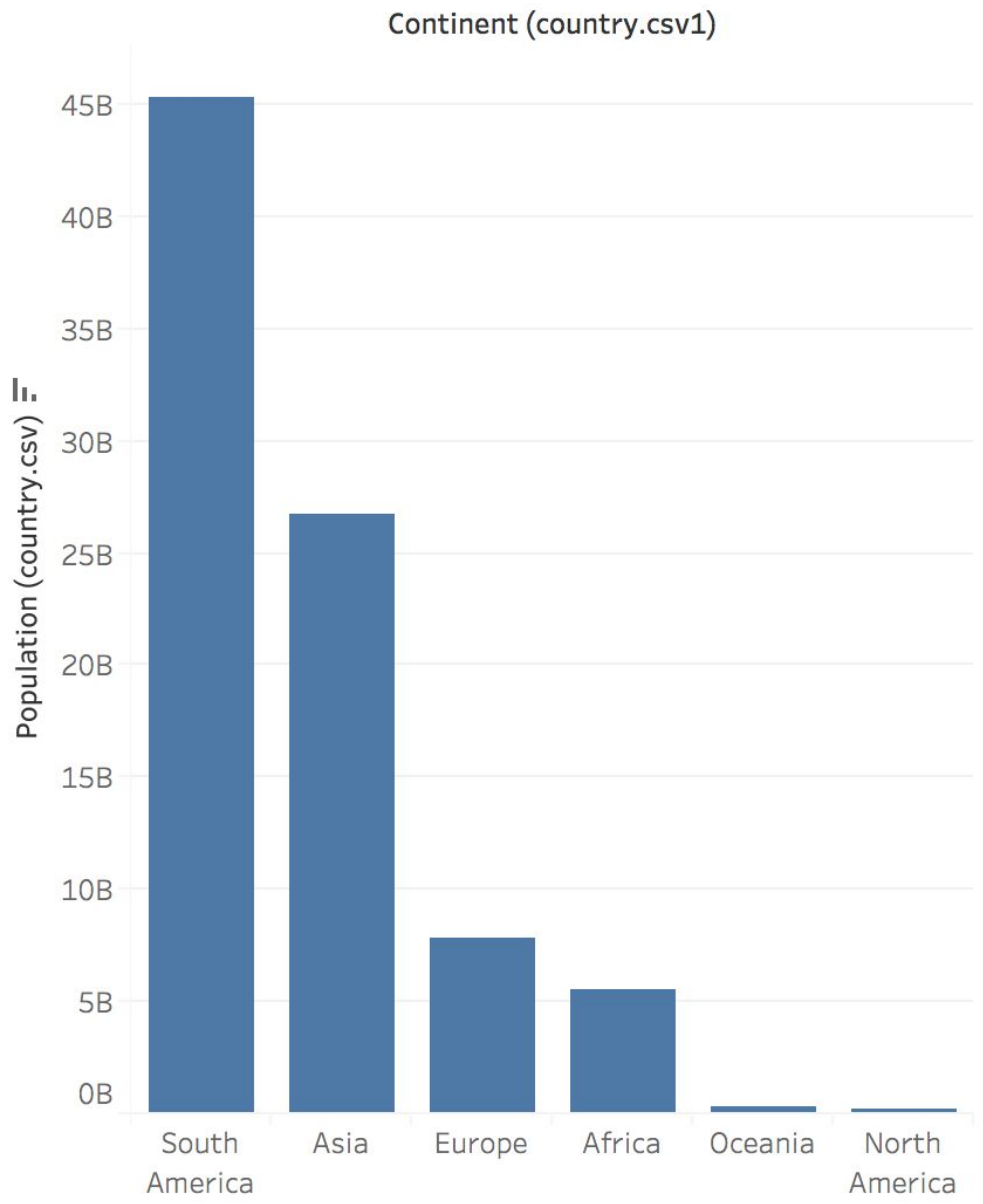
Data integrity

- Does this data make sense?
  - Are there more people in South America than Asia?

- What could be wrong?

- In today's Exercises, we will ask you to:
  - Rebuild these figures in your book.
  - Look for data integrity issues.

- We will come back to fix them.

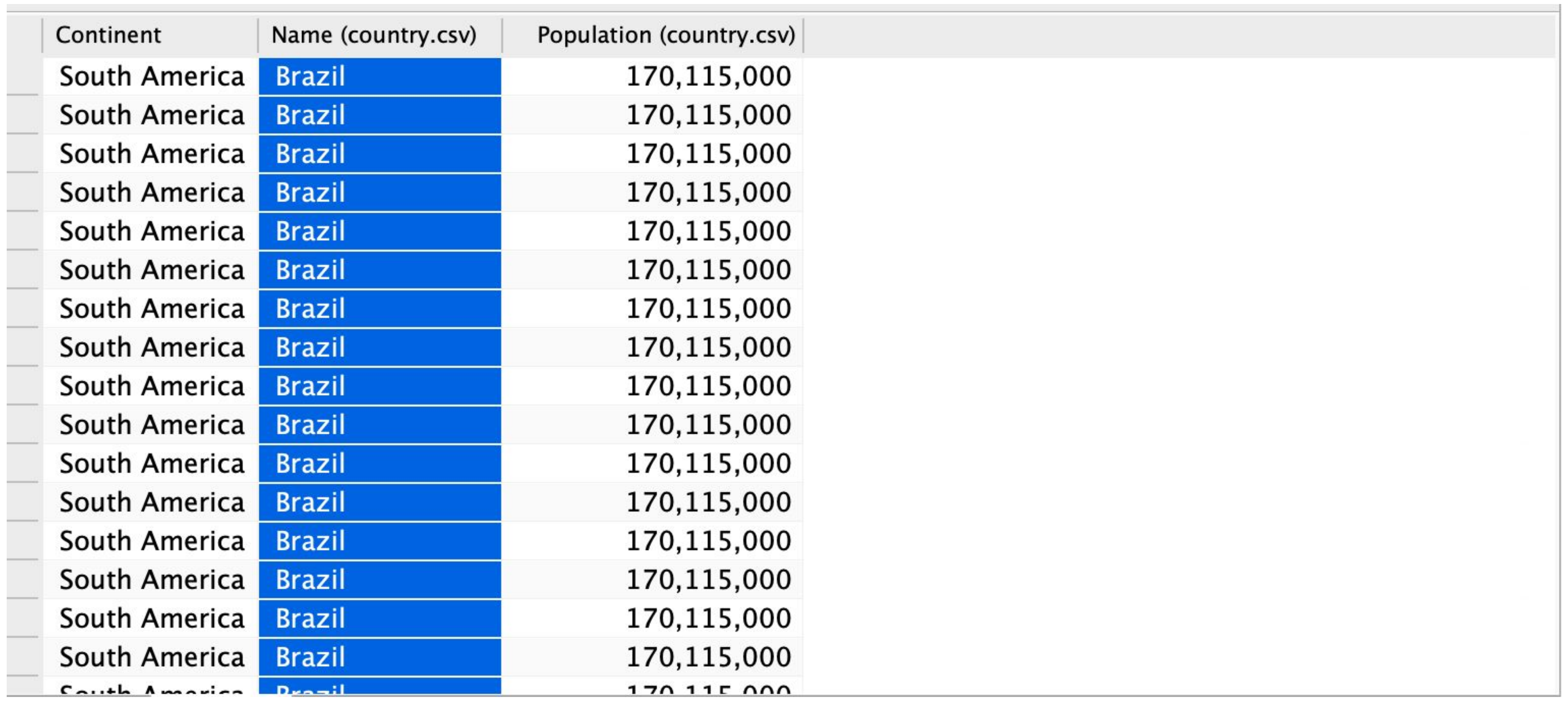


Aggregation in the data

- Check the data – the only way such high numbers are arising is that the **aggregation** of population is wrong.

- We want to calculate the population per continent with each country only represented **once**.

- We can apply the **ATTR (Attribute)** argument to tell Tableau that the populations are an attribute of each country.



Continent	Name (country.csv)	Population (country.csv)
South America	Brazil	170,115,000
South America	Brazil	170,115,000
South America	Brazil	170,115,000
South America	Brazil	170,115,000
South America	Brazil	170,115,000
South America	Brazil	170,115,000
South America	Brazil	170,115,000
South America	Brazil	170,115,000
South America	Brazil	170,115,000
South America	Brazil	170,115,000
South America	Brazil	170,115,000
South America	Brazil	170,115,000
South America	Brazil	170,115,000
South America	Brazil	170,115,000
South America	Brazil	170,115,000
South America	Brazil	170,115,000
South America	Brazil	170,115,000

Note: Brazil has 170 million people.

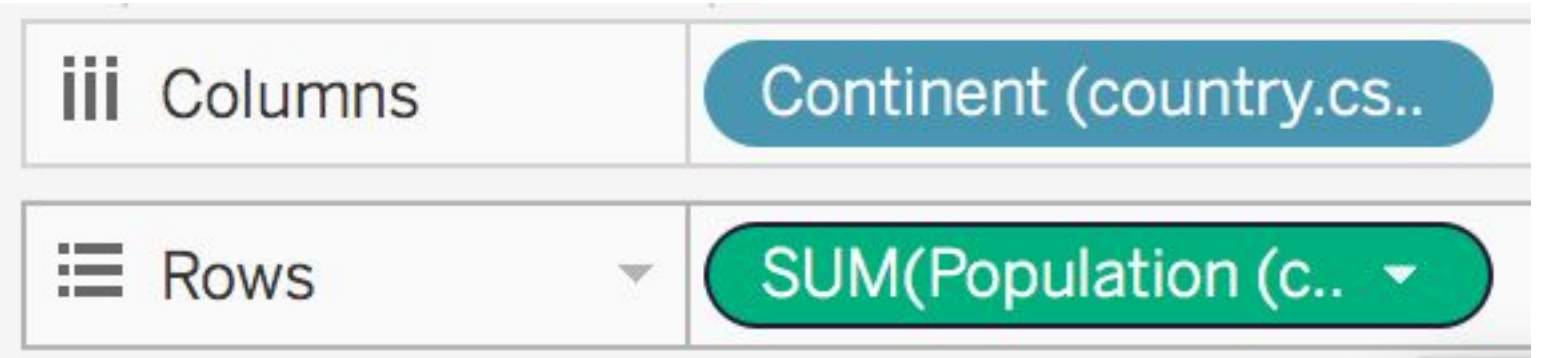
Aggregation and attribute

- Look closely at the row and column shelves in the pills above the graph.

- Population is being aggregated as a **sum** across all rows of each country.

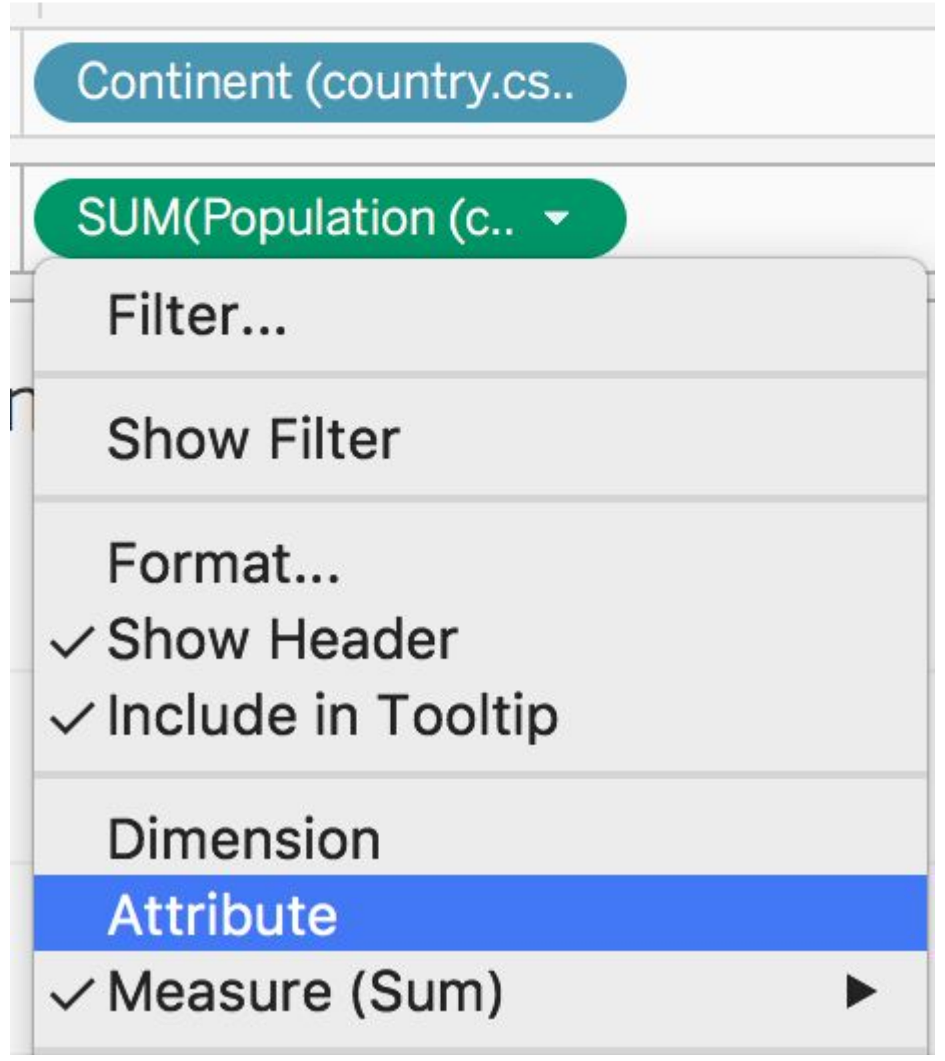
- Use the drop down in the pill and switch the aggregation to **“Attribute.”**

- This means that the country level values are taken as an **attribute** of the data rather than being further aggregated.



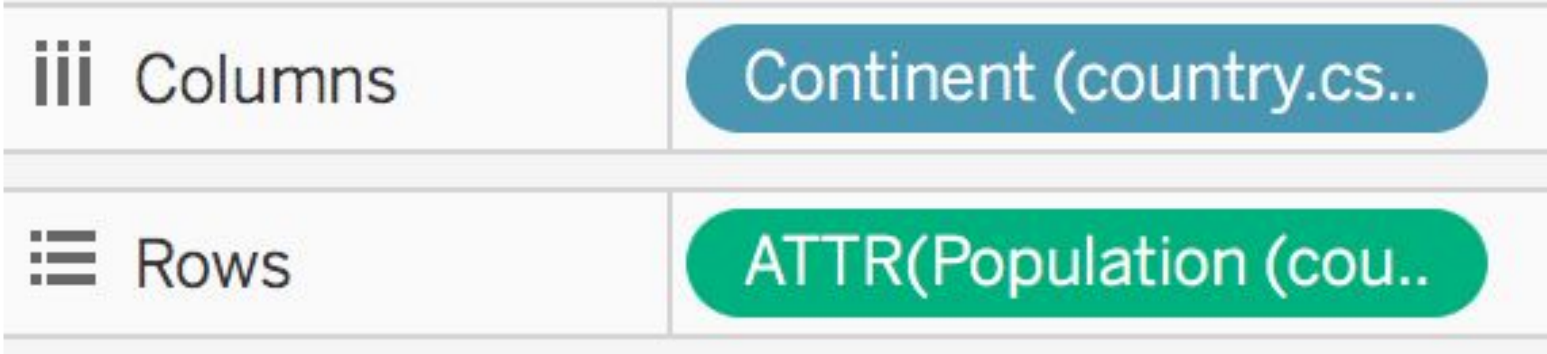
Columns: Continent (country.cs..

Rows: SUM(Population (c..



Continent (country.cs..

SUM(Population (c..



Columns: Continent (country.cs..

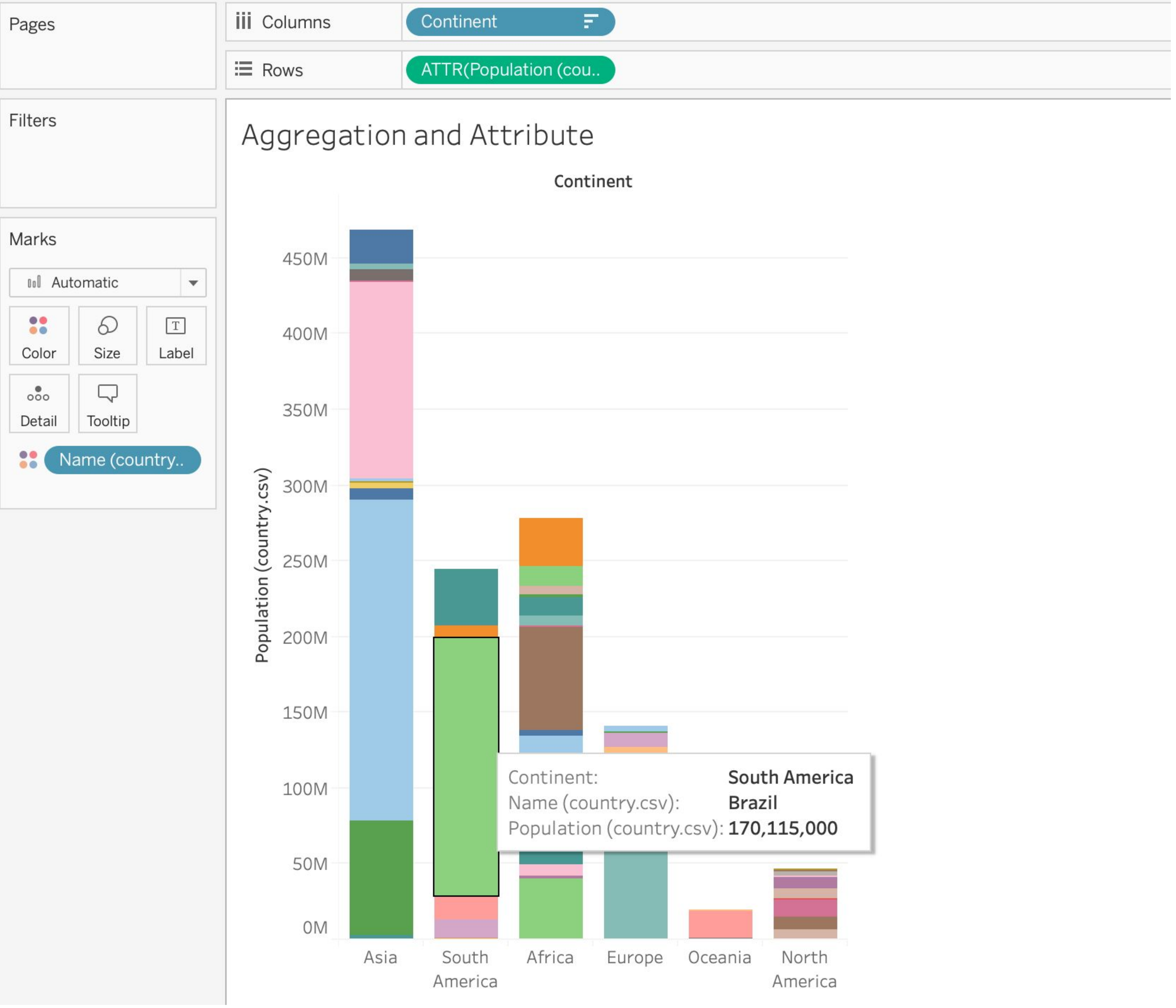
Rows: ATTR(Population (cou..

How has our visualization changed?

- Now we are treating the country populations on each row as the whole population of each country, in other words as an **attribute** of the country.

- Before, Tableau was treating each row as a part of each country's population and summing it together.

- We verify this by noting that the country populations are now as expected.



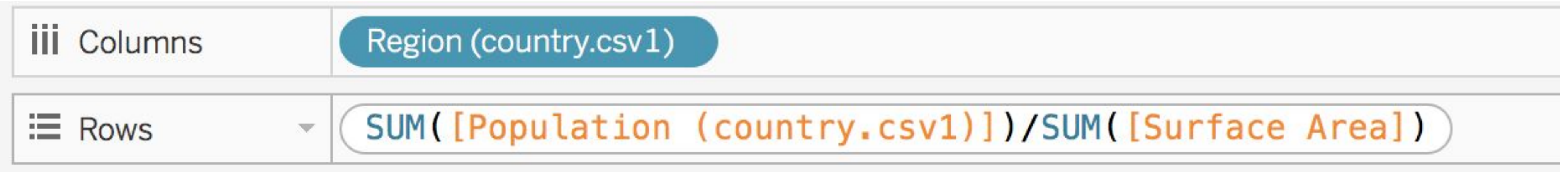
Columns: Continent
Rows: ATTR(Population (cou..

Aggregation and Attribute

Continent: South America
Name (country.csv): Brazil
Population (country.csv): 170,115,000

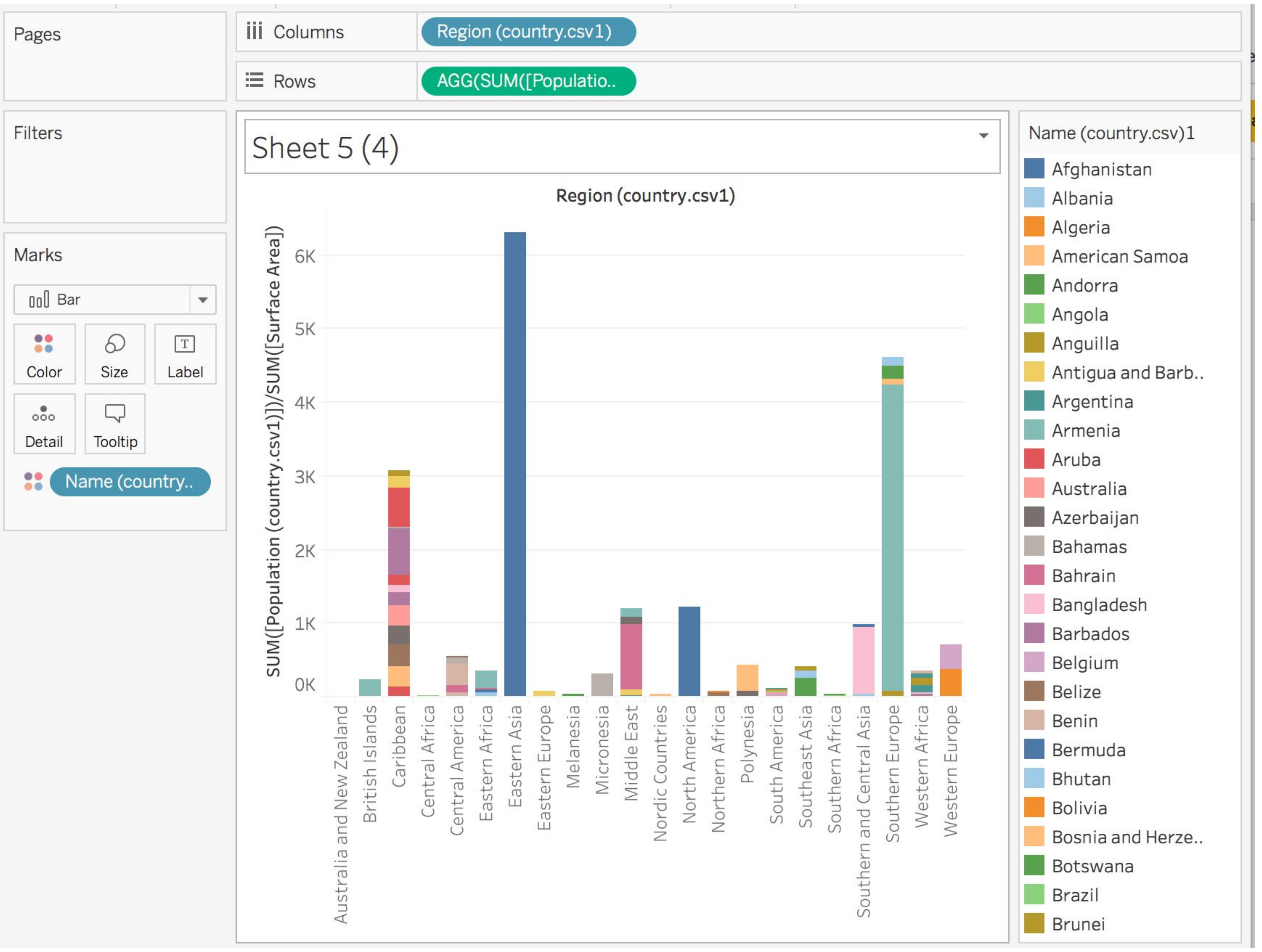
Aggregation: get population density

- Let’s try plotting **population density**:
  - First, we add an aggregating dimension in the **Columns** field.
  - Then we write the aggregating formula in the **Rows** field.
  - When we press enter, Tableau makes an **aggregation formula** that calculates the new value per region.
    - Note that this is *not* a new column – rather, it is cast as **“AGG().”**



Columns: Region (country.csv1)

Rows: SUM([Population (country.csv1)])/SUM([Surface Area])

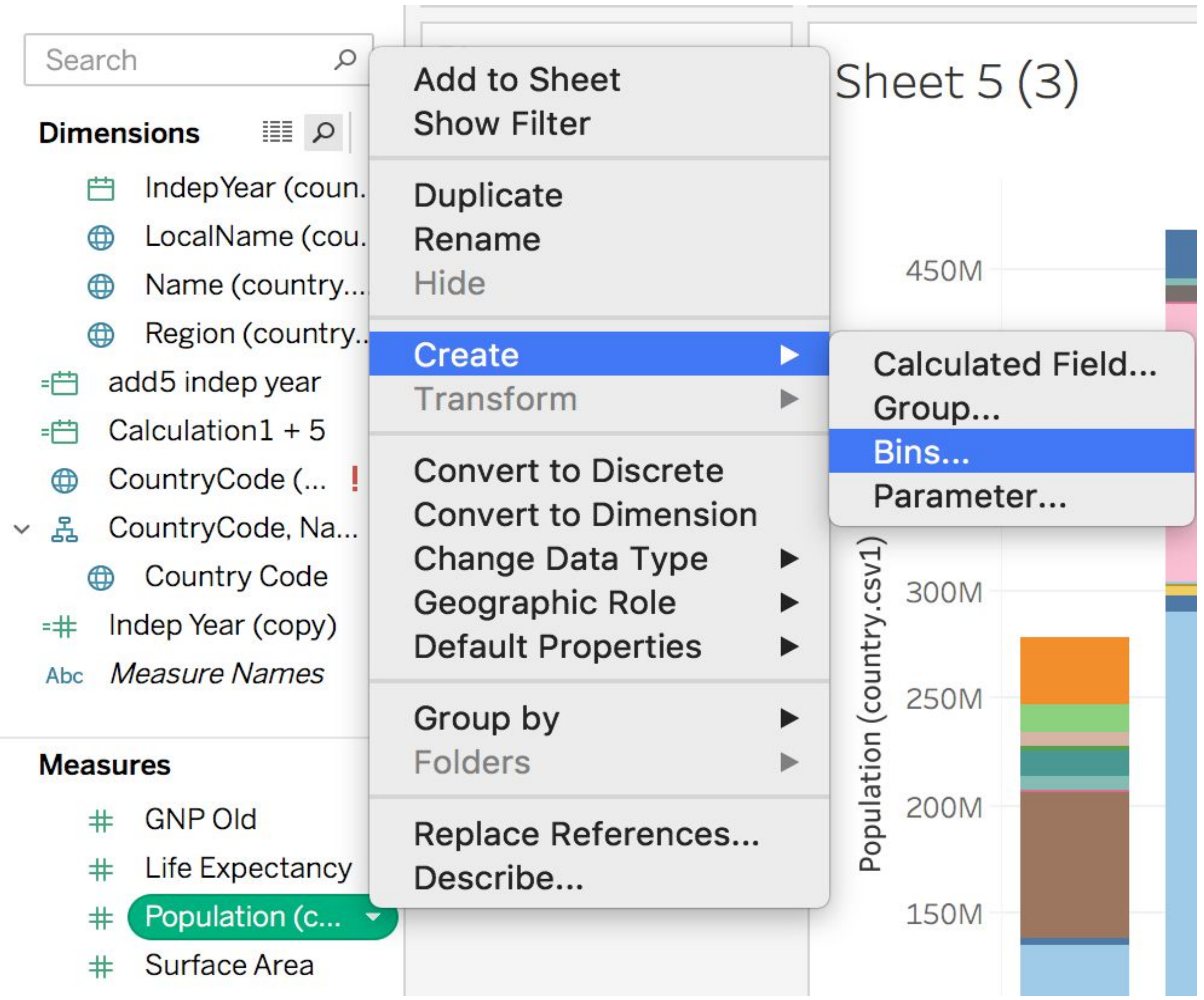


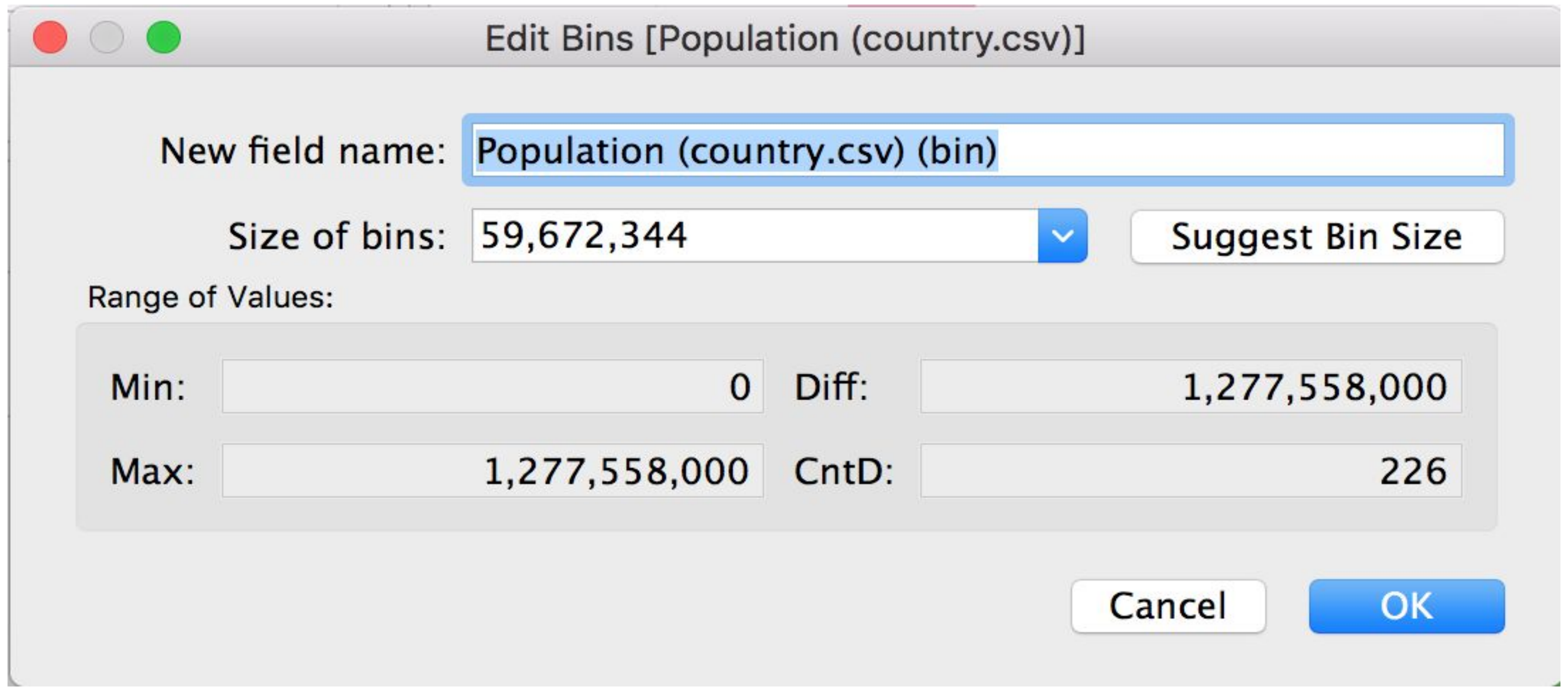
Module completion checklist

Objective	Complete
Manually join tables using joins	✓
Introduce the concept of aggregating, binning, and grouping	✓
Create and use bins and groups	
Explore filtering capabilities of Tableau	

Binning

- Now we will create **bins** based on the country's population.
- Binning can help us group many **continuous values** into smaller bins, or numerical ranges, for easier analysis.




Edit Bins [Population (country.csv)]

New field name: Population (country.csv) (bin)

Size of bins: 59,672,344

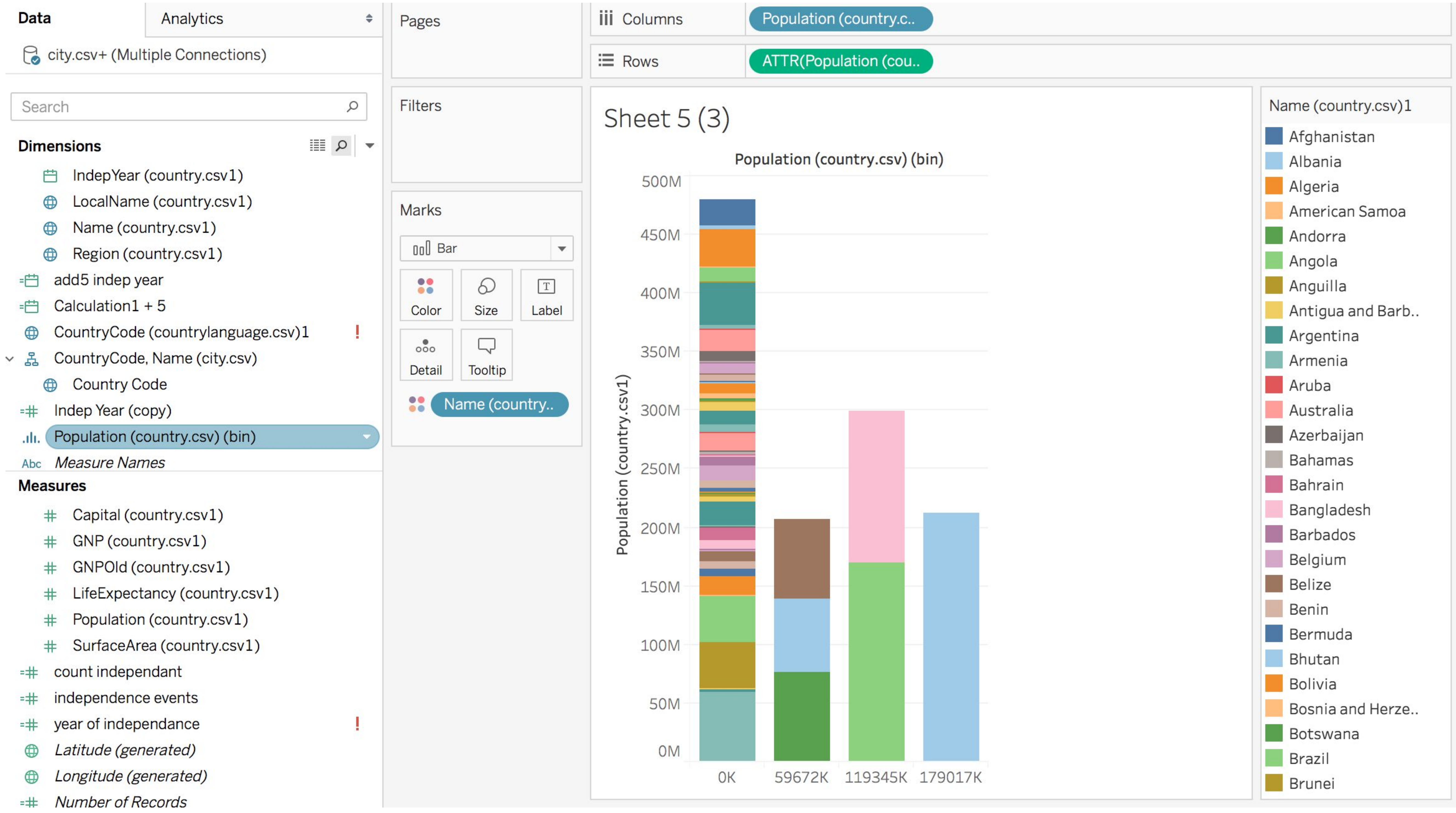
Range of Values:

Min:	0	Diff:	1,277,558,000
Max:	1,277,558,000	CntD:	226

Binning, cont'd.

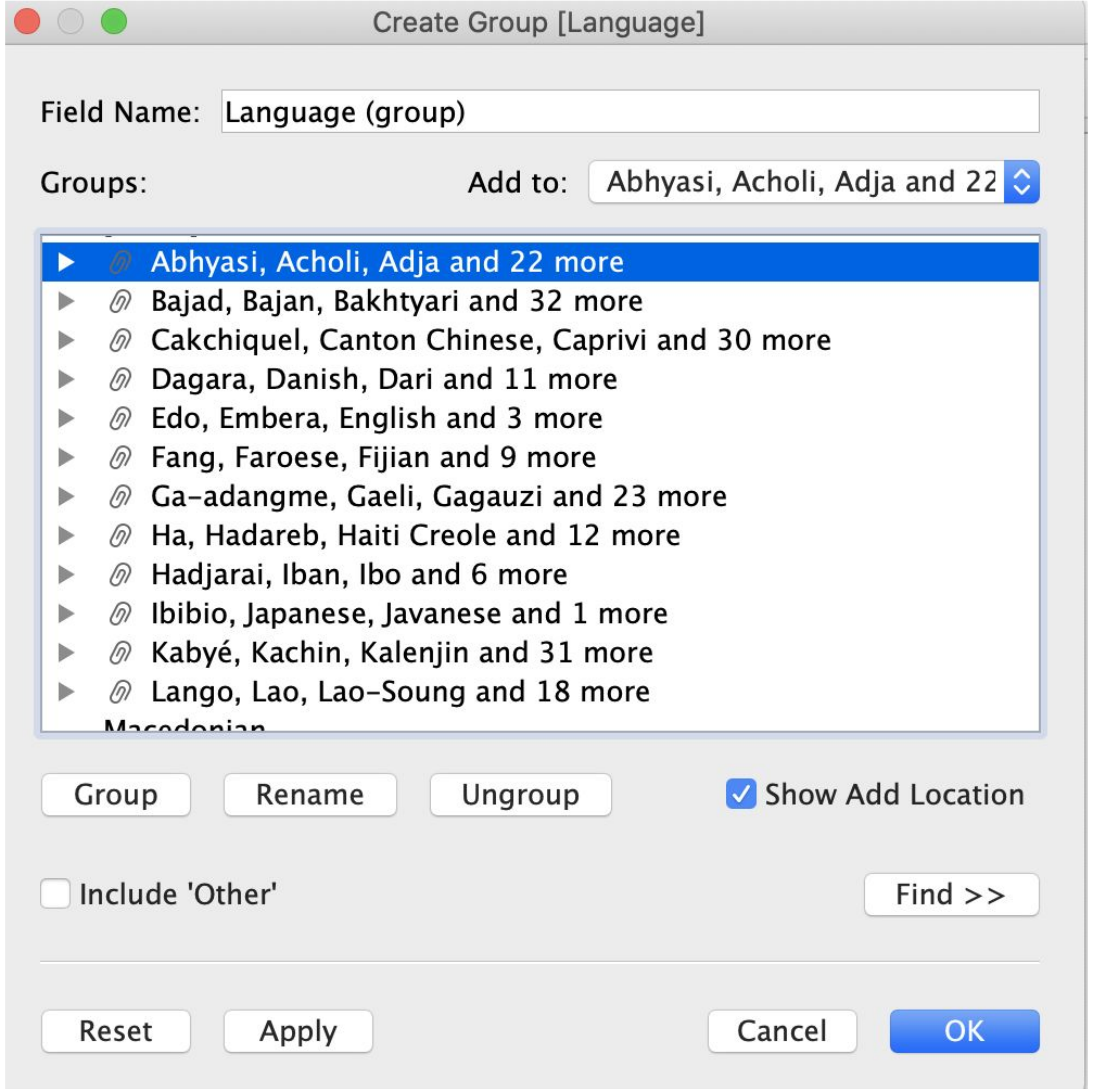
- We can now can see that most countries fall into the bin with the smallest values.

- Note that we are using the population attribute to keep Tableau from counting countries multiple times.



Grouping

- If your data can be classed into obvious or natural categories, you might want to organize them using **grouping.**
- You can specify groups manually.
- For instance, languages might be grouped into different classes alphabetically.



Create Group [Language]

Field Name: Language (group)

Groups: Add to: Abhyasi, Acholi, Adja and 22

- Abhyasi, Acholi, Adja and 22 more
- Bajad, Bajan, Bakhtyari and 32 more
- Cakchiquel, Canton Chinese, Caprivi and 30 more
- Dagara, Danish, Dari and 11 more
- Edo, Embera, English and 3 more
- Fang, Faroese, Fijian and 9 more
- Ga-adangme, Gaeli, Gagauzi and 23 more
- Ha, Hadareb, Haiti Creole and 12 more
- Hadjarai, Iban, Ibo and 6 more
- Ibibio, Japanese, Javanese and 1 more
- Kabyé, Kachin, Kalenjin and 31 more
- Lango, Lao, Lao-Soung and 18 more
- Macedonian

Knowledge check 3

Exercise 3

Module completion checklist

Objective	Complete
Manually join tables using joins	✓
Introduce the concept of aggregating, binning, and grouping	✓
Create and use bins and groups	✓
Explore filtering capabilities of Tableau	

End of Part 3