Designing Drug Overdose Surveillance Dashboards: **Workbook**

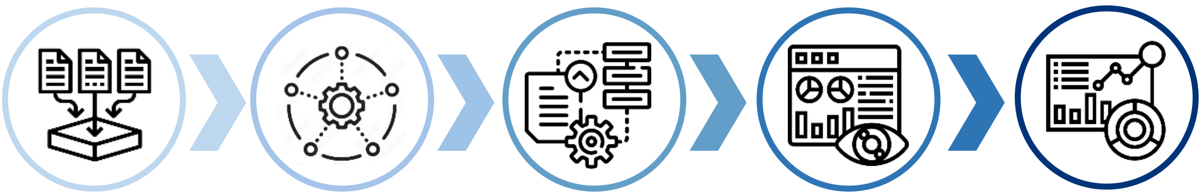
This project was funded through the Centers for Disease Control and Prevention's National Center for Injury Prevention and Control, Cooperative Agreement Number NU38OT000297-02-00.

Content was developed in collaboration with CSTE members, CDC Subject Matter Expert contributors and Data Scientists (Dr. Somya D Mohanty and Dr. Prashanti Manda). Views expressed do not necessarily represent the views of the Centers for Disease Control and Prevention.

About this document

This workbook will guide its readers to through a process towards developing a dashboard using Tableau. (Please refer "Dashboards: Topics in Design, Evaluation, and Maintenance for effective insights of drug overdose surveillance " for additional insights in data preparation and drug overdose dashboard development)

Within the workbook, we are going to use data from multiple sources, combine them, and develop interactive visualizations from the data. Readers will also learn approaches needed to understand and evaluate their data including common data issues. Next readers will be guided through the process of starting with multiple datasets, merging them on specific attributes, and utilizing the developed dataset to create a single interactive dashboard. We will evaluate several approaches to develop visualizations with different data attributes/variables which will help inform readers on appropriate data dashboard development techniques.



The workbook explores development of visualizations using opioid mortality data, county indicators, and prescription dispensing dataset to describe different patterns that exist in these datasets. The workbook uses county level indicators to show how to create visualizations with Tableau. Using the same workflow readers can use their jurisdiction specific datasets, i.e. Emergency Department, hospital discharge, syndromic surveillance datasets, to develop their own dashboards.

A complementary guide for non-Tableau users is also included which is an additional resource (Please see the course website for supplemental materials). The guide is based in Python for more programming-oriented users and explores similar visualization and dashboard development.

TABLE OF CONTENTS

UNDERSTANDING YOUR DATA

CONNECT TO THE DATA SOURCE (csv, xls, database, etc.)

OPEN TABLEAU: On the left panel you should be able to see options to connect to a dataset as shown below:

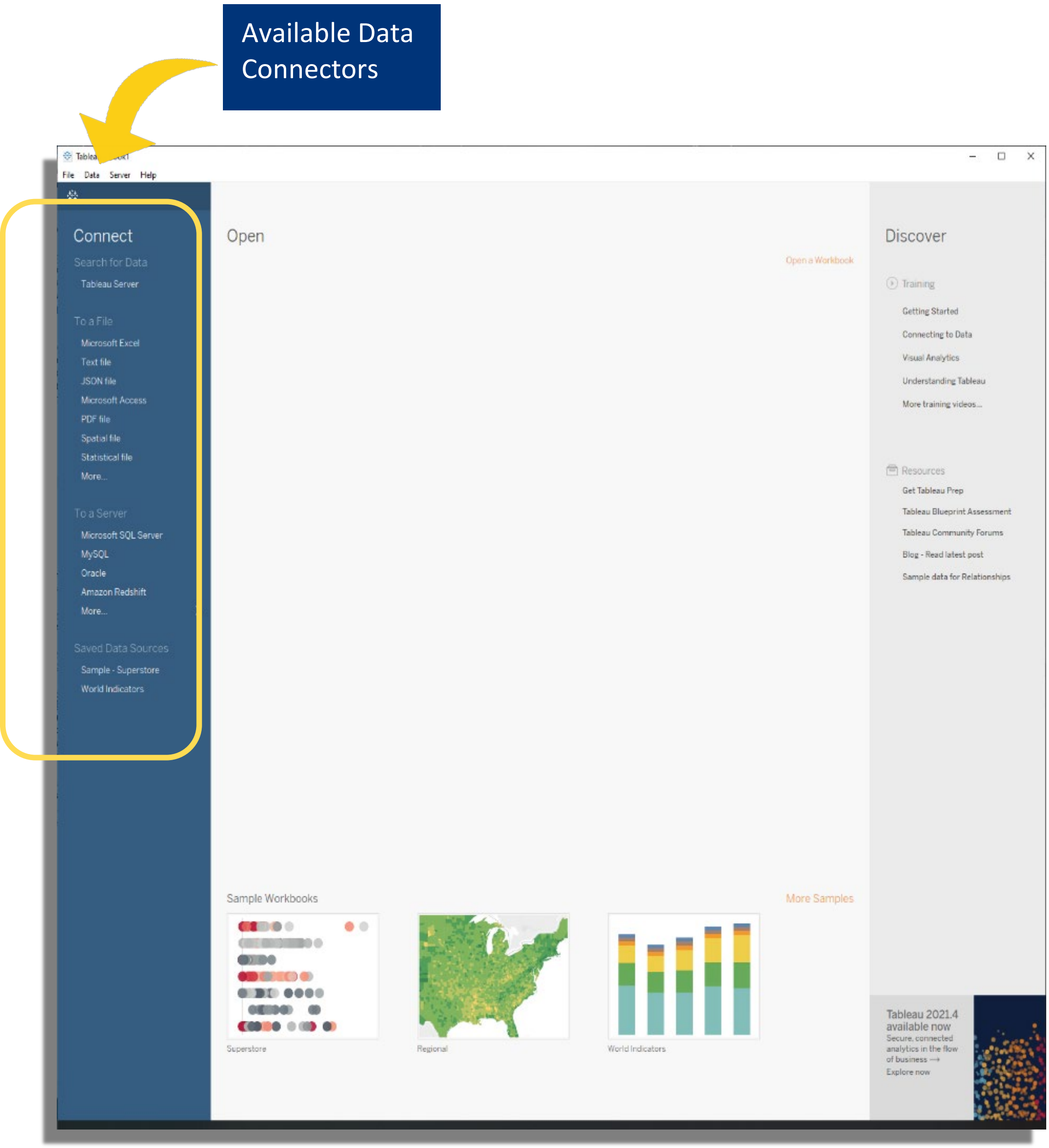


Figure 1: Tableau Desktop Data Connectors

VIEW YOUR DATA – Simple visualizations/graphs

Here we are going to use three datasets and explore them

1

DATASET 1: DRUG OVERDOSE DATASET

- The overdose death/cause dataset was obtained from CDC Wonder (https://wonder.cdc.gov/ucd-icd10.html). The data is from the Underlying Cause of Death database, which contains mortality and population counts for all U.S. counties over multiple years. The counts are based on death certificates for U.S. residents, where death certificates identify the underlying cause of death for a person and their demographic data.
  - From this data we obtained the Drug/Alcohol Induced causes data for 2009 - 2019 across all counties in US. - CDC Wonder Database – Drug/Alcohol Causes

2

DATASET 2: COUNTY HEALTH RANKINGS AND ROADMAPS

- The County Health Rankings contains multiple indicators of population for each geographic county region in U.S. The project is developed by University of Wisconsin Population Health Institute with a goal to analyse indicators and explore ways to improve health of underlying population. Some of the indicators (from over 500) in the data include, obesity, smoking, unemployment, access to healthy foods, the quality of air and water, and income inequality.
  - County Health Rankings
  - From this data we obtained the measures data for 2021 across all counties in US - County Health Ranking Documentation
  - County Health Ranking - Data Dictionary

3

DATASET 3: COUNTY OPIOID DISPENSING RATES

- The third dataset is the Opioid Dispensing Rate dataset. The data is summarized by CDC from the IQVIA Xponent database. The dataset has data for opioid prescriptions dispensed per 100 persons for years of 2006–2019 per county in U.S.
  - We utilize County Opioid Dispensing Rates for 2019
  - The first step is to **import the data** and view them. Here we use Tableau desktop to connect to the data and view it.

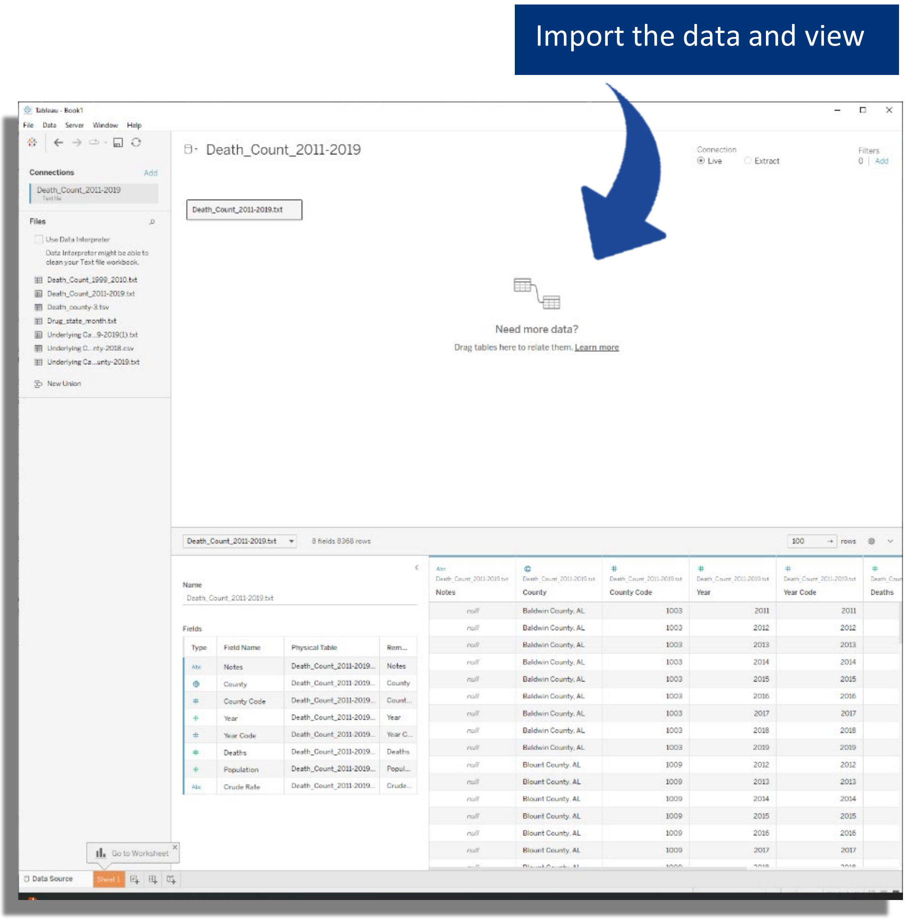


Figure 2: Data view of 2019 Opioid Mortality Data

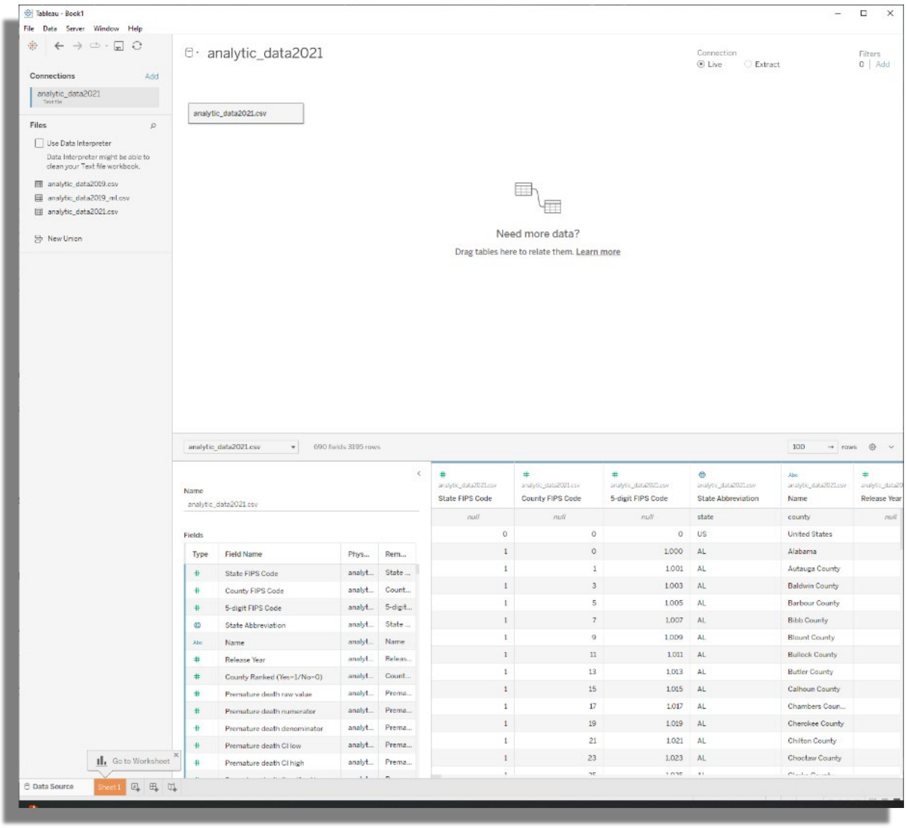


Figure 3: Data view of 2021 County Health Indicators

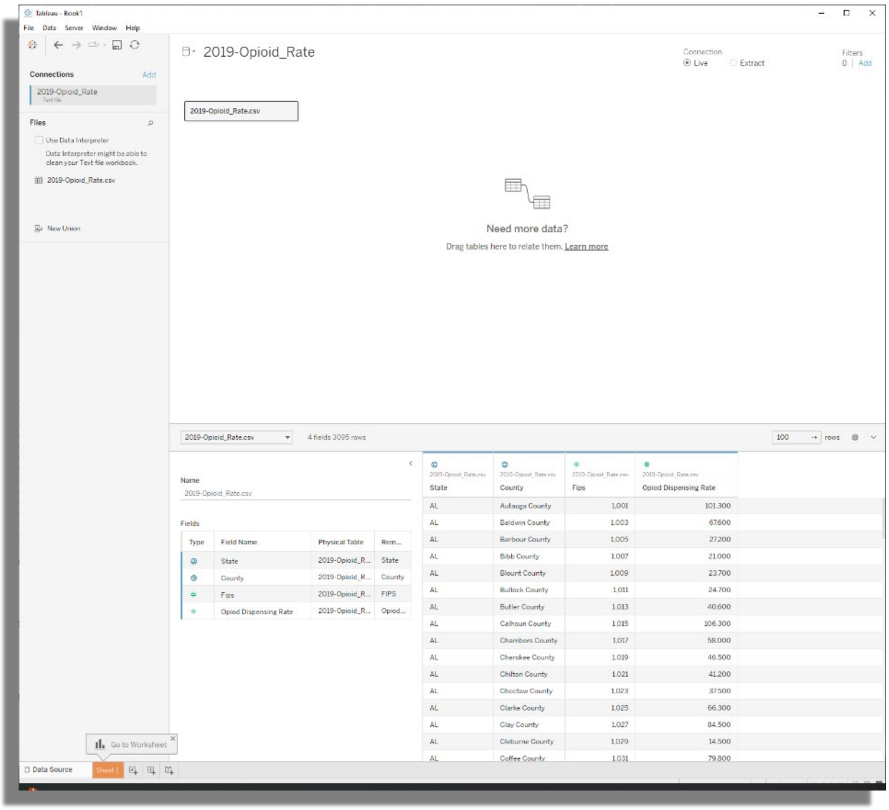


Figure 4: 2019 Opioid Dispensing data

DATA ATTRIBUTES AND DEVELOPING YOUR DATA DICTIONARY

The imported data has multiple variables and each variable as an associated datatype assigned by Tableau.

For example, within the opioid dispensing dataset, we observe there are 4 distinct variables in the data –

- **STATE**
- **COUNTY**
- **FIPS**
- **OPIOD DISPENSING RATE**

Each of those variables can be associated with the following datatypes.

Icon	Data type
Abc	Text (string) values
	Date values
	Date & Time values
#	Numerical values
T|F	Boolean values (relational only)
	Geographic values (used with maps)
	Cluster Group

We refer the reader to visit:

- [Tableau Datafields and Roles](#) for more information about Tableau datatypes.

While these datatypes were automatically assigned by Tableau during the reading of the source files, we can alter the datatype to suit the need of future analysis by clicking on the datatype icon (as shown in the figure)

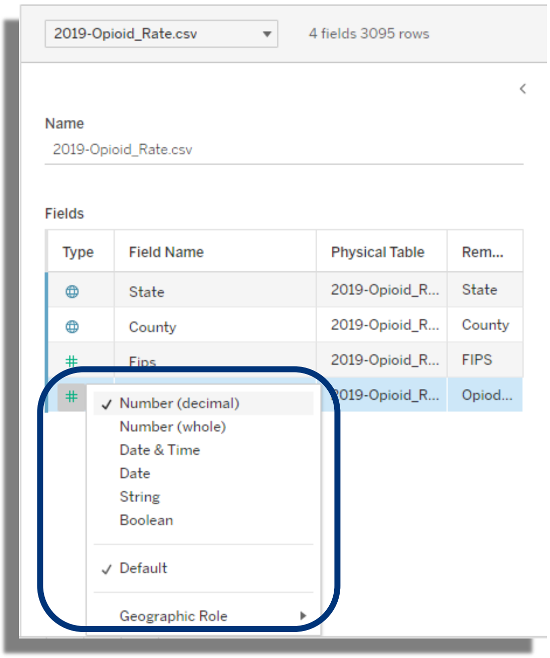


Figure 5: Changing Datatype of a variable in Tableau

We can also extract additional metadata information of the data variables using the **Describe feature** (via the dropdown on the variable names). This can be used to create a starting point for a data dictionary.

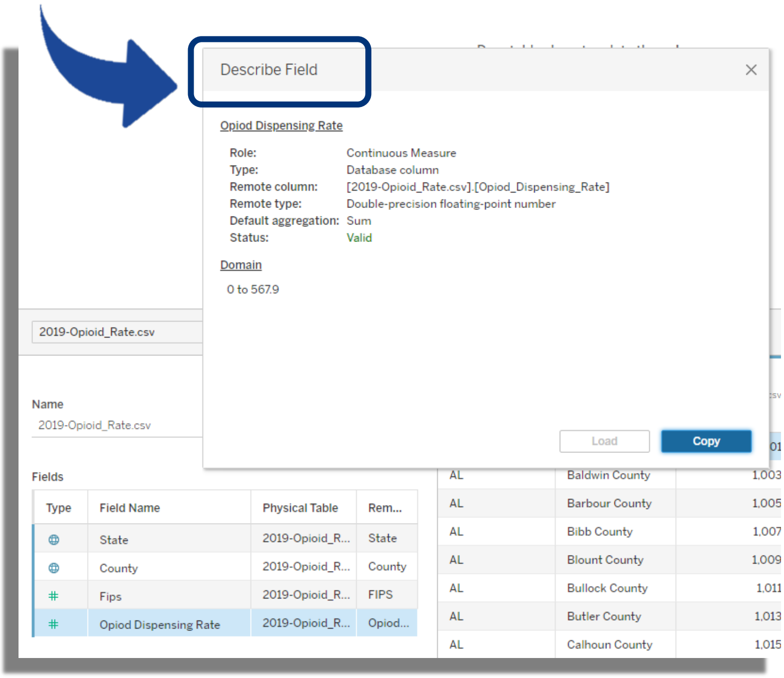
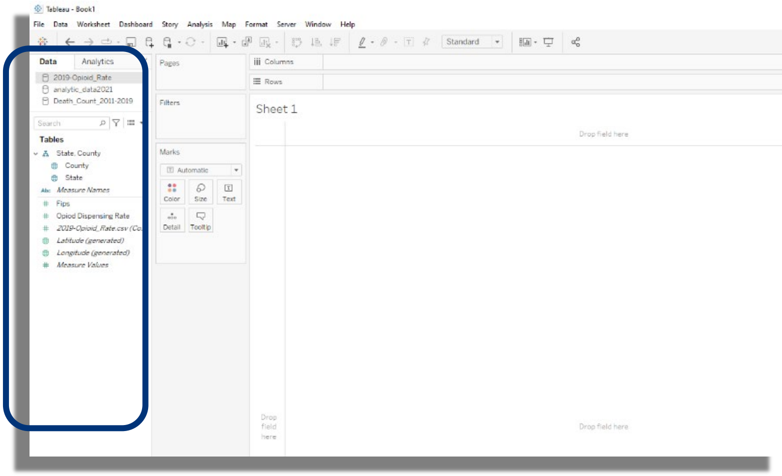


Figure 6: Metadata for a variable in Tableau

VIEW THE DATA ON A WORKSHEET

Now that we have the data imported, we can view the data on a Worksheet. Tableau uses a workbook and sheet file structure, much like Microsoft Excel. A workbook contains sheets. A sheet can be a worksheet, a dashboard, or a story. A worksheet contains a single view along with shelves, cards, legends, and the Data and Analytics panes in its side bar. Below we show the worksheet for the Opioid Dispensing dataset.

Here we can observe that the sheet has a left pane which contains the variable arranged by what it thinks could be indexes and measures. In indexes we see hierarchy of variables where, state and county are grouped together. Measures, include the Opioid dispensing rate and Geographic variables (Latitude and Longitude) generated from the State and County variables.



- Our first step can be to explore the data, for which we can use the **Table view** of a worksheet.

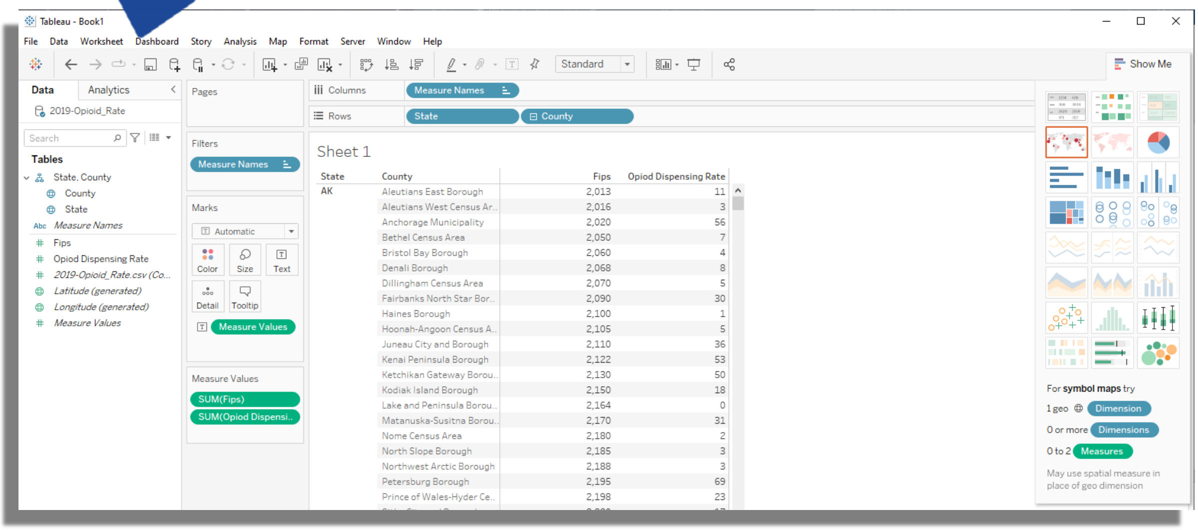


Figure 7: Table view of the Opioid dispensing rate.

- Here we use the geography (County and State) as our row indexes and the columns to be the Measure Names of FIPS and Opioid Dispensing Rate. \

- We will discuss how the Worksheet components work in the later sections, but for now we already observe issues with the imported data.

- The FIPS code is not in the standard 5-digit form. The data was read in as numbers from the csv file, but even if we convert it to string format (during import), the csv file itself has an issue where it had saved the FIPS variable as a number.

To fix such issues, explore data summaries, and perform data wrangling we will use **Tableau Prep Builder**, an add-on data pre-processing software.

DATA ISSUES

CHECKING DATA ISSUES

We read in the dataset via the **Tableau Prep Builder** software and are able to preview the data as such.

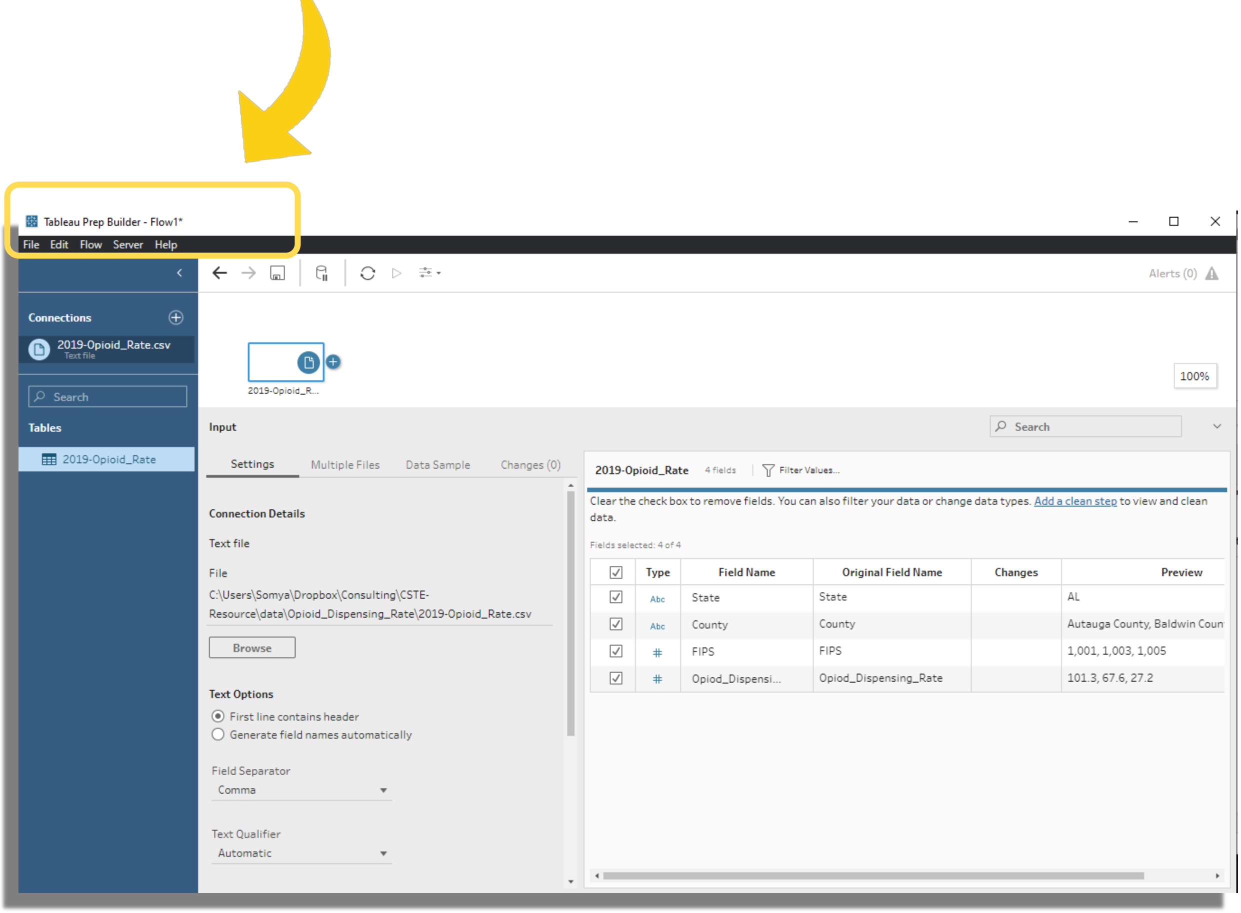


Figure 8: Tableau Prep Builder import of Opioid Dispensing Rate file

Here we observe the FIPS variable is still imported as a number, which can be changed to a string by updating the Type column. Following this we can add a **Clean step** (with the plus button), which shows more details about the dataset.

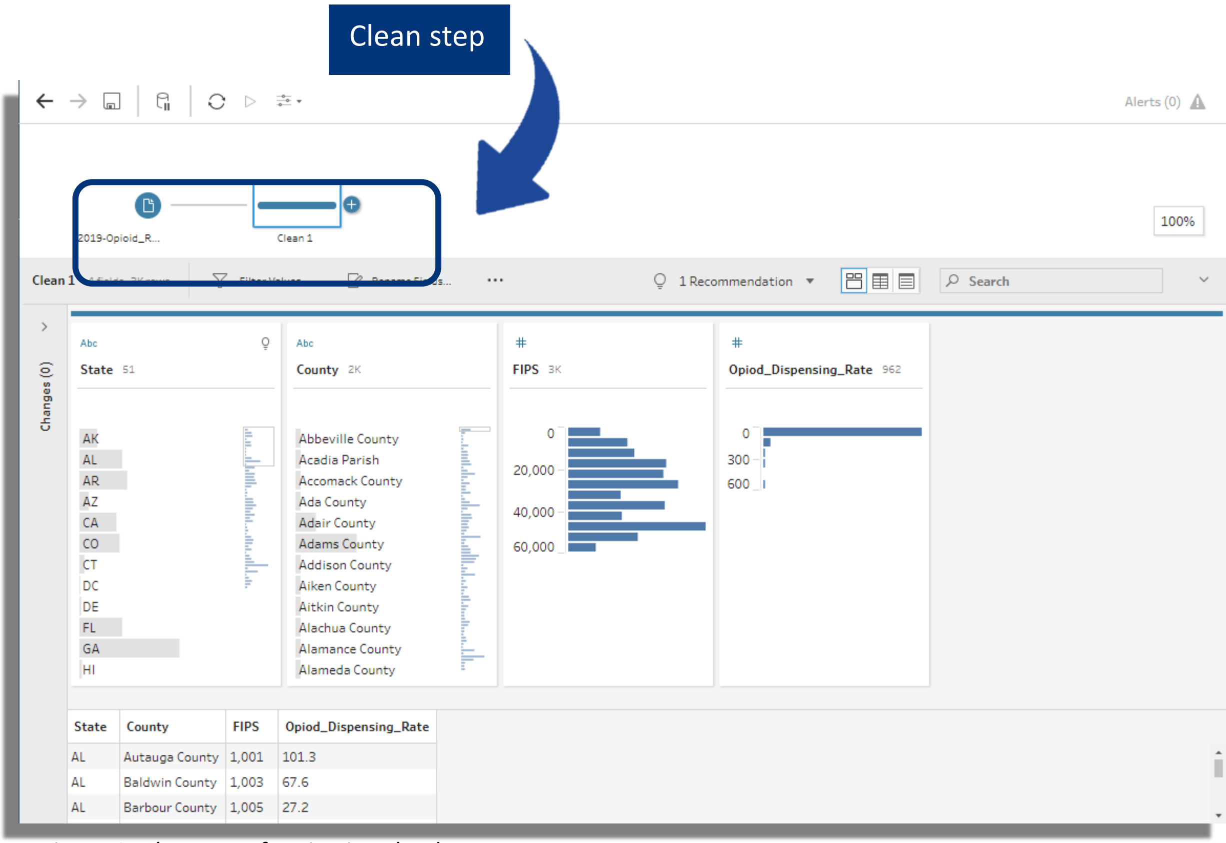
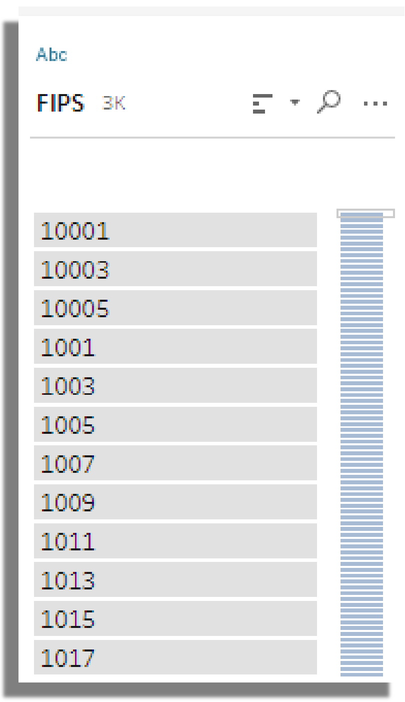


Figure 9: Clean step for viewing the data.

INCORRECT REPRESENTATION

The clean step presents a view of the raw data with its potential issues. Here we observe the issue with FIPS codes, where if it is read in as a number cannot uniquely differentiate each county. To merge multiple datasets (i.e., with the County Health Indicators dataset and the opioid mortality dataset) using the FIPS codes, we must ensure they are in the same format in each dataset. The first step is to convert the data into string, but even with that we observe that the FIPS is not converted into their proper 5-digit format, as shown in the figure:



The issue here is the missing 0 in front of the 4 digit FIPS codes. This can be handled in the clean step by creating a Calculated Field.

Figure 10: String conversion of FIPS code

We are going to create a new variable and call it **FIPS_STR**, and use the following function to perform the cleaning operation:

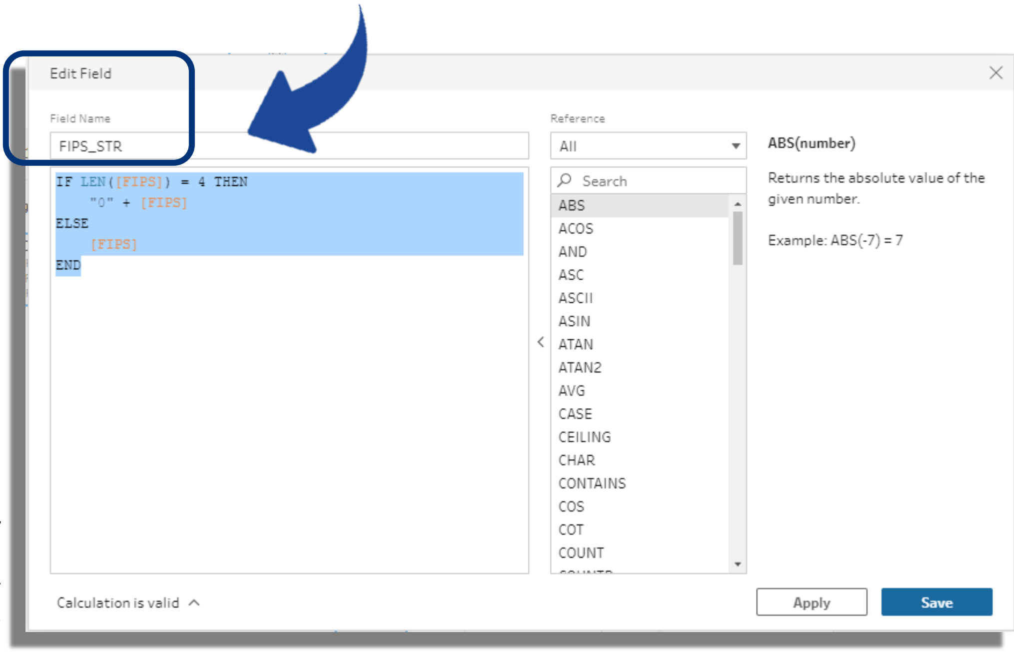


Figure 11: Create a new variable FIPS_STR by appending 0 to the beginning of FIPS codes

This operates row wise and creates the values of FIPS_STR by appending '0' if the length of FIPS string is 4, else if the string length is 5, it keeps it as it is. See here for additional examples on how to create a Calculated Field from your data:

https://help.tableau.com/current/pro/desktop/en-us/calculations_calculatedfields.htm

COMPLEX CONFIGURATION

Next, we import Dataset 2: County Health Rankings and Roadmaps. Here we observe data issues during import as shown below. The data here has two headers and the second header is being treated as a row of observation.

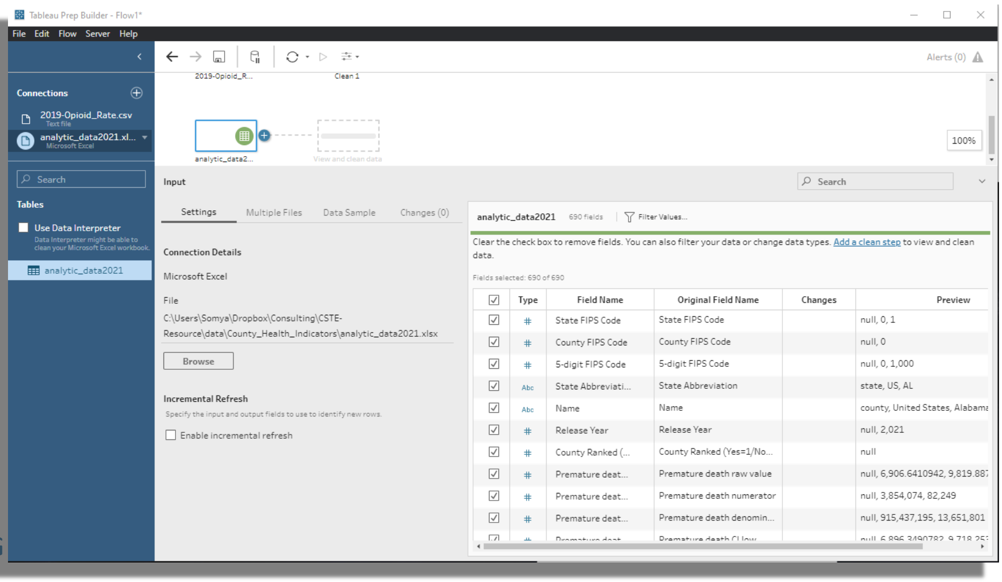


Figure 12: County Health Rankings data import

We can use the Clean with Data Interpreter to handle the configuration of import and get the data read in the right format. See here for capabilities of the Data Interpreter: Tableau Data Interpreter.

ANOMALIES:

After importing the data and creating the Calculated field of FIPS_STR (as we did in i), we can observe the variable summaries. Here we notice another issue with the FIPS_STR. We still have some observations with 0 as the FIPS code.

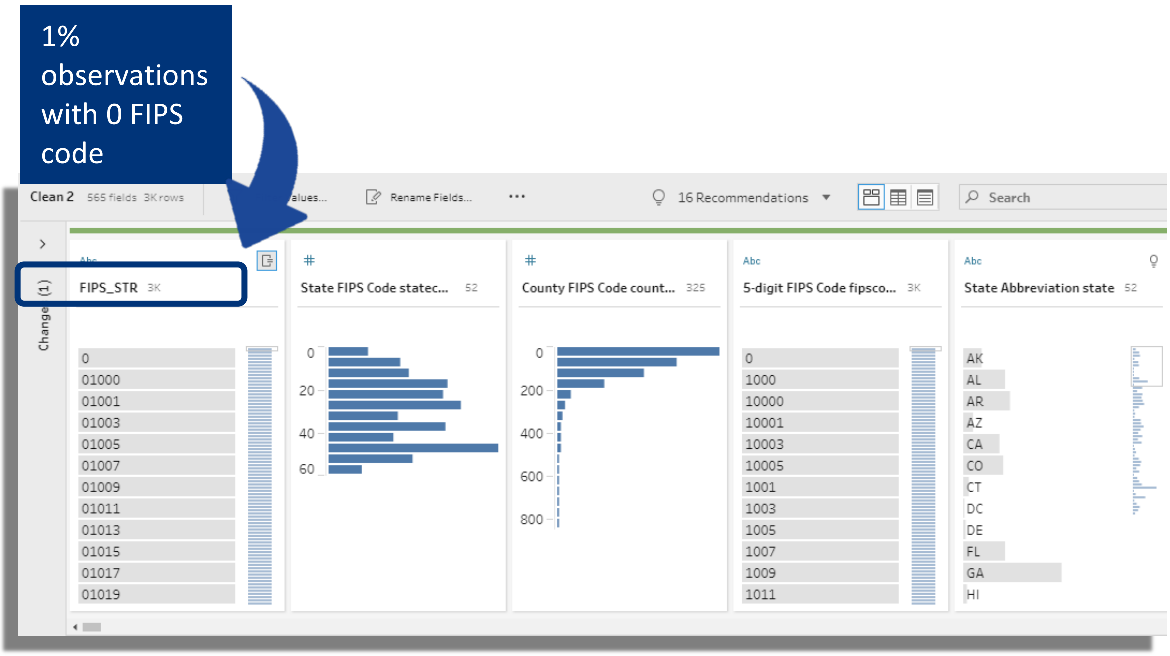


Figure 13: Dataset 2 variable summary after FIPS_STR creation

Going to the data view we can see that these 0 FIPS code values are coming from the rows which have aggregates for the country and the states.

FIPS_STR	State FIPS Code statecode	County FIPS Code countycode	5-digit FIPS Code fipscode	State Abbreviation state	Name county	Release Year year	County Ranked (Yes=1/No=0) county_ranked
0	0	0	0	US	United States	2,021	null
01000	1	0	1000	AL	Alabama	2,021	null
01001	1	1	1001	AL	Autauga County	2,021	1
01003	1	3	1003	AL	Baldwin County	2,021	1
01005	1	5	1005	AL	Barbour County	2,021	1
01007	1	7	1007	AL	Bibb County	2,021	1
01009	1	9	1009	AL	Blount County	2,021	1

Figure 14: Data view of Dataset 2

The dataset has a **County Ranked** variable which can be used to remove the aggregate observations to create a clean dataset of just the county level data.

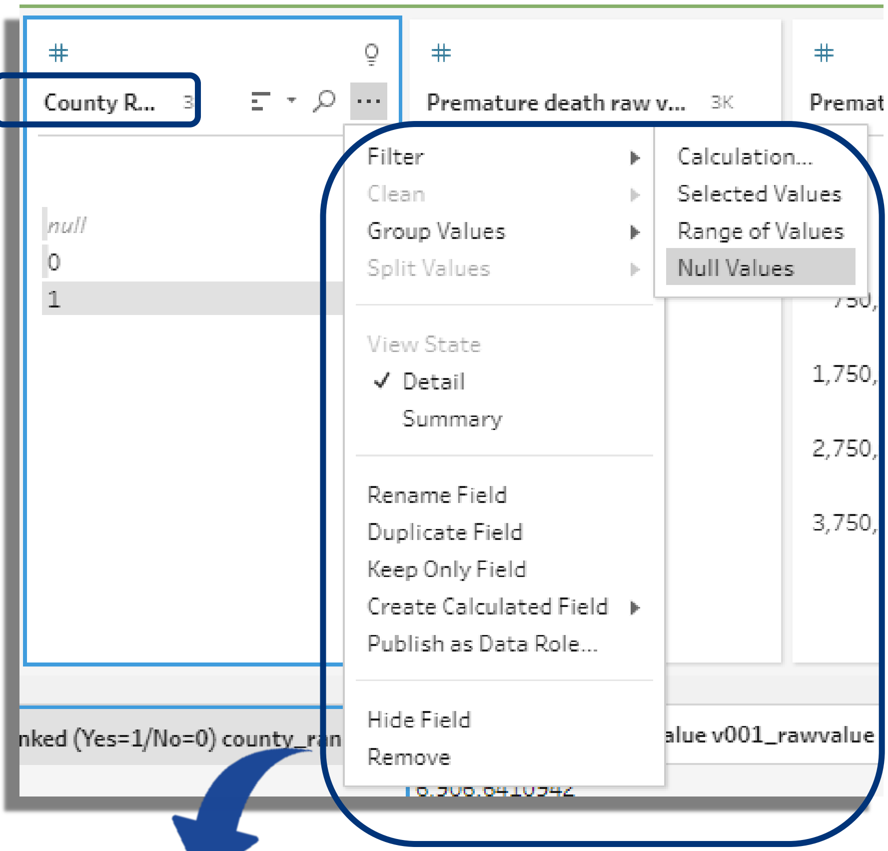


Figure 15: Filter Dataset 2 for Null values on County Ranked variable

Here we can use the **Filter** operation to remove any **Null values** which represent aggregate level data for County Ranked Variable

MISSING DATA

The variable summary section can be used to observe missing or inconsistencies with the imported data.

For example, in the Figure below we show that the Premature death variable has 2% of the observations with **Null or missing value** in them.

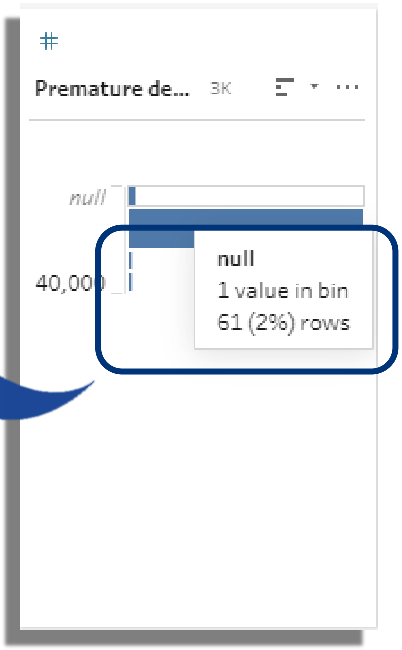


Figure 16: Premature death variable summary

We can choose to filter the observations out or make that decision in the later stages of the analysis. As a rule of thumb variables with over 10% missing values can cause issues in downstream analysis or dashboard development. In such situations, we can either remove the observations which have such values by using the Filter operations as above or we can remove the variable column entirely from the dataset. Filtering the observations will lead to records bring omitted for counties, whereas removing the variable excludes one facet of data for all the counties. The decision then relies on the overall goal of the dashboard as presented towards their intended users. Keeping Null values can be another option where in the final form of presentation the visualization can take it into account and present as data not available.

SEE LINKS BELOW FOR ADDITIONAL CLEANING OPERATIONS YOU CAN PERFORM WITH TABLEAU PREP BUILDER:

- Data Cleaning
- Best Practices of Data Preparation

DATA WRANGLING

MERGING MULTIPLE DATASETS[1]

Now that we have all three of our datasets filtered and checked for missing/null values, we can start with the data wrangling stage of the process.

- **KEYS**: The first step in the process is to identify data keys. Data keys are unique attributes of the data which can be used to merge different datasets together. In other words, we need to identify a variable/column from the datasets which can be used to combine the datasets together.
  Let's start with the Opioid Dispensing Rate and the County Health Indicators data. As we have discussed before we are trying to **combine these datasets** on the FIPS code of the counties, and we have cleaned and established the FIPS code as an index/key we can use.

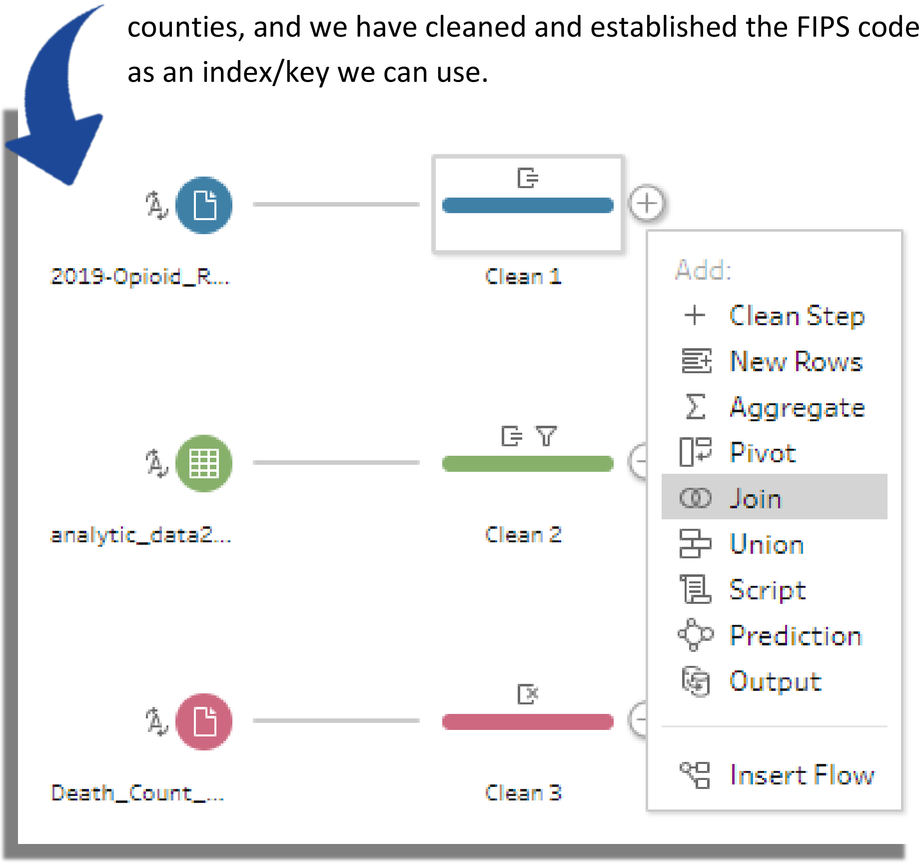


Figure 17: Insert Join/Merge Flow

[1] This step may not apply to jurisdictions who have already developed their data set prior to starting the dashboard developmental process. Skip over to Section: View Your Data for visualization techniques.

- **JOINS**[2]: First we create a new flow from the Clean 1 (representing the cleaned version of the Opioid Dispensing Rate) dataset to start the merge/join process (Figure 17). Then we can drag the Clean 2 (representing the County Indicators) into the join flow. The result is shown in Figure 18, where there are options of choosing what variables to join on (Applied Join Clauses) and the Join Type.

Joins between Datasets Can Be Of Two Main Types – A) Inner & 2) Outer

INNER represents intersection of data, where only common/matching key rows are kept in the join, whereas Outer represents a union, where all elements of datasets are kept.

In an **OUTER JOIN**, as it keeps elements which are not common, null values are associated with non-intersecting rows. Outer join can also be sub-divided into Left and Right outer joins, where in Left Outer join all elements of the left dataset are kept and only the intersecting elements of the right dataset are included. Right Outer join is the opposite of it, where all elements of the right dataset are kept and only intersecting elements of left data are included. Figure 19 describes an example of different types of joins.

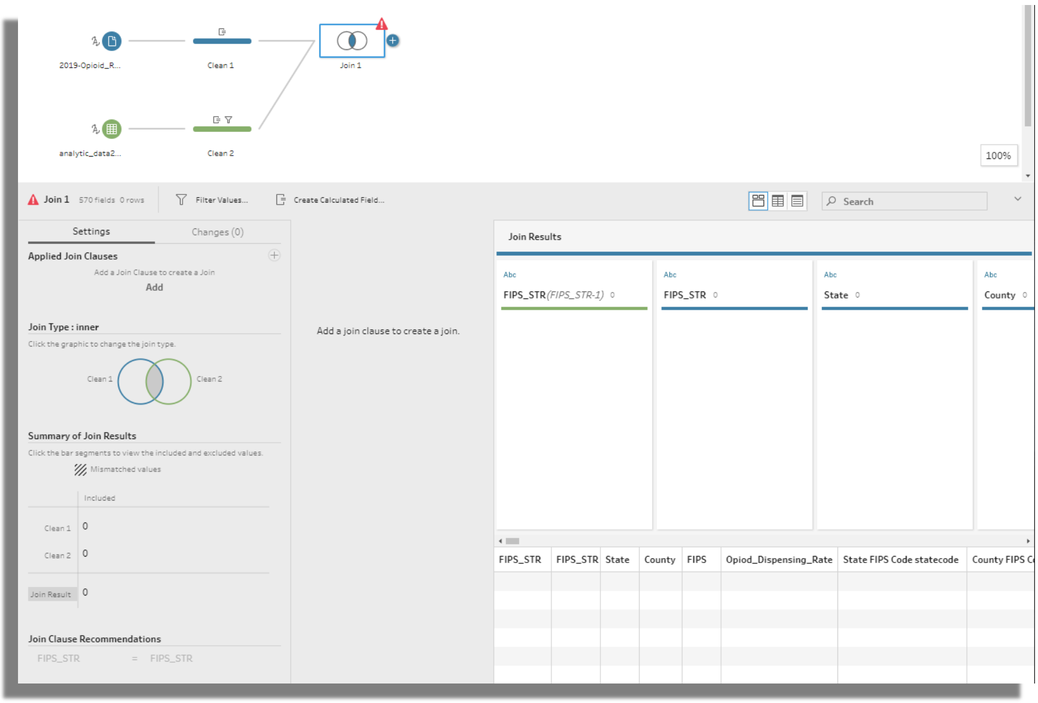


Figure 18: Initial Join Configurator

[2] Tableau uses terminology of Relationships, which is similar to merging/joining of datasets. See here for more details: Tableau Relationships vs Joins (https://help.tableau.com/current/pro/desktop/en-us/joining_tables.htm)

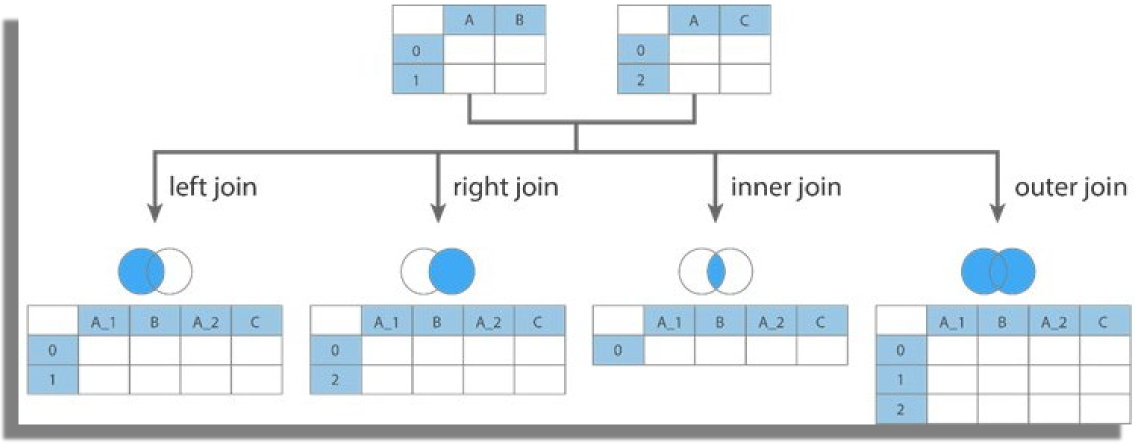


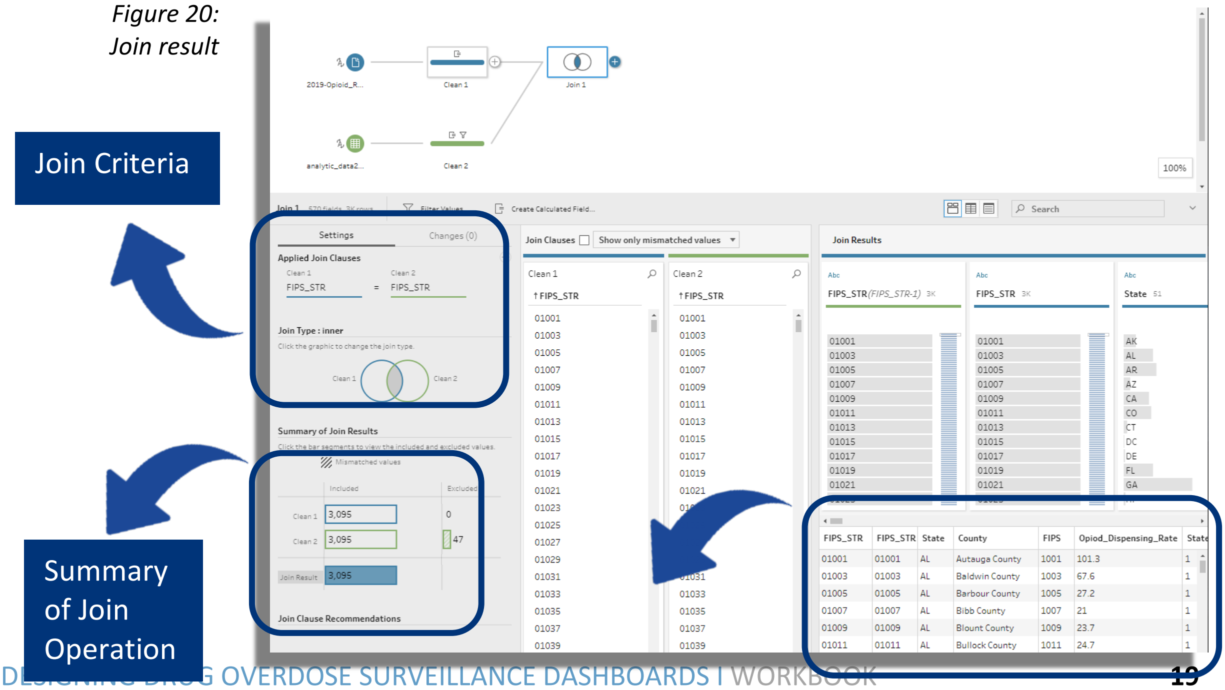
Figure 19: Join types

In our example, the goal is to develop a resulting dataset which contains the common data from the different datasets, thus we will use an Inner join. We select the FIPS_STR variable (calculated field created in Checking for Data Issues) as the Join Clause for both the datasets.

EVALUATING THE JOIN/MERGE

Figure 20 shows the outcome of the joined data. The Summary section describes the number of matched rows/observations (3,095) and the number of unmatched (47). It also show that these unmatched elements are coming from Clean 2 which represents the County Indicators dataset, which suggests that these counties do not have data for Opioid Dispensing Rate. We can select the green bar of the excludes to check which counties did not match. The bottom right section of the Figure shows the joined data including all of the variables from both the datasets.

Figure 20: Join result



We can also **merge the third dataset**, Opioid Mortality Data, into the joined data using the FIPS_STR and County Code (Opioid Mortality Data) as our keys.

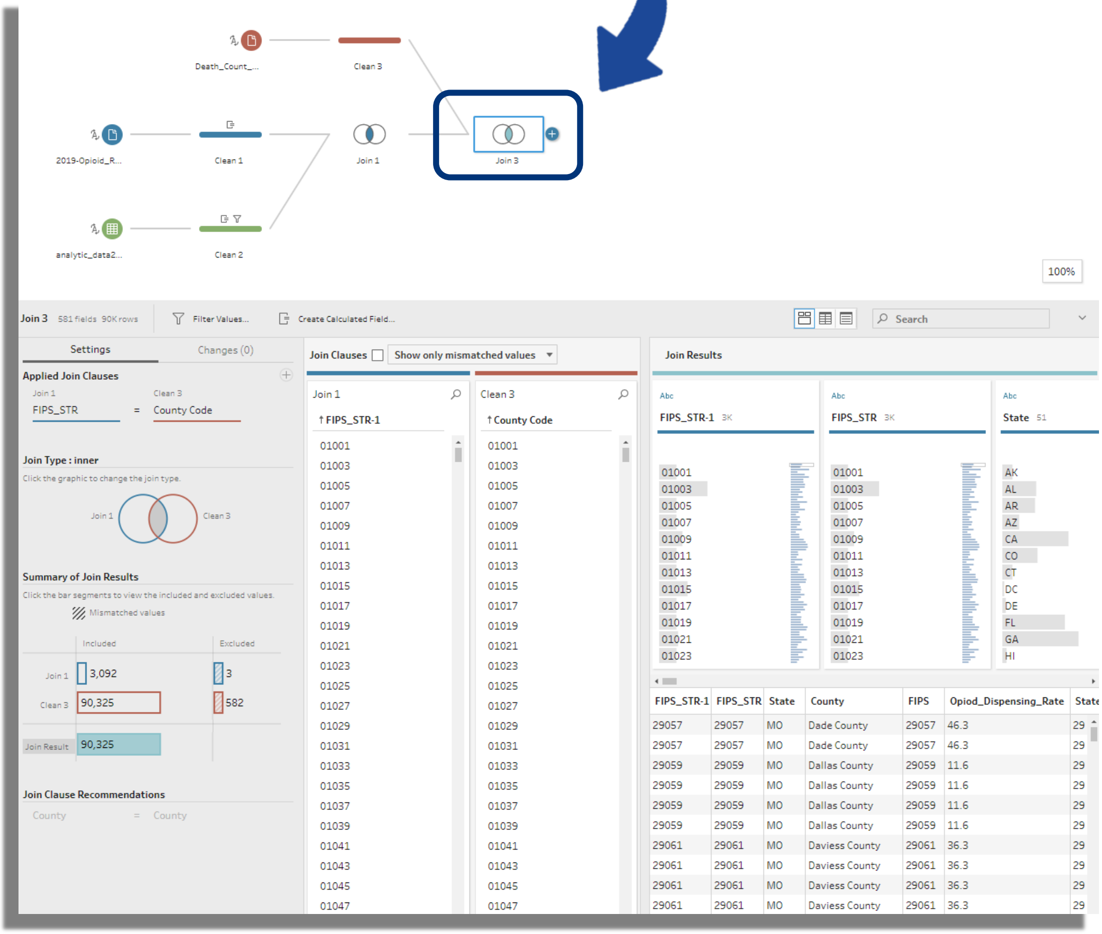


Figure 21: Join Opioid Mortality Data with the Joined data of Opioid Dispensing Rate and County Indicators

Figure 21 shows the Join pathways and the summary result. Here we observe that the 3 rows were excluded from the Join 1 (Opioid Dispensing Rate and County Indicators) and 582 rows are excluded from Clean 3 (Opioid Mortality Data), which suggests data was not available for all counties.

STORING THE RESULTING DATA

We can add an Output flow to the resulting join for storage of the data and use with Tableau. Figure 22 and 23 show the addition of Output flow for storage and the storage options. We can store the joined data as a Tableau data extract, which is efficient to be used in Tableau or as csv/xlsx files to be used in any other analytical software.

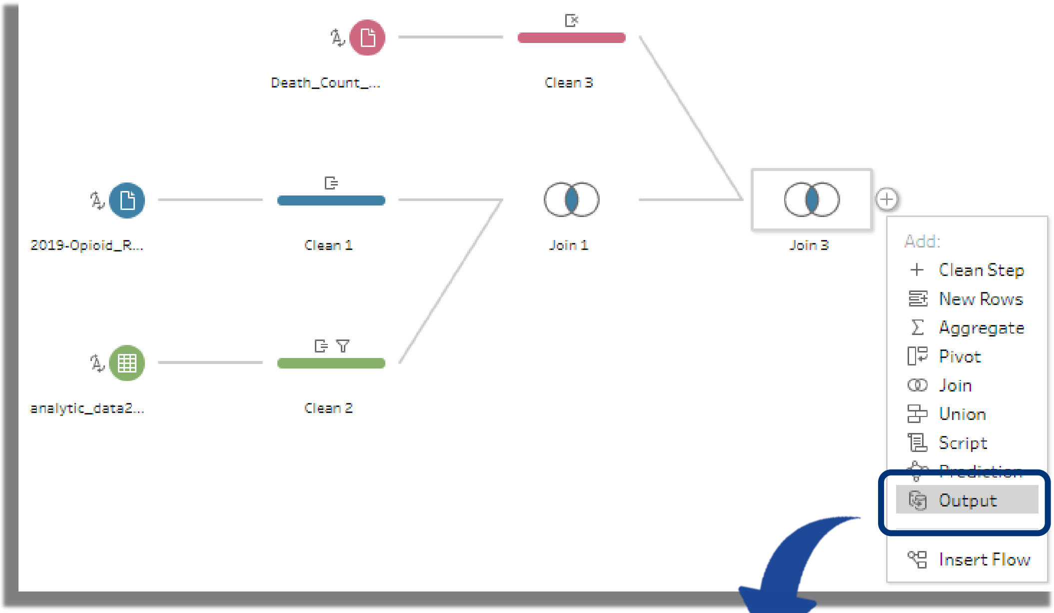


Figure 22: Storing the data with Output flow

Add an Output flow

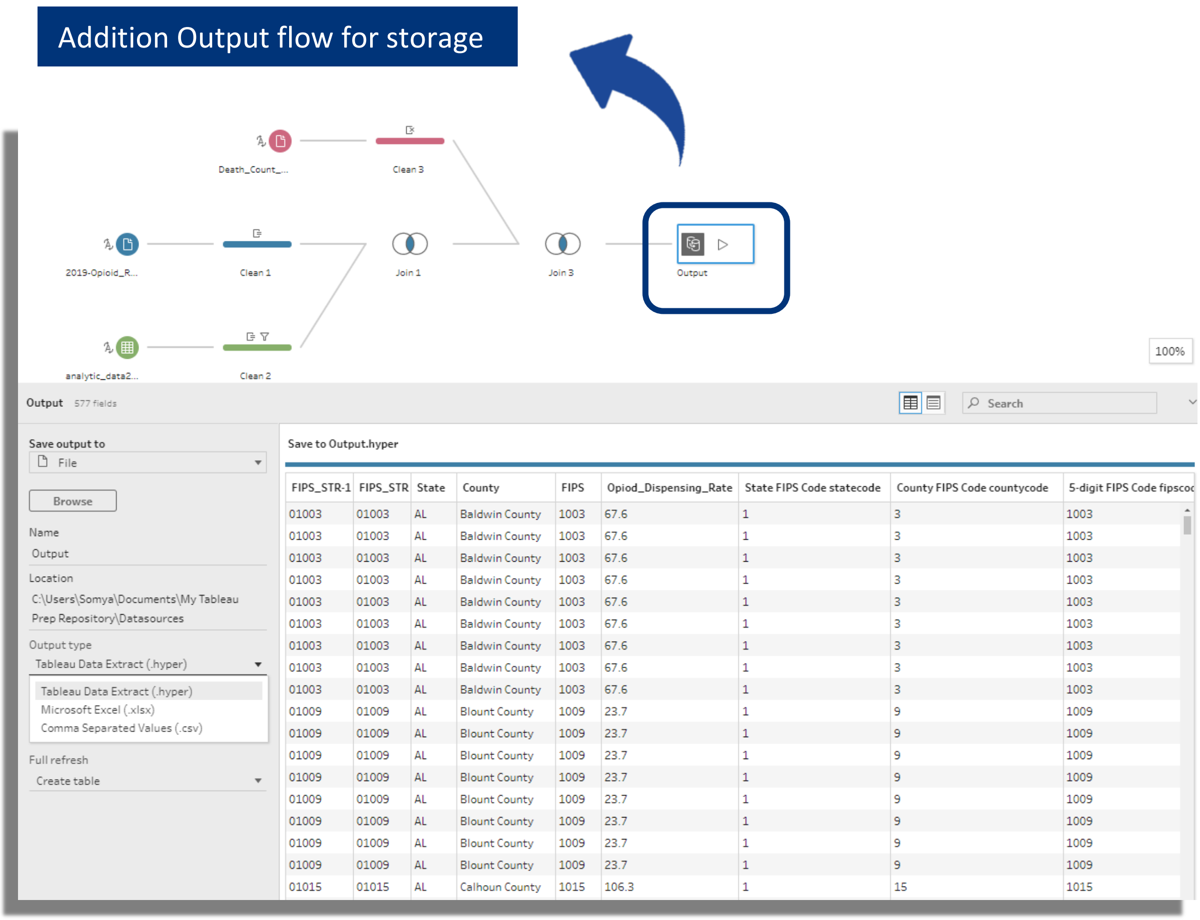


Figure 23: Configuration of Storage

VIEW YOUR DATA: Exploring basic graphs with the new data

MERGING MULTIPLE DATASETS

In order to perform some basic visualization, we load up the Tableau Data Extract file (from Tableau Prep.) that we created into the Tableau Desktop. Once the file is loaded the data source interface shows a similar interface as described in Section 1.ii. From here you can select the variables you would like to use in your visualization. All of the visualization is done either as a Dashboard or a Worksheet. Figure 24, highlights how you can create a new worksheet to work on a basic visualization.

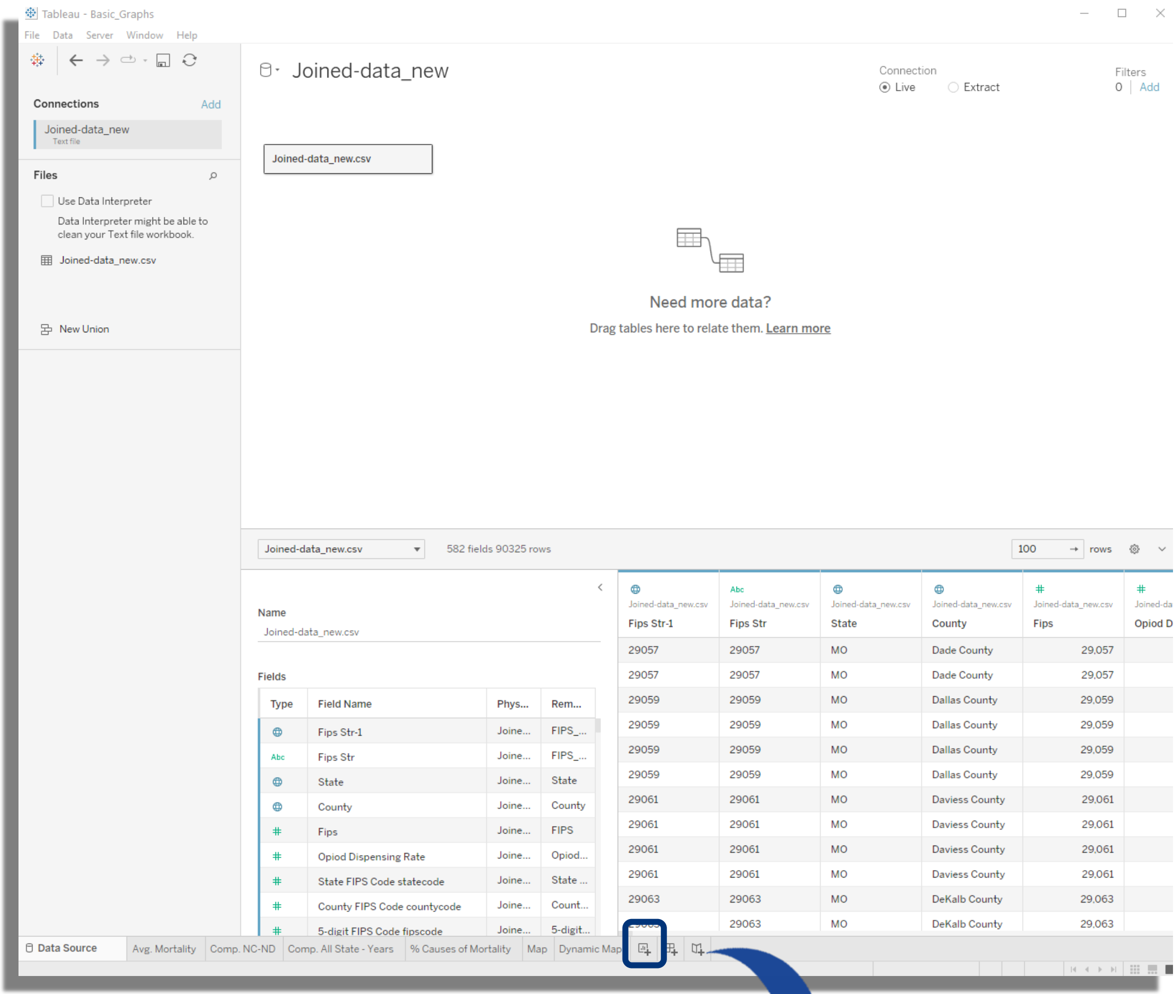


Figure 24: Loading merged data to Tableau Desktop

Create a new Worksheet

- **SIMPLE BAR GRAPH**: Here we are going to use **Bar graph** to visualize opioid related mortality and compare it by state. In a new worksheet, the first step is to select the row and column variables. Here we choose the State variable to be our column and the created variable Population Normalized Mortality (opioid mortality rate, per 10,000 persons) as our rows. As we are aggregating to State level, we average the county data to represent the state.

  Within the Worksheet, the State variable can be dragged to the columns field and the Population Normalized Mortality is moved to the rows, it automatically uses AVG (average) to aggregate the values. The function can be changed by clicking on the rows variable as shown in Figure 26.

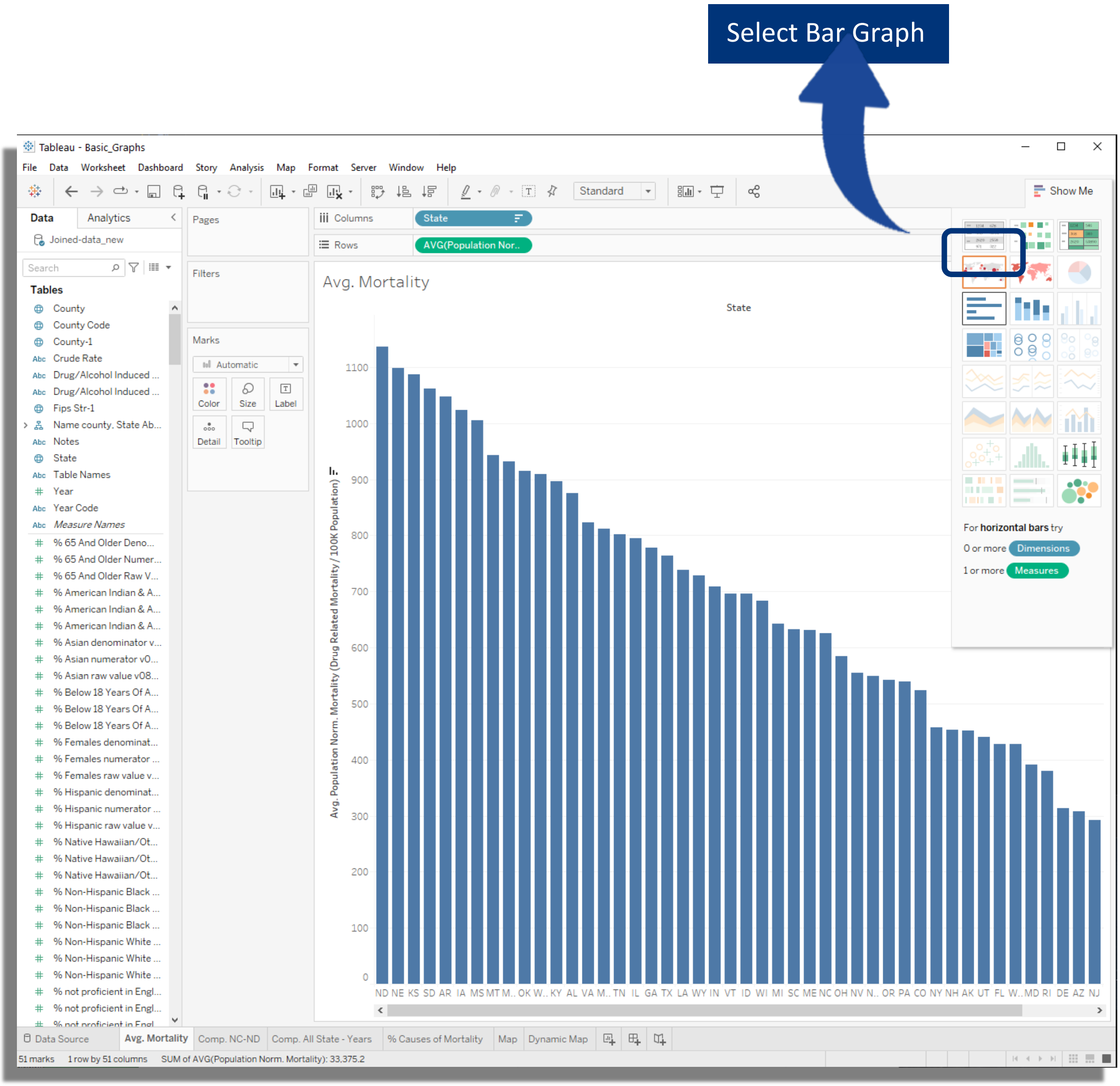
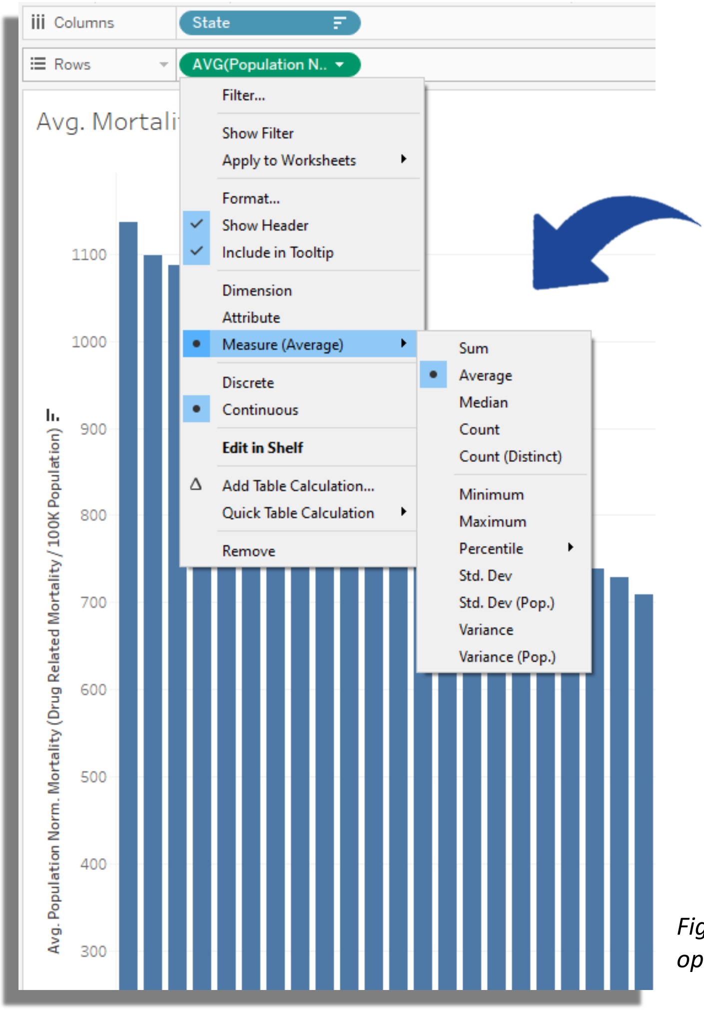


Figure 25: Average Opioid Mortality per State - Normalized for 100,000 population



Variable can be dragged to the columns field and when the Population Normalized Mortality is moved to the rows, it automatically uses AVG (average) to aggregate the values.

Figure 26: Changing the operation on a measure variable

- **SCATTER PLOT WITH FILTERING:**

  In this visualization we create a scatter plot of Opioid related mortality versus population for counties present in North Carolina and North Dakota.

  a) Here we first select the Population and the Population Normalized Mortality as our columns and rows respectively.
  b) Now to select specific states, we need to add the State variable to the Filter section. This opens a new window, where you can choose which States to visualize for.
  c) In the Marks section, we can drag the State and County variables, this is needed by the scatter plot to color the data points on the plot and position the points.
  d) On the State Marks, one can click on the left side of the variable to select State as a color indicator.
  e) After all of this is set, we can then select the scatter plot as the type of visualization. The resulting visualization and its different sections are shown in Figure 27 and 28.

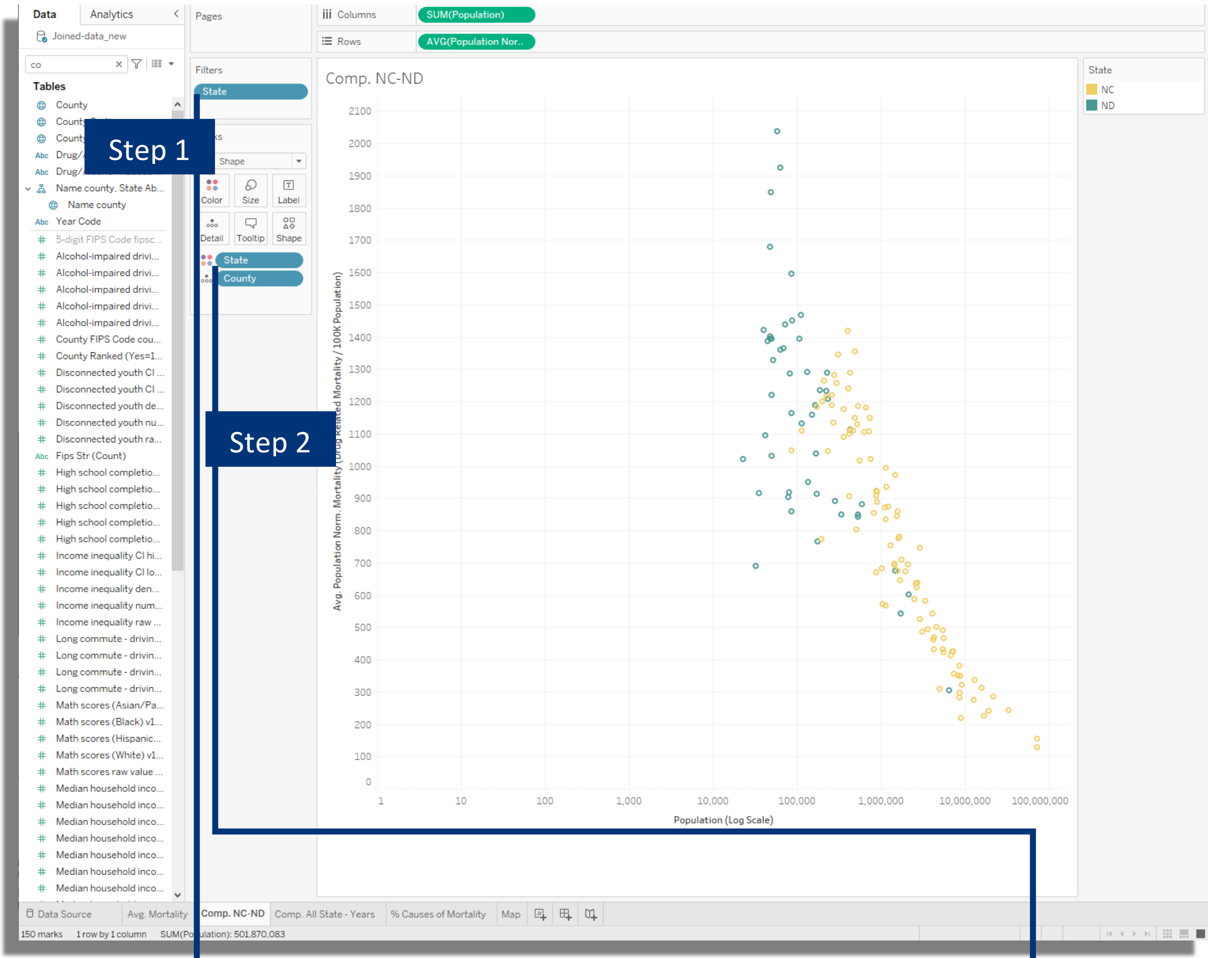
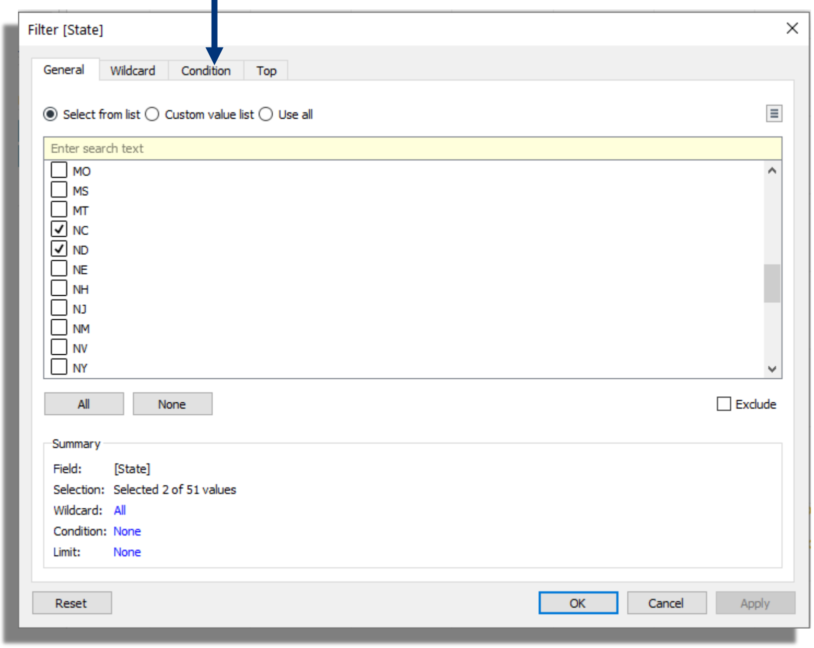


Figure 27: Comparing Opioid Related Deaths by County for NC and ND



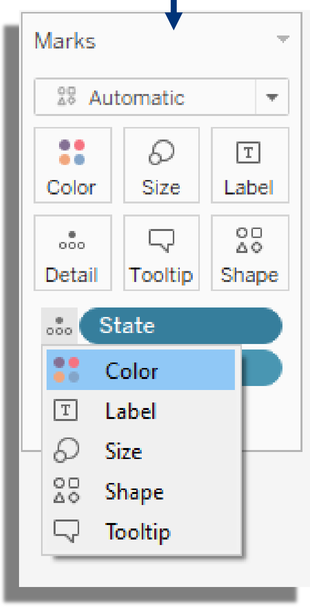


Figure 28: Filter and Marks section operations

- **LONGITUDINAL ANALYSIS WITH LINE PLOT:**
  Display trends of Opioid mortality across the years of 1999-2020 for different states can be done with line plots.

  Here we choose Year variable as our column and the rows variable to be the Population Normalized Opioid Mortality. State variable is chosen as the Marks in order to differentiate the trends of different states and Population is chosen as text output for hover. Line plot is selected to display the visualization as shown in Figure 29. The graph also displays an interactive legend to select and deselect states on the right panel. This can be done by allowing a legend for the visualization, as shown in Figure 30.

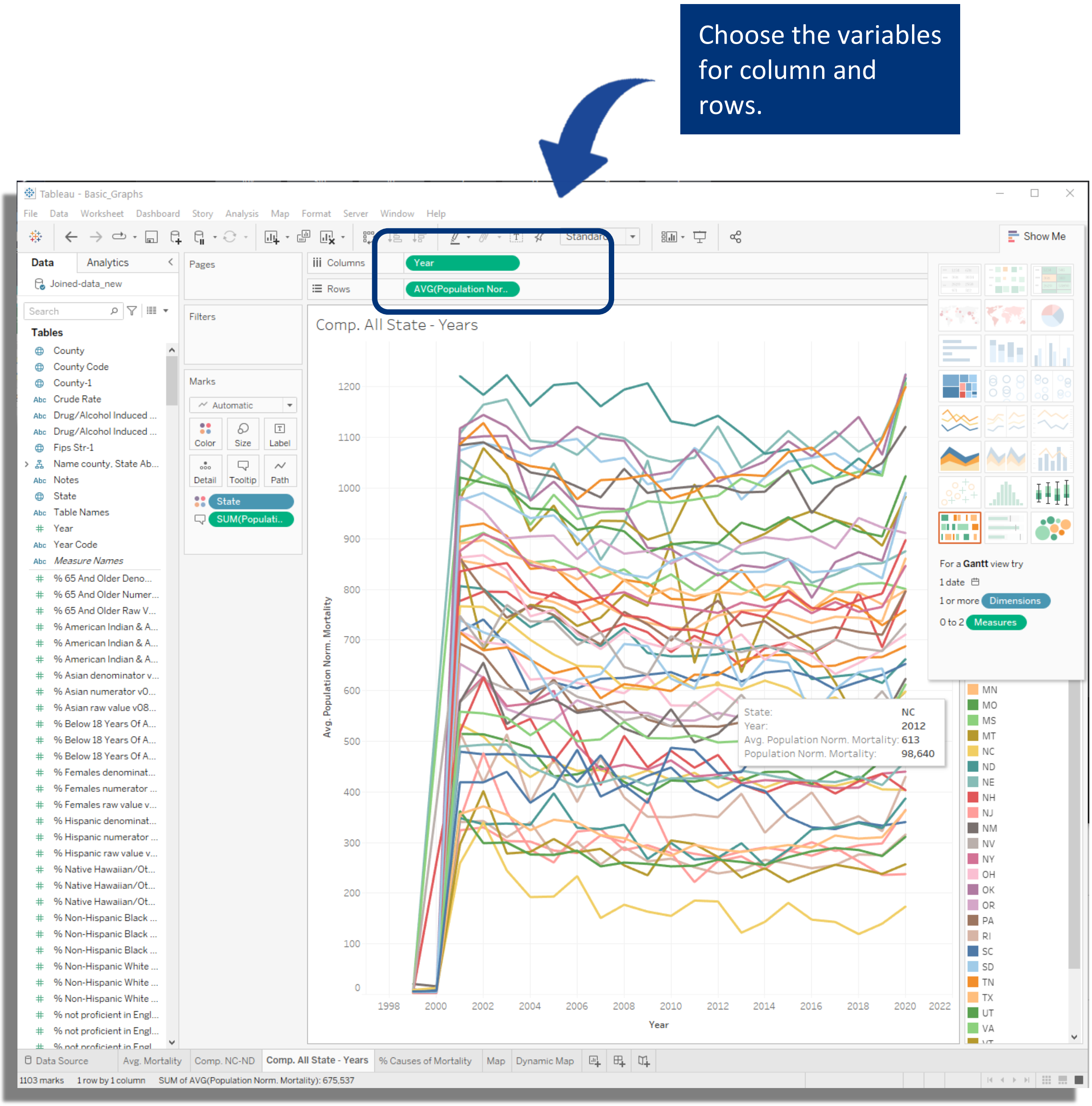


Figure 29: Longitudinal Trends (1999-2020) of Opioid Related Deaths for States

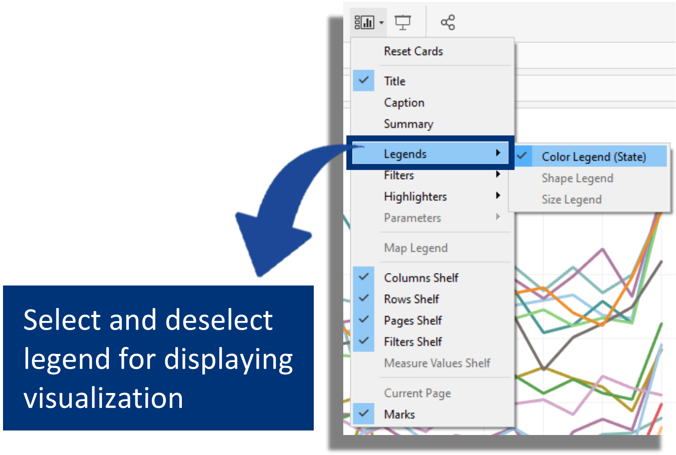


Figure 30: Setting a legend for a visualization

- **PIE CHART PLOT:**

  We can use a Pie chart visualization to proportions of drug use mortality. Here we select Drug/Alcohol Induced Cause as our Filter variable, and Population Normalized Mortality as our Marks for the Pie chart proportions. We have also included the Population Normalized Mortality as a text output in the Marks in order to display the values besides the Pie chart. The Filter attribute allows for interactivity as a user can select a particular cause and highlight it, as shown in Figure 31.

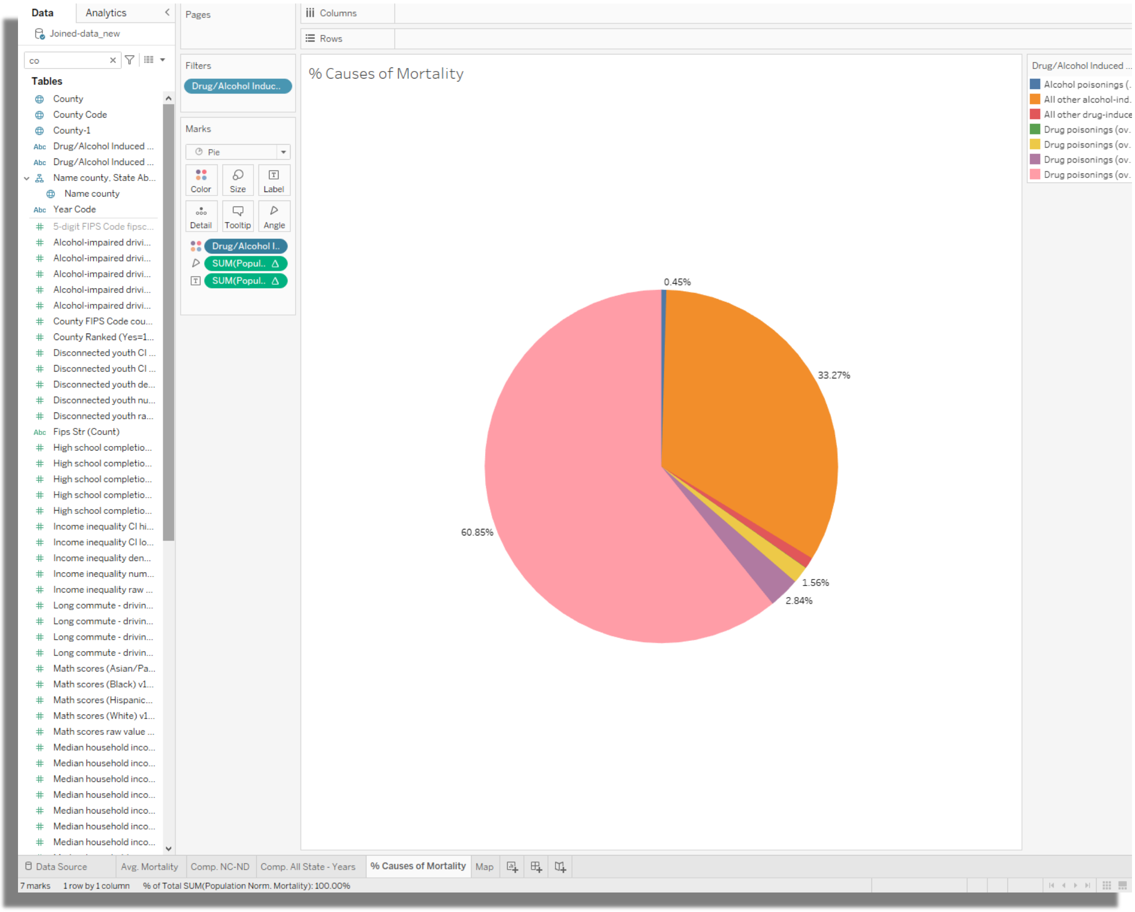
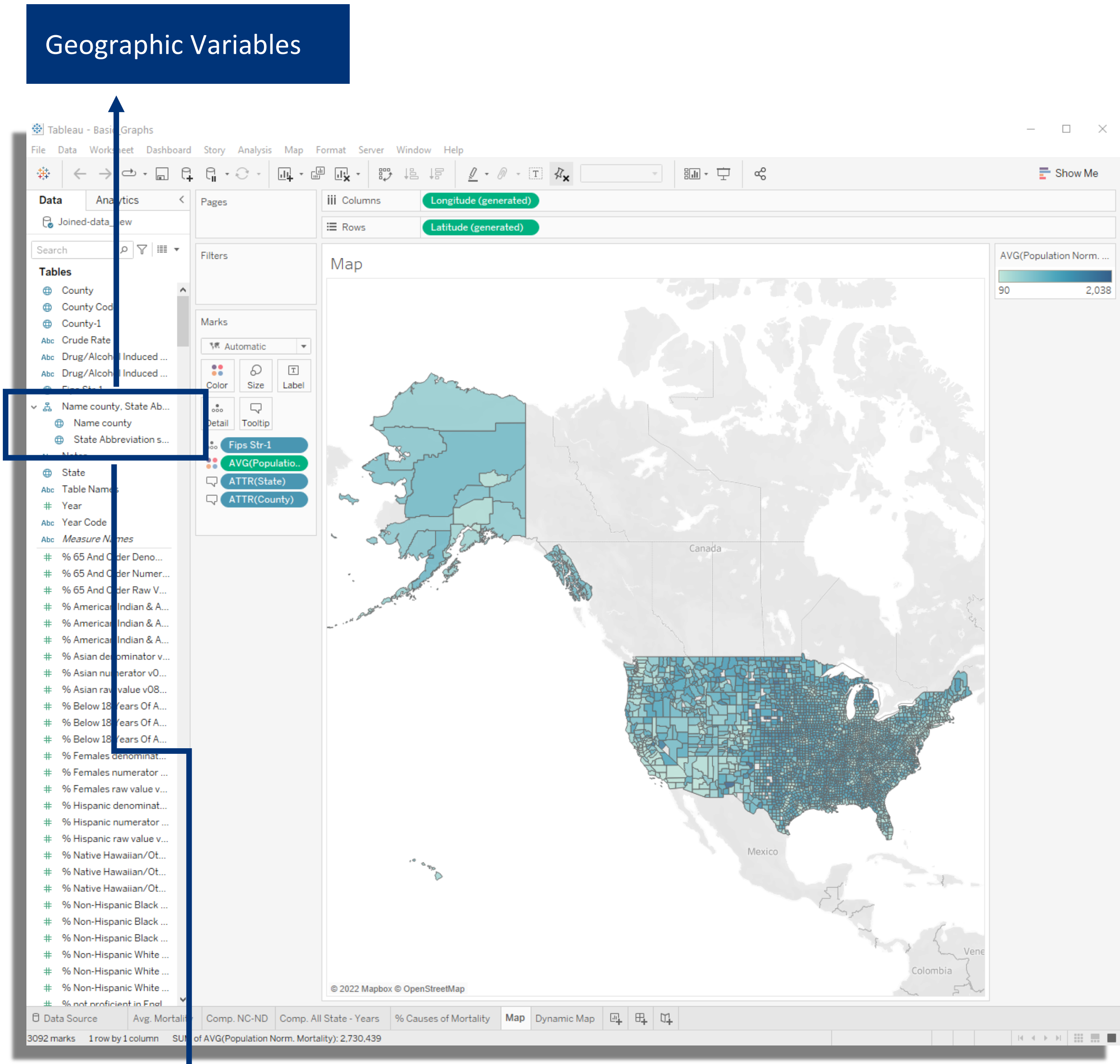


Figure 31: Underlying Cause of Death - Opioid Mortality

- **GEO-SPATIAL PLOT:**
  As our data is at County level we can visualize the Opioid mortality geographically. Figure 32, shows the geographic visualization of the data.



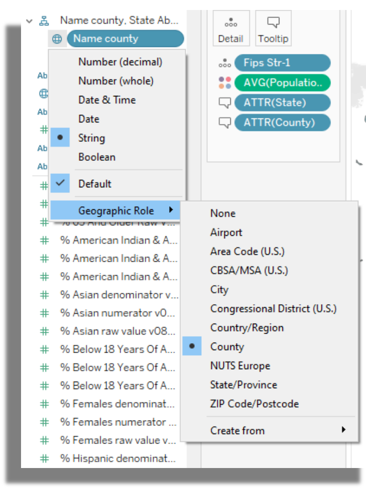


Figure 32: County level map of Opioid Mortality (per 100,000 population) - 1999-2020

a) Figure 32, shows the geographic visualization of the data, here we take the Population Normalized Mortality variable calculated by each county and average it across the years.
b) Prior to this we need to select the Geographic variables of County and State.
c) Tableau allows for automatic geographic recognition of variables by selecting the appropriate designation of geographic specificity in the variable attribute.
d) By dragging the Name County – State Attribute variable to the rows and columns section, the area is auto populated with Latitude and Longitude attributes of the variable.
e) Following this the map visualization can be selected to display the data.
f) In this example we have also added hover over information by including the FIPS code, County, State, and Mortality variable in the Marks section of the visualization

CREATING A DASHBOARD

In this section we develop a simple interactive dashboard for our data. This is going to be based on the visualizations developed in the previous section and here we discuss how to develop the designs and add multi-level interactions in a dashboard.

WIREFRAMING

The first step in developing a dashboard is to create a mockup/wireframe model of how the dashboard is going to look. There are several tools which can help you get started on developing a wireframe design. The simplest of which is MS PowerPoint, which is used as a starting point for a lot of dashboard designs. More powerful wireframing tools exist such as Balsamiq (https://balsamiq.com/) which can provide interactivity even in the wireframe designs. This link shows video tutorials guiding you though the process of designing and developing dashboard wireframes (via Balsamiq):

Balsamiq Tutorials

Also refer Dashboards: Topics in Design, Evaluation, and Maintenance for Effective Insights of the Drug Overdose Surveillance

DESIGNING AN EFFECTIVE DASHBOARD

Apart from the visual appeal of a visualization/dashboard, we also need to consider how effective it is in conveying the intended message. Here we can take insights from design concepts such as layered designs and/or hierarchy of information when designing a dashboard. The concepts outline how the dashboard can invite the users to dive into the interaction by creating tiered levels of information that are displayed to the user.
Below we link useful resources towards this concept:

a) Layered Grammar of Visualization
b) How to use hierarchy of information for Dashboards
c) Balsamiq Data Table Design

WIREFRAME/DESIGN EVALUATION

One of the key benefits of developing a wireframe is to evaluate how effective is the design via user testing. This is an iterative loop where a participating user group is shown the wireframe designs and surveyed to ascertain the overall usability of the dashboard and catch issues with the design prior to deployment. The design evaluation can be divided into two different categories: 1) A/B testing and 2) Overall usability evaluation. A/B testing or split testing is used to compare different versions of the dashboard/visualization, by surveying effectiveness of each design to different user groups. In usability survey, a single design is shown to participating users as an initial prototype and surveyed for more open ended questions such as,

1. Does the dashboard so far provide all the information you need on the topic? If not, what is missing?
2. Was the navigation smooth and clear? If not, why?
3. Generally speaking, do figures seem correct to you? If not, please list the potential errors you have spotted. This can be an iterative process where the feedback from the respondents are taken into account, changes made to the design, and the users surveyed again.

Below we link some of the useful resources regarding the topic:

- Use case matrix and A/B testing
  a) A-B testing with Tableau
  b) Balsamiq Usability Testing

- Example surveys
  a) Engaging users via dashboard testing
  b) Creating a usability dashboard

CREATING DASHBOARDS WITH TABLEAU

Here we will create a simple dashboard with Tableau. First, we develop a wireframe design with Balsamiq.

Figure 33 Shows The Design Where We Have 4 Sections Within The Dashboard:

1) Choropleth Map (county level) of Opioid Mortality
2) Data Table
3) Bar Chart of Population
4) Selector/Filter for States.

The dashboard will have interactivity where based on the selection of each section other areas will be highlighted and the Selector section allows for a group based filtering of attributes.

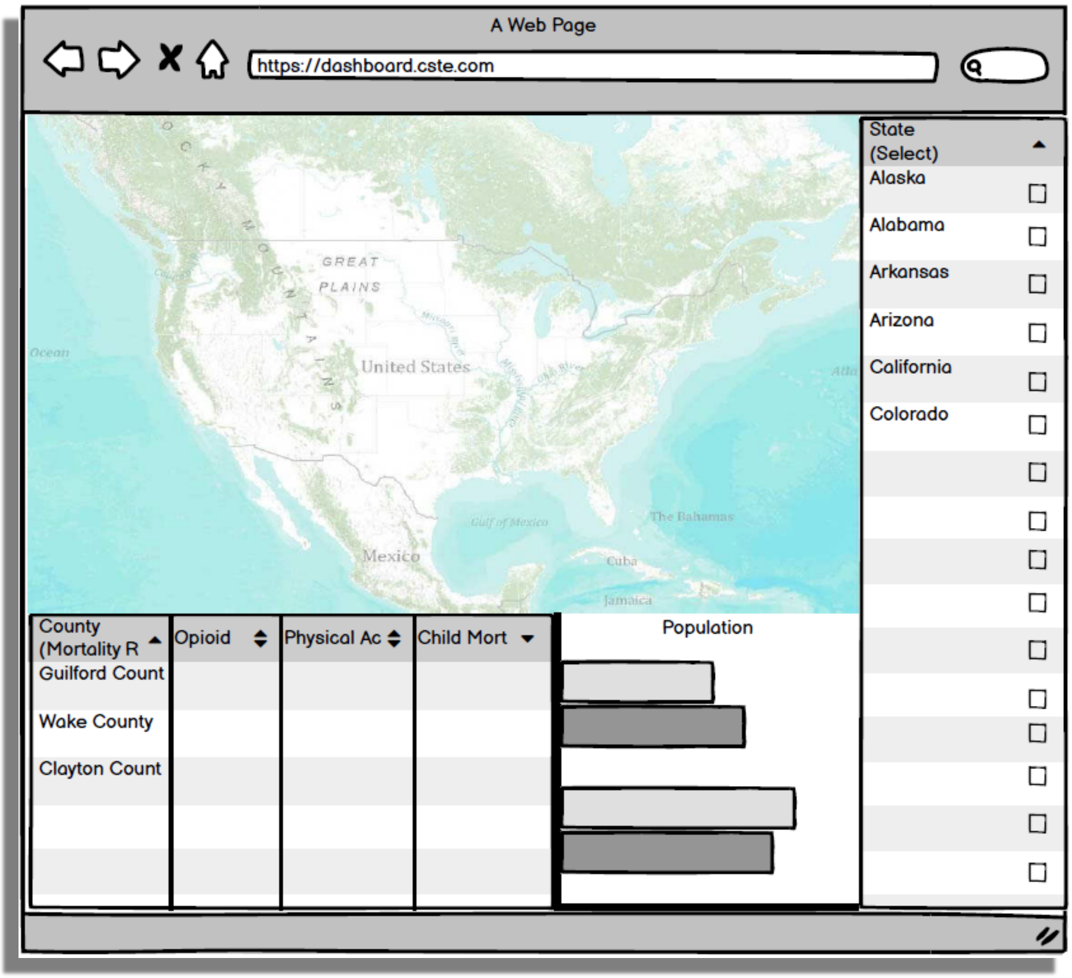


Figure 33: Wireframe design of dashboard

CREATING THE SHEETS FOR DASHBOARD

As the dashboard is going to have three different visualizations:

1) MAP
2) DATA TABLE
3) BAR GRAPH

We need to create three different sheets for each of them.

1

Figure 34 shows the choropleth map where we utilize the geographic attributes of State and County (as shown in Section 4.V) and the variable Population Normalized Opioid Mortality to create our visualization.

While a national level map may not be useful for local jurisdictions. Chloropleth maps indicate distinction between different areas, which can be used to provide end users comparative visualization between different counties or even census-tracts.

We also add the State and County to our Filters section to allow for selection of states and their counties.

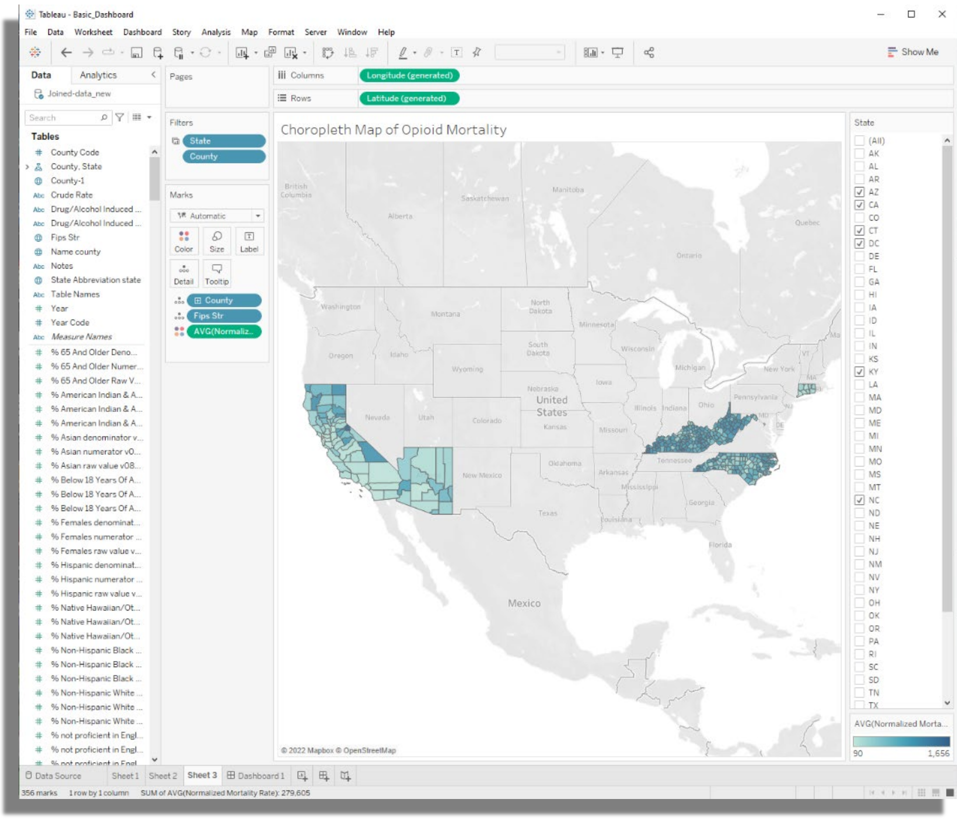


Figure 34: Choropleth Map of Opioid Mortality

2

In the second worksheet (Figure 35), we create a data table with the following variables:

- State and County (as our Rows),
- Population Normalized Mortality,
- Opioid Dispensing Rate,
- Physical Inactivity &
- Child Mortality as Measures variables.

Here we also add the State and County geographic variables to the Filters section.

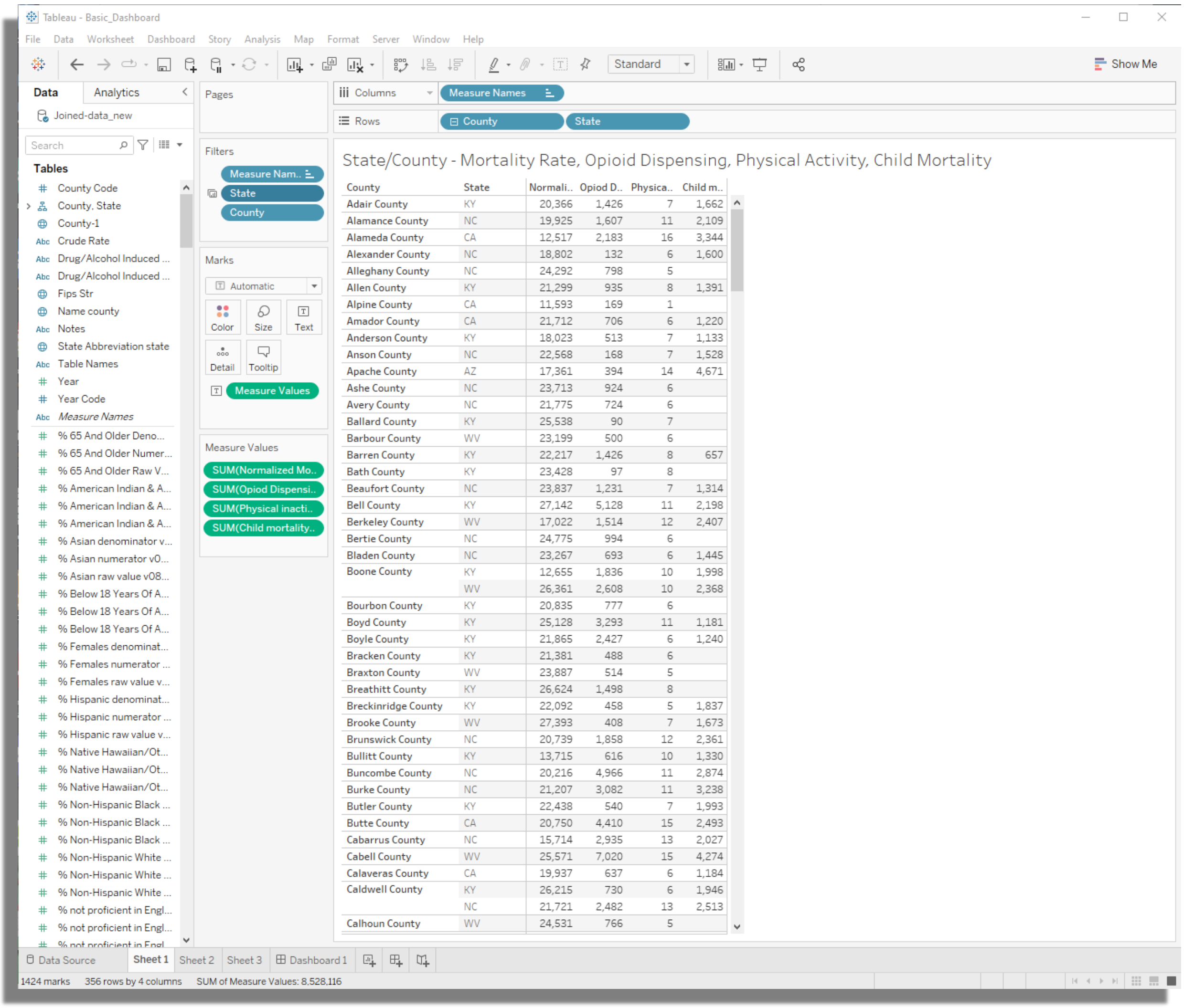


Figure 35:Data table of Mortality Rate, Opioid Dispensing, Physical Activity and Child Mortality

3

Finally, for the third worksheet (Figure 36) we create a bar plot by taking the Population variables as our Column and State as our Rows. This graph will be used to show aggregate population of state in the dashboard development section. Here we also add County and State to the Filters section of the worksheet. We now have three distinct components to build a dashboard, a geographic map highlighting county level visual information, a data table providing granular data by county, and an aggregate level visualization of population per state. Within the workbook we are showcasing examples on how different dashboard components can be developed using Tableau. As our readers might have different use cases, they can choose to develop different visualizations and/or information displays components for their dashboards. We refer our readers to Resource Guide - Dashboards: Topics in Design, Evaluation, and Maintenance for Effective Insights of Drug Overdose Surveillance, on how to select different indicators for their dashboard.

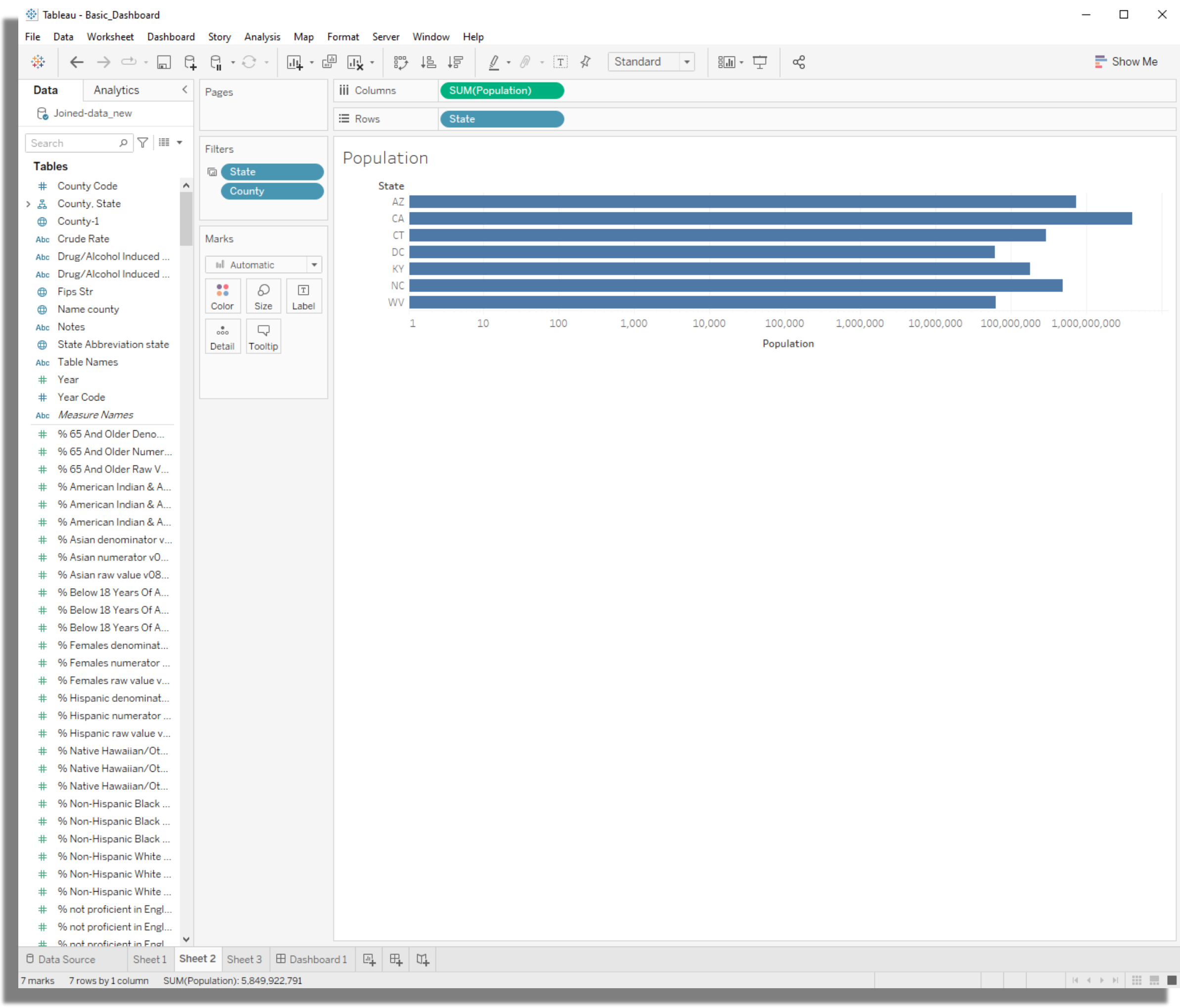


Figure 36: Horizontal Bar Graph for State Population

Once the worksheets are completed, we go back to the first worksheet of the map visualization. Here in the filters section, we right click and select Apply to Worksheets – Selected Worksheets. This will pull up a menu which allows you to select which worksheets will also filter for the same attributes as selected by the primary worksheet (map). As we want this to be applied to all of our worksheets we select all of them.

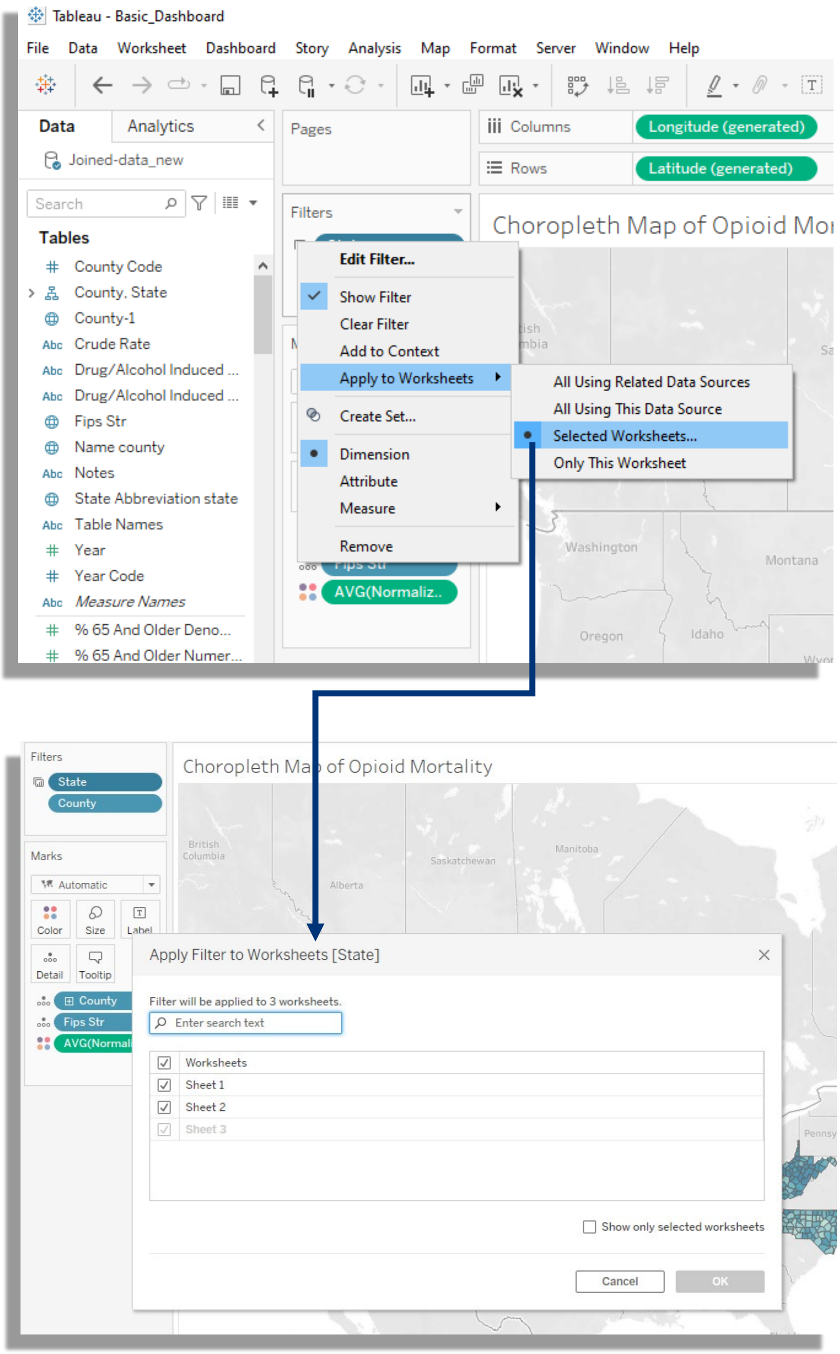


Figure 37: Connecting worksheets to have a single filter.

ADDING SHEETS TO A DASHBOARD

Once the sheets are completed, we can now create a dashboard from them. Using the bottom row of the Tableau interface, we can add a new dashboard as shown in Figure 38. After this you will have a blank canvas of a dashboard which allows you to drag and drop sheets that you have created (Figure 39).

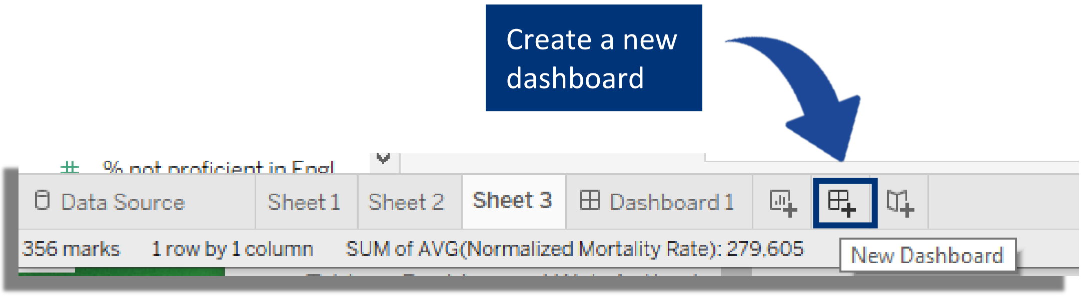


Figure 38: Create a Dashboard

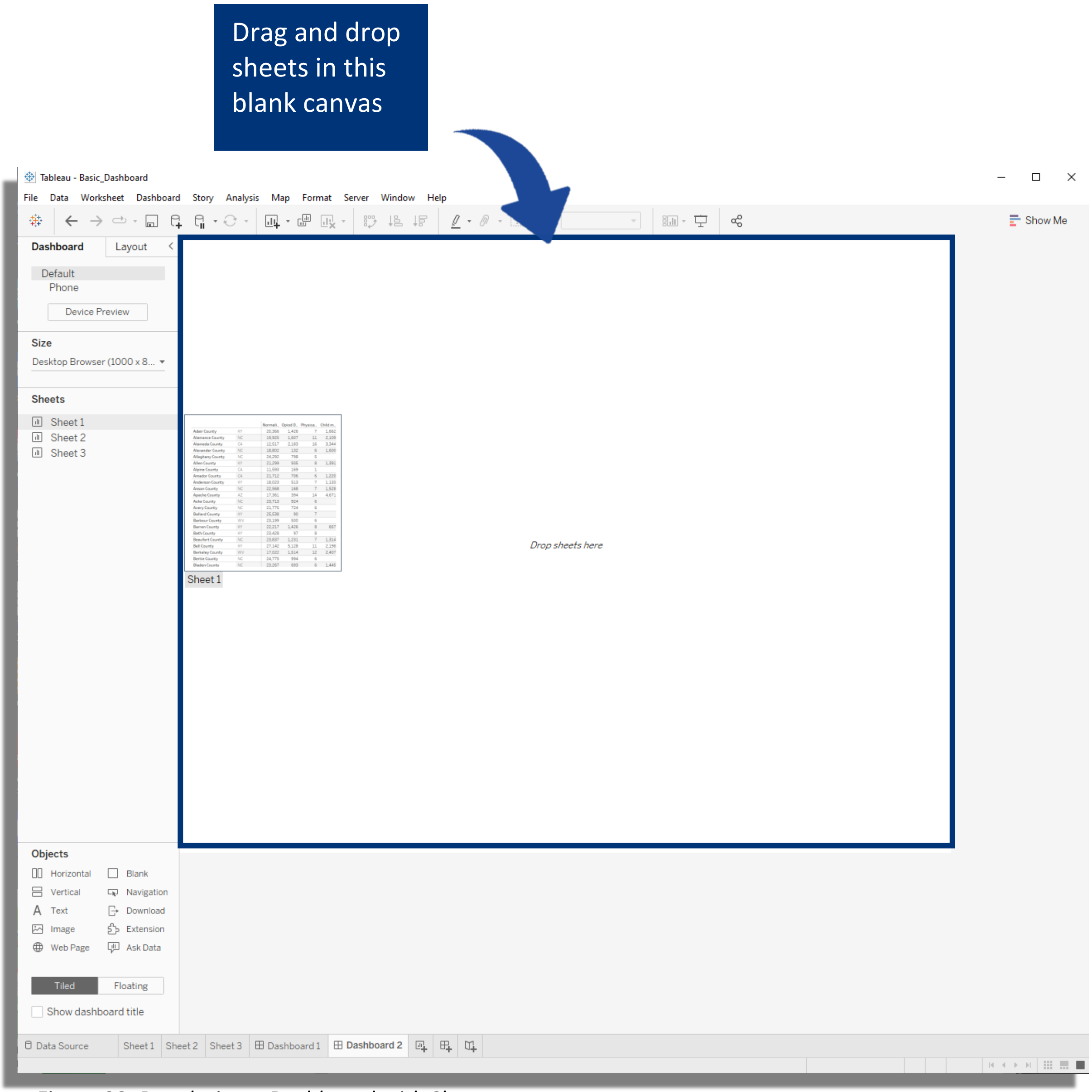


Figure 39: Populating a Dashboard with Sheets

Figure 40 shows the completed dashboard with the different sheets pulled in.

> The focus of the dashboard is the map, with the data table and bar graph providing additional information.

The State selector allows for dynamic interactivity, where based on the selection of the state, the values are populated in the map, data table, and the bar graph.

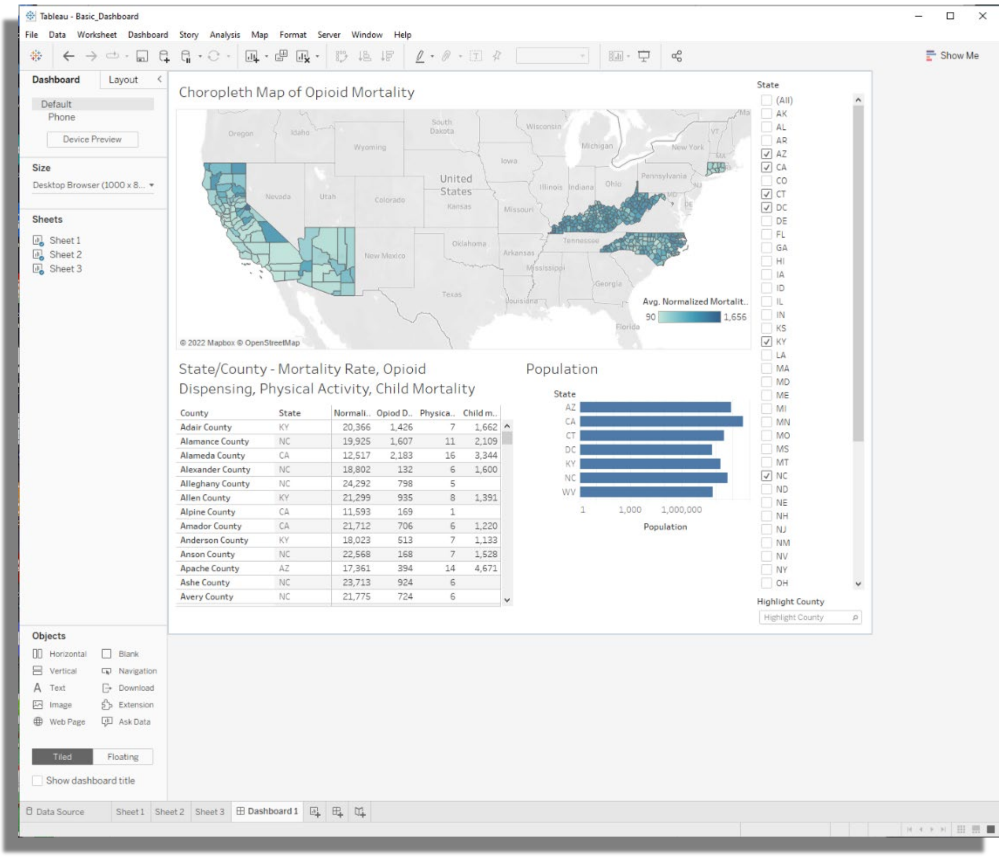


Figure 40: Dashboard with map, data table and bar graph

ADDITIONAL INTERACTIVITY

Apart from the filter attribute we can also provide additional interactivity to the dashboard with user actions. Here we can select a hover action (Tooltip), where if a user hovers over a particular attribute, the linked attributes are also highlighted in other visualizations. Figure 41 shows an example, where if we hover over NC in the bar population visualization then the data table highlights the counties in NC and their attributes.
This can be created by selecting actions in the worksheet menu and adding an action (Figure 42). Here we select run action on Hover and select all the sheets to be affected. The target highlighting is selected to all fields but one can select specific fields if needed in this section. Once the action is saved, it allows for highlighting of the different attributes in the visualizations on hover over one.

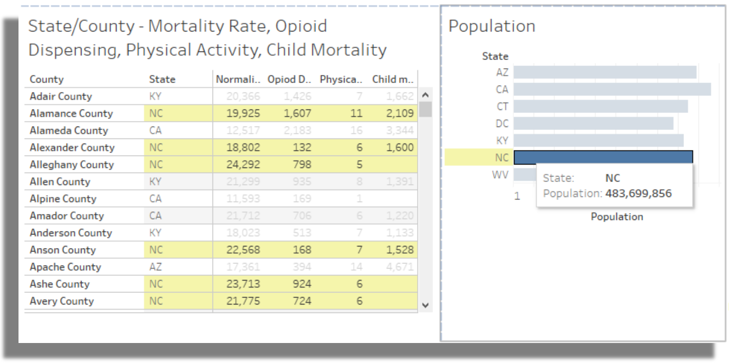


Figure 41: On hover highlighting

A dashboard needs to be inviting for end-users to interact with. Including information about goals of the dashboard, instructions on how to interact, providing tool tips (information about values), and documentation, aids in the effort. See here for more information on [Visual Best Practices using Tableau](). Resource Guide 1 also explores best practices in visualization and dashboard development.

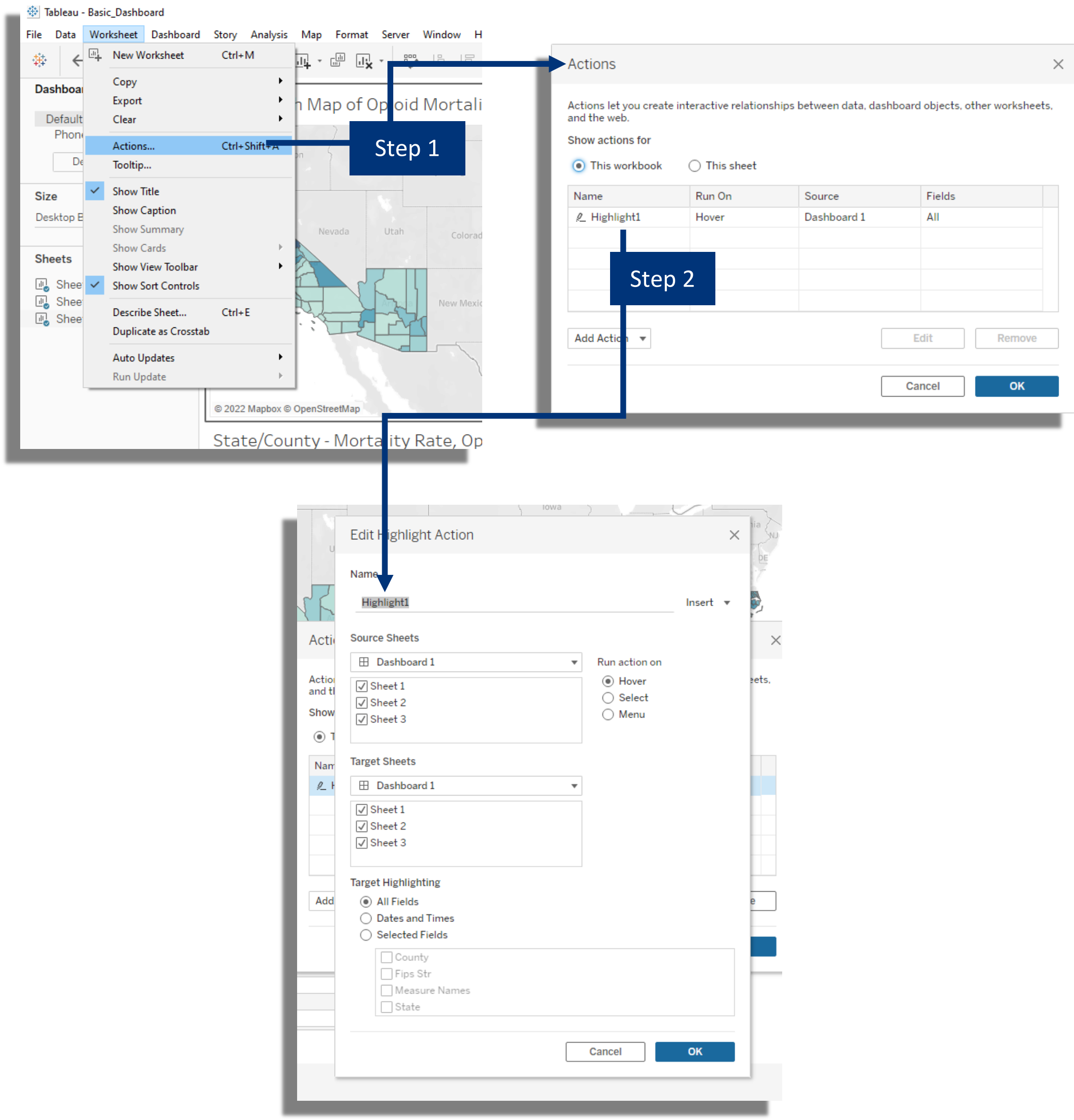


Figure 42: Highlight on hover for all worksheets

PUBLISHING A DASHBOARD

- When you want to share a workbook with your colleagues, you can publish it to Tableau Server or Tableau Online via the share button. There, other people can view it, interact with it, and even edit it if their server permissions allow. The following link can guide you through the process:
    a) Publishing Tableau Workbooks

- You can embed interactive Tableau views and dashboards into web pages, blogs, wiki pages, web applications, and intranet portals. Embedded views update as the underlying data changes, or as their workbooks are updated on Tableau Server or Tableau Online. Embedded views follow the same licensing and permission restrictions used on Tableau Server and Tableau Online. That is, to see a Tableau view that's embedded in a web page, the person accessing the view must also have an account on Tableau Server or Tableau Online.
  The following links guide you through the process:
    b) Tableau Embed Views onto Webpages
    c) Tableau Embed Dashboards

- Dashboards can include layouts for different types of devices that span a wide range of screen sizes. When you publish these layouts to Tableau Server or Tableau Online, people viewing your dashboard experience a design optimized for their phone, tablet, or desktop. As the author, you only have to create a single dashboard and deliver a single URL. Please refer to the following links which show to cater to different device layouts in Tableau Server:
    a) Tableau Dashboard Browser Rendering
    b) Making a mobile friendly data visualization
    c) Create Dashboard layouts for different devices

- If you are interested in analyzing how well the dashboard is doing in real world interactions after deployment Google Analytics is a good way to track users. Here we share some additional resources to allow Google Analytics integrations with a Tableau Dashboard:
    a) Tableau and Google Analytics
    b) Tips for Tableau and Google Analytics

UPDATING A DASHBOARD TO NEW DATA

- Tableau dashboards can be updated to new data. As Tableau connects dashboards directly to the data sources a simple Refresh operation can be performed when new data is available. The following link can guide you through the process:
    a) Refresh Data Sources Tableau

SUPPLEMENTAL RESOURCE

Video resource guides walking through each step of the development process. (Short 5-10 mins videos)

- Audio voice over
- Closed captions