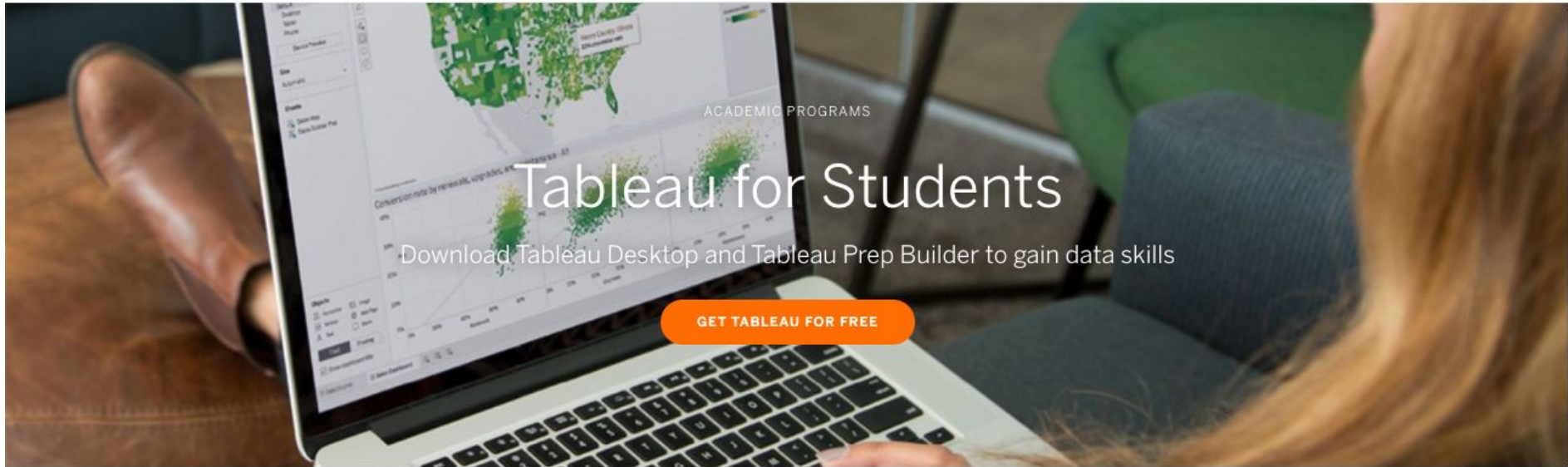


# ORIE 3120

## Lecture 8: Tableau and Database Connections



[FREE STUDENT LICENSE](#)

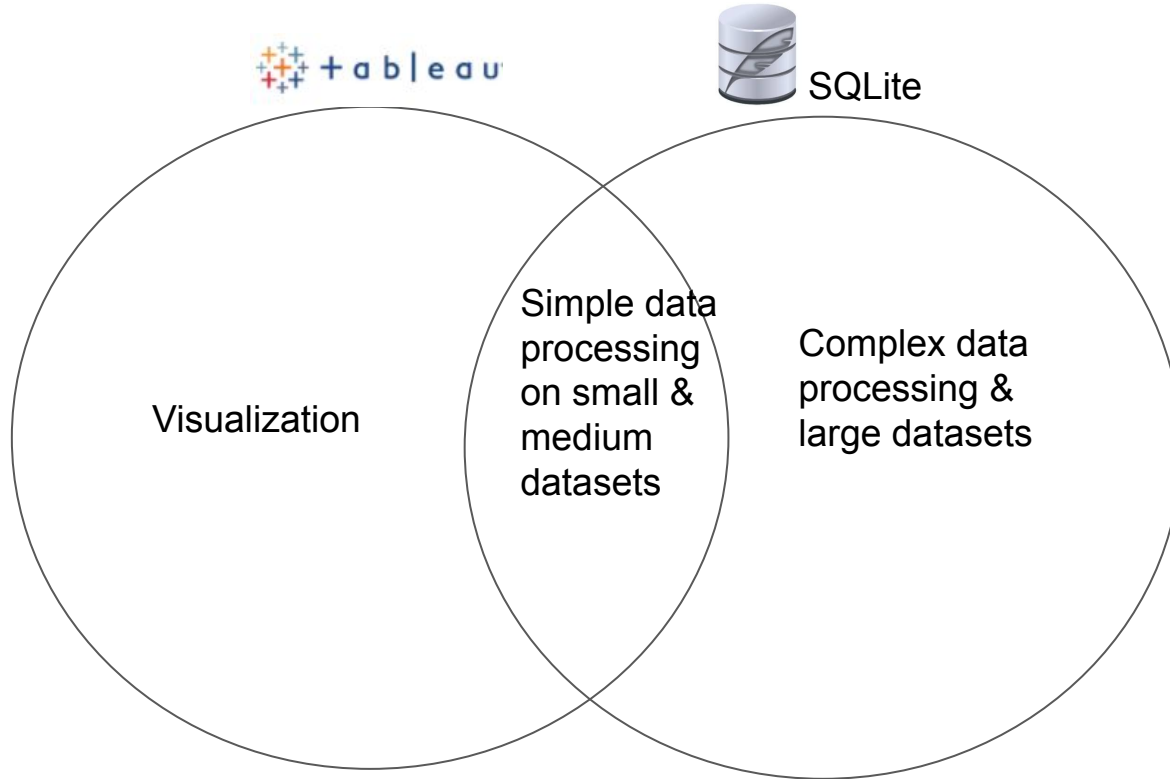


We offer free one-year Tableau licenses to students at accredited academic institutions through our Tableau for Students program. Receive access to our entire eLearning suite once verified.

# How to Install Tableau

- Tableau is a tool for data visualization
- There are two pieces of software: **Tableau Desktop** & Tableau Prep Builder.  
We'll focus on the first one, Tableau Desktop
- Instructions:
  - Click on [this link](#) and select "Download Tableau Desktop" (we don't need "Tableau Prep Builder"). On the form, enter your Cornell email address for Business E-mail and enter Cornell for Organization.
  - Activate with the ORIE 3120 your product key: TCPD-8E63-6AF0-65E2-52BF
  - Already have a copy of Tableau Desktop installed? Update your license in the application: Help menu → Manage Product Keys
- This key is specific to our course. There is also 1 year free license for students.  
Visit [tableau.com/academic/students](https://tableau.com/academic/students) to download the software & get the license.

# What's the relationship between Tableau & SQLite?



# What's the relationship between Tableau & SQL?



Pros of **Tableau** for data analysis

- Graphical user interface is intuitive & easier to learn



SQLite

Pros of **SQL** for data analysis

- Can analyze datasets with hundreds of millions of records
- Supports complex tasks
- Integrates with real-time operations and many users

# Advice

If your analysis isn't too complex & your data isn't too big:

- Do all your analysis in Tableau

If your analysis is too slow or complex to do in Tableau, or you need to collaborate:

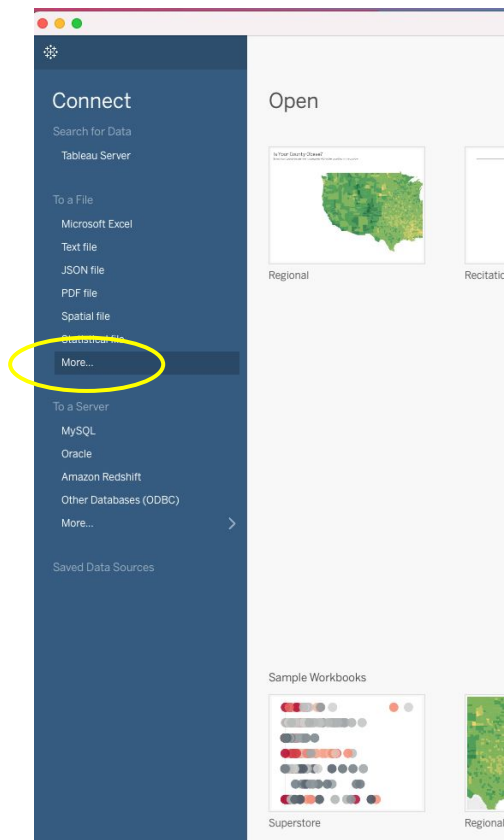
- Do your analysis in SQL and then visualize it in Tableau
- To use data from SQL in Tableau, either export as a CSV or create a direct database connection

# Visualization in Tableau

# Importing data

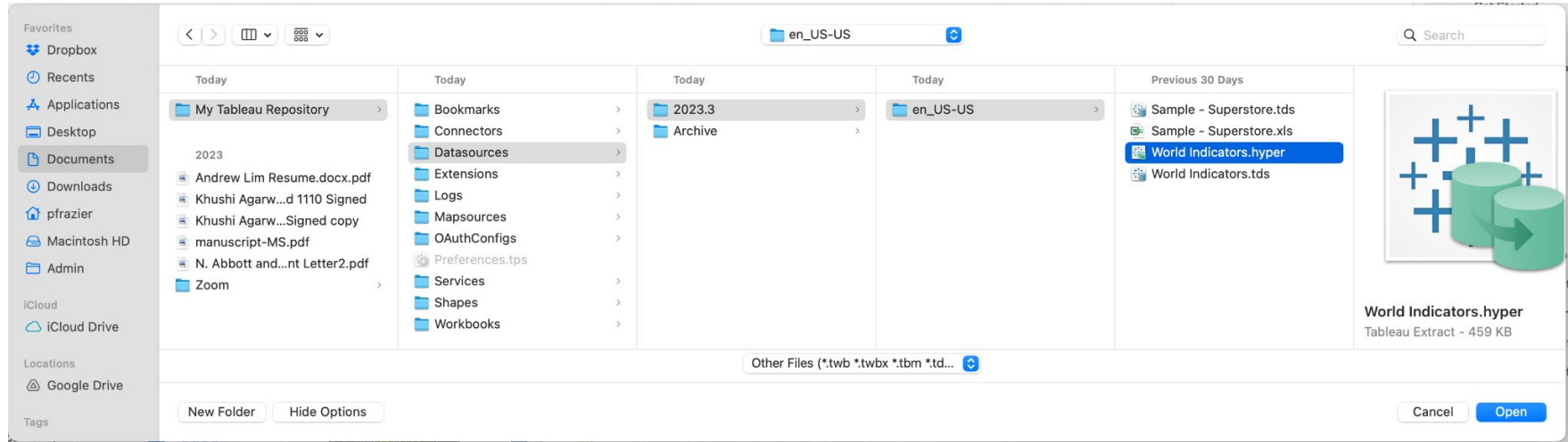
On the left side of Tableau's menu, choose the type of data you are trying to import.

Click "More..."





# We'll use the World Indicators sample data source



# It should look like this

Tableau - Book1 - Tableau license expires in 13 days

World Indicators Extract

Connection: ☒ Live ☐ Extract

Filters: 0 | [Add](#)

Extract

Need more data?  
Drag tables here to relate them. [Learn more](#)

Click this to move the panel out of the way

Extract 26 fields 2691 rows 100 rows

Name	Type	Field Name	Physical Table	Rem...	World Indicators.csv	World Indicators.csv	World Indicators.csv	World Indicators.csv	World Indicator
Extract					Birth Rate	Business Tax Rate	CO2 Emissions	Country/Region	Days to Sta
	#	Birth Rate	World Indicato...	Birth ...	0.0200000	null	87,931	Algeria	
	#	Business Tax Rate	World Indicato...	Busin...	0.0500000	null	9,542	Angola	
	#	CO2 Emissions	World Indicato...	CO2 E...	0.0430000	null	1,617	Benin	
	#	Country/Region	World Indicato...	Count...	0.0270000	null	4,276	Botswana	
	#	Days to Start Business	World Indicato...	Days t...	0.0460000	null	1,041	Burkina Faso	
	#				0.0420000	null	301	Burundi	
	#				0.0410000	null	3,432	Cameroon	
	#				0.0390000	null	268	Central African Republic	

Go to Worksheet

Data Source Sheet 1

# Now it will look like this

World Indicators Extract

Extract

Need more data?

Drag tables here to relate them. [Learn more](#)

World Indicators...

World Indicators.csv

World Indicators.csv

World Indicators.csv

World Indicators.csv

World Indicators.csv

World Indicators.csv

World Indicators.csv

World Indicators.csv

World Indicators.csv

World Indicators.csv

#	#	#	#	#	#	#	#	#	#	#
World Indicators...	World Indicators.csv	World Indicators.csv	World Indicators.csv	World Indicators.csv	World Indicators.csv	World Indicators.csv	World Indicators.csv	World Indicators.csv	World Indicators.csv	World Indicators.csv
Birth Rate	Business Tax Rate	CO2 Emissions	Country/Region	Days to Start Busi...	Ease of Business	Energy Usage	GDP	Health Exp % G...	Health Exp/Cap...	Hours to do T...
0.0200000	null	87,931	Algeria	null	null	26,998	54,790,058,957.00	0.035000	60	null
0.0500000	null	9,542	Angola	null	null	7,499	9,129,594,819.00	0.034000	22	null
0.0430000	null	1,617	Benin	null	null	1,983	2,359,122,303.00	0.043000	15	null
0.0270000	null	4,276	Botswana	null	null	1,836	5,788,311,645.00	0.047000	152	null
0.0460000	null	1,041	Burkina Faso	null	null	null	2,610,959,139.00	0.051000	12	null
0.0420000	null	301	Burundi	null	null	null	870,486,066.00	0.063000	7	null
0.0410000	null	3,432	Cameroon	null	null	6,310	9,287,367,569.00	0.044000	26	null
0.0390000	null	268	Central African Repu...	null	null	null	914,500,332.00	0.043000	11	null
0.0510000	null	176	Chad	null	null	null	1,385,058,212.00	0.063000	10	null
0.0390000	null	84	Comoros	null	null	null	201,899,884.00	0.035000	13	null
0.0480000	null	1,646	Congo (Kinshasa)	null	null	16,679	19,088,046,305.00	0.048000	14	null
0.0390000	null	1,049	Congo (Brazzaville)	null	null	814	3,219,910,666.00	0.021000	22	null
0.0380000	null	6,791	Cote d'Ivoire	null	null	6,734	10,417,060,605.00	0.065000	42	null
0.0300000	null	403	Djibouti	null	null	null	551,230,862.00	0.058000	44	null
0.0250000	null	141,326	Egypt	null	null	40,658	99,838,540,997.00	0.054000	79	null

World Indicators Extract

Extract

Need more data?

Drag tables here to relate them. [Learn more](#)

World Indicators...

World Indicators.csv

World Indicators.csv

World Indicators.csv

World Indicators.csv

World Indicators.csv

World Indicators.csv

World Indicators.csv

World Indicators.csv

World Indicators.csv

World Indicators.csv

#	#	#	#	#	#	#	#	#	#	#
World Indicators...	World Indicators.csv	World Indicators.csv	World Indicators.csv	World Indicators.csv	World Indicators.csv	World Indicators.csv	World Indicators.csv	World Indicators.csv	World Indicators.csv	World Indicators.csv
Birth Rate	Business Tax Rate	CO2 Emissions	Country/Region	Days to Start Busi...	Ease of Business	Energy Usage	GDP	Health Exp % G...	Health Exp/Cap...	Hours to do T...
0.0200000	null	87,931	Algeria	null	null	26,998	54,790,058,957.00	0.035000	60	null
0.0500000	null	9,542	Angola	null	null	7,499	9,129,594,819.00	0.034000	22	null
0.0430000	null	1,617	Benin	null	null	1,983	2,359,122,303.00	0.043000	15	null
0.0270000	null	4,276	Botswana	null	null	1,836	5,788,311,645.00	0.047000	152	null
0.0460000	null	1,041	Burkina Faso	null	null	null	2,610,959,139.00	0.051000	12	null
0.0420000	null	301	Burundi	null	null	null	870,486,066.00	0.063000	7	null
0.0410000	null	3,432	Cameroon	null	null	6,310	9,287,367,569.00	0.044000	26	null
0.0390000	null	268	Central African Repu...	null	null	null	914,500,332.00	0.043000	11	null
0.0510000	null	176	Chad	null	null	null	1,385,058,212.00	0.063000	10	null
0.0390000	null	84	Comoros	null	null	null	201,899,884.00	0.035000	13	null
0.0480000	null	1,646	Congo (Kinshasa)	null	null	16,679	19,088,046,305.00	0.048000	14	null
0.0390000	null	1,049	Congo (Brazzaville)	null	null	814	3,219,910,666.00	0.021000	22	null
0.0380000	null	6,791	Cote d'Ivoire	null	null	6,734	10,417,060,605.00	0.065000	42	null
0.0300000	null	403	Djibouti	null	null	null	551,230,862.00	0.058000	44	null
0.0250000	null	141,326	Egypt	null	null	40,658	99,838,540,997.00	0.054000	79	null

This is like an SQL Table or an Excel Sheet

You can sort data, create calculated fields, & do some other operations

You can also add other tables with joins & relationships (more soon)

# Creating calculated fields

## Step 1:

	# World Indicators.csv Life Expectancy Fe...	# World Indicators.csv Life Expectancy M...	# World Indicators.csv Mobile Phone Us
...			
3000	71		0.
3200	47		0.
	57		0.
5500	52		0.
	52		0.
5800	49		0.
2000	53	51	52.0000
2000	45	42	43.5000
2000	48	46	47.0000
2000	59	56	57.5000
	48	45	46.5000

## Step 2:

Life Expectancy

$$([Life\ Expectancy\ Male] + [Life\ Expectancy\ Female]) / 2$$

The calculation is valid.

Apply

OK

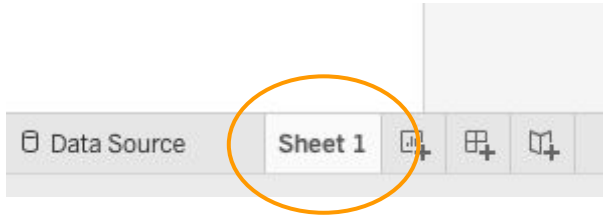
# Creating calculated fields

Here's the result:

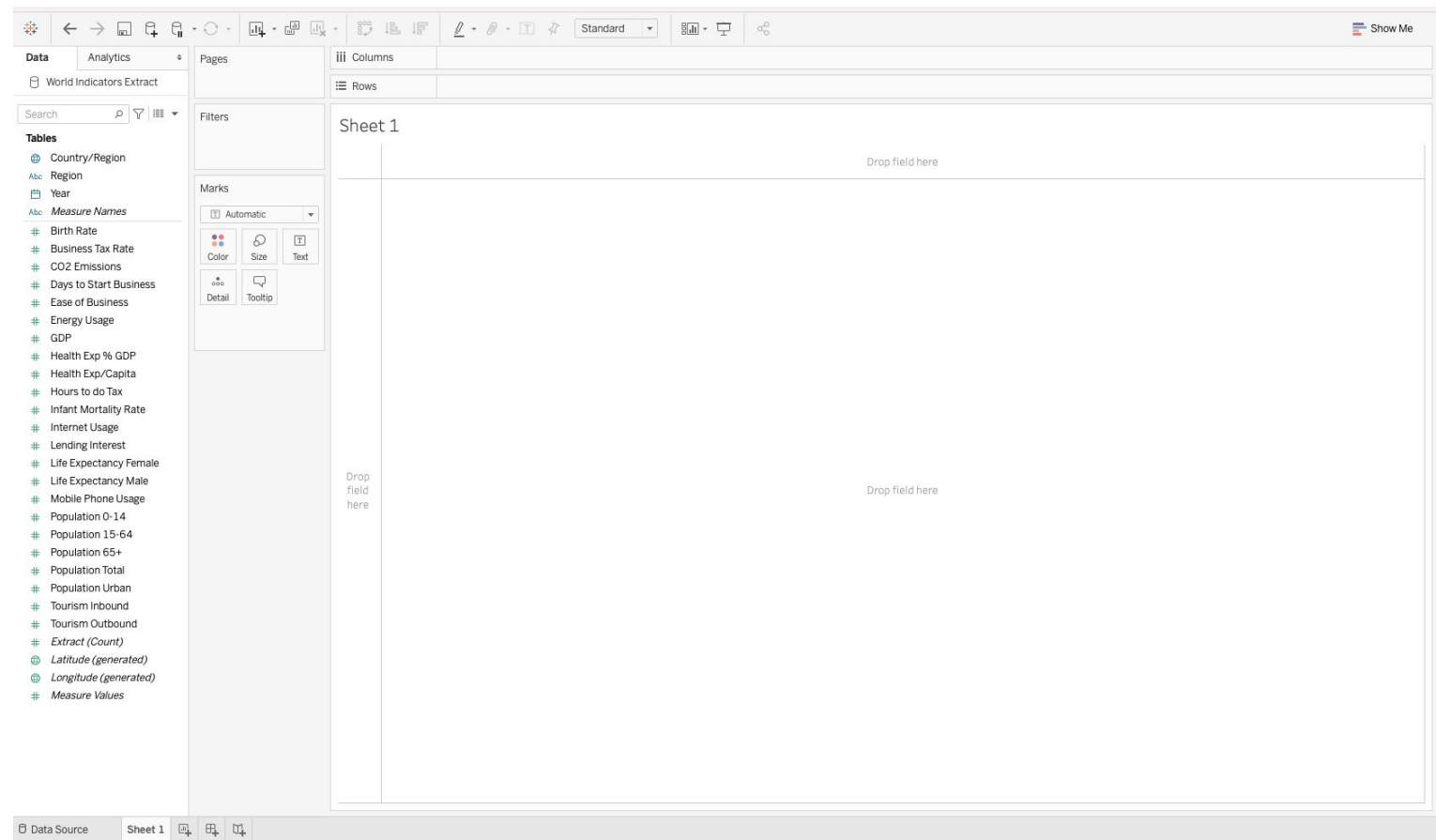
# World Indicators.csv Life Expectancy Female	=# Calculation Life Expectancy	# World Indicators.csv Life Expectancy Male
71	69.0000	67
47	45.5000	44
57	55.0000	53
52	50.5000	49
52	50.5000	49
49	48.0000	47
53	52.0000	51
45	43.5000	42
48	47.0000	46
59	57.5000	56
48	46.5000	45
53	52.0000	51
47	46.5000	46
59	57.0000	55
71	68.5000	66
49	47.5000	46

# We will make some graphs. Go to Sheet 1

Click in the lower left of the screen



It should look like this



## Tables

Country/Region  
Region  
Year  
*Measure Names*

## Dimensions

Birth Rate  
Business Tax Rate  
CO2 Emissions  
Days to Start Business  
Ease of Business  
Energy Usage  
GDP  
Health Exp % GDP  
Health Exp/Capita  
Hours to do Tax  
Infant Mortality Rate  
Internet Usage  
Lending Interest  
Life Expectancy Female  
Life Expectancy Male  
Mobile Phone Usage  
Population 0-14  
Population 15-64  
Population 65+  
Population Total  
Population Urban  
Tourism Inbound  
Tourism Outbound  
*Extract (Count)*  
*Latitude (generated)*  
*Longitude (generated)*  
*Measure Values*

## Measures

# Explanation for what we see

- Each field from the data source is listed with an icon indicating its type
- A field's color indicates whether it is “discrete” or “continuous”
- Dimensions are “qualitative” values that you might use to categorize or segment data , e.g., the name of a country or region.
  - Dimensions are usually discrete.
  - Think of them as “dimensions along which we can categorize”
- Measures are quantitative values that you can aggregate and plot as a number, e.g., the GDP
  - Measures are usually continuous.
  - Think of them as “things we’ve measured”
- You can change whether a field is a dimension or a measure by mousing over it and clicking on the arrow. You can also change its data type, whether it is treated as discrete or continuous, and some other properties.
- There are some other fields in italics. These are auto-generated. Don't worry about them for now.

[Tableau documentation](#) for more detail



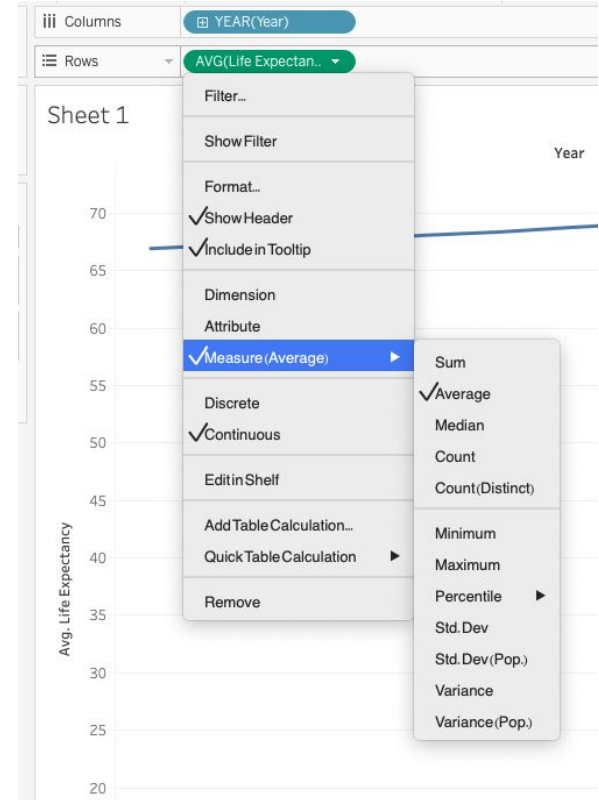
# Plotting Life Expectancy vs. Year

Drag “Year” onto the “Columns”

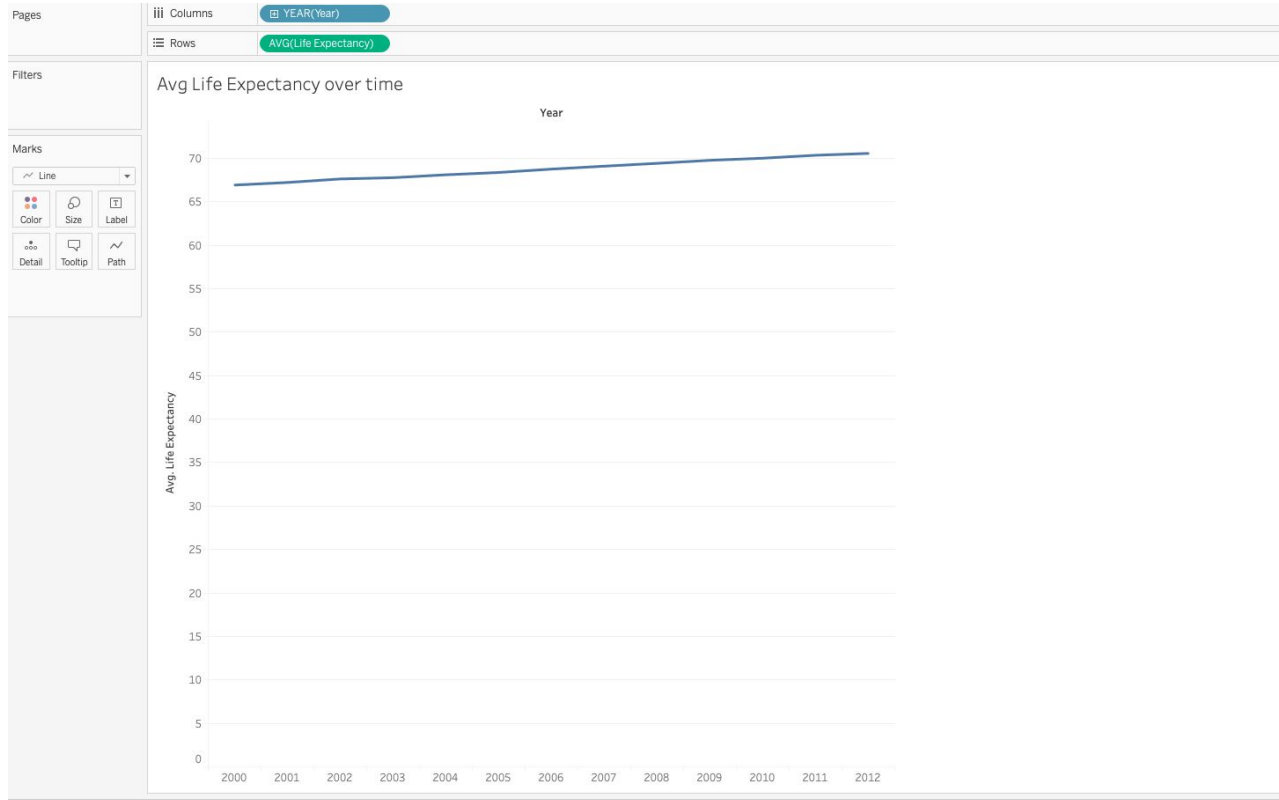
Drag “Life Expectancy” onto the “Rows”. It will become “SUM(Life Expectancy)”.

Tableau is doing a GROUP BY and plotting the results of  
`SELECT Year, SUM([Life Expectancy]) FROM [World Indicators] GROUP BY Year`

Change “SUM” to “AVG” so that it plots the average over countries instead of the SUM.



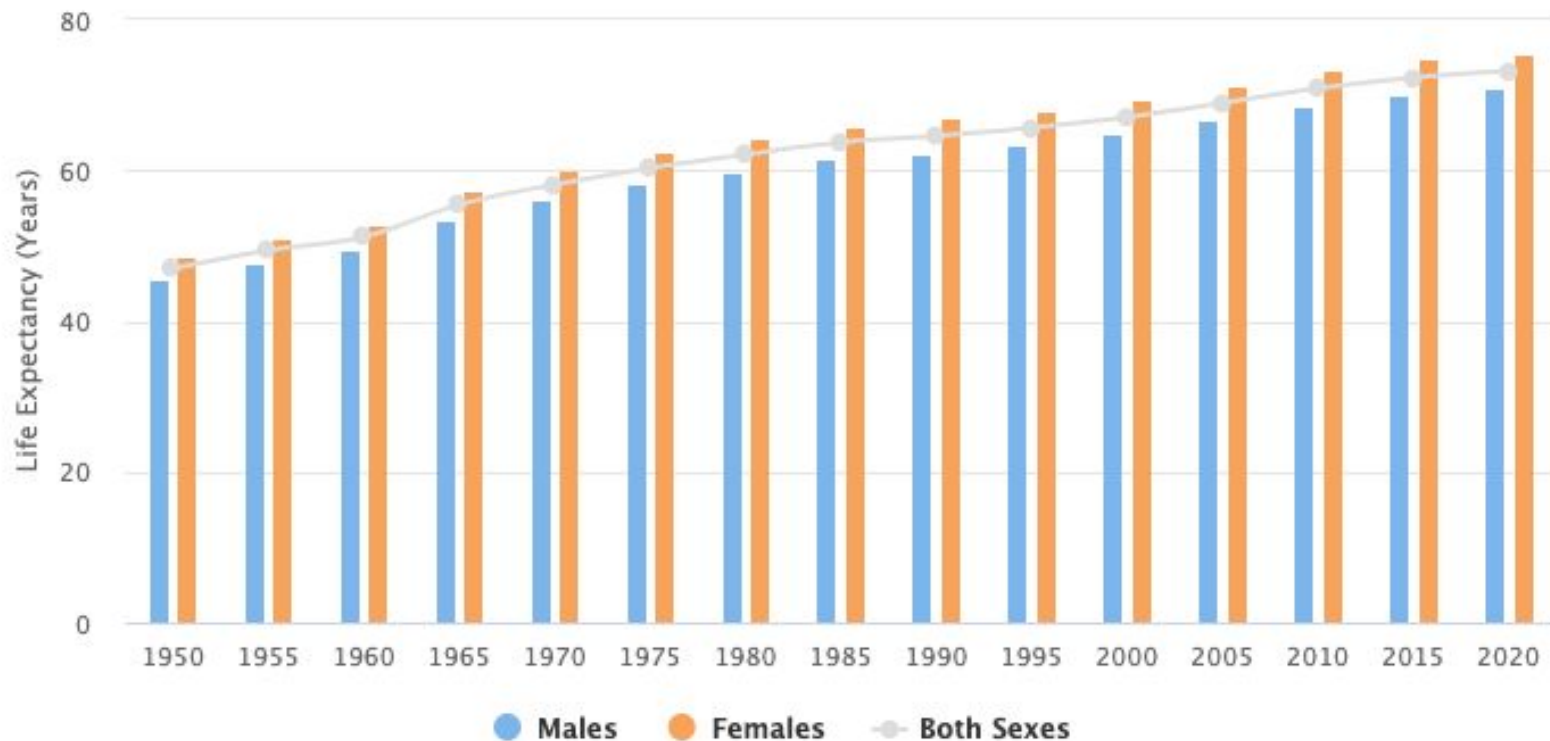
# Here's the result (unweighted average)



Except for COVID, life expectancy across the world has been going up for a long time

## Life Expectancy in the World from 1955 to Present

Males, Females, and Both Sexes combined



# How would we get the weighted average?

The unweighted average isn't quite right because we'd like to weight our average by a country's population, and this is weighting equally.

To fix this, we'd need to either use SQL or Tableau Prep to compute the fraction of the world's population in each country in each year and then join to World Indicators

# Q1: Which query gives us the fraction of the world's population in a country?

```
WITH T AS (  
  SELECT Year, SUM([Total Population]) As WorldPop FROM [World Indicators] GROUP BY Year  
)
```

```
(a) SELECT Country, [Total Population] *1.0 / T.WorldPop  
FROM [World Indicators]  
INNER JOIN T ON T.Year = [World Indicators].Year
```

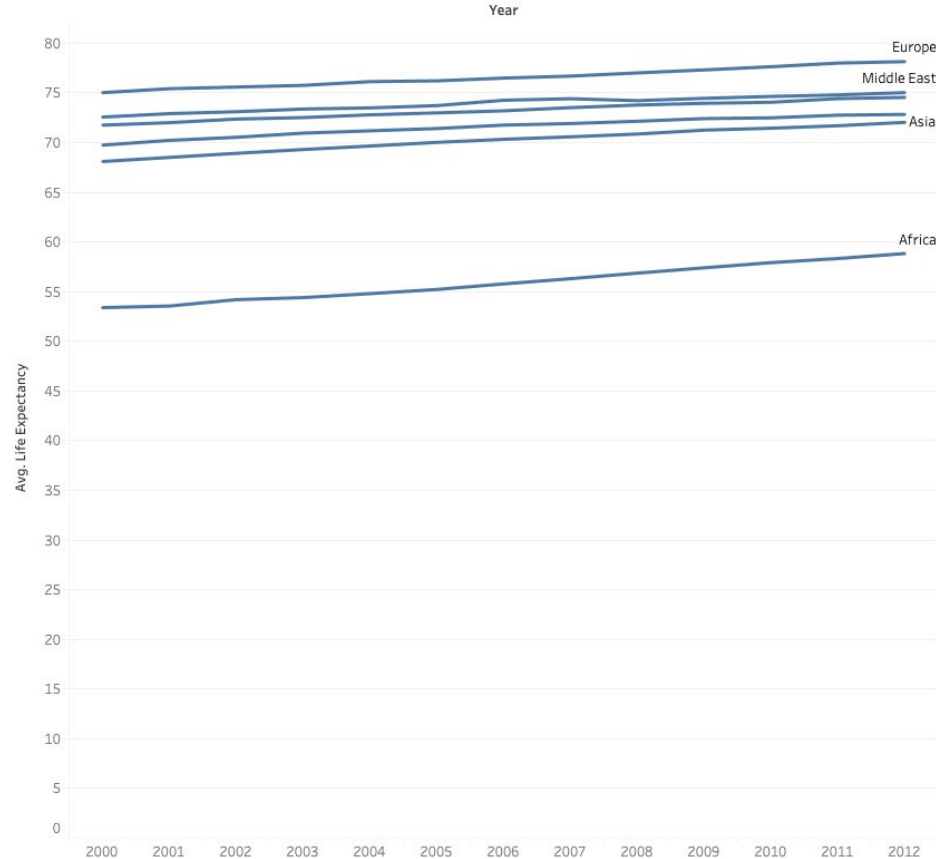
```
(b) SELECT T.Year, Country, [Total Population] *1.0 / T.WorldPop  
FROM [World Indicators]  
INNER JOIN T ON T.Year = [World Indicators].Year
```

```
(c) SELECT T.Year, Country, [Total Population] *1.0 / T.WorldPop  
FROM [World Indicators]  
INNER JOIN T ON T.Year = [World Indicators].Year AND T.Country = [World Indicators].Country
```

```
(d) SELECT Country, [Total Population] *1.0 / T.WorldPop  
FROM [World Indicators]  
INNER JOIN T ON T.Year = [World Indicators].Year AND T.Country = [World Indicators].Country
```

# Let's plot separately by region

Avg Life Expectancy over time



Drag “Region” to the  
“Label” square in the  
“Marks” panel

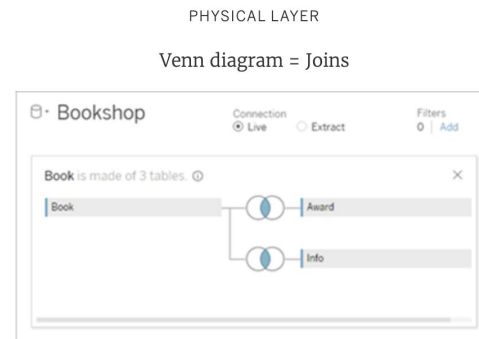
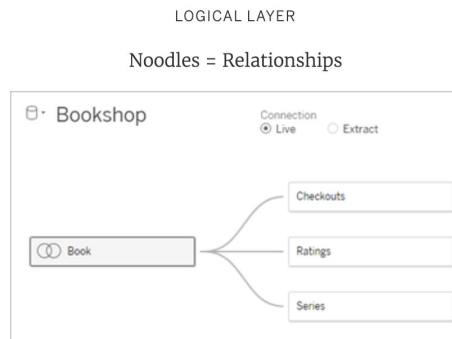


# Joins in Tableau

# Tableau data has 2 layers:

## logical & physical

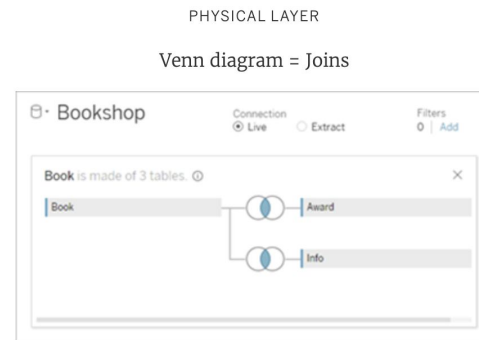
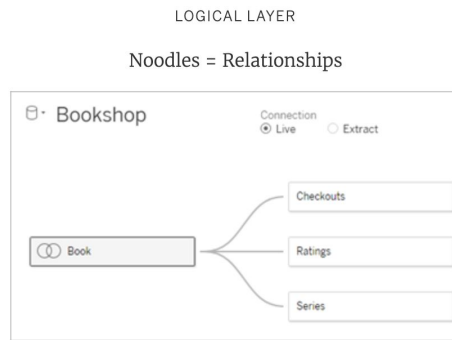
- Logical layer
  - Seen by default in Tableau
  - Logical tables can be combined via “relationships”
- Physical layer
  - Each logical table consists of one or more physical tables
  - Physical tables can be combined via “joins” within one logical table
  - Get to this layer by double-clicking on a logical table





# Combining data in Tableau

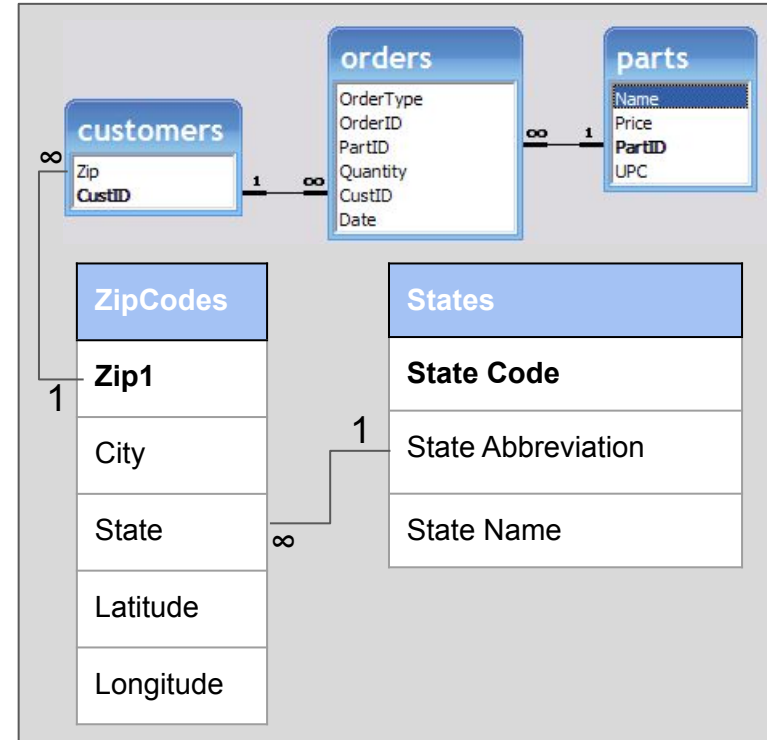
- Combining data via “relationships” at the logical layer
  - Tableau tries to guess what you want
  - Requires less user knowledge
  - Harder to get Tableau to do exactly what you want
  - Slower
- Combining data via “joins” at the physical layer
  - Closer to SQL
  - Requires you to know what an Inner Join / Left Join / Right Join / Outer Join are
  - You get more control
  - Faster



We'll show you how to do joins (physical layer) using an example from Recitation 4

# Example from Recitation 4

- Zip = 5 digit zipcode of dealer
- OrderType = “RR” (replenishment order)  
or “EO” (emergency order)
- Quantity = quantity ordered by dealer
- Date = date ordered by dealer
- Price = unit price of part
- Name = part description
- UPC = uniform product code  
(a useful classification of parts)



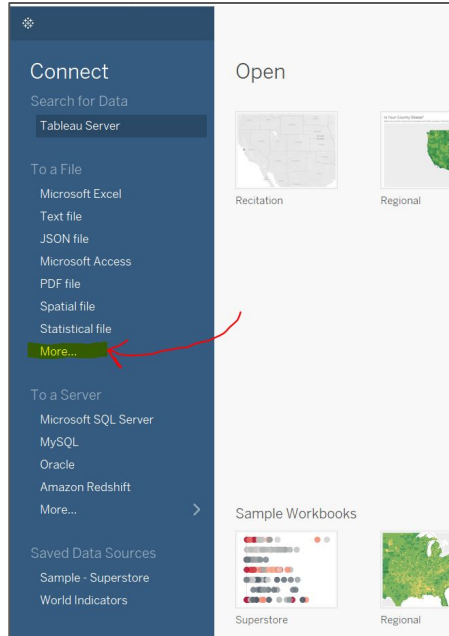
# Getting the data for this example

Download the Recitation 4 zip file from Canvas, put it in an easy to access directory, and unzip it.

You should now have 5 CSV files.

After downloading, open the program.

# Importing data

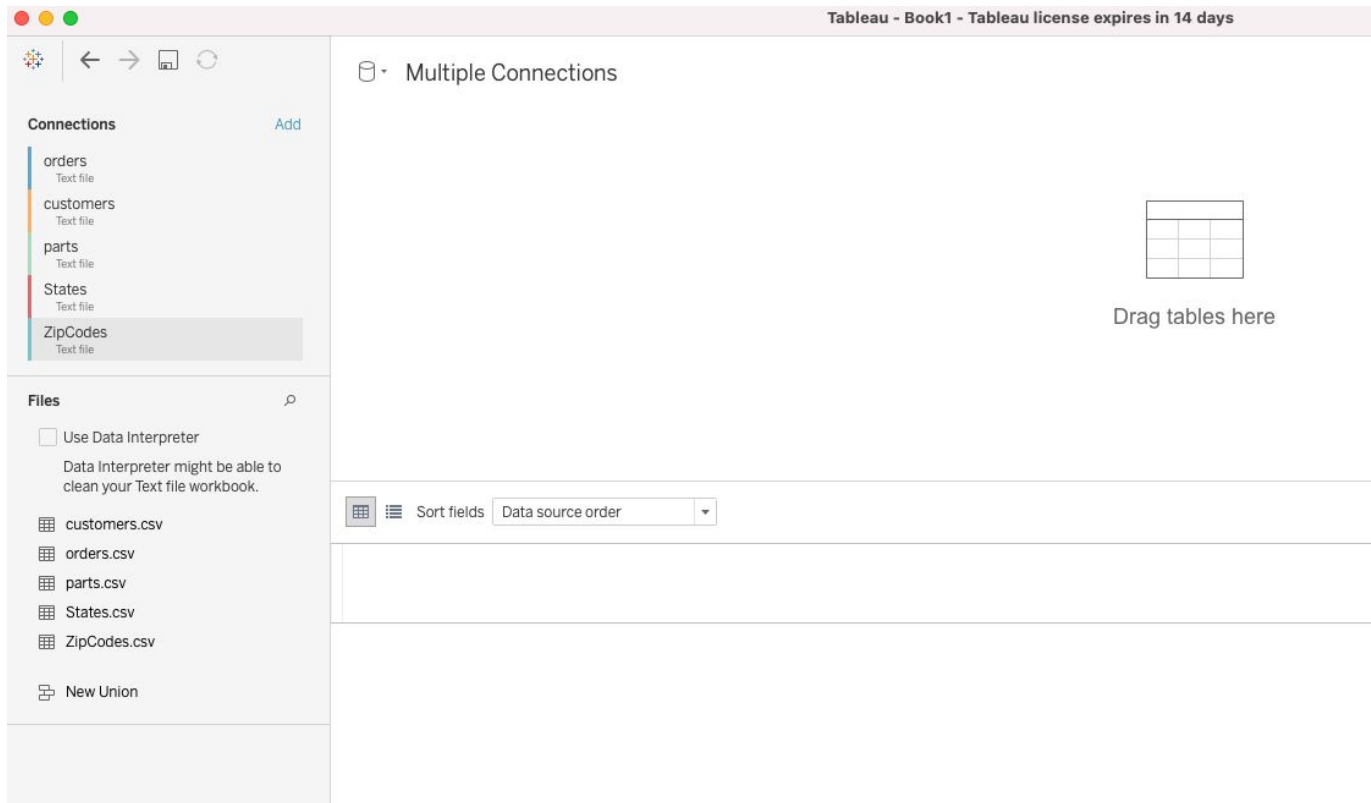


On the left side of Tableau's menu, choose the type of data you are trying to import.

For this lab, we are using csv files. Choose "more" to find that option.

Import all csv files for the lab.

# Here's what it should look like



# Double click on a table to get to the physical layer

The screenshot shows a data tool interface. On the left, a sidebar contains 'Connections' (customers, Text file) and 'Files' (customers.csv, orders.csv, parts.csv, States.csv, ZipCodes.csv, New Union, New Table Extension). A blue circle highlights the 'customers.csv' file in the 'Files' section. The main area shows the 'customers' table selected, with a dropdown menu displaying 'customers.csv'. A tooltip for 'customers.csv' states: 'Logical Table: customers.csv. Double-click this logical table to see its physical table.' Below this, a diagram shows two tables connected by a line, with the text 'Need more data? Drag tables here to relate them. [Learn more](#)'. At the bottom, a table view for 'customers.csv' is shown, displaying 2 fields and 1478 rows. The table has columns for 'Name' (customers.csv), 'Zip' (customers.csv), and 'Cust ID' (customers.csv). The data rows are: 30341, 41015; 30076, 41017; 37917, 41031; 37422, 41042.

customers

Connection: ☒ Live ☐ Extract Filters: 0 | [Add](#)

Logical Table: customers.csv  
Double-click this logical table to see its physical table.

Need more data?  
Drag tables here to relate them. [Learn more](#)

customers.csv 2 fields 1478 rows 100 rows

Name	Zip	Cust ID
customers.csv	customers.csv	customers.csv
30341	41015	
30076	41017	
37917	41031	
37422	41042	

Type	Field Name	Physical Table	Remote Field Name
	Zip	customers.csv	Zip

# How to join tables

The screenshot shows a data tool interface. On the left, a sidebar contains a 'Files' section with a list of CSV files: customers.csv, orders.csv, parts.csv, States.csv, and ZipCodes.csv. The main workspace displays a diagram where 'customers.csv' is joined to 'ZipCodes.csv'. A 'Join' dialog box is open, showing four join types: Inner (selected), Left, Right, and Full Outer. Below the join types, there is a table for 'Add new join clause' with columns for 'Data Source' and 'ZipCodes.csv'. The bottom panel shows the 'customers.csv' table structure with fields: Zip, Cust ID, Zip1, City, and State. The 'Zip' field is highlighted, and its details are shown in the right pane.

customers.csv is made of 2 tables. ①

customers.csv

ZipCodes.csv

Join

Inner Left Right Full Outer

Data Source ZipCodes.csv

Add new join clause

customers.csv

Name: customers.csv

Fields

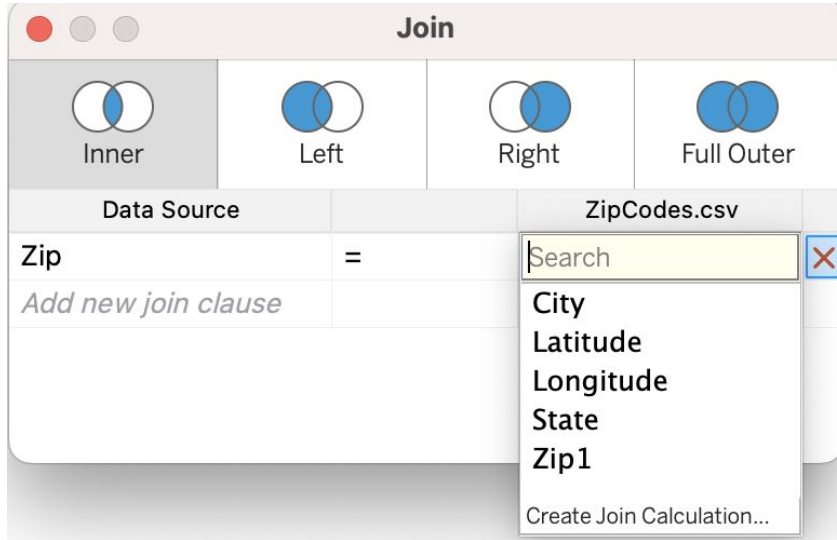
Type	Field Name	Physical Table	Remote Field Name
📍	Zip	customers.csv	Zip
#	Cust ID	customers.csv	CustID
📍	Zip1	ZipCodes.csv	zip
📍	City	ZipCodes.csv	city
📍	State	ZipCodes.csv	state

Data preview unavailable

- Drag ZipCodes from “Files” on the left to Customer table’s physical layer
- This dialog box will pop up
- You will click “Add new join clause” to specify the join clause
- You can also change the kind of join from Inner to Left / Right / Full Outer



# Specifying the Join Clause



- After you click “Add new join clause”, select two fields to use and a logical operator (e.g., =)
- You can use a calculated field by selecting “Create Join Calculation”
- You can also add additional join clauses — these are combined using AND

# Specifying the Join Clause

- Here we are doing  
SELECT \* FROM  
Customers  
INNER JOIN  
ZipCodes ON  
Customers.Zip =  
ZipCodes.Zip1

customers.csv is made of 2 tables. @

customers.csv

ZipCodes.csv

Join

Inner Left Right Full Outer

Data Source

ZipCodes.csv

Zip = Zip1

Add new join clause

customers.csv 7 fields 1448 rows

100 rows

Name

customers.csv

Fields

Type	Field Name	Physical Table	Remote Field Name
📄	Zip	customers.csv	Zip
#	Cust ID	customers.csv	CustID
📄	Zip1	ZipCodes.csv	zip
📄	City	ZipCodes.csv	city
📄	State	ZipCodes.csv	state

customers.csv	customers.csv	ZipCodes.csv	ZipCodes.csv	ZipCodes.csv	ZipCodes.csv	ZipCodes.csv
Zip	Cust ID	Zip1	City	State	Latitude	Longitude
00613	49986	00613	Arecibo	PR	18.4581	-66.7327
00613	49978	00613	Arecibo	PR	18.4581	-66.7327
00660	49994	00660	Hormigueros	PR	18.1391	-67.1209
00725	55021	00725	Caguas	PR	18.2339	-66.0450
00725	49966	00725	Caguas	PR	18.2339	-66.0450
00731	53837	00731	Ponce	PR	18.0773	-66.6119
00731	50035	00731	Ponce	PR	18.0773	-66.6119
00802	52491	00802	St Thomas	VI	18.3223	-64.9637

# Specifying the Join Clause

Below Tableau is showing the results from

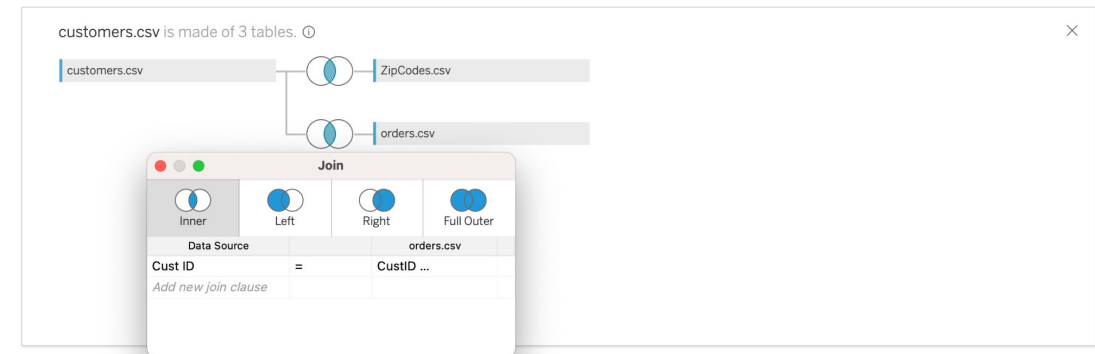
```
SELECT *  
FROM Customers  
INNER JOIN ZipCodes  
ON Customers.Zip =  
ZipCodes.Zip1
```

The screenshot shows the Tableau interface. At the top, a window titled 'customers.csv is made of 2 tables.' displays a join configuration between 'customers.csv' and 'ZipCodes.csv'. The 'Join' dialog box shows the 'Inner' join type selected, with the join condition 'Zip = Zip1'. Below this, the main Tableau view shows a data table with 7 fields and 1448 rows. The table has columns for 'Zip', 'Cust ID', 'Zip1', 'City', 'State', 'Latitude', and 'Longitude'. The 'Fields' pane on the left lists the fields and their physical and remote names.

Type	Field Name	Physical Table	Remote Field Name
📄	Zip	customers.csv	Zip
#	Cust ID	customers.csv	CustID
📄	Zip1	ZipCodes.csv	zip
📄	City	ZipCodes.csv	city
📄	State	ZipCodes.csv	state

customers.csv	customers.csv	ZipCodes.csv	ZipCodes.csv	ZipCodes.csv	ZipCodes.csv	ZipCodes.csv
Zip	Cust ID	Zip1	City	State	Latitude	Longitude
00613	49986	00613	Arecibo	PR	18.4581	-66.7327
00613	49978	00613	Arecibo	PR	18.4581	-66.7327
00660	49994	00660	Hormigueros	PR	18.1391	-67.1209
00725	55021	00725	Caguas	PR	18.2339	-66.0450
00725	49966	00725	Caguas	PR	18.2339	-66.0450
00731	53837	00731	Ponce	PR	18.0773	-66.6119
00731	50035	00731	Ponce	PR	18.0773	-66.6119
00802	52491	00802	St Thomas	VI	18.3223	-64.9637

# Joining Multiple Tables



customers.csv 13 fields 545370 rows 100 rows

Name	Type	Field Name	Physical Table	Remote Field Name
customers.csv	Zip	Zip	customers.csv	Zip
customers.csv	Cust ID	CustID	customers.csv	CustID
ZipCodes.csv	zip	zip	ZipCodes.csv	zip
ZipCodes.csv	city	city	ZipCodes.csv	city
ZipCodes.csv	state	state	ZipCodes.csv	state

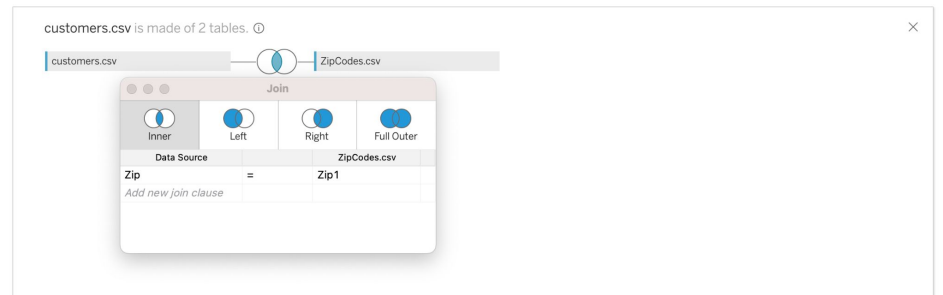
customers.csv	customers.csv	ZipCodes.csv	ZipCodes.csv	ZipCodes.csv	ZipCodes.csv	ZipCodes.csv	order
Zip	Cust ID	zip	city	state	latitude	longitude	Ord
37408	50859	37408	Chattanooga	TN	35.0309	-85.3100	RR
32082	51326	32082	Ponte Vedra Beach	FL	30.1899	-81.3821	RR
70119	49306	70119	New Orleans	LA	29.9745	-90.0875	RR
70119	49306	70119	New Orleans	LA	29.9745	-90.0875	RR
70119	49306	70119	New Orleans	LA	29.9745	-90.0875	RR
37013	49385	37013	Antioch	TN	36.0551	-86.6478	RR
39601	49604	39601	Brookhaven	MS	31.5743	-90.4503	RR
70433	49702	70433	Covington	LA	30.4708	-90.1049	RR

- Drag orders onto the customers physical table and set up the join condition
- Tableau may set up the join condition automatically — double check that it is right
- Tableau is showing the results from

```
SELECT * FROM Customers
INNER JOIN ZipCodes ON
Customers.Zip = ZipCodes.Zip1
INNER JOIN Orders ON
Customers.[Cust ID] =
Orders.CustID
```

# Multiple Join Clauses, Calculated Fields

- Here we are doing INNER JOIN ON Customers.Zip = ZipCodes.Zip1



customers.csv 7 fields 1448 rows 100 rows

Name	customers.csv
Zip	00613
Cust ID	49986
Zip1	00613
City	Arecibo
State	PR
Latitude	18.4581
Longitude	-66.7327

Type	Field Name	Physical Table	Remote Field Name
🔍	Zip	customers.csv	Zip
#	Cust ID	customers.csv	CustID
🔍	Zip1	ZipCodes.csv	zip
🔍	City	ZipCodes.csv	city
🔍	State	ZipCodes.csv	state

customers.csv	#	ZipCodes.csv	ZipCodes.csv	ZipCodes.csv	ZipCodes.csv	ZipCodes.csv
Zip	Cust ID	Zip1	City	State	Latitude	Longitude
00613	49986	00613	Arecibo	PR	18.4581	-66.7327
00613	49978	00613	Arecibo	PR	18.4581	-66.7327
00660	49994	00660	Hormigueros	PR	18.1391	-67.1209
00725	55021	00725	Caguas	PR	18.2339	-66.0450
00725	49966	00725	Caguas	PR	18.2339	-66.0450
00731	53837	00731	Ponce	PR	18.0773	-66.6119
00731	50035	00731	Ponce	PR	18.0773	-66.6119
00802	52491	00802	St Thomas	VI	18.3223	-64.9637

# Here are all 5 csv files, joined together

customers+

Connection  
☒ Live ☐ Extract

Filters  
0 | [Add](#)

customers.csv is made of 5 tables. ⓘ



This is equivalent to:

```
SELECT * FROM Customers  
INNER JOIN ZipCodes ON Customers.Zip = ZipCodes.Zip1  
INNER JOIN States ON ZipCodes.state = States.[State Abbreviation]  
INNER JOIN Orders ON Customers.[Cust ID] = Orders.CustID  
INNER JOIN Parts ON Orders.[Part ID] = Parts.PartID
```

customers.csv 20 fields 545370 rows

100 rows

Name  
customers.csv

Fields			
Type	Field Name	Physical Table	Remote Field Name
📍	Zip	customers.csv	Zip
#	Cust ID	customers.csv	CustID
📍	zip	ZipCodes.csv	zip
📍	city	ZipCodes.csv	city
📍	state	ZipCodes.csv	state

customers.csv Zip	customers.csv Cust ID	ZipCodes.csv zip	ZipCodes.csv city	ZipCodes.csv state	ZipCodes.csv latitude	ZipCodes.csv longitude	Abc Order
37204	50700	37204	Nashville	TN	36.1121	-86.7763	RR
35233	50712	35233	Birmingham	AL	33.5096	-86.7997	RR
30666	50739	30666	Statham	GA	33.9577	-83.5849	RR
33404	50765	33404	West Palm Beach	FL	26.7821	-80.0653	RR
37408	50859	37408	Chattanooga	TN	35.0309	-85.3100	RR
37408	50859	37408	Chattanooga	TN	35.0309	-85.3100	RR
37408	50859	37408	Chattanooga	TN	35.0309	-85.3100	RR
29406	50869	29406	Charleston	SC	32.9188	-80.0228	RR

# Relationships in Tableau

# This can also be done via relationships

- The concepts are very similar, but Tableau decides for you whether the underlying join should be an INNER / LEFT / RIGHT / FULL OUTER JOIN
- Pros:
  - Good for people who don't know about join types (INNER / LEFT / RIGHT / FULL OUTER)
- Cons:
  - Slower
  - Less Control



# How to create relationships between tables

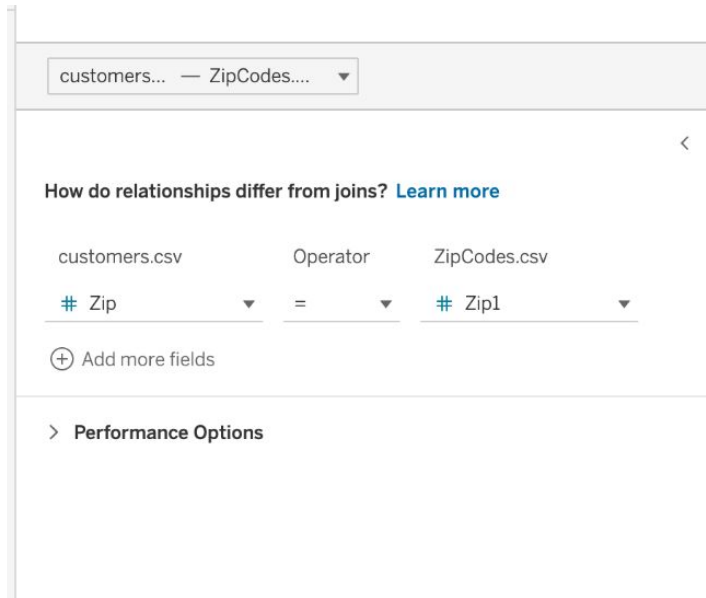
The screenshot shows the Tableau interface. On the left, the 'Connections' pane lists 'customers' as a Text file. Below it, the 'Files' pane lists several CSV files: customers.csv, orders.csv, parts.csv, States.csv, and ZipCodes.csv. In the center, a diagram shows 'customers.csv' connected to 'ZipCodes.csv' by an orange line. On the right, the 'Connection' type is set to 'Live'.

- Drag customers & ZipCodes from “Files” on the left to the top right
- You should see an **orange** line between the two tables (the relationship) & a panel open below

The screenshot shows the relationship configuration panel in Tableau. At the top, a dropdown menu shows 'customers... — ZipCodes...'. Below this, a section titled 'How do relationships differ from joins? [Learn more](#)' contains a visual representation of the relationship: 'customers.csv' followed by 'Operator' and 'ZipCodes.csv'. Below this, a dropdown menu shows '# Zip' followed by an equals sign and another dropdown menu showing '# Zip1'. Below this, there is a link to 'Add more fields'. At the bottom, there is a section titled '> Performance Options'.

ZipCodes.csv Zip1	ZipCodes.csv City	ZipCodes.csv State	ZipCodes.csv Latitude	ZipCodes.csv Longitude
00210	Portsmouth	NH	43.0059	-71.01320
00211	Portsmouth	NH	43.0059	-71.01320
00212	Portsmouth	NH	43.0059	-71.01320
00213	Portsmouth	NH	43.0059	-71.01320
00214	Portsmouth	NH	43.0059	-71.01320
00215	Portsmouth	NH	43.0059	-71.01320
00501	Holtsville	NY	40.9223	-72.63708
00544	Holtsville	NY	40.9223	-72.63708

# Changing the relationship condition



The screenshot shows a configuration panel for a relationship between two tables. At the top, a dropdown menu shows 'customers...' and 'ZipCodes...' with a downward arrow. Below this, a link reads 'How do relationships differ from joins? [Learn more](#)'. The main configuration area has three columns: 'customers.csv', 'Operator', and 'ZipCodes.csv'. Under 'customers.csv', there is a field '# Zip' with a dropdown arrow. Under 'Operator', there is an equals sign '=' with a dropdown arrow. Under 'ZipCodes.csv', there is a field '# Zip1' with a dropdown arrow. Below these fields is a button with a plus icon and the text 'Add more fields'. At the bottom, there is a section header '> Performance Options'.

Edit this panel to change the relationship, e.g., by:

- changing the operator used to do the comparison (currently =)
- changing the fields used (currently customers.Zip and ZipCodes.Zip1)
- clicking “add more fields” to add additional conditions

# For faster performance, indicate foreign key relationships

Diagram illustrating a relationship between two data sources:

- customers.csv** (Left Table)
- ZipCodes.csv** (Right Table)

The relationship is defined by the following fields:

- customers.csv**: Zip
- ZipCodes.csv**: Zip1

The relationship is a one-to-many relationship, indicated by the infinity symbol ( $\infty$ ) on the customers side and the 1 on the ZipCodes side.

**Edit Relationship**

How do relationships differ from joins? [Learn more](#)

customers.csv      ZipCodes.csv

# Zip      =      # Zip1

(+) Add more fields

^ Performance Options

These settings help Tableau optimize queries during analysis. The default settings are recommended, if you aren't sure what to choose. [Learn more](#)

Cardinality

Many      One

Referential Integrity

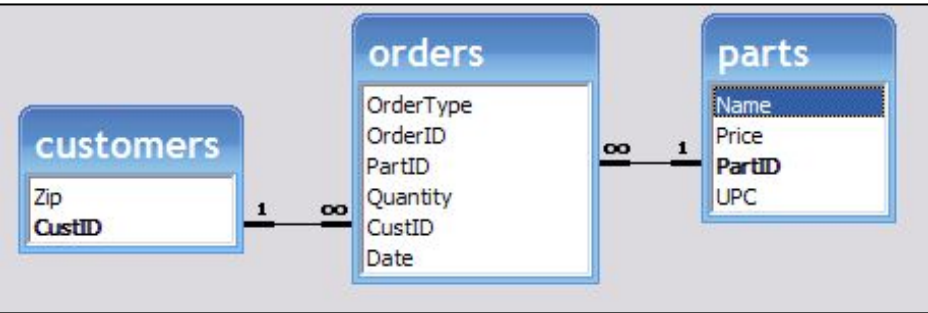
All records match      Some records match

Revert to Default

**ZipCodes**

- Zip
- CustID
- Zip1
- City
- State
- Latitude
- Longitude

# Q2: Which is correct?



A Orders.PartID

Cardinality

Many

Referential Integrity

Some records match

B Orders.PartID

Cardinality

One

Referential Integrity

Some records match

C Orders.PartID

Cardinality

One

Referential Integrity

All records match

D Orders.PartID

Cardinality

Many

Referential Integrity

All records match

Parts.PartID

One

All records match

Parts.PartID

Many

All records match

Parts.PartID

Many

Some records match

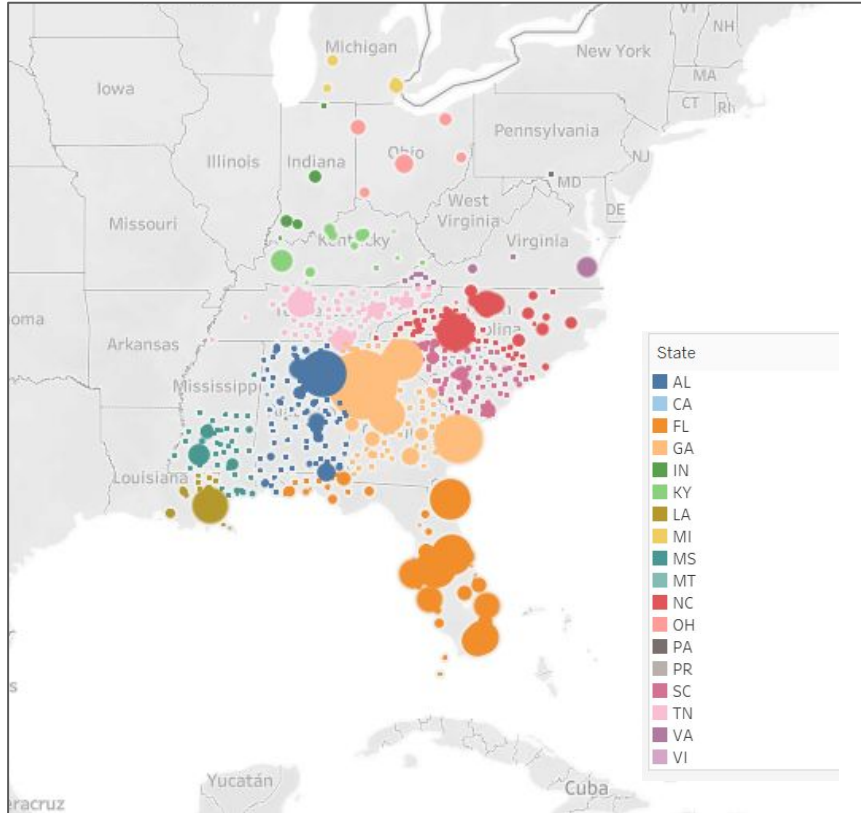
Parts.PartID

One

Some records match

# Next Steps

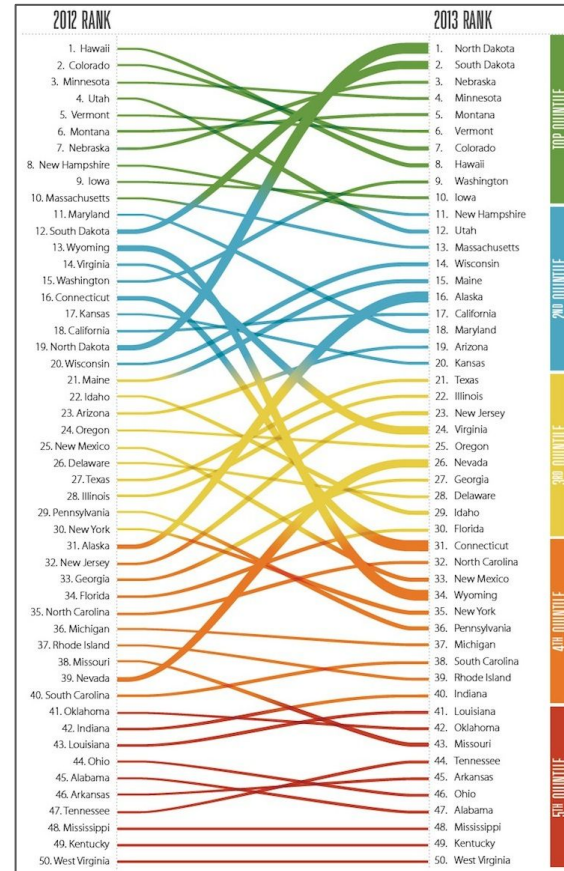
# In Recitation 4, you'll learn to explore data using maps



- Each dot is a car dealership
- Its size indicates its sales volume for spare parts
  - Its color indicates its state

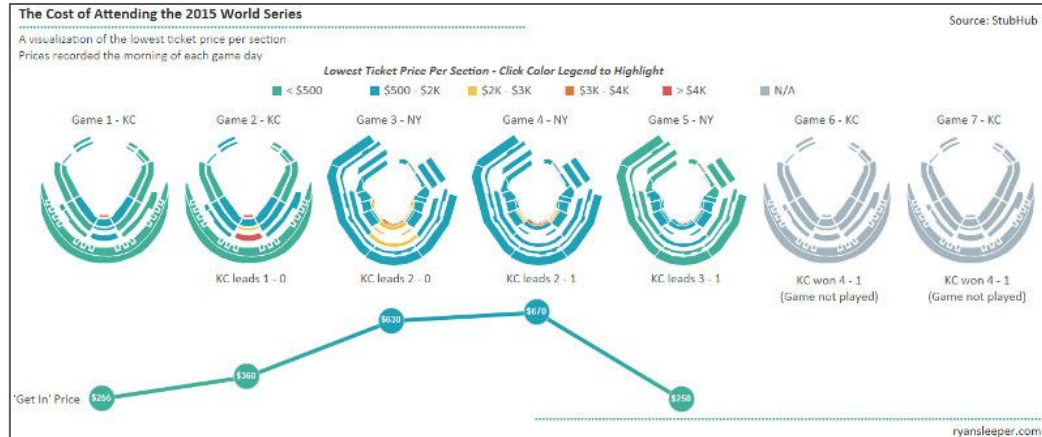
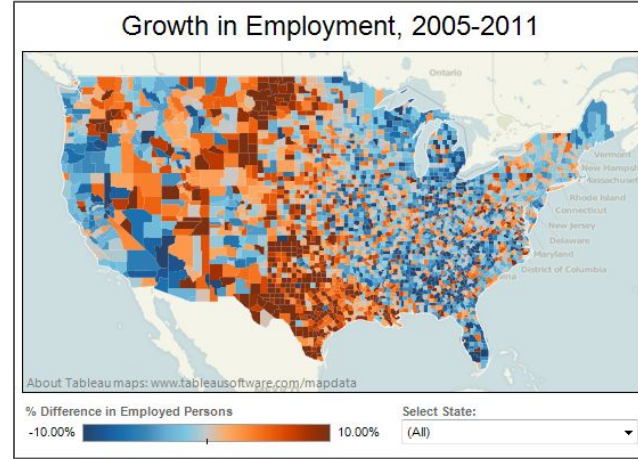
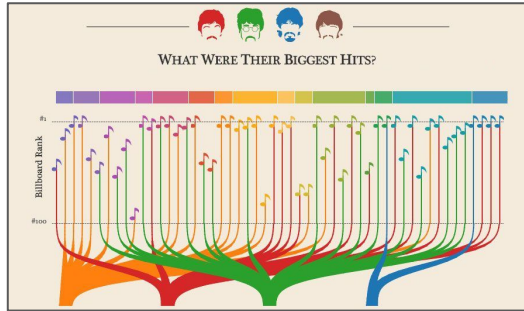
This was a brief introduction to Tableau.

You can do incredible things with tableau like create a map of a states “happiness” ranking across years...



# And...

## ...visualizations such as:





# Or entire dashboards such as:



## Monitoring Public Health

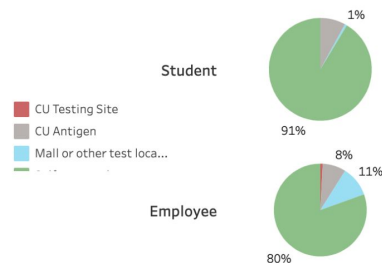
Cornell University MPH Dashboard

### Recent COVID activity

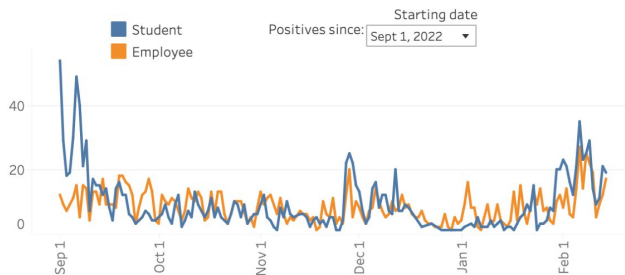
#### Confirmed Positives

		Positives listed by <b>test result</b> (reporting) date:							
		Tue Feb 7	Feb 8	Feb 9	Feb 10	Feb 11	Feb 12	Feb 13	Tue Feb 14
<b>University Total</b>		37	50	51	33	14	20	33	<b>36</b>
<b>Student</b>	Subtotal	23	25	29	14	9	11	21	<b>19</b>
	Undergrad	18	20	22	12	6	10	16	<b>11</b>
	Grad/prof	5	5	7	2	3	1	5	<b>8</b>
<b>Employee</b>	Subtotal	14	25	22	19	5	9	12	<b>17</b>
	Faculty	1	2	3	4				<b>2</b>
	Staff	13	23	19	16	5	9	10	<b>15</b>

#### Positive test provenance: prior eight days



### COVID over time



#### All Positives, by term

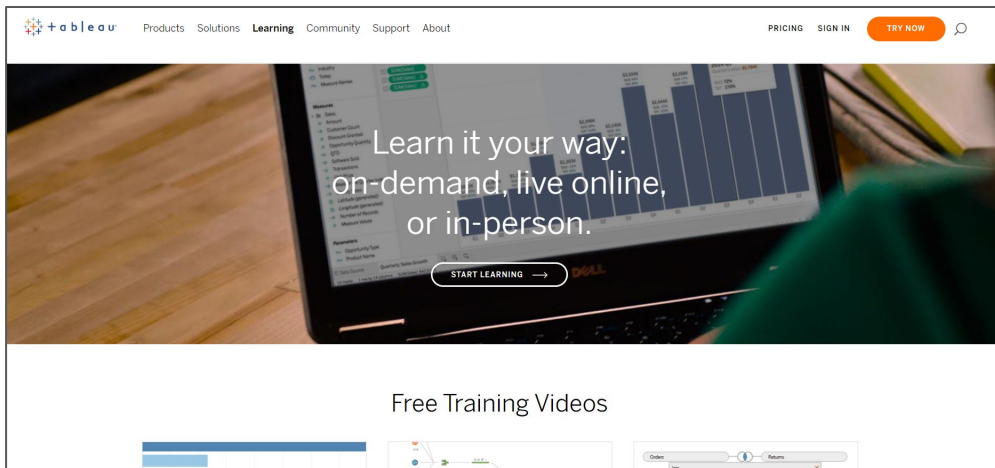
(where Spring is Jan 1 - July 30 and Fall is July 1 - Dec 31)

	Before Jan 1 '21	Spring '21	Fall '21	Spring '22	Fall '22	Spring '23	Ever
Student	260	570	3,020	4,633	1,950	428	10,861
Employee	157	162	553	2,129	1,644	415	5,060
Total	417	732	3,573	6,762	3,594	843	15,921

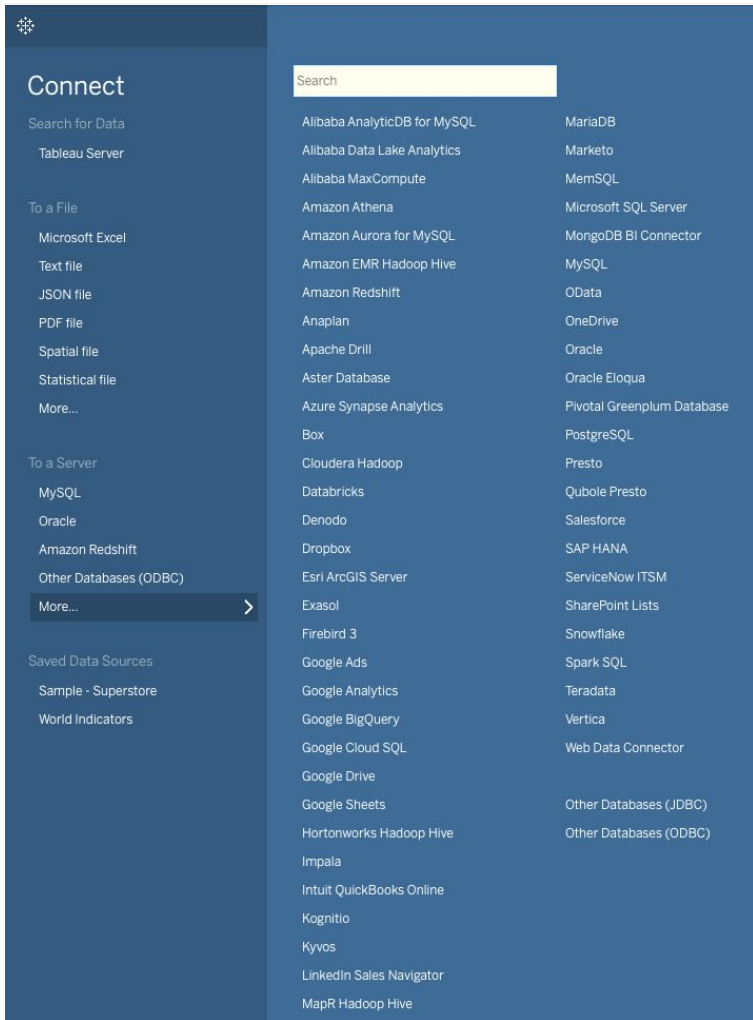
# Resources

There are many online resources and forums for Tableau communities - a great starting place is:

<https://www.tableau.com/learn>



# Appendix: Connecting Tableau to SQL Databases

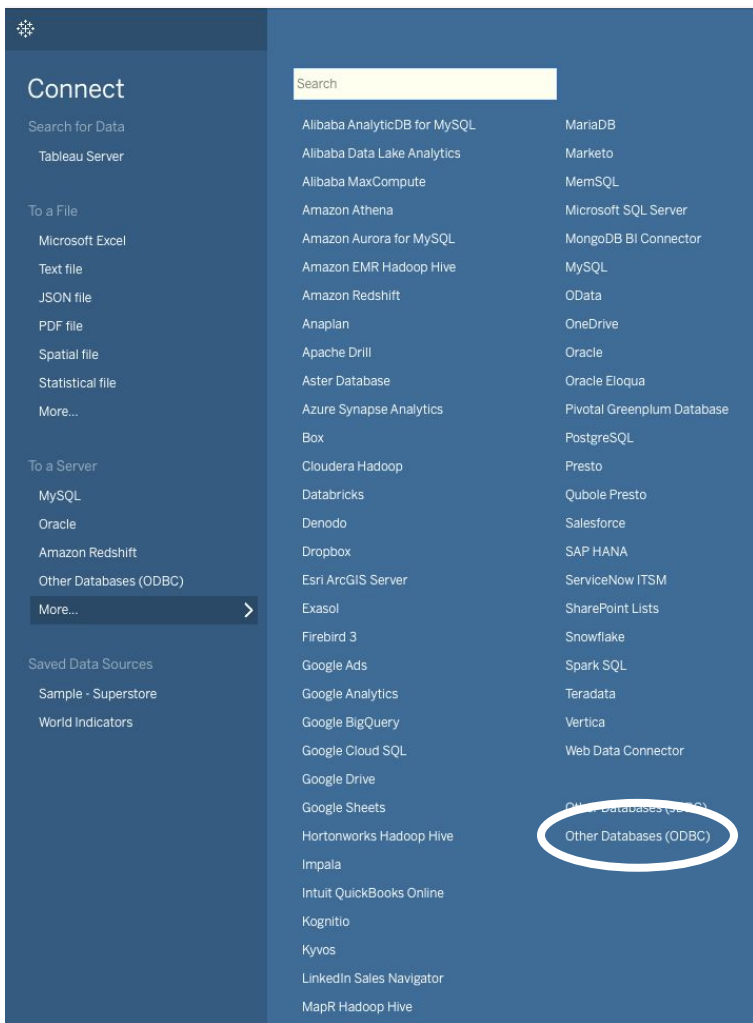


# Connecting to Tableau...

Is really easy for industrial-scale SQL databases you're likely to use in an internship or job

- Microsoft SQL Server
- Oracle
- PostgreSQL
- Teradata
- MySQL

Is also really easy for many other datasources



# Connecting to Tableau...

For SQLite, we need to do a few extra steps

But we'll learn something along the way

# ODBC is a generic way to connect to a database

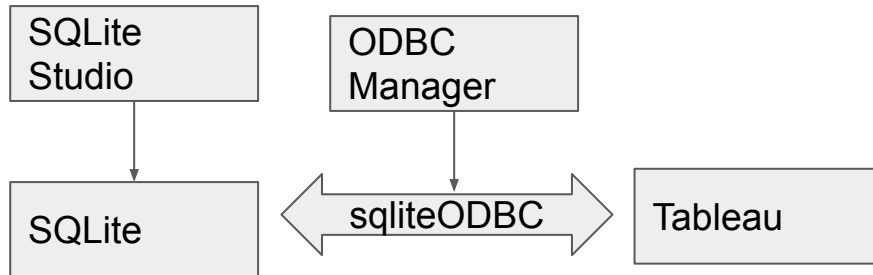
- ODBC stands for “open database connectivity”
- ODBC is an “API (application programming interface)”. It is a low-level set of functions in databases that can be called by other software.
- To use ODBC you may need some or all of the following:
  - A database (in our case SQLite)
  - An ODBC Driver for your database type (in our case, SQLite)
  - An ODBC Manager that helps you store and manage database “connections”
  - A piece of client software that can use the ODBC driver

# Today: Connecting Tableau to SQLite using ODBC on a Mac

Next week: Connecting python to SQLite using ODBC

We need:

- Database: SQLite
- ODBC Driver: sqliteODBC
- ODBC Manager: ODBC Manager from unixODBC
- Software to connect to database: Tableau



# Step 1: install ODBC driver & manager

Install homebrew (<https://brew.sh/>)

In Terminal: `brew install sqlitedbc`

(also installs unixodbc)

## Caveat:

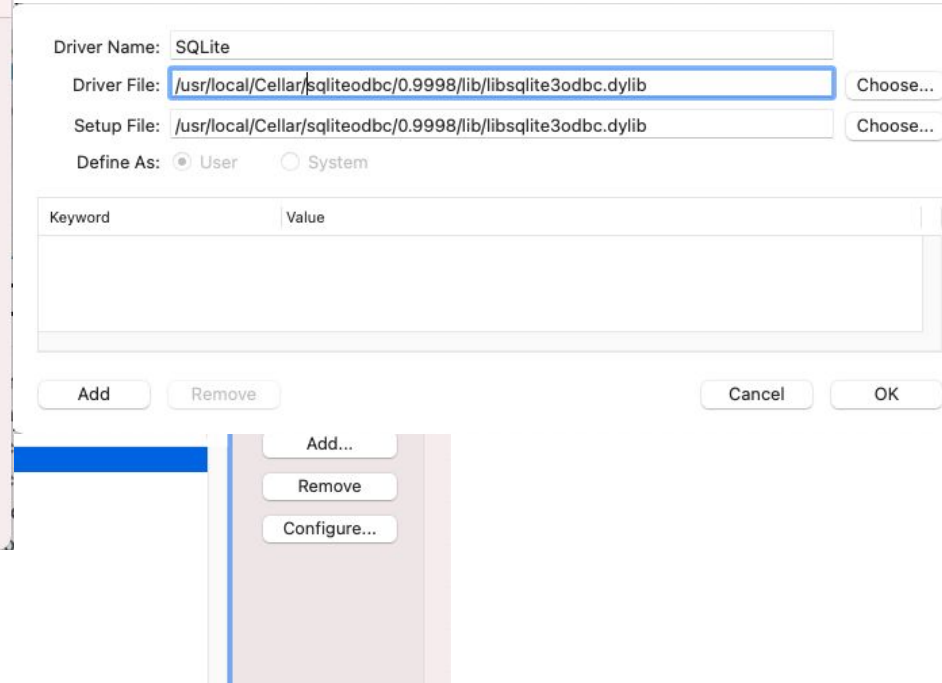
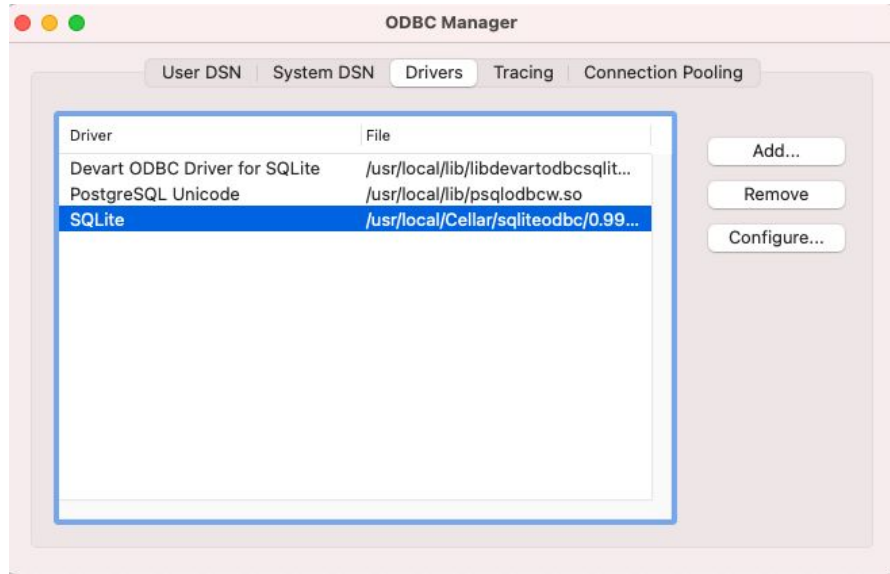
- If these steps don't work for you, it's ok
- Connecting SQLite to Tableau is **not required** for the homework or recitation
- I'm showing this to you to explain how everything is connected



## Step 2: Run ODBC Manager

Make sure that your SQLite driver is listed

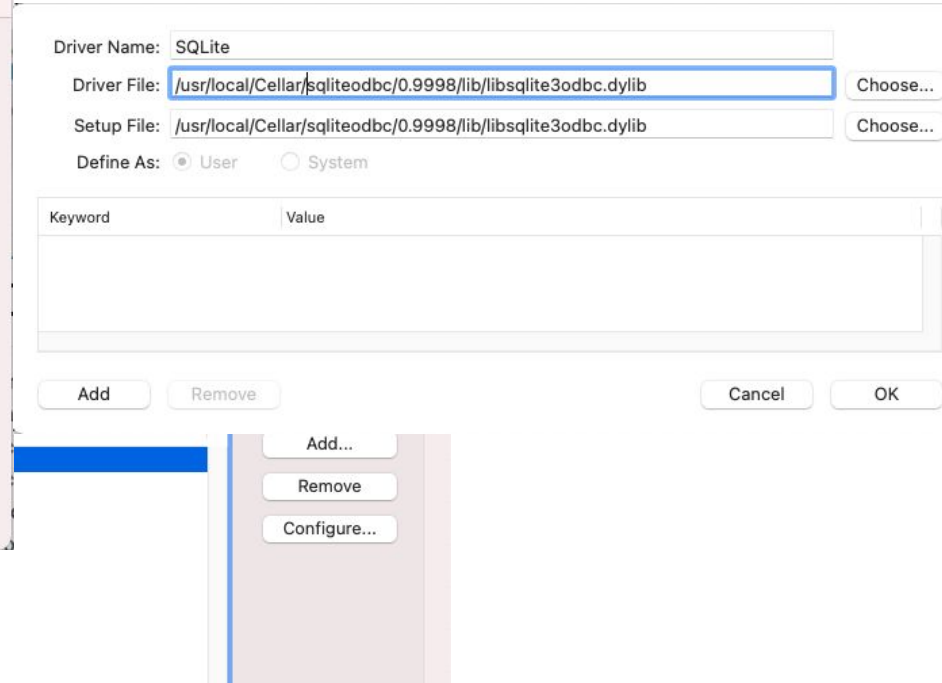
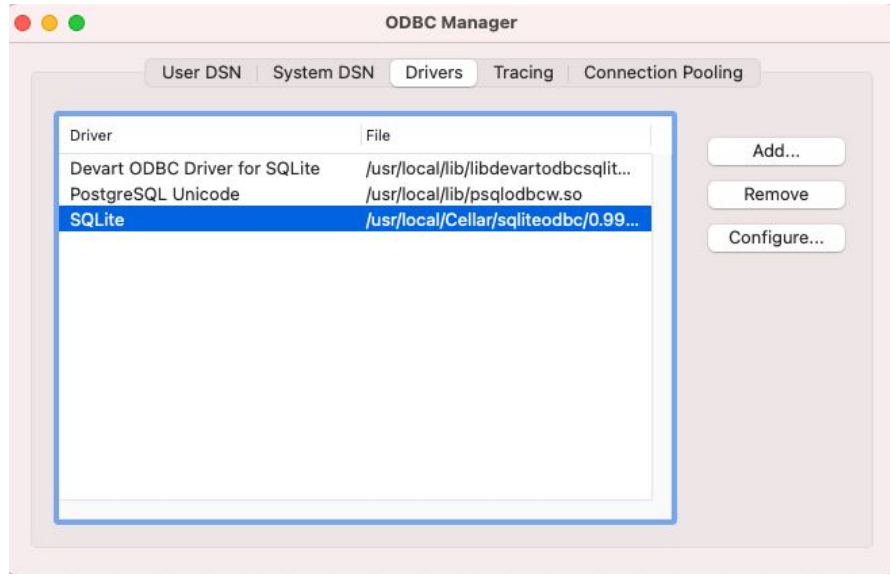
- DSN (data source name) is a string that describes a database



## Step 2: Run ODBC Manager

Make sure that your SQLite driver is listed

- DSN (data source name) is a string that describes a database



# If you want to connect ODBC to a SQLite database on a Mac

1. Download & install commercial ODBC SQLite driver with 30-day free trial from <https://www.devart.com/odbc/sqlite/>
2. Download & install free iODBC driver manager from <http://www.iodbc.org/dataspace/doc/iodbc/wiki/iodbcWiki/Downloads>
3. Run iODBC driver manager and create a User DSN referring to the sqlite file you want to access. “User DSN” means that this is a DSN created by and available to one specific user (you).



Search or jump to...



Pull requests

Issues

Marketplace

Explore



softace / **sqliteodbc**

Watch ▾

<> Code

! Issues 6

🔗 Pull requests

▶ Actions

📁 Projects

📖 Wiki

🛡 Security



🔗 master ▾

🔗 4 branches

🏷 0 tags

Go to file

Add file ▾

↓ Code ▾



chw and jarl-dk Imported sqliteodbc-0.9998.tar.gz



debian

Imported sqliteodbc-0.9998.tar.gz



missing

Imported sqliteodbc-0.60.tar.gz



osx

Imported sqliteodbc-0.9998.tar.gz



tccex

Imported sqliteodbc-0.78.tar.gz



ChangeLog

Imported sqliteodbc-0.9998.tar.gz



Makefile.in

Imported sqliteodbc-0.9993.tar.gz

2 years ago



Makefile.mingw-cross

Imported sqliteodbc-0.9993.tar.gz

2 years ago



Makefile.mingw64-cross

Imported sqliteodbc-0.9993.tar.gz

2 years ago



Clone



HTTPS SSH GitHub CLI

git@github.com:softace/sqliteodbc.g



Use a password-protected SSH key.

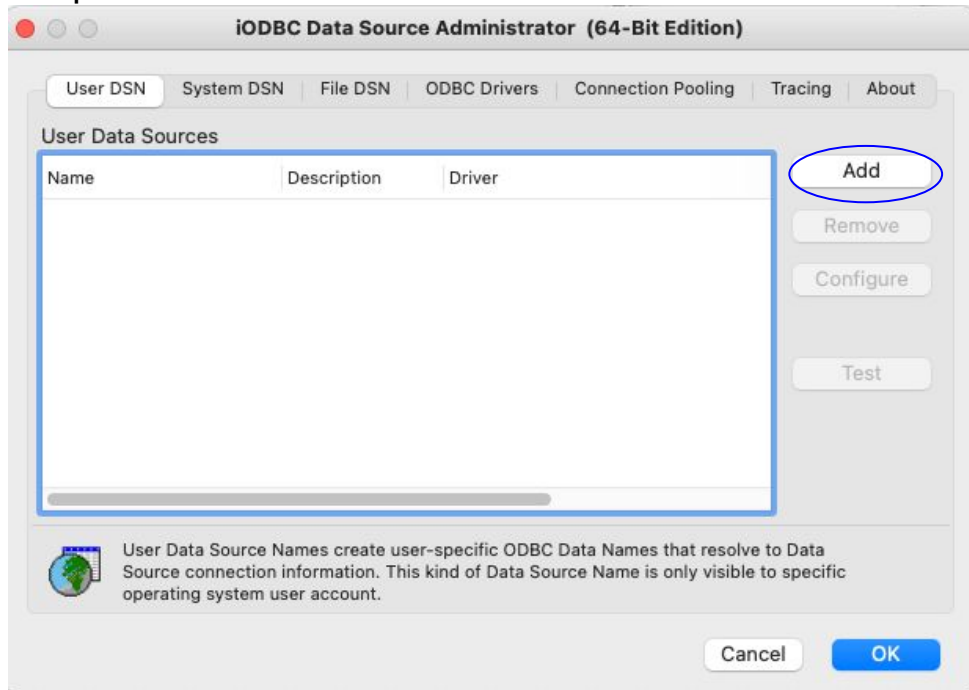


Open with GitHub Desktop



Download ZIP

### Step 3a:



### Step 3b:

