

Geometry Data Type

Oracle Analytics Product Documentation

29th October 2024 | Version 1.6

Contents

Disclaimer	2
Revision History	2
Summary	3
Geometry Data Type	3
Business Value	3
Accelerate Map Design and Visualization	3
Optimize Map Layer Maintenance	3
Support Spatial Calculations	3
Visualize Geometry Columns in Oracle Analytics	4
About the Geometry Data Type	4
Enable Geometry Data Type in System Settings	4
Prepare Geometry Columns for Visualizing Maps	4
Add Geometry Columns to a Workbook	5
Visualize Geometry Column Data	7
Work with Data Layers and Layer Properties	8
Work with Background Map Layers and Layer Properties	13
Work with Toggle Off, Toggle On, and Legends	16
Work with the Selection Tool	18
Work with the Information Option	21
Spatial Calculations in Oracle Analytics	23
Calculations That Require Only One Geometry Column	24
How to Calculate the Area of a Geometry	24
How to Calculate the Length of a Geometry	25
How to Generate a Spatial Data Object Definition in Standard Text Format	26
Calculations That Require Two Geometry Columns	27
How to Calculate the Distance Between Two Geometry Data Types	27
How to Relate Two Geometry Data Types	28
How to Determine if Two Geometry Data Types Are Within a Specific Distance	29

Limitations	31
Performance Considerations	32
Appendix 1 – Oracle Spatial Calculations	33

Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates. This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Revision History

DATE	REVISION
November 2024	Initial Publication

Author: Gautam Pisharam

Contributing Authors: Philippe Lions, Anna Moat, Nick Fry

Summary

Geometry Data Type

Oracle Analytics natively supports the geometry data type which lets users visualize and work with geometry columns. Oracle Analytics now automatically recognizes geometry columns from Oracle databases and CSV files, so users can easily import and connect to spatial data. A geometry data type in Oracle Analytics can represent a wide range of spatial shapes, including points, polygons, and lines, enabling detailed and dynamic geographic visualizations. Geometry data types can be used for visualizing data on maps or as inputs for advanced spatial calculations, enabling detailed spatial analysis. This document provides information for content authors about using the geometry data type in their workbooks.

Business Value

Accelerate Map Design and Visualization

You can quickly visualize geographic information by dragging and dropping geometry columns onto a canvas where the shapes render automatically as a map visualization, without having to manually set up any map layers. Spatial data objects, whether sourced directly from an Oracle database or a CSV file, are effortlessly rendered in Oracle Analytics as maps, allowing you to quickly create dynamic spatial insights. This native feature boosts visual performance and ensures spatial data objects are rendered smoothly and responsively.

Optimize Map Layer Maintenance

In scenarios where geometry definitions frequently change—such as delivery routes, flood zones, or custom sales regions—using the geometry data type ensures that maps in Oracle Analytics are automatically updated with the latest shape information. There's no need to preconfigure or ensure a specific map layer exists in your Oracle Analytics environment, the geometry data automatically shows in a map visualization. There's also no need to manually update any data or map layers to reflect changes, as the maps dynamically adjust by querying the updated geometry data. This eliminates extra maintenance of data layers or map layers and ensures that maps always display the most current data.

Support Spatial Calculations

You can compute calculations to perform spatial measurements using the geometry data type. Spatial calculations enable you to calculate the area and length of geometry shapes, measure distances between two geographic data types (assuming proper joining is defined), determine spatial relationships, and perform condition-based calculations that return true or false. These calculations help you analyze geographic data more effectively, leading to powerful spatial data-driven decisions.

Visualize Geometry Columns in Oracle Analytics

About the Geometry Data Type

Geometry columns are special first-class objects in Oracle Analytics and can be used only for the following operations:

- To create map visualizations
- In custom calculations for spatial measurements and correlations (either on a canvas or in a data flow)
- As part of custom SQL sourcing (meaning that you can create a dataset using Spatial SQL defined through the Manual SQL query mode)

Consequently, note that many other operations are disabled or extremely limited when using a geometry column. Geometry columns cannot be used as filters in a workbook or as components (visualization definition grammar) to any visualizations other than maps. The geometry data type cannot be manually set by users on a data column, it must be recognized as a geometry data type when importing as a dataset.

Enable Geometry Data Type in System Settings

Your administrator must enable the geometry data type for your Oracle Analytics environment.

1. On the Home page, click the **Navigator**, and then click **Console**.
2. Click **System Settings**.
3. Click **Preview** and turn on **Enable Geometry Data Type**.

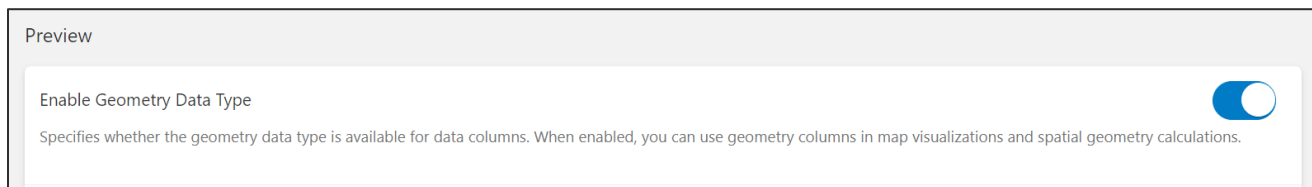


Figure 1 - Example shows how to enable Geometry Data Type in Preview in System Settings.

4. Click **Apply**.

Prepare Geometry Columns for Visualizing Maps

Once the geometry data type is enabled, you can prepare and connect to geometry data types in Oracle Analytics from either an Oracle database (using the Spatial Data Object Geometry data type (SDO.Geometry)) or a CSV file. The CSV source data must include geometry data in the Well-Known Text (WKT) format to be recognized by Oracle Analytics. For data sourcing from an Oracle database, this can be done via a direct connection or through a semantic model. The value stored in geometry columns appears as [GEOMETRY] and is useable only in map visualizations or custom calculations.

For direct connections, follow these steps:

1. On the Home page, click **Create** and then click **Dataset**.
2. In the Create Dataset dialog, select an already configured connection to an Oracle database which has the SDO.Geometry column or select a CSV file with a WKT column.

GEOM	STATE	ORDER_LINE_ID	ORDER_ID	ORDER_PRIORITY	CUSTOMER_ID	CUSTOMER_SEGMENT	PRODUCT_CATEGORY	PRODUCT_SUB_CATEGORY	PRODUCT_CONTAINER	SHIP_MODE	ORDER_DATE
[GEOMETRY]	Kansas	5611	85835	Low	2066	Corporate	Office Supplies	Binders and Binder Accessories	Small Box	Regular Air	10/19/2013 12:00:00.000 AM
[GEOMETRY]	North Carolina	9277	91437	High	3359	Home Office	Office Supplies	Binders and Binder Accessories	Small Box	Regular Air	10/28/2013 12:00:00.000 AM
[GEOMETRY]	Alabama	2079	90961	Low	785	Small Business	Furniture	Bookcases	Jumbo Box	Delivery Truck	01/05/2013 12:00:00.000 AM
[GEOMETRY]	Missouri	4711	87244	Low	1749	Home Office	Furniture	Office Furnishings	Medium Box	Regular Air	02/13/2013 12:00:00.000 AM
[GEOMETRY]	Maryland	3381	10435	Medium	1280	Corporate	Office Supplies	Pens & Art Supplies	Small Pack	Regular Air	08/10/2014 12:00:00.000 AM
[GEOMETRY]	Missouri	1973	87725	Low	744	Corporate	Office Supplies	Binders and Binder Accessories	Small Box	Regular Air	01/27/2013 12:00:00.000 AM
[GEOMETRY]	Kentucky	5928	89465	Medium	2178	Small Business	Office Supplies	Pens & Art Supplies	Wrap Bag	Regular Air	07/27/2013 12:00:00.000 AM
[GEOMETRY]	Idaho	3610	89996	High	1365	Consumer	Furniture	Tables	Jumbo Box	Delivery Truck	07/03/2014 12:00:00.000 AM
[GEOMETRY]	Kansas	2117	90048	Medium	803	Small Business	Technology	Telephones and Communication	Small Box	Regular Air	10/29/2013 12:00:00.000 AM
[GEOMETRY]	Georgia	3840	89077	Low	1442	Corporate	Furniture	Office Furnishings	Small Box	Express Air	07/30/2013 12:00:00.000 AM
[GEOMETRY]	Michigan	5446	86878	Medium	2012	Corporate	Office Supplies	Binders and Binder Accessories	Small Box	Regular Air	08/22/2014 12:00:00.000 AM
[GEOMETRY]	Georgia	8545	86221	High	3096	Consumer	Office Supplies	Appliances	Small Box	Express Air	05/24/2013 12:00:00.000 AM
[GEOMETRY]	Ohio	3069	87523	Critical	1171	Consumer	Furniture	Chairs & Chairmats	Medium Box	Regular Air	07/25/2014 12:00:00.000 AM
[GEOMETRY]	Louisiana	75	87652	Not Specified	27	Corporate	Furniture	Office Furnishings	Small Box	Regular Air	11/02/2013 12:00:00.000 AM
[GEOMETRY]	Georgia	1934	89345	High	717	Corporate	Technology	Computer Peripherals	Small Pack	Regular Air	02/11/2014 12:00:00.000 AM

Figure 2 - Example shows the Sample Order Lines dataset with the geometry column GEOM. The data is coming from an Oracle database.

WKT	STATE	ORDER_LINE_ID	ORDER_ID	ORDER_PRIORITY	CUSTOMER_ID	CUSTOMER_SEGMENT	PRODUCT_CATEGORY	PRODUCT_SUB_CATEGORY	PRODUCT_CONTAINER	SHIP_MODE	ORDER_DATE	SHIP_DATE
[GEOMETRY]	North Carolina	9325	91476	Low	3372	Home Office	Technology	Computer Peripherals	Small Box	Regular Air	25-Jul-14	28-Jul-14
[GEOMETRY]	Kansas	25	86837	Critical	15	Small Business	Office Supplies	Envelopes	Small Box	Regular Air	08-Apr-13	10-Apr-13
[GEOMETRY]	Illinois	1762	88475	Medium	669	Home Office	Office Supplies	Paper	Small Box	Regular Air	20-Sep-13	22-Sep-13
[GEOMETRY]	Puerto Rico	4397	89065	Not Specified	1657	Small Business	Office Supplies	Paper	Small Box	Regular Air	13-Jun-14	16-Jun-14
[GEOMETRY]	Alabama	2995	88102	Low	1132	Home Office	Office Supplies	Binders and Binder Accessories	Small Box	Regular Air	15-Feb-13	23-Feb-13
[GEOMETRY]	Kansas	5555	88220	High	2049	Corporate	Technology	Computer Peripherals	Small Pack	Regular Air	21-Dec-13	24-Dec-13
[GEOMETRY]	Texas	3173	89124	Critical	1208	Home Office	Office Supplies	Paper	Small Box	Regular Air	07-Apr-14	09-Apr-14
[GEOMETRY]	Iowa	241	87178	Medium	92	Corporate	Office Supplies	Scissors, Rulers and Trimmers	Small Pack	Express Air	07-Dec-13	10-Dec-13
[GEOMETRY]	Kansas	272	87306	Not Specified	97	Home Office	Technology	Telephones and Communication	Small Box	Regular Air	03-Nov-13	05-Nov-13
[GEOMETRY]	Oklahoma	7914	87632	Critical	2883	Consumer	Office Supplies	Paper	Small Box	Regular Air	09-May-13	10-May-13

Figure 3 - Example shows the Sample Order Lines dataset with the geometry column WKT. The data is coming from a CSV file.

Joins between geometry columns from two different datasets aren't supported. For use cases where you need to relate between the geometry columns, you must create and establish a distinct join using existing columns.

Add Geometry Columns to a Workbook

To start visualizing shapes from a geometry column, you need a dataset that holds geometry data, see [Prepare Geometry Columns for Visualizing Maps](#). A geometry column can be consumed from a dataset created based on a SDO.Geometry data type from an Oracle database or a Well-Known Text (WKT) format from a CSV file.

1. On the Home page, click **Create** and click **Workbook**.
2. In the Add Data dialog, click a dataset with a geometry column, then click **Add to Workbook**.

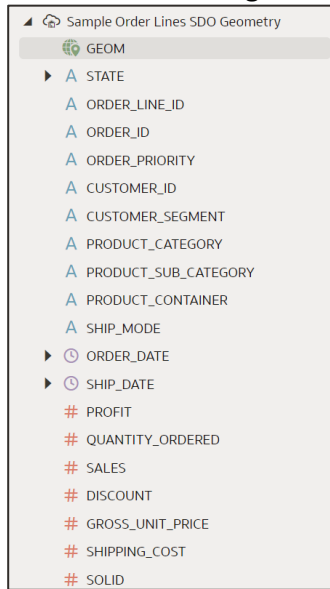


Figure 4 - Example of a geometry data type column with the name GEOM in the Sample Order Lines dataset created from an Oracle database.

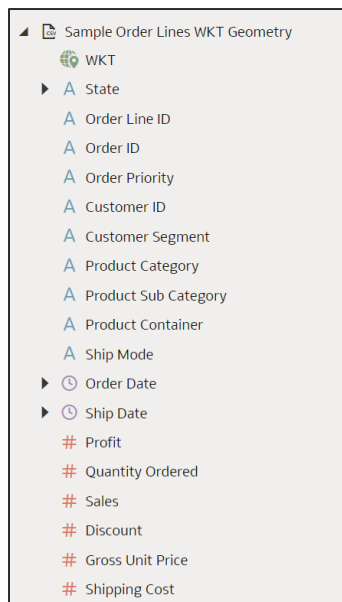




Figure 5 - Example of a geometry data type column with the name WKT in the Sample Order Lines dataset created from a CSV file.

3. Click **Save**.

Once the dataset is added to your workbook, you can simply drag and drop the geometry column  onto the Visualize canvas to start creating visualizations, as described in the next section.

Visualize Geometry Column Data

In the Data panel, locate the geometry column  and drag it to the Visualize canvas to start building a visualization.

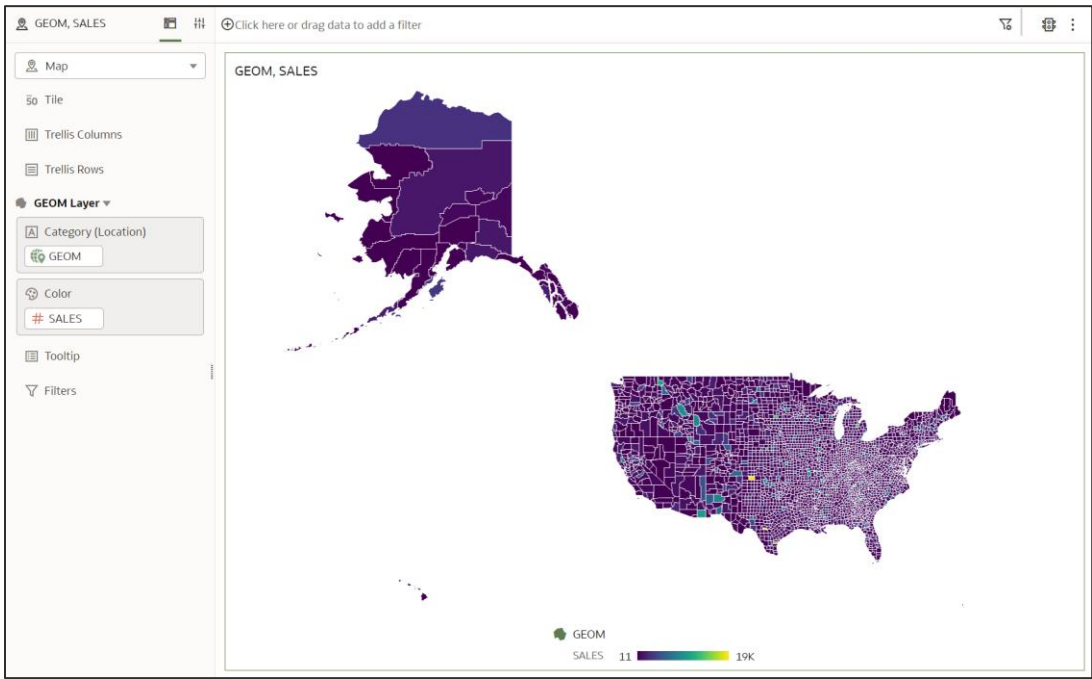


Figure 6 - Example of a geometry data type column from an Oracle database visualizing the shape of US Counties with the metric Sales for Color.

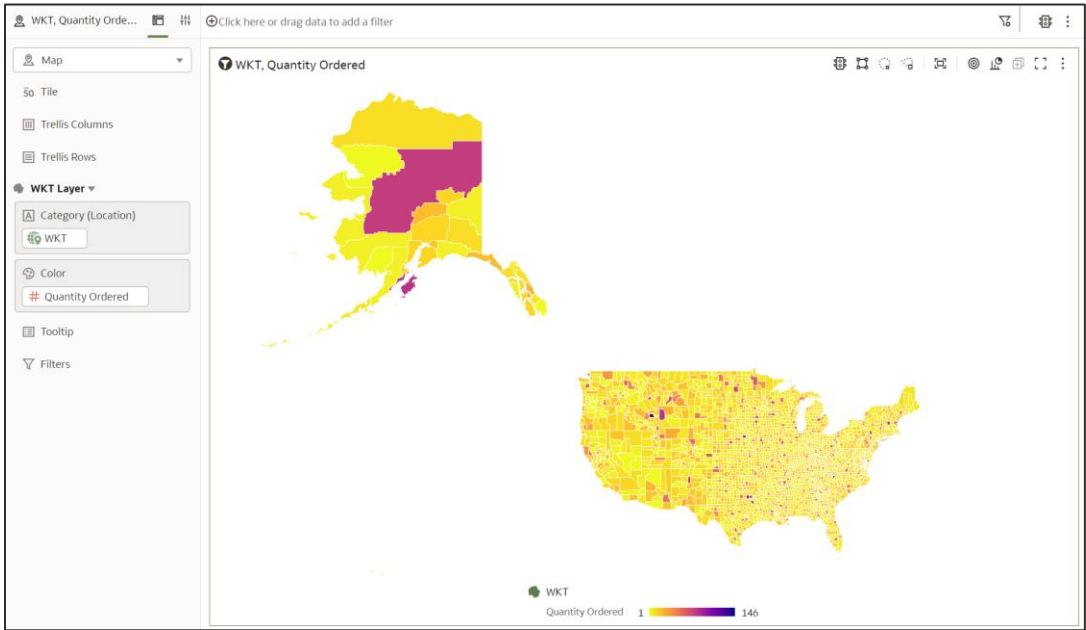


Figure 7 - Example of a geometry data type WKT column from a CSV file visualizing the shape of US Counties with the metric Quantity Ordered for Color.

By default, the map visualization renders without any background map layer. If required, you can overlay the map shape on a background map layer.

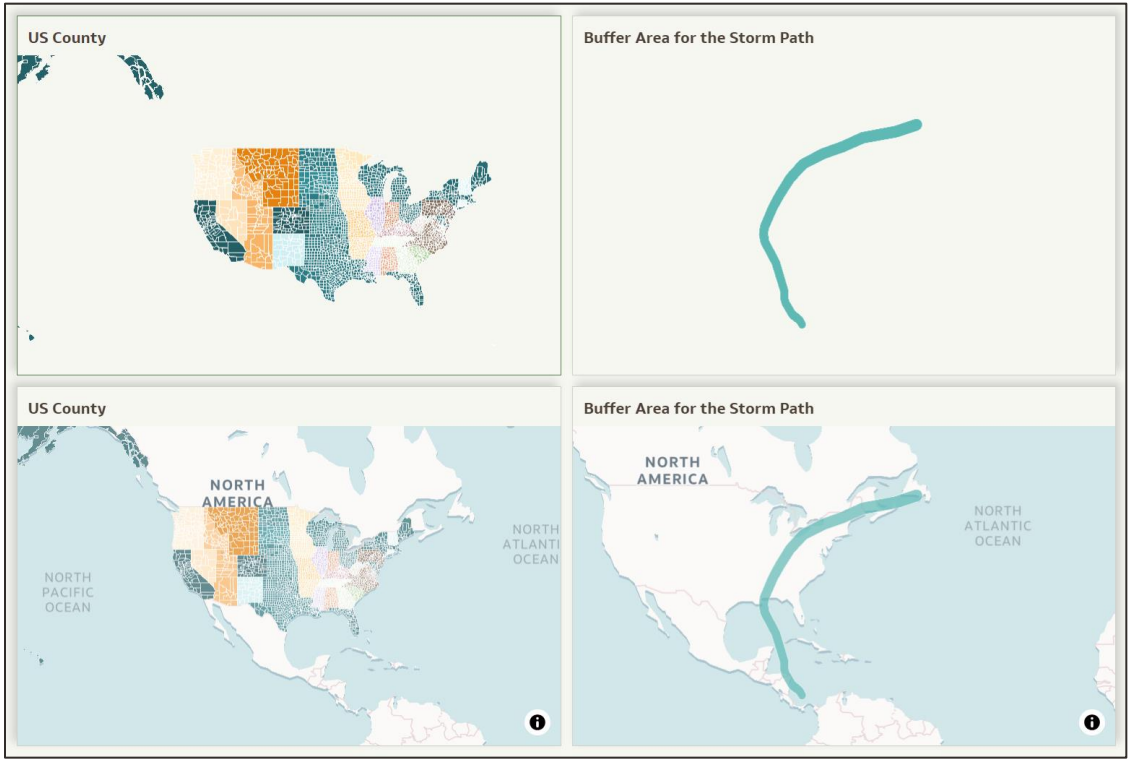


Figure 8 - Example of US Counties and the buffer area from a storm path visualized with no background layer in the top row of the figure, and with the Oracle DataViz map background layer in the bottom row of the figure.

Work with Data Layers and Layer Properties

You can overlay multiple data layers with geometry columns.

1. In the Grammar pane, click **Layer Options** for an existing layer, click **Add Layer**, and then select **Add Data Layer**.

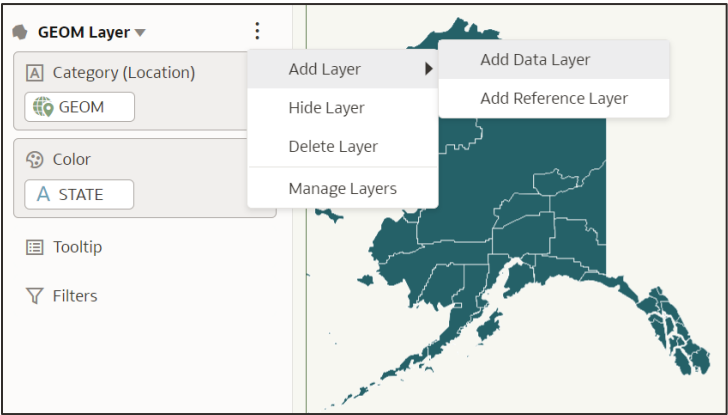


Figure 9 - Example of adding a new data layer for the map visualization.

Note that geometry columns can be used only in a data layer. If you need to use them in a reference layer, you must generate a GeoJSON file to support that reference layer.

2. In the Data pane, locate the dataset from which the new layer needs to be overlaid and drag the geometry column to the new data layer.

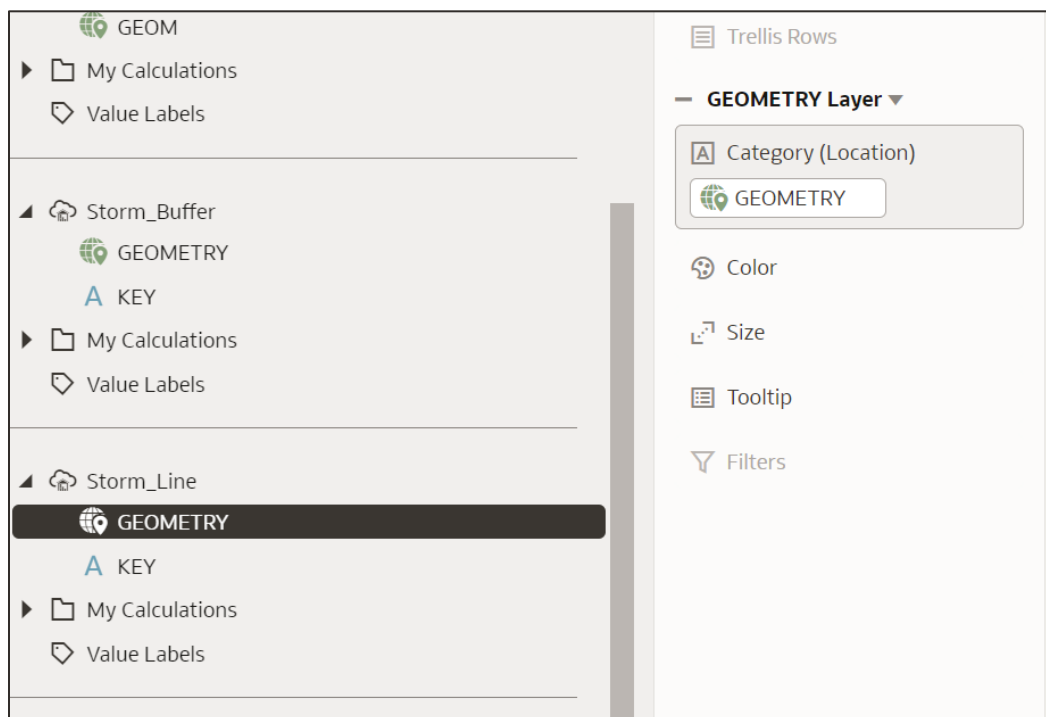


Figure 10 - Example of adding a new geometry column over an existing data layer. In this example, the geometry of a storm line is added on top of an existing layer which has the shapes of US Counties.

This visualizes the new data layer over the existing one.

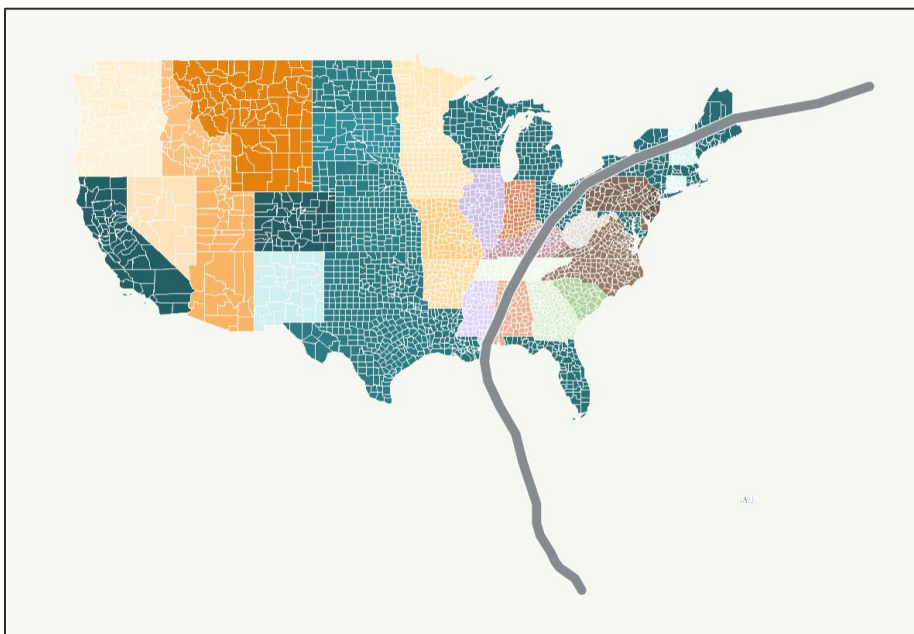


Figure 11 - Example shows how the Storm Line layer is added on top of US Counties.

The geometry data type supports existing data layer properties.

If your geometry is a line or a point, you can adjust the size of the geometry.

1. In the Properties pane, click **Layers**, and then locate and expand the layer you want to adjust the size for.
2. Click the **Size** field and use the slider to adjust the size.

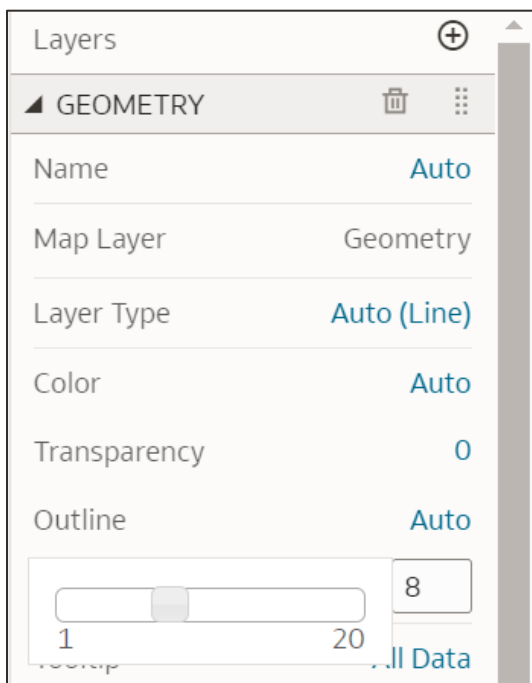


Figure 12 - Example shows how the Storm Line size can be adjusted.

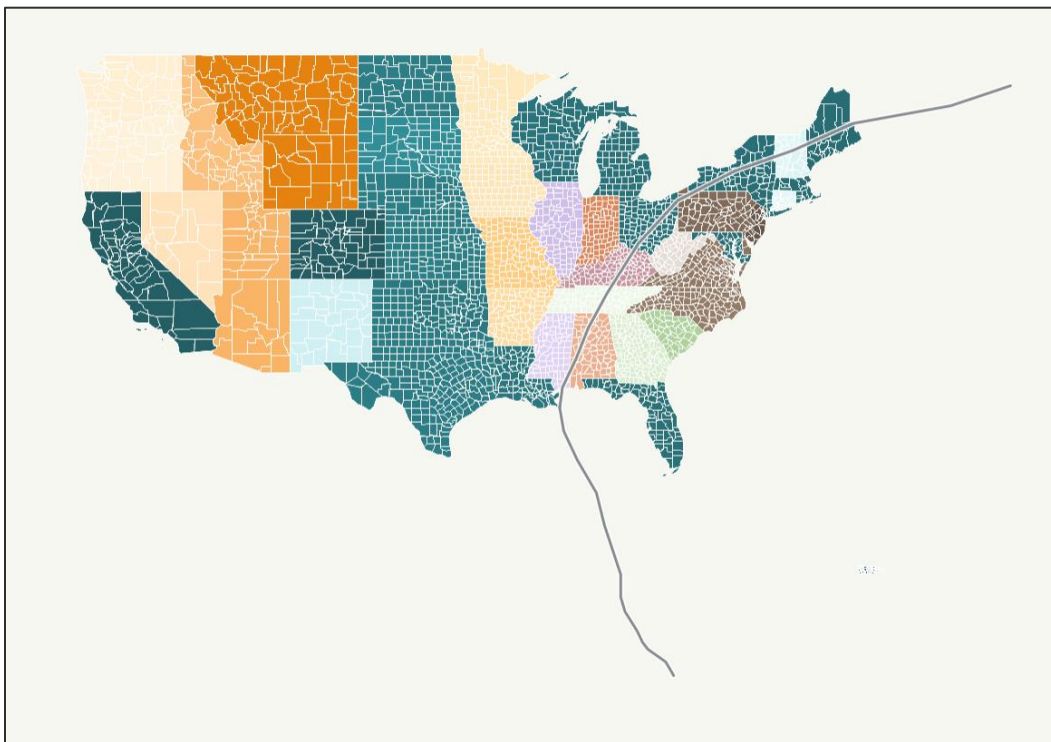


Figure 13 - Example shows how the Storm Line size is now adjusted in the visualization. The point size now has a numerical value of 2.

You can change the colors of the geometry shapes if no default colors are specified in the Grammar pane.

1. Right-click on the map visualization and select **Layer**.
2. Select a color, and then click **OK**.

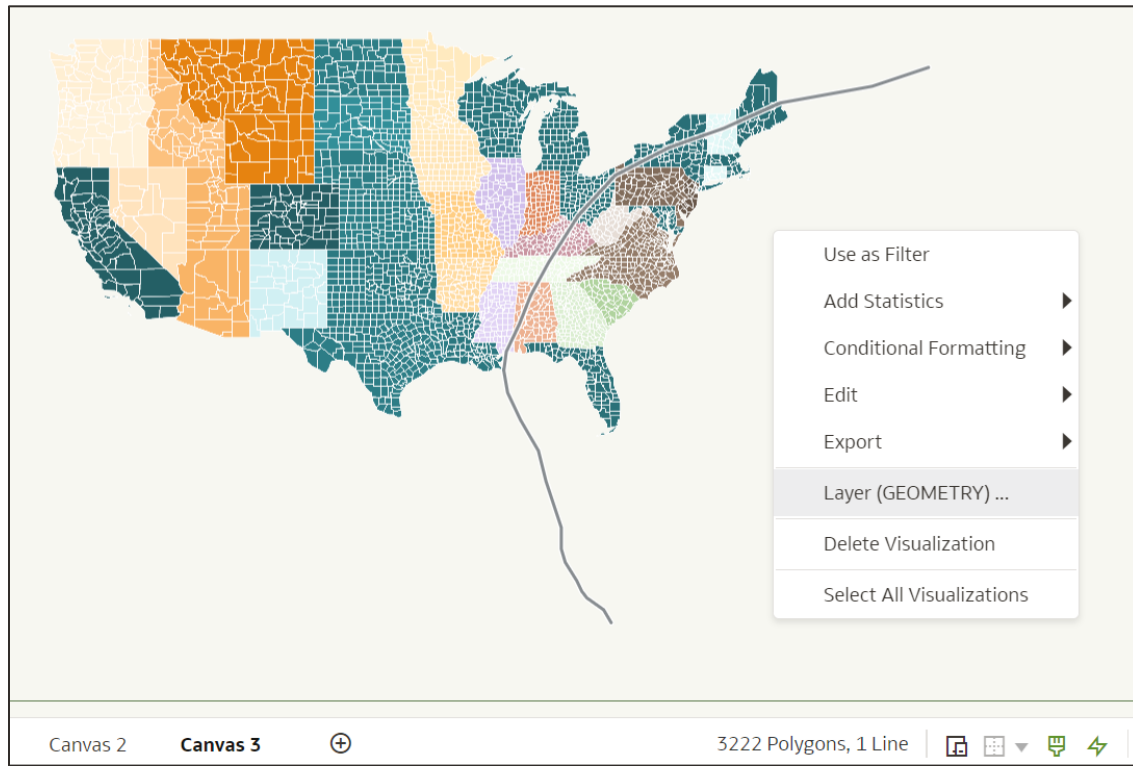


Figure 14 - Example shows how to adjust the color for the storm line overlaid on top of US Counties.

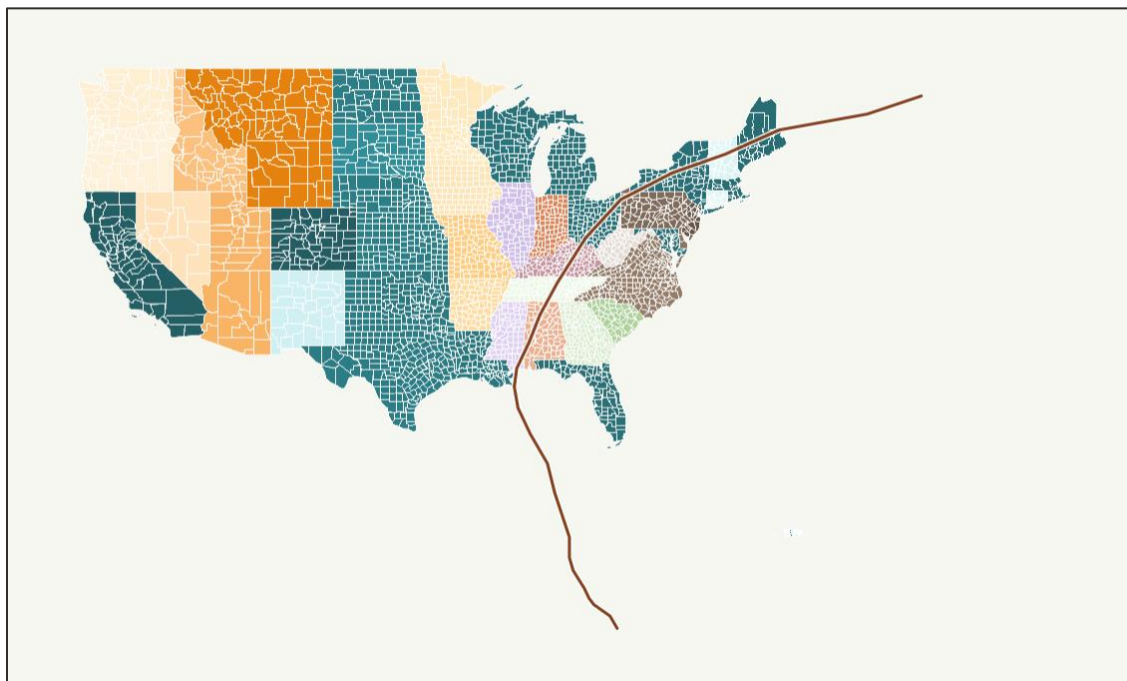


Figure 15 - Example shows the storm line is now red.

You can order the layers if there are more layers to be overlaid.

1. In the Grammar pane, click **Layer Options** for an existing layer, and then click **Order Layer**.
2. Click **Send Backward**. The new layer is sent backwards behind the previous layer, and the previous layer moves in front of the latest layer added.

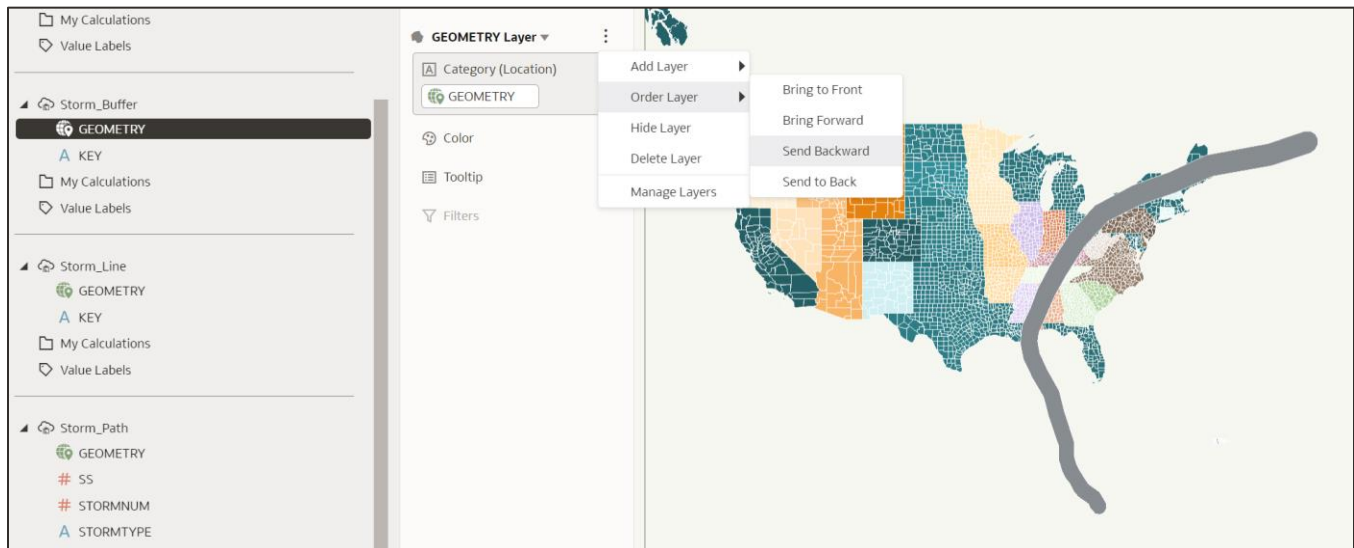


Figure 16 - Example shows how a new layer for Storm Buffer is added on top of an existing layer which has the geometry column from Storm Line. Since it's overlaid on top of an existing layer, the storm line is no longer visible.

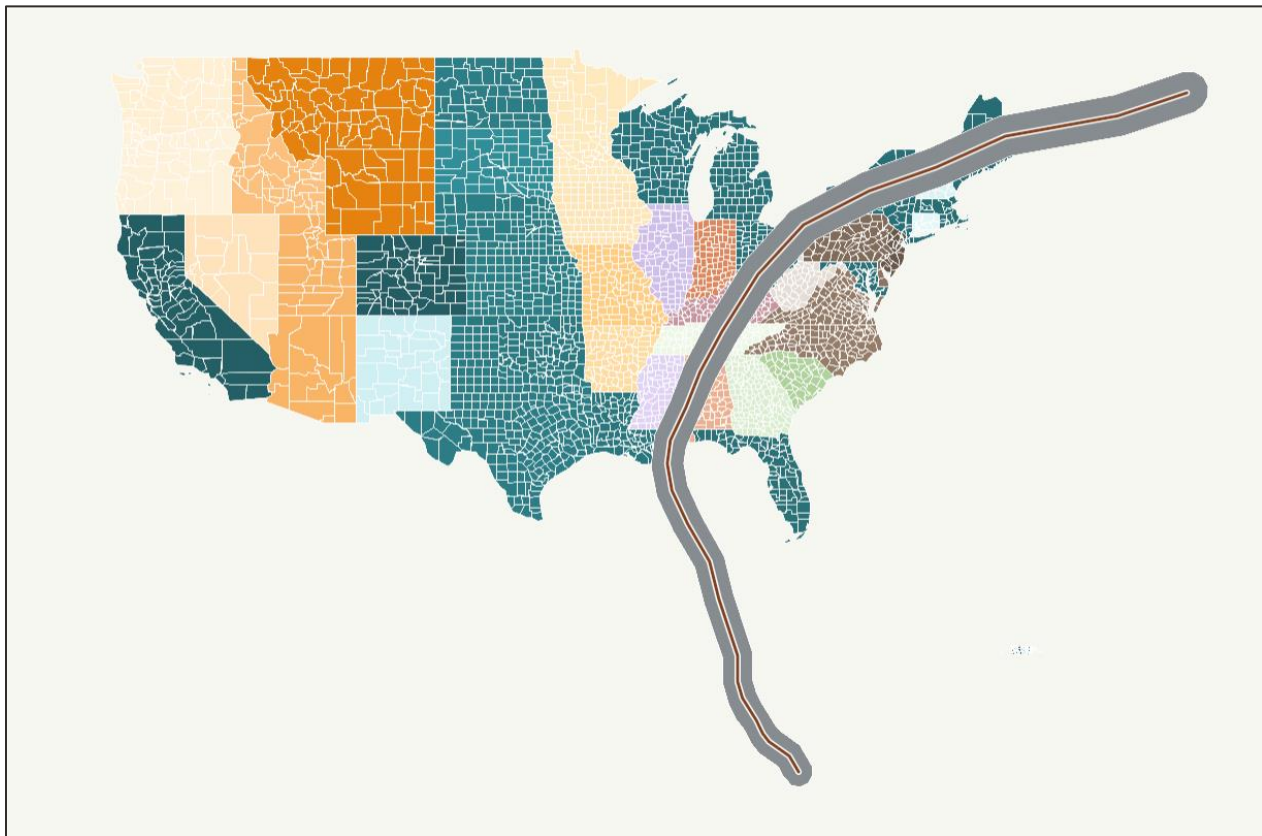


Figure 17 - Example shows the map layers for US Counties, Storm Line, and Storm Buffer in the right order.

Work with Background Map Layers and Layer Properties

You can apply a new background map layer for all the data layers.

1. In the Properties pane, click **Map**.
2. Click the **Background Map** field and select a layer.

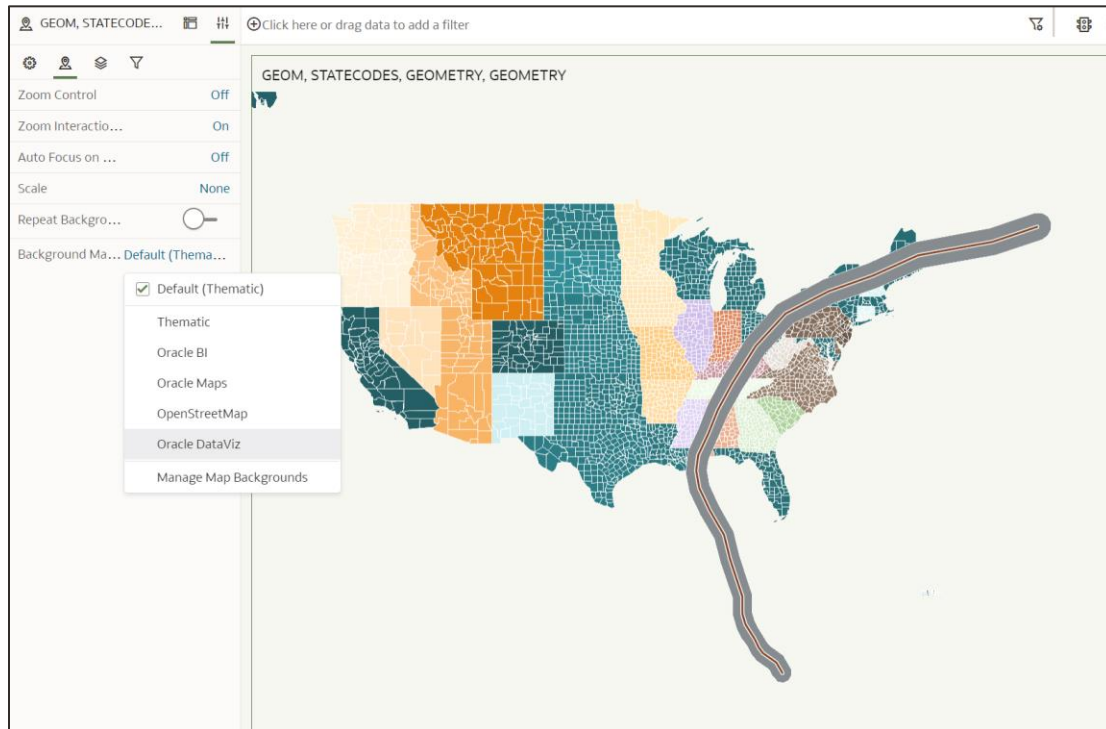


Figure 18 - Example shows how to apply a background map layer. In this example Oracle DataViz is selected.

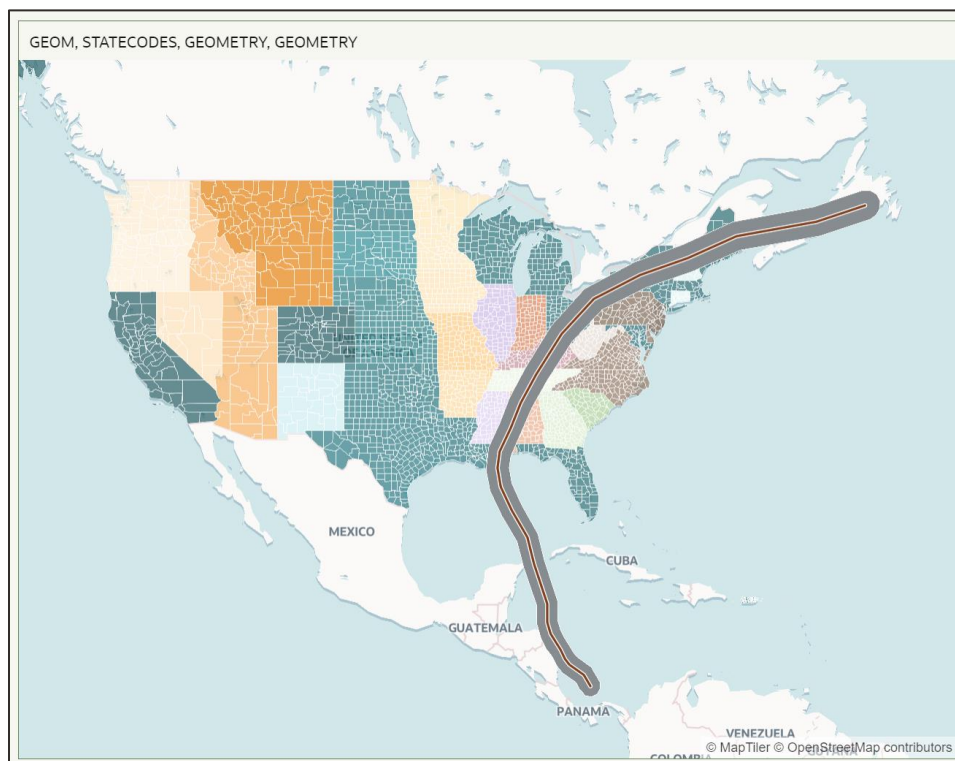


Figure 19 - Example shows the Oracle DataViz background map layer applied to the visualization.

You can still make changes to the layers by modifying the properties of each data layer.

1. In the Properties pane, click **Layers**, and expand the layer you want to modify.
2. Click the **Color** field and select **Custom**.
3. Click the **Fill Color** field and choose a color.

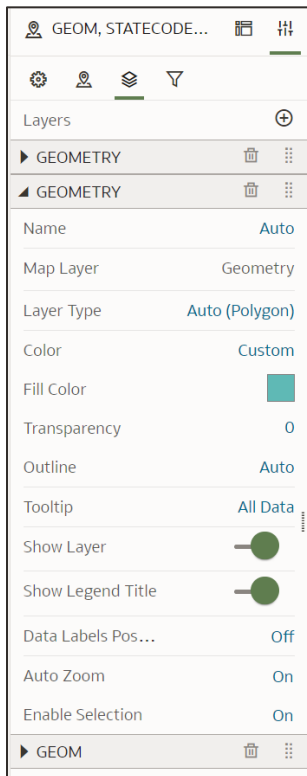


Figure 20 - Example shows how to apply a custom color to a data layer in the Properties pane for the map visualization.

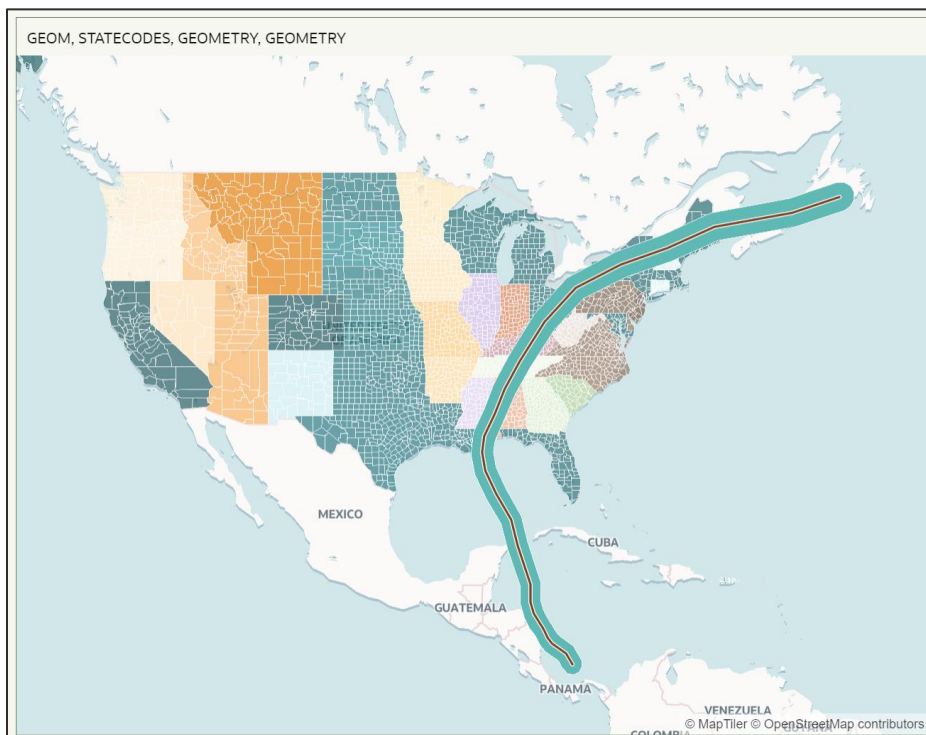


Figure 21 - Example shows the Storm Buffer layer with the new color.

You can see the statistics on how many shapes are used in the visualization at the bottom of the Visualize page.

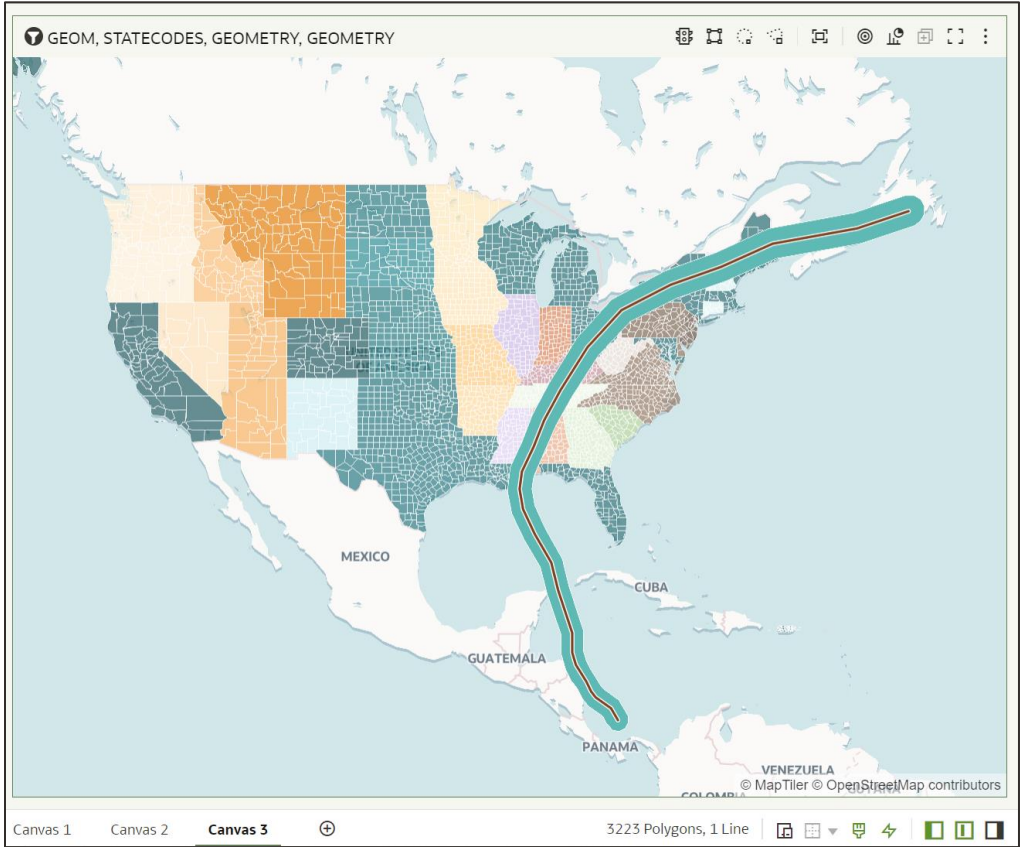


Figure 22 - Example shows that there are 3,223 Polygons and one Line used as part of all the data layers in the map.

As a best practice, name each data layer with a unique name.

1. In the Properties pane, click **Layers**.
2. Click the **Name** field, choose **Custom**, and provide a name for the layer.

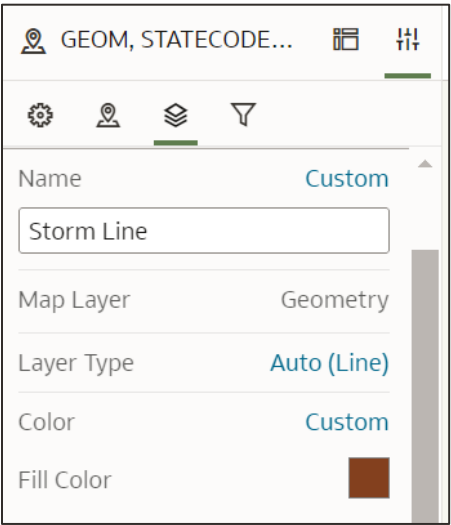


Figure 23 - Example shows the custom name Storm Line for the layer with the geometry column.

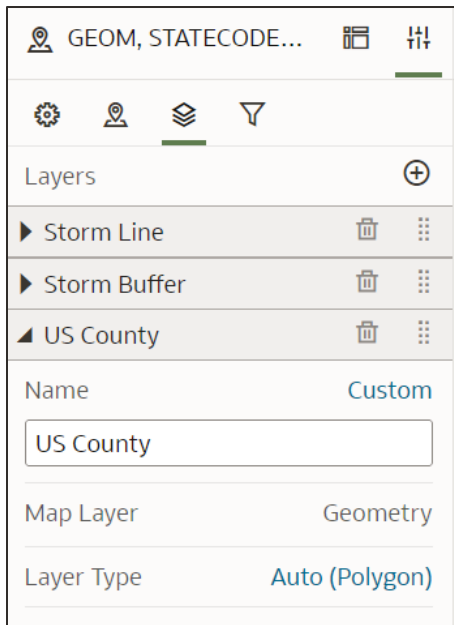


Figure 24 - Example shows the custom names given for all the data layers.

Work with Toggle Off, Toggle On, and Legends

You can toggle the data layers on and off when the map legend is enabled.

1. In the Properties pane, click **General**, and check the **Legend Position** field. The legend is displayed by default.

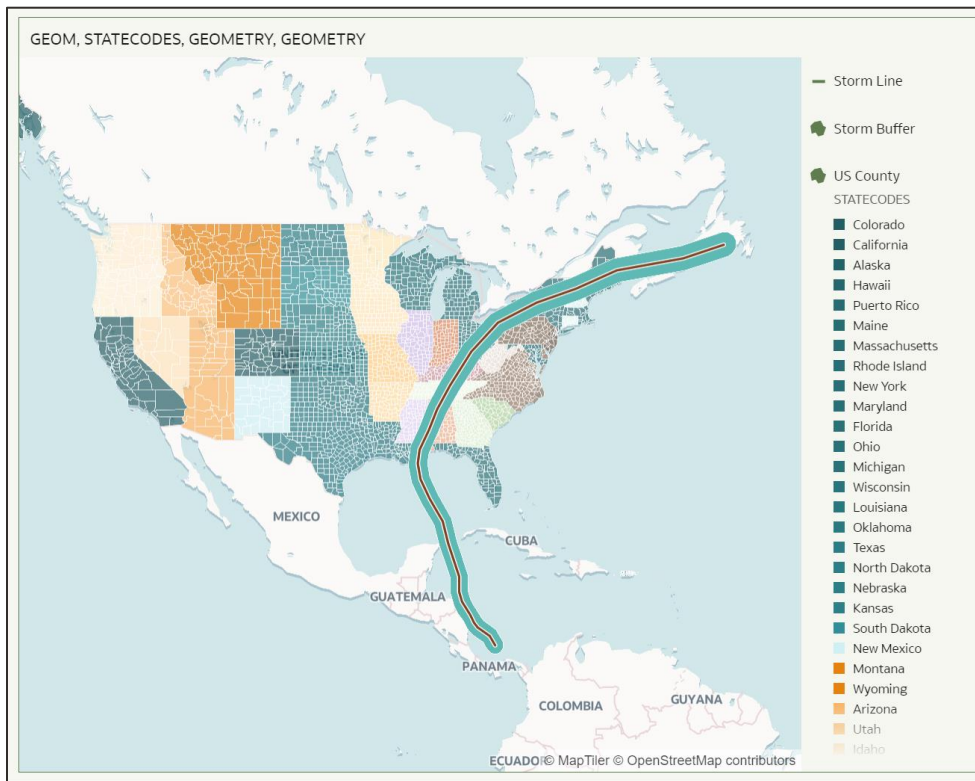


Figure 25 - Example shows the legend and layer names for the map visualization.

2. In the map legend, click a data layer to toggle it on or off.

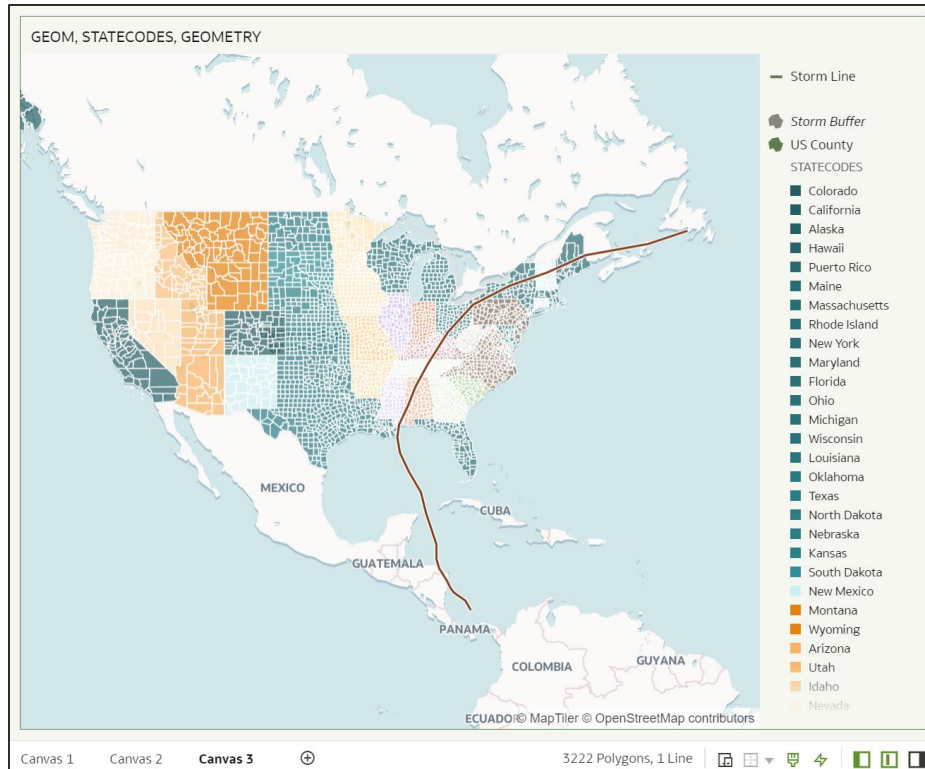


Figure 26 - Example shows the Storm Buffer data layer toggled off, and Storm Line with US County data layers toggled on. Statistics now shows 3,222 Polygons and one Line. One Polygon for the Storm Buffer is now excluded from the visualization.

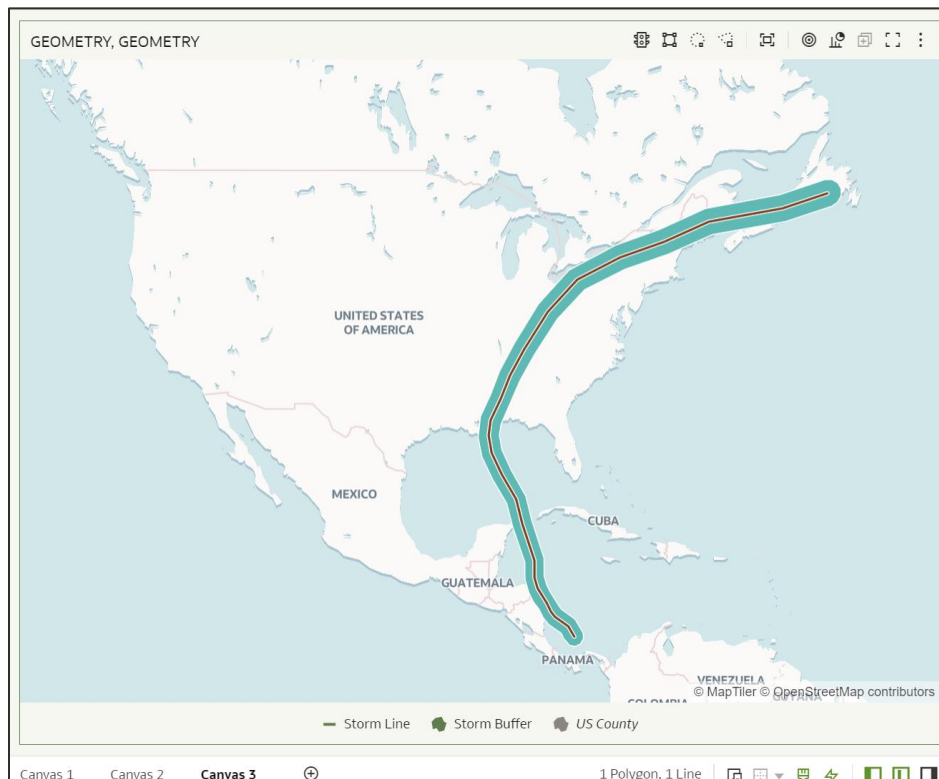


Figure 27 - Example shows the US County data layer toggled off, and Storm Line with Storm Buffer data layers toggled on. Statistics now shows one Polygon and one Line. 3,222 Polygons for the US County data layer are now excluded from the visualization.

Work with the Selection Tool

You can use the Selection tool to interact with the map layers. To select content based on geometry columns, you must have an attribute ID column (for example the name of the shape, or its ID) added to the map visualization. For example, only once an attribute is added as a Color, Tooltip, or Shape, are the Keep Selected and Remove Selected options available.

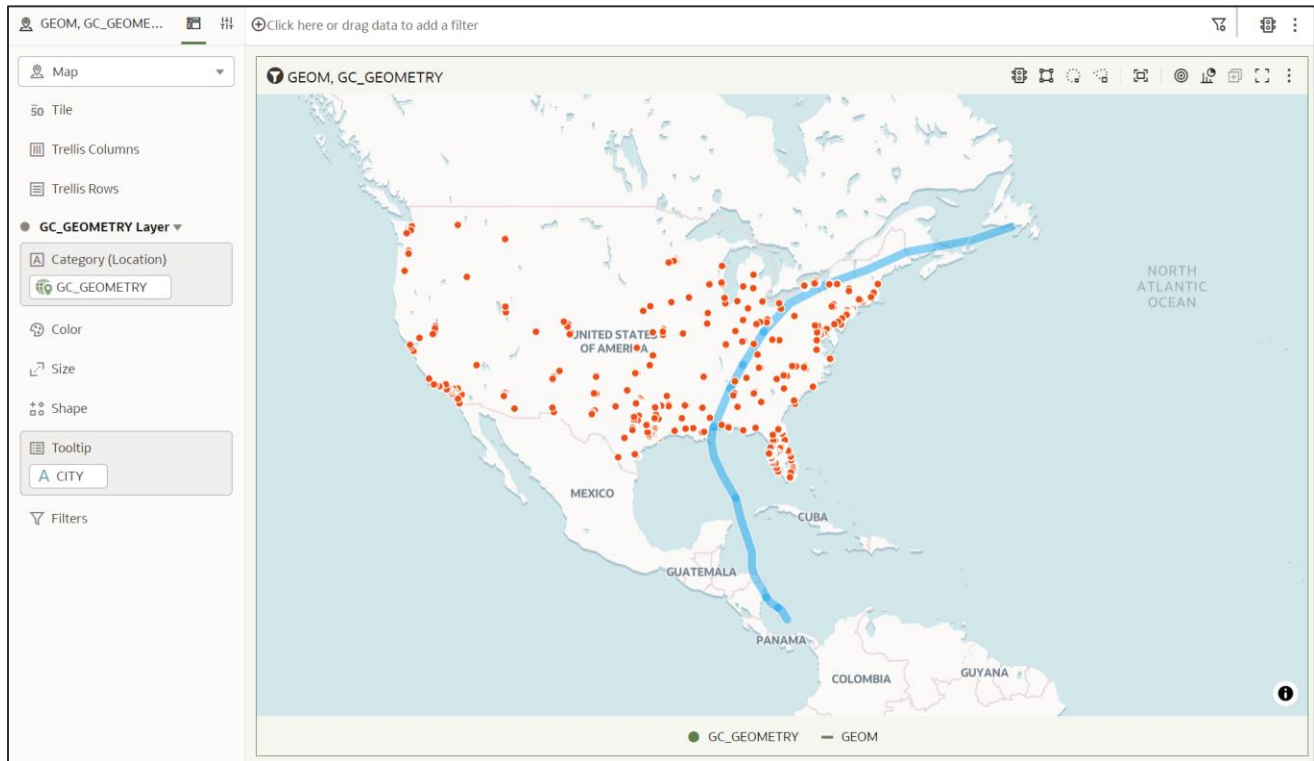



Figure 28 – Example shows the Storm Line layer and a layer with assets in different cities. City as an attribute column is used as a tooltip here to enable Keep Selected and Remove Selected.

1. To apply a selection to a new area, hover over the top right of the map visualization and locate the options for selection: Rectangle, Radial, and Polygon .
2. Click an option to select it, then drag and draw a custom selection area on top of the data layer.

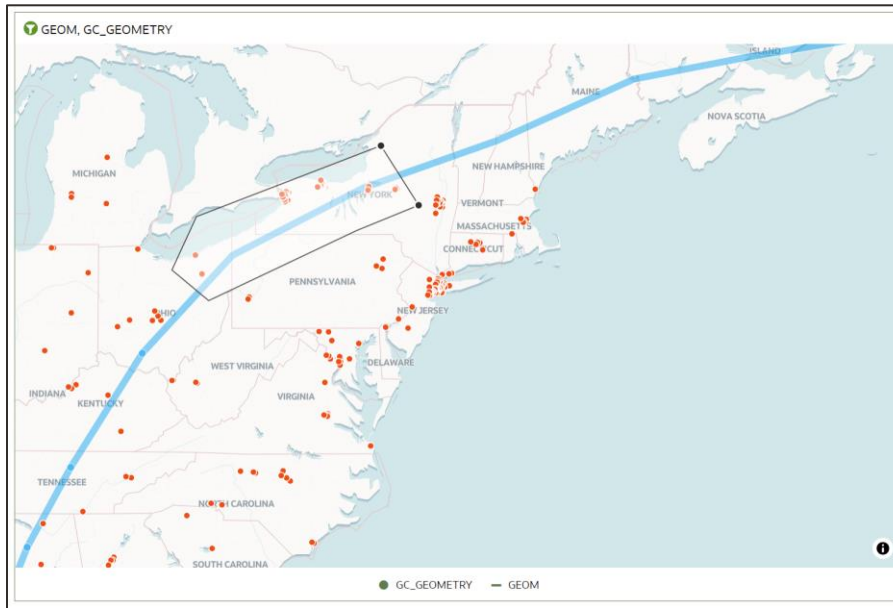


Figure 29 – Example shows how a Polygon selection is used and enables users to draw and select specific attributes.

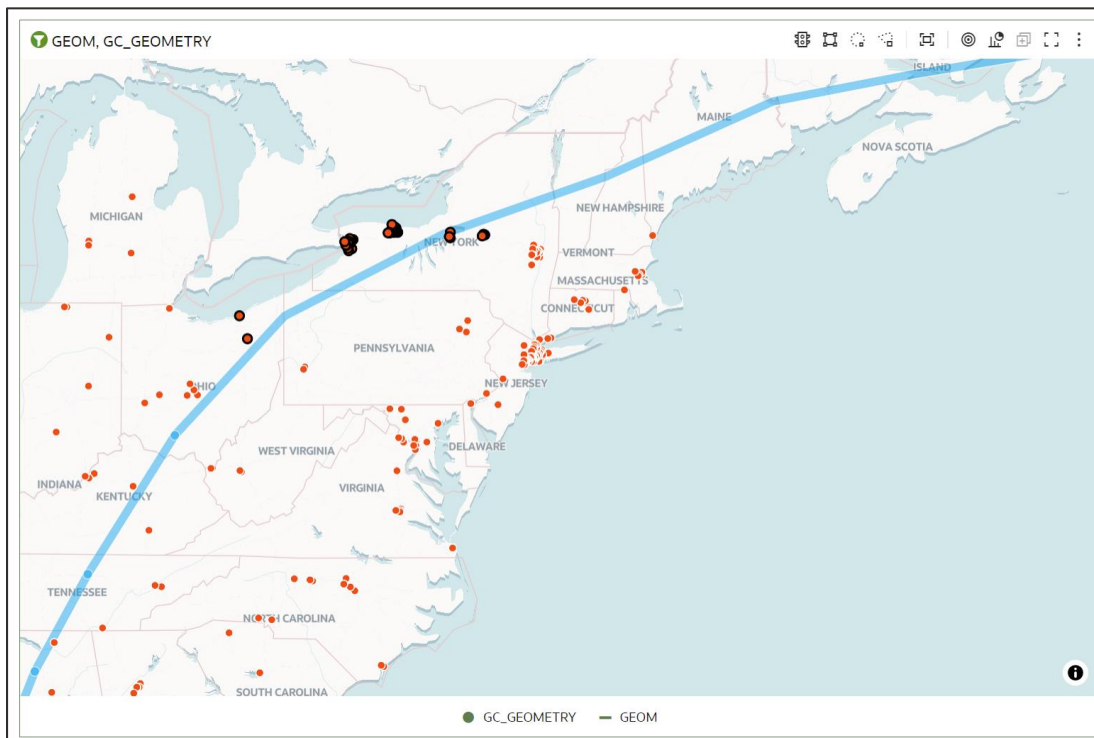


Figure 30 – Example shows how selected locations are highlighted once the polygon drawing is released.

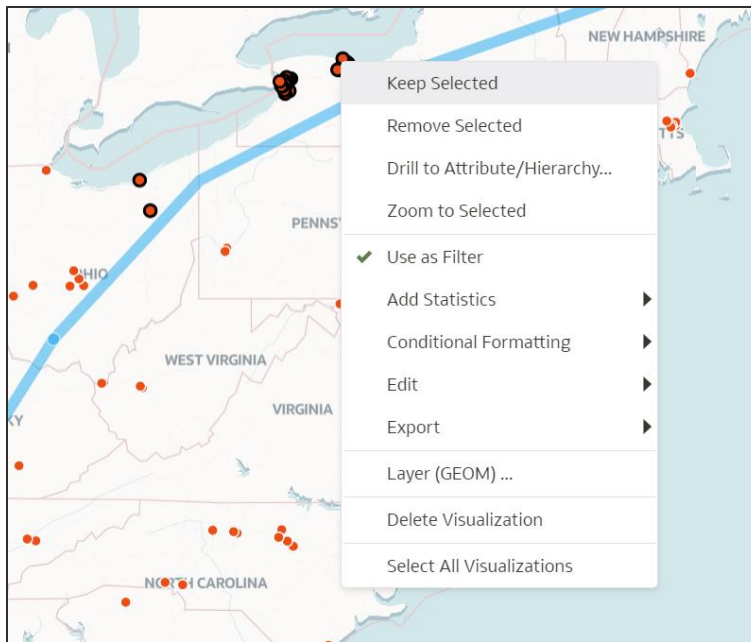


Figure 31 – Example shows how users can interact with the custom selection by either keeping the selection or removing the selection.

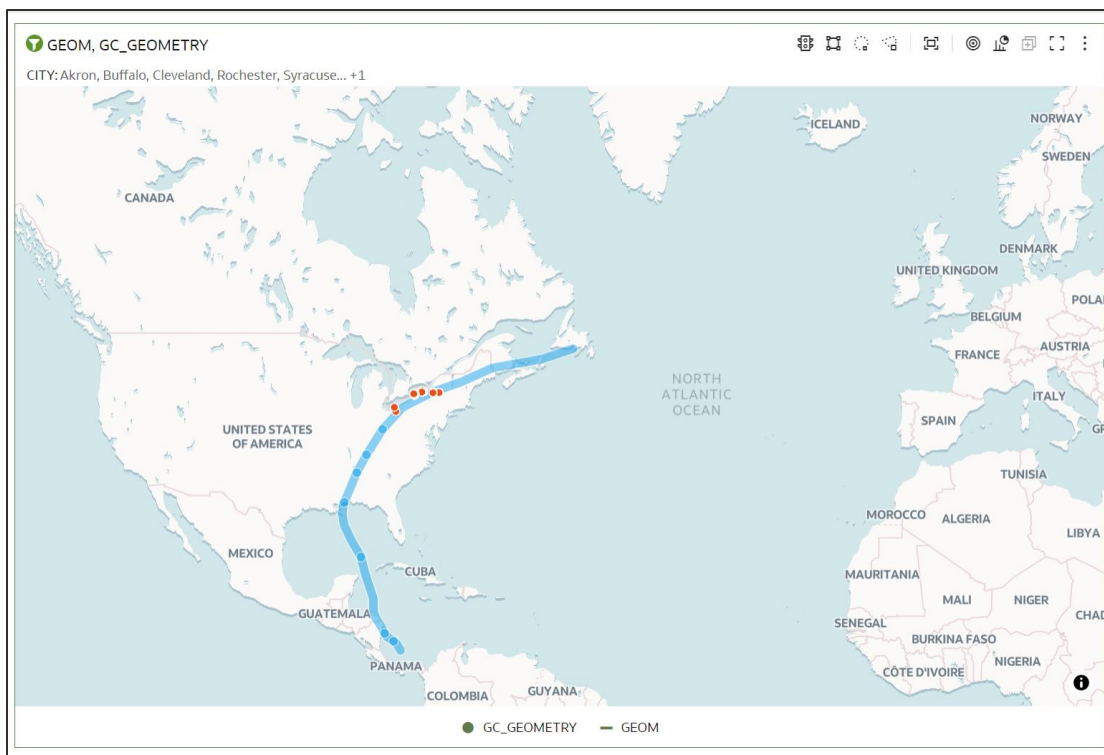


Figure 32 – Example shows how the option Keep Selected keeps only the filtered members.

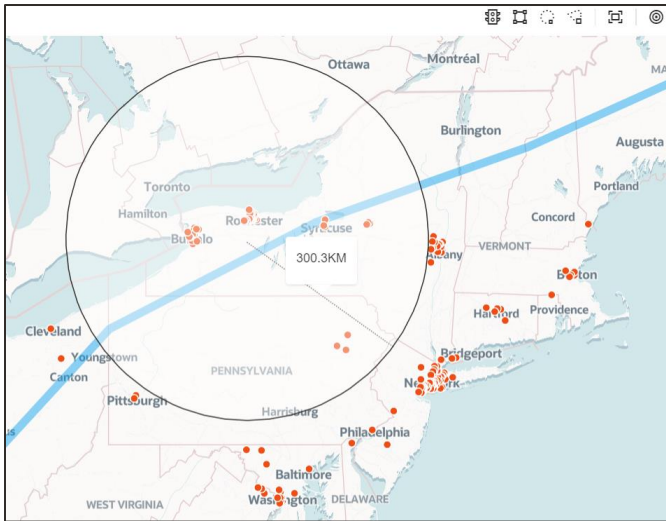


Figure 33 – Example shows the Radial selection.

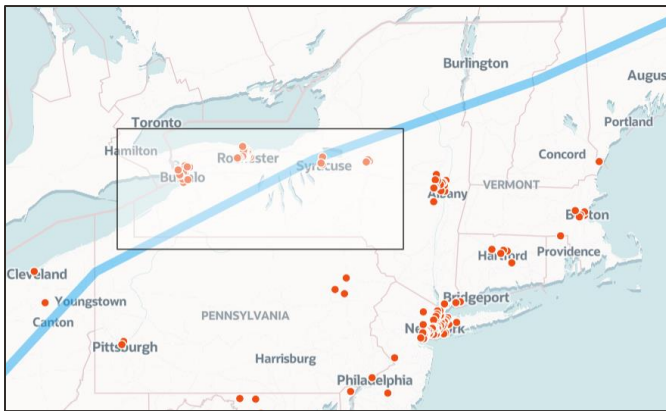




Figure 34 – Example shows the Rectangle selection.

Work with the Information Option

The Information option for a map visualization allows you to get information about geometry shapes that may not be displayed on the map. For example, shapes or polygons that have a disk size above 128 KB (per record) aren't displayed in the map visualization. In this case, the Information icon  appears towards the bottom right of the map visualization. When you click the icon, it turns green  and turns on the Information option near the map visualization title. Then, when you hover over this option, you can see which shapes didn't render as part of the layer. The information displayed is more detailed if the attribute ID column is part of the map visualization (as a tooltip, color, or shape).

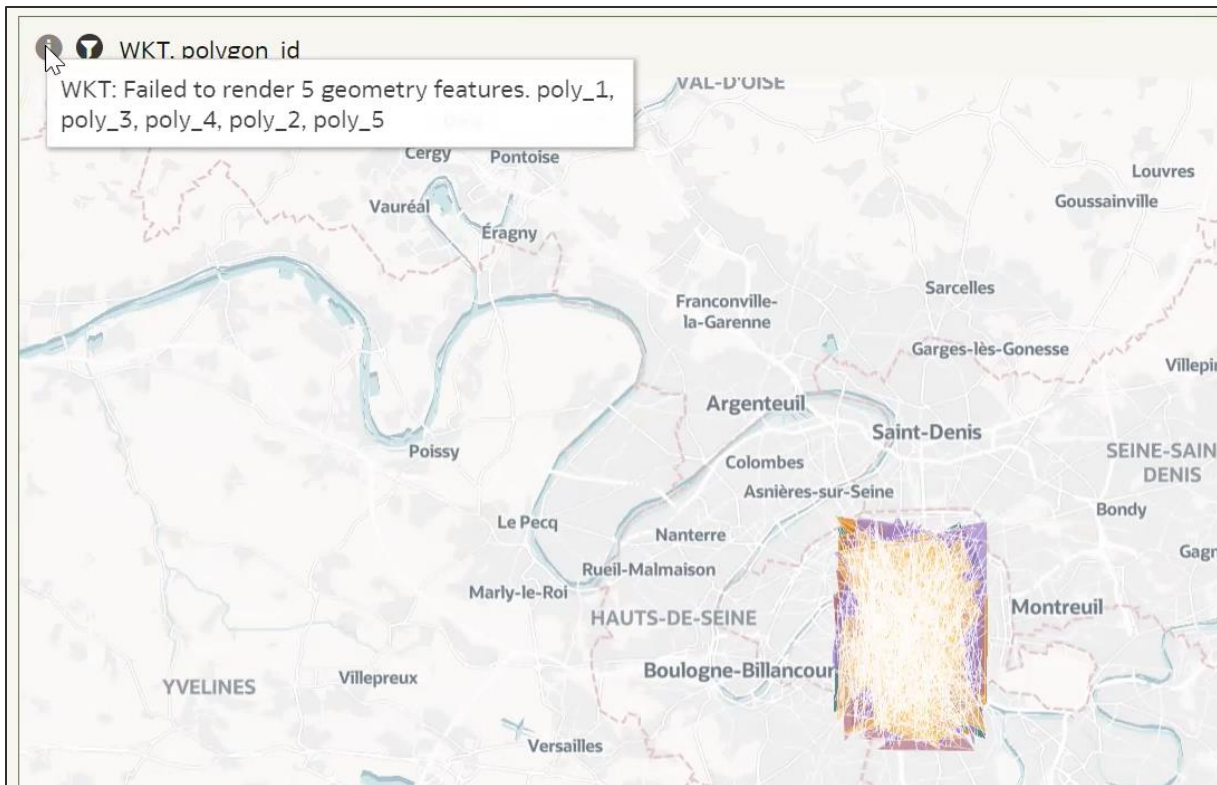


Figure 35 - Example shows how the Information option provides more information on shapes that have issues or that didn't render.

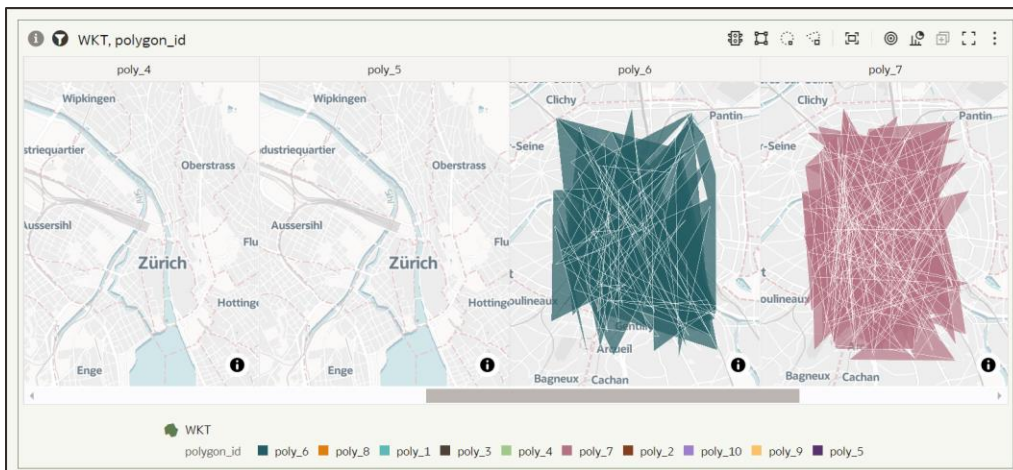


Figure 36, Example where the polygons are shown in Trellis Column, and polygons poly_4 and poly_5 aren't displayed due to shape size beyond 128 KB, whereas poly_6 and poly_7 are displayed since the shape size is within 128 KB.

Spatial Calculations in Oracle Analytics

You can create custom calculations to perform spatial measurements using geometry data types. Spatial calculations enable you to measure the area and length of geometry shapes, measure distances between two geographic data types, determine spatial relationships, and perform condition-based calculations that return true or false. These calculations help you analyze geographic data more effectively, leading to spatial data-driven decisions. Cartesian joins may be required for spatial calculations like `GeometryDistance`, `GeometryLength`, and `GeometryRelate`. The geometry calculations for area and length are computed in meters, ensuring accurate measurements for shapes based on their spatial coordinates.

1. In the Data pane, click **Create Calculation** .

You can also right-click the My Calculations folder and click **Create Calculation**.

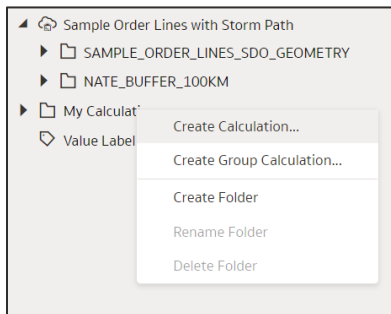


Figure 37 – Example showing the Create Calculation option from the My Calculations folder.

2. In the Create Calculation dialog, expand **Spatial** under the functions area and select the calculation you need.

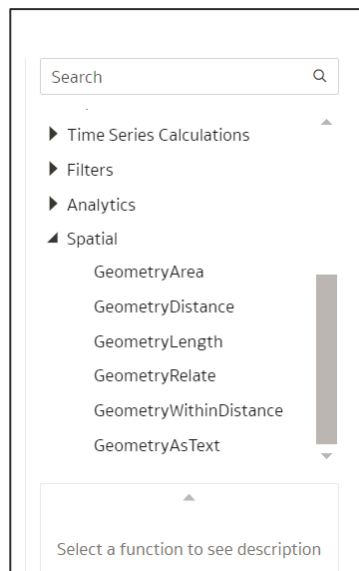


Figure 38 – Example showing the different spatial calculations available within Oracle Analytics.

Or you can type `geometry` in the calculation area and Oracle Analytics suggests calculations for you.

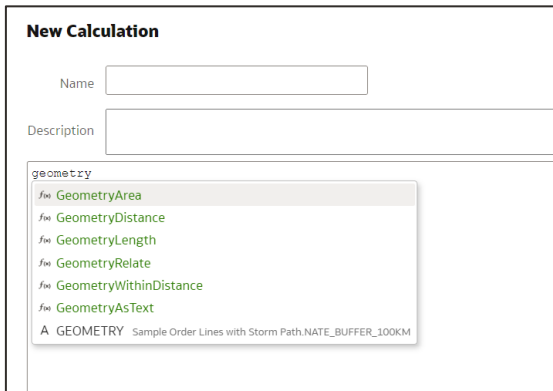


Figure 39 – Example showing how to select the desired calculation in the calculation area.

There are two types of spatial calculations available in Oracle Analytics:

- Calculations that require only one geometry column as an argument
- Calculations that require two geometry columns as arguments to compute the relationship

Calculations That Require Only One Geometry Column

How to Calculate the Area of a Geometry

The function `GeometryArea()` is used to calculate the area that a shape occupies.

Usage: `GeometryArea(geometry_column)`

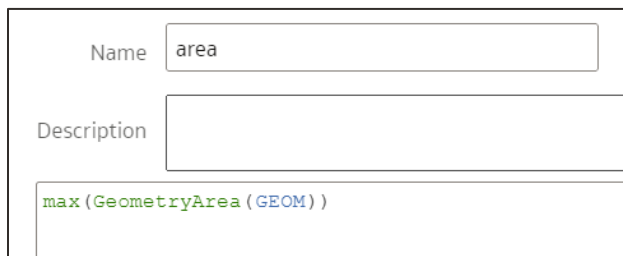


Figure 40 – Example showing an aggregate function `max()` applied on the `GeometryArea()` calculation.

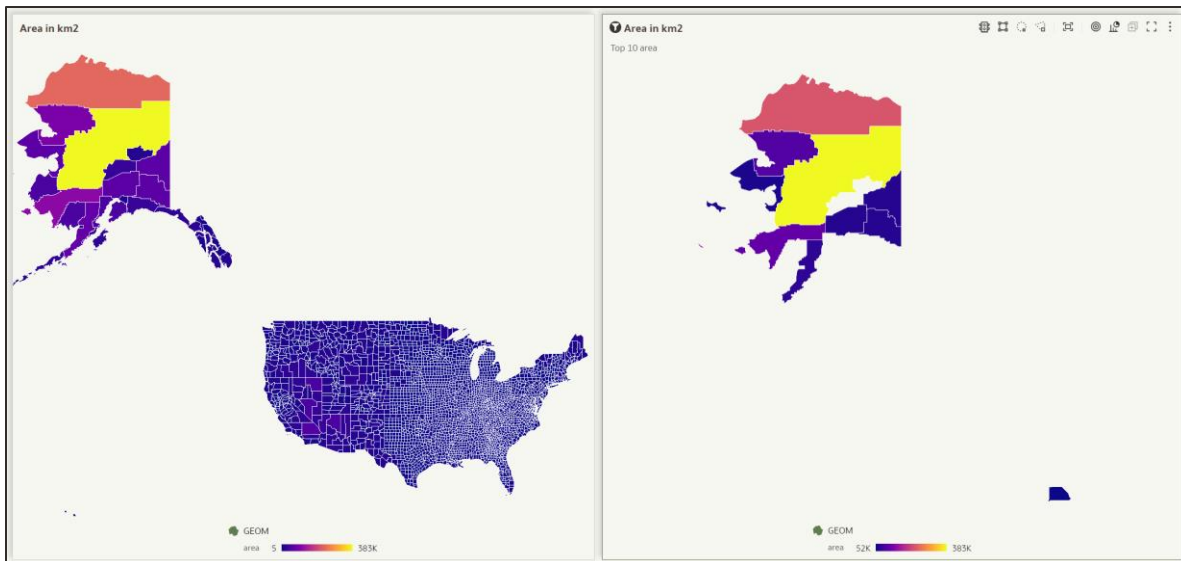


Figure 41 – Example showing how Top 10 counties in US based on area is visualized alongside all counties colored by the calculated area.

How to Calculate the Length of a Geometry

The function `GeometryLength()` is used to calculate the circumference of a shape.

Usage: `GeometryLength(geometry_column)`

Name	<input type="text" value="length"/>
Description	<input type="text"/>
	<pre>max (GeometryLength (GEOM))</pre>

Figure 42 – Example showing an aggregate function `max()` applied on the `GeometryLength()` calculation.

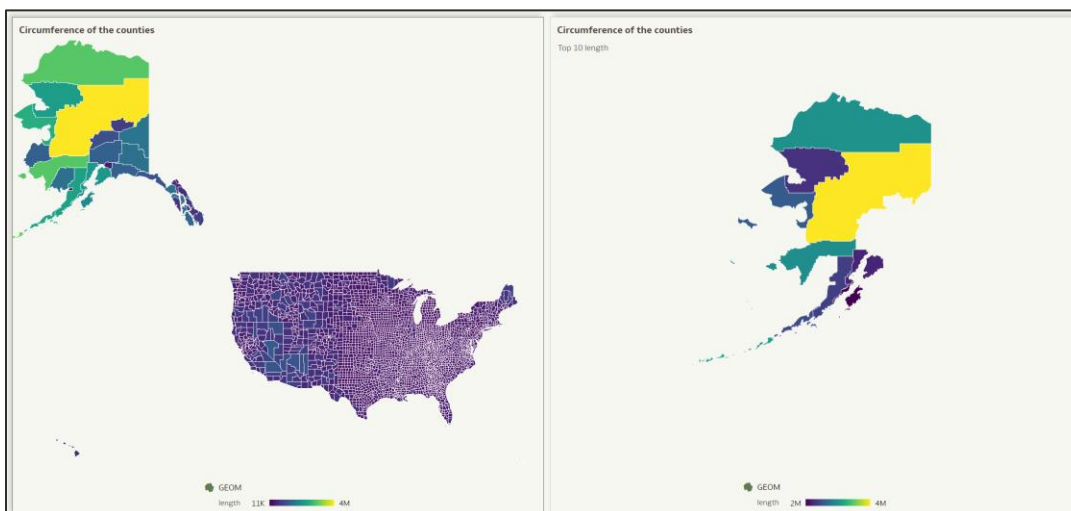


Figure 43 – Example showing the visual on the right selecting top 10 counties based on `GeometryLength()` alongside all counties colored by length of the shape.

How to Generate a Spatial Data Object Definition in Standard Text Format

The function `GeometryAsText()` is used to convert the longvarbinary values of the geometry column into Well-Known Text (WKT) format, a standard format for representing geometry objects. This function also is also useful to troubleshoot when a specific shape isn't being represented in the visualization.

Usage: `GeometryAsText(geometry_column)`

Name	AsText
Description	<pre>GeometryAsText (GEOMETRY)</pre>

Figure 44 – Example showing usage of `GeometryAsText()`.

Geometry As Text	
KEY	AsText
1	POLYGON((-54.65617 47.32792, -54.63506 47.38534, -54.61918 47.44305, -54.60861 47.50111, -54.60341 47.55921, -54.60362 47.61711, -54.60925 47.6746, -54.6203 47.73145, -54.63675 47.78744, -54.65855 47.84234, -54.68562 47.89595, -54.71789 47.94804, -54.75523 47.99842, -54.79752 48.04689, -54.84459 48.09524, -54.89627 48.1373, -54.95237 48.17888, -55.01268 48.21783, -55.07694 48.25398, -55.14495 48.28718, -55.21635 48.3173, -55.29094 48.34423, -55.36839 48.36784, -55.44839 48.38804, -55.53062 48.40476, -55.61473 48.41791, -55.7004 48.42745, -55.78726 48.43535, -55.87497 48.44353, -55.96316 48.45405, -56.05147 48.45888, -56.13952 48.46005, -56.22702 48.460759, -56.31353 48.45915, -60.40595 47.40966, -60.40618 47.01905, -67.82417 46.51271, -67.88531 46.50255, -67.94577 46.49033, -68.00541 46.47607, -68.06408 46.45981, -72.44755 45.0646, -76.72965 43.96701, -76.78486 43.95071, -76.83892 43.93244, -81.08 42.51219, -81.14592 42.28342, -81.20918 42.2515, -81.26953 42.21658, -81.32675 42.1788, -81.38054 42.13831, -84.27611 39.71675, -84.31698 39.67913, -84.35506 39.59778, -84.39024 39.59884, -86.66708 36.68022, -86.6815 36.66052, -86.69527 36.64054, -88.07755 34.52938, -88.10524 34.48264, -88.12972 34.43463, -89.00174 32.54792, -89.0156 30.98082, -89.01618 30.98023, -89.05426 30.93945, -89.08947 30.89688, -89.0217 30.85267, -89.95085 30.80696, -89.97682 30.75992, -89.99953 30.71149, -90.01892 30.66244, -90.03492 30.61233, -90.04748 30.56153, -90.05656 30.51021, -90.24526 29.20863, -90.24925 29.15327, -90.25127 29.09773, -90.24931 29.04221, -90.24534 28.98693, -90.23556 28.93208, -89.91962 27.45429, -89.90828 27.38615, -89.89597 27.33865, -89.87674 27.29194, -89.85663 27.24614, -89.83372 27.20141, -88.82174 25.30783, -88.7995 25.26734, -88.77462 25.22793, -87.42797 23.13758, -86.84683 21.0549, -86.83188 21.00554, -86.81397 20.95711, -86.11348 19.18295, -85.6523 17.73397, -85.64506 16.26695, -85.64332 16.21293, -85.63841 16.15922, -85.63054 16.10798, -85.61914 16.0574, -85.60487 16.00165, -85.58089 15.00307, -85.28386 14.95197, -85.26386 14.90205, -85.24096 14.85348, -84.83872 14.05509, -84.81346 14.00802, -84.7855 13.9626, -84.75494 13.91897, -84.72187 13.87728, -84.2858 13.55643, -83.92973 12.4484, -83.90655 12.60492, -83.88111 12.56283, -83.85348 12.52226, -83.82374 12.48331, -83.42354 11.98453, -83.3861 11.94061, -83.34622 11.89964, -83.30386 11.86119, -83.25917 11.8254, -82.89776 11.52651, -82.82092 11.49089, -82.78049 11.47544, -82.31659 11.19643, -82.07461 11.01411, -82.02822 10.94478, -81.96747 10.76592, -81.94997 10.71602, -81.93 10.66924, -81.90762 10.62372, -81.88288 10.5796, -81.85587 10.53701, -81.82666 10.49406, -81.82636 10.09688, -81.4899 10.05131, -81.45074 10.0083, -81.40902 9.96803, -81.36491 9.93065, -81.31858 9.89631, -81.27022 9.86514, -81.22002 9.83727, -81.16817 9.81281, -81.11488 9.79185, -81.06035 9.77448, -81.0048 9.76076, -80.94845 9.75075, -80.89151 9.74448, -80.83422 9.74198, -80.77678 9.74326, -80.71944 9.74831, -80.6624 9.75712, -80.60591 9.76965, -80.55017 9.78585, -80.4954 9.80564, -80.44183 9.829, -80.38965 9.85577, -80.33908 9.88587, -80.29051 9.91919, -80.24354 9.95559, -80.19894 9.99493, -80.1567 10.03706, -80.11698 10.0818, -80.07994 10.12899, -80.04572 10.17844, -80.01446 10.22995, -79.98428 10.28332, -79.96413 10.33835, -79.93962 10.3948, -79.92132 10.45248, -79.90447 10.51131, -79.89515 10.57054, -79.88738 10.63048, -79.88122 10.69066, -79.88266 10.75096, -79.88573 10.81103, -79.89241 10.87068, -79.90267 10.92966, -79.91648 10.98775, -79.93579 11.04471, -79.95452 11.10033, -79.9786 11.15437, -80.00593 11.20663, -80.03641 11.2569, -80.06992 11.30498, -80.27817 11.38351, -80.32895 11.75481, -80.34584 11.78359, -80.36512 11.8293, -80.38673 11.87582, -80.41063 11.91703, -80.43673 11.9598, -80.46333 12.229, -80.46556 12.30029, -80.46906 12.3398, -80.47047 12.37743, -80.46596 12.41305, -80.46033 12.44656, -80.45422 12.47787, -81.24506 12.77836, -81.27853 12.80337, -81.31477 12.8266, -81.37469 13.10249, -82.04472 13.50544, -82.10771 13.63666, -82.66496 14.35052, -82.68883 14.59499, -82.71513 14.438, -82.74578 14.47941, -82.77468 14.51911, -83.20552 15.03805, -83.51969 15.67041, -83.75438 16.46388, -83.74745 17.92706, -83.74883 17.97969, -83.7555 18.05206, -83.76085 18.08401, -83.77144 18.13537, -83.78506 18.18597, -84.27747 19.78784, -84.28486 19.81032, -84.29286 19.8336, -84.9651 21.58762, -85.5384 23.74081, -85.52125 23.78955, -85.54645 23.83014, -85.58726 25.87355, -85.60852 25.91594, -85.6322 25.95726, -86.99673 26.12601, -87.90433 27.87922, -88.14076 29.12139, -87.98964 30.04182, -87.94071 30.08595, -87.93469 30.13138, -87.02087 31.82468, -87.01025 31.84474, -87.00021 31.86502, -86.10577 35.70954, -84.73258 35.73284, -82.47722 38.53288, -79.76881 40.7533, -75.82796 42.24003, -71.6368 43.31831, -71.54556 43.33699, -71.49586 43.35882, -67.26128 46.70442, -65.73974 45.18227, -63.73047 45.18343, -60.28721 45.5788, -60.22113 45.58673, -60.15547 45.59495, -60.0904 45.60943, -60.02066 45.62414, -55.69155 46.61071, -53.61156 46.62906, -53.53289 46.65035, -53.4558 46.67505, -53.38058 46.70255, -53.30749 46.73288, -53.23678 46.76399, -53.16871 46.80174, -53.10552 46.84001, -53.04143 46.88066, -54.98269 46.92356, -54.9275 46.96855, -54.87607 47.01547, -54.82858 47.06417, -54.78522 47.11445, -54.74616 47.16616, -54.71154 47.2191, -54.6815 47.27308, -54.65617 47.32792))

Figure 45 – Example showing the standard format of the Storm Buffer geometry column.

You can also use standard calculation functions along with the `GeometryAsText` calculation. For example, you can use the `String` function `Length` to calculate the size of a shape, which is useful when identifying if a shape is beyond the acceptable limit of 128 KB.

Usage: `Length(GeometryAsText(Geometry_Coumn))` can return the size of the geometry shape in bytes.

Calculations That Require Two Geometry Columns

How to Calculate the Distance Between Two Geometry Data Types

The function `GeometryDistance()` is used to calculate the distance between two shapes. If your data holds two geometry columns in different datasets, then this function requires you to properly join the two datasets so that all geometry records inter-relate (mimicking a spatial join). It's advisable to establish the join while preparing and creating the dataset, and avoid blending the columns within the workbook canvas as an ad hoc mashup. It's critical that you always consume these joins with some filter value on one side of the join. Either design the dataset with an embedded filter value or design the canvas with an explicit filter value when using these in advanced geometry calculations. A good practice is to always filter the values while using calculations like `GeometryDistance`, `GeometryRelate`, and `GeometryWithinDistance`.

Usage: `GeometryDistance(geometry_column1, geometry_column2)`

Name

distancekm

Description

```
max (GeometryDistance (GEOMETRY, GEOM) )
```

Figure 46 – Example showing an Aggregate function `max()` applied on the `GeometryDistance()` calculation.

SAMPLE_ORDER_LI...

NATE_BUFFER_100KM

Join SAMPLE_ORDER_LINES_SDO_GEOMETRY - NATE_BUFFER_100KM

Full Outer

SAMPLE_ORDER_LINES_SDO_GE...

NATE_BUFFER_100KM

LOOKUPID

=

LOOKUPID

Add Join Condition

GEOM	STATE	LOOKUPID	ORDER_LINE_ID	ORDER_ID	ORDER_PRIORITY	CUSTO
This column contains Geometry data	Texas	1		86510	Medium	1193
	Vir...			86997	High	102
	Ge...			87456	Low	1228
	Ka...			11206	Not Specified	1723
	Illin...			32164	Critical	2006
	Ke...			34017		2333
	Mis...			35840		2530
	Ten...			42528		2882
	Ind...			43526		1014
	Ohio			44002		1085
GEOM	STATE	LOOKUPID	ORDER_LINE...	ORDER_ID	ORDER_PRIO...	CUST
[GEOMETRY]	Mississippi	1	8760	86492	Medium	3168
[GEOMETRY]	New Mexico	1	7735	87721	Medium	2828
[GEOMETRY]	Michigan	1	2460	86566	Critical	946
[GEOMETRY]	South Carol...	1	2654	89009	Medium	1028

Figure 47 - Example shows how a join is defined between two different database tables when creating the dataset.

How to Relate Two Geometry Data Types

The function `GeometryRelate()` is used to determine whether one shape is related to another shape.

A shape from one geometry column could be inside, outside, adjacent inside, adjacent outside, or touching another shape from a different geometry column. The shapes are categorized as Inside, Outside, or Touch. This function may require a Cartesian join.

Usage: `GeometryRelate(geometry_column1,geometry_column2)`

Name	<input type="text" value="relate"/>
Description	<input type="text"/>
	<code>GeometryRelate(GEOM, GEOMETRY)</code>

Figure 48 – Example showing usage of `GeometryRelate()`.

The sequence in which the geometry columns are selected within the `GeometryRelate` calculation has a direct impact on the result in which shapes are identified as relating to each other.

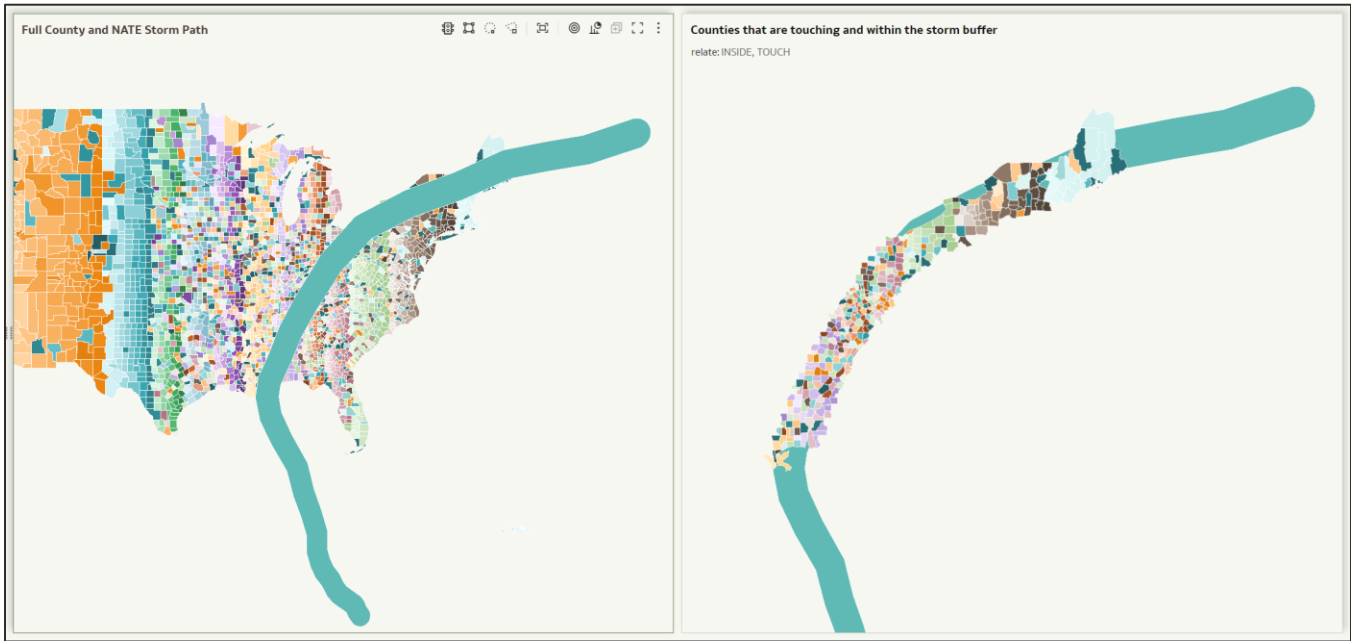


Figure 49 – Example showing the visual on the right relating all the counties that are touching the storm buffer or inside the storm buffer for potential impact.

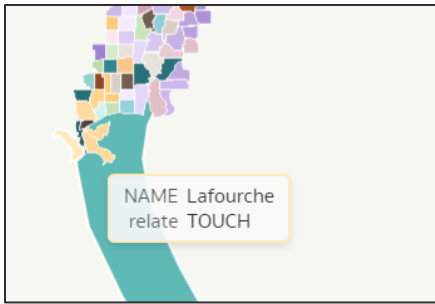


Figure 50 – Example showing shapes from US counties that are touching the storm buffer shape, calculated using GeometryRelate ().



Figure 51 – Example showing shapes from US counties that are inside the storm buffer shape, calculated using GeometryRelate ().

How to Determine if Two Geometry Data Types Are Within a Specific Distance

The function `GeometryWithinDistance()` is used to determine whether two shapes are within a specified distance of each other. Based on the calculation, this function returns a True or a False as the result, and the visualizations can be designed based on the result. This function may require a Cartesian join. The sequence in which the geometry columns are selected within the calculation has a direct impact to the result.

Usage: `GeometryWithinDistance(geometry_column1, geometry_column2, distance_in_meter)`

Name	<input type="text" value="withindistance"/>
Description	<input type="text"/>
<pre>GeometryWithinDistance(GEOM, GEOMETRY, 350000.0)</pre>	

Figure 52 – Example showing usage of GeometryWithinDistance () with two geometry columns within aa numerical value of 350,000 meters or 350 km.



Figure 53 – Example shows the visual on the right with all US Counties within a 350 km range of the storm buffer. Only those counties that return the result as True are filtered and shown.

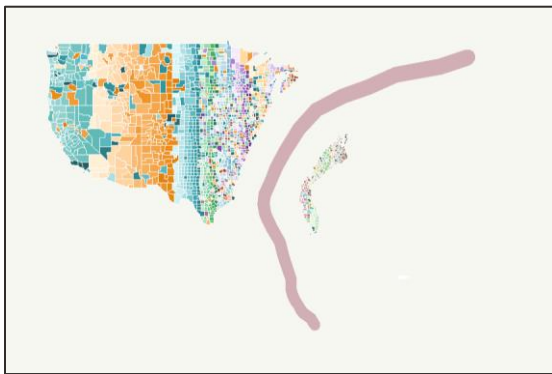


Figure 54 – Example shows all the US Counties that return False and hence don't have any direct impact from the storm.

Limitations

The geometry data type in Oracle Analytics offers exciting possibilities for map-based visualizations and advanced spatial analysis. While there are some limitations, understanding them helps you make the most of its capabilities.

- Geometry columns cannot be used as joins between database tables or as objects to drill on and cannot be visualized in a visualization other than a map.
- Geometry columns can be placed only in the Category grammar field for a map visualization. The content of a geometry column is like binary data to the human eye and is only valid when used in a map or as an argument for a spatial calculation.
- Geometry columns cannot be dragged into any Filter grammar field or used to pass as a context of a custom data action. The current version of the geometry data type doesn't natively support a use case requiring users to click on achieve filtering on a selection of shapes.
- Maps showing geometry columns aren't yet rendered on mobile devices.
- Geometry columns have limitations for use in data flows.
- Advanced analytics functions like Forecasting, Clustering, Outlier/Non-Outlier, Reference Line, and Trend Line aren't supported for geometry columns.
- Spatial joins or joins between geometry data types from different database tables or datasets aren't supported.
- Geometry data type records must be under 128 KB individually. If a record in a data source has a geometry expression that sizes above 128 KB, the projection for this record doesn't show on a map. For example, suppose a table contains data about US states (50 rows), each containing a geometry data type column with the polygon representing the state shape. Assume one of these states has a very high resolution for its border definition, and the size of the geometry object string for this state amounts to more than 128 KB. The dataset is created in Oracle Analytics and queries to the dataset honor every single column from the dataset, but if geometries are projected on a map, then only 49 states are shown on the map. The one larger than 128 KB isn't shown on the map, and a warning is displayed in the dialog for the Information icon on the map.

If the size of a geometry shape is over 128 KB, then you can try the following alternatives as possible workarounds:

- Simplify the definition of the polygon (reduce the number of points to delimit the polygon) and therefore reduce its technical storage.
- Split the polygon into two or more distinct polygons to break down the size of each piece.
- Oracle Analytics supports 2-D map coordinate systems only. Any known coordinate system defined in the data source is projected to WGS 84 (EPSG 4326).
- File-based data sources containing geometry data must be Text/CSV files with the geometry field expressed as Well-Known Text (WKT) formats. WKT is a text markup language for representing vector geometry objects. It can represent distinct geometry objects such as Points, MultiPoints, LineString, Polygons, and MultiPolygons. WKT format syntaxes such as GeometryCollection or FeatureCollection aren't supported.
- Files with the GeoJSON format with geometry data aren't supported for importing as datasets with geometries in Oracle Analytics Cloud. In order to create a dataset with a geometry column, GeoJSON files must first be converted into CSV with WKT geometry formats or loaded into an Oracle database with the SDO.Geometry format.

Performance Considerations

- Physical size of the data retrieved by the map
 - For large datasets with many distinct geometry records, the performance of the map visualization rendering is directly impacted by the size of the geometry data transiting over the network to the Oracle Analytics browser. The performance is a factor of how many distinct geometries are represented on the map and the average size (in KB) of each geometry. As it builds, the map visualization fetches the geometry definitions via the network for each distinct shape on the map, for each distinct layer. Very large lists of geometries take longer to render when geometries have a very fine definition (large average size).
 - **Mitigation:** Avoid representing very high numbers of geometries (>10,000) on a map, even if the system supports up to 125,000, the performance experience to render this is slow. If you do need to show many shapes at once, filter the map visualization with specific areas and enable consumer users to filter one area at a time.
- Spatial Cartesian joins
 - Spatial calculation functions such as GeometryDistance, GeometryLength, GeometryRelate, and GeometryWithinDistance, may require you to have two joined datasets. These functions require a proper data model to avoid ambiguity in execution, that is, a many-to-many distance calculation, which may lead to exceeding the query limit and poor performance.
 - **Mitigation:**
 1. Use a data model, and most importantly, data filtering on geometry records to avoid Cartesian joining effects.
 2. When using a database, spatial calculations ship down the function to the database and let Oracle Analytics retrieve the results of the calculation. Be cautious when optimizing performance and ensure this function-shipping is happening properly. There may be cases where other calculations are added (in data flows or workbooks) that prevent Oracle Analytics from pushing down the work to the database. In the context of troubleshooting performance, it's critical to check the context that allows function shipping.

Appendix 1 – Oracle Spatial Calculations

The following is a list of Spatial Functions from the Oracle Analytics Calculation Library.

Spatial Function	Syntax	Description
GeometryArea	GeometryArea(shape_expr)	Calculates the area that a shape occupies.
GeometryLength	GeometryLength(shape_expr)	Calculates the circumference of a shape.
GeometryAsText	GeometryAsText(shape_expr)	Converts the longvarbinary values of the geometry column into Well-Known Text (WKT) format, a standard format for representing geometry objects.
GeometryDistance	GeometryDistance(shape_expr1, shape_expr2)	Calculates the distance between two shapes.
GeometryRelate	GeometryRelate(shape_expr1, shape_expr2)	Determines whether one shape is inside another shape.
GeometryWithinDistance	GeometryWithinDistance(shape_expr1, Shape_expr2, distance_expr)	Determines whether two shapes are within a specified distance of each other.

shape_expr represents any valid geometry column or shape expression.

shape_expr1, shape_expr2 represents any valid geometry column or shape expression.

distance_expr is any expression that evaluates to a numeric value in meters.