

# **GeoAnalytics Enablement**

# Exercises

September, 2018

Renato Vieira rvr@qlik.com



qlik.com



# **Table of Contents**

Version History	3
Installing GeoAnalytics	4
Exercises Part 1 – Using Extensions and Location Services	6
Exercise 1: Preparing application data	6
Exercise 2: Area Layer to display Countries	9
Exercise 3: Bubble Layer to display sales per customer	11
Exercise 4: Drill from one layer to another based on user interactions	14
Exercise 5: Chart Layer to display sales by product category on the map	
	15
Exercise 6: Line Layer to shipping from warehouses to customers	16
Exercises Part 2 – Using the GeoAnalytics Connector	20
Exercise 1: Load geographical data from an external source (GeoJSON)	
	20
Exercise 2: Using TravelAreas	25
Exercise 3: Using Binning	31
Exercise 4: Using Within	39
Exercises Part 3 – Additional GeoAnalytics Connector exercises	44
Exercise 1: Cluster	44
Exercise 2: Routes	53
Exercise 3: Closest	58
Exercise 4: Dissolve	61
Exercise 5: Intersect	67
Mobile "Near me" Location	71
Additional Guidelines	77
How to perform GeoCoding	77
How to perform Reverse GeoCoding	78
Enablement Materials	80

# **Version History**

Date	Contents	Author
13/04/2017	Initial Version: Exercises Part 1, Exercises Part 2 and Additional Guidelines	Renato Vieira
25/09/2018	<ul> <li>Updated</li> <li>Exercises Part 1 and 2 – added notes regarding official documentation</li> <li>Additional Guidelines: Enablement Materials</li> </ul>	Renato Vieira
	Added:	
	<ul><li>Exercises Part 3</li><li>Mobile "Near me" Location</li></ul>	

# **Installing GeoAnalytics**

In this section, we will quickly cover how to install GeoAnalytics on your machine with Qlik Sense. My recommendation is to use it on a server version, but Qlik Sense Desktop can work as well.

**Download** the Qlik GeoAnalytics software from our download site (<u>http://eu-a.demo.qlik.com/download/</u>) and select the most recent version. At the time of the redaction of this document it's version 5.8.0.

Downloa	ad Site	)						Bo	okmarked Dow	vnloads ★
Qlik <sup>®</sup> Sense	<b>QlikView</b> ®	Olik Connectors	Qlik NPı	rinting	Qlik	GeoAn	alytics	View	All / Sea	arch
Filters		Downloads								
Clear Filte	ers	File Name	Versio	Release Number	Release Stage	Build	Operating System	Language	Last Modified	Size
<b>Preferred Language</b> Chinese (中文) Dutch (Nederlands)	۵	GeoAnalyticsForQlikSenseDesktopS 5.8.0.exe	etup- 5.8.0	Initial Release	Public	31130	Windows x64 (64bit)	English	2017-04-10	940 KB
English	Ψ	GeoAnalyticsForQlikSenseServer- 5.8.0.zip	5.8.0	Initial	Public	31130	Windows x64 (64bit)	English	2017-04-10	862 KB
Product Qlik GeoAnalytics		GeoAnalyticsForQlikViewDesktop- 5.8.0.zip	5.8.0	Initial	Public release (GA)	31130	Any	English	2017-04-10	911 KB
		GeoAnalyticsForQlikViewServer-5.8.	<u>0.zip</u> 5.8.0	ninitiai	Public release (GA)	31130	Any	English	2017-04-10	914 KB
Version No 5.8.0 1.7.0 4.3.0 Relea	<b>ise Number</b> Initial Release									
File Function Installation program Documentation and Tuf Misc Release notes Mobile client	torial									

**Execute** the installer and follow the guided steps.



For Sense Deskop:

• accept the default path suggestions for both the GeoAnalytics Maps (Extensions) as well as the GeoAnalytics Connector.

 Accept the default URL suggestions for both the GeoAnalytics Maps and GeoAnalytics Connector servers



•

.

#### For Sense Server:

- Unzip the file and import the extensions via QMC
- Execute the GeoAnalyticsSetup\_5.8.0.exe file to install GeoAnalytics Connector
- Accept the default URL suggestion for the GeoAnalytics Connector servers

**Apply** your internal license when the popup appears. You may get your license here at our Internal Licenses Page:

https://q.qliktech.com/Intranet/Departments/Products/RnD/QlikView%20Licences.aspx#4QlikGeoAnalytics

Qlik GeoAnalytics License Activation Tool	- 🗆 ×
Owner name: Owner organization:  Serial: Choose Installation: Olik Sense Desktop	Instructions 1) Enter license serial and control number. 2) Enter author and organisation name. 3) Select Installation that the license should be activated for. 4) Press Activate/Refresh. 5) Verfly in the log window to make sure that the license has been applied. 6) Done.
Path: [C\Users\viei\Documents\Qiik\Sense\Extensions\ Installation Log	
Activate/Refresh License	Close

Important note: **select the correct product** in the *Choose Installation* option to avoid malfunction such as *Invalid Key* error when adding a map to your application.

# **Exercises Part 1 – Using Extensions and Location Services**

In this section, we will cover a few exercises to let you know the extensions a little bit. No GeoAnalytics connector will be used in here. Only direct requests to our hosted server using the location services intelligence already developed within the extensions.

Note: always check the GeoAnalytics documentation page for full details of the product's capabilities: <u>https://bi.idevio.com/products/idevio-maps-for-glik-sense/documentation</u>

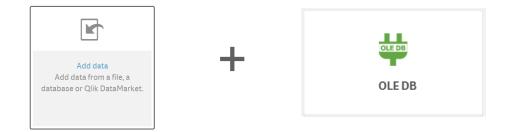
The *Exercises Part 1* uses contents available with more detail at the following sections:

- User Guide: <u>https://bi.idevio.com/wp-</u> content/qlik/qliksense/releases/IdevioMapsForQlikSense-5.13.0/user\_guide-September\_2018.html
- Properties: <u>https://bi.idevio.com/wp-</u> <u>content/qlik/qliksense/releases/IdevioMapsForQlikSense-5.13.0/imap\_reference-qs-</u> <u>September\_2018.html</u>

### **Exercise 1: Preparing application data**

We will create an application from scratch, so you can get familiar with the data. This is a standard application used for workshops and activities on the likes. The data represents an example of a clothing stores franchise that is multi-region. Its database, in the Sales.mdb file, contains all their sales history.

Start by **creating a new application and connecting to the Sales.mdb** file via OLE DB Jet 4.0. This database is open, so there is no need to specify administrators nor passwords. We will use the Bubbles Interface to speed up the datamodel creation process.



#### Create new connection (OLE DB)

Provider							
Microsoft Jet 4.0 OLE DB Provider(32-bit)							
Data source (file path or server name) C:\Users\rviei\OneDrive - QlikTech Inc\Workshops and Events\20170413 - GeoAnalytics							
C. OSEIS (MEROHEDI VE - QIIKTECI		vents (20170415 - GeoAnalytics					
Windows integrated security							
O Specific user name and passw	ord						
Username							
Password	Password						
<ul> <li>Test successful.</li> </ul>		Test connection					
Database							
Load	Select databa	ISE v					
NI							
Name ColorDB							
SalesDB							

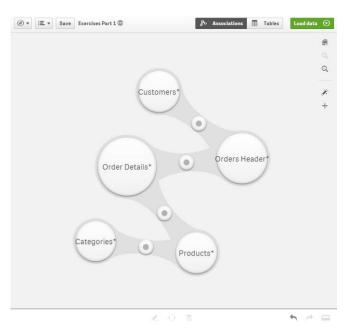
#### Select the following tables to load:

Tables	
<b>Q</b> Filter tables	
Categories	3
Customers	10
✓ Order Details	7
✓ Orders Header	6
Products	4

You may load the tables with all the fields. Once the tables are selected choose the **Prepare Data** option.



Click on the *Magic Wand* so the data preparation functionality suggests the datamodel. You should get the same result as indicated in the image below:



Load the data to your application.

#### Load data 🕟

Once the data is loaded we will quickly **create a sheet** that has some fields and KPIs. This way we will be able to see the associative model work and how it interacts with the map functionalities we will fiddle with.

#### Add 4 filter panes:

- Country
- City
- NomeCat
- ProductName

Add 2 KPIs (recommended via Measures Master Item)

- # of customers *count(Customer)*
- Sales sum(Sales)

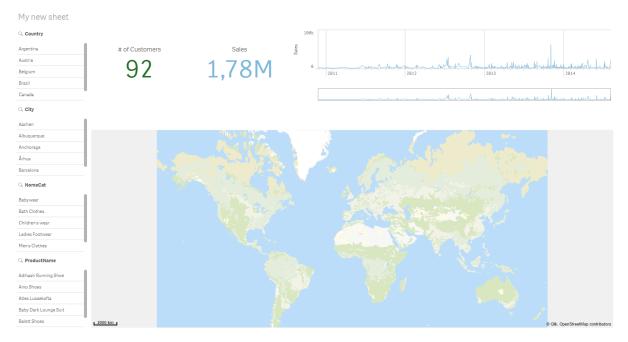
Add 1 Line chart with the sales trend.

- Dimension: OrderDate
- Measure: *sum(Sales)*

Add the GeoAnalytics Map extension to your sheet.



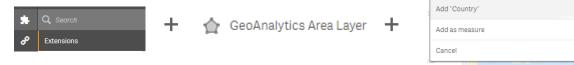
Here is an **example** of what your sheet could look like:



### **Exercise 2: Area Layer to display Countries**

This exercise will allow you to add a layer to your map: The Area Layer. The objective will be to draw all the countries where customers exist and color the area by the number of customers.

Add the GeoAnalytics Area Layer to your sheet and drag the Country field to it.

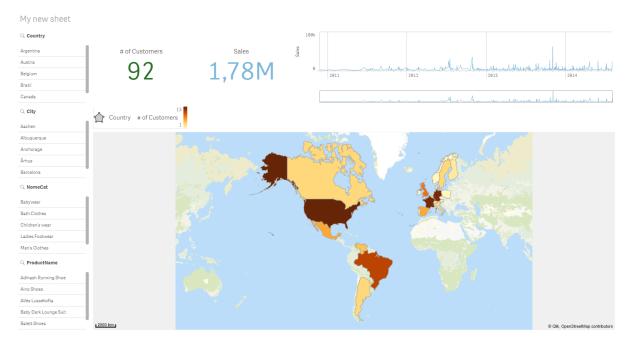


You will immediately see on the map the areas of all the Countries that exist in the Country field.



Now we can **color** all these countries based on a measure. Go to the GeoAnalytics Area Layer extension properties, select **Appearance**, expand **colors** and **deactivate** the **Auto** option. **Select** By GeoAnalytics Enablement | 9

*measure*. If you have created a # of customers Master Item select it, otherwise add the count(Customer) expression. Here is how your final result should look like:



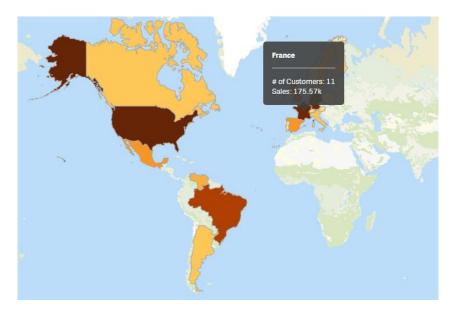
Notice that **you can make this layer visible or not** at will by clicking on the extension and checking/unchecking the *visible* checkbox.



There are also many options to customize the appearance and behavior of the extension. One example could be to **customize the Info Bubble** that appears when a user hovers the area. Go to the GeoAnalytics Area Layer extension properties, select **Appearance**, expand **Info Bubble** and **deactivate** the **Auto** option. Note: If you get the Add measure to use expression warning message go to the **Location ID** pane, add **Country** as a measure (only the field, no aggregations) and return to the previous Info Bubble configuration area. Click on the **fx** option to open the full expression editor and **add a custom expression** at your choosing. **The Info Bubble is affected by HTML tags** to customize its layout and you can even add external images. Add an expression like:

```
'<h1>'&Country&'</h1><hr /># of Customers: '&count(Customer)&'<br />Sales: '&sum(Sales)
```

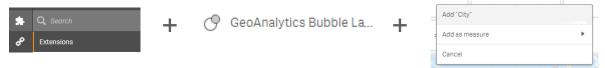
And it is expected you get an output like the image bellow when you hover a country:



### Exercise 3: Bubble Layer to display sales per customer

This exercise will allow you to add a layer to your map: The Bubble Layer. The objective will be to draw all the customers' cities on the map and both color as well as affect the bubble size based on the overall sales made to that city.

Add the GeoAnalytics Bubble Layer to your sheet and drag the City field to it.



You will immediately see on the map the bubbles of all the Cities that exist in the City field.



At a first glance, this representation may look good. But there is always a chance that a City name can be the same in more than one Country. To make sure that we are addressing to the representation of the correct City we can add some control to what is sent to the Location Service.

To **improve our representation accuracy**, we will create a measure that sends a <City, Country> representation instead of only sending a City name.

Go to the GeoAnalytics Bubble Layer extension properties, select **Location,Size** and click on **Add measure.** Add the following expression to concatenate the City to Country separated by a comma:

City&','&Country

And compare the result:



For more details about the Location Options specifications you can refer to <u>https://bi.idevio.com/wp-content/qlik/qliksense/releases/IdevioMapsForQlikSense5.13.0/imap\_reference-qs-September\_2018.html#Bubble\_20Layer</u>

Now that we are representing the cities correctly, we can **add color and size** to the bubbles, based on the overall sales for each city. To do so, an easy way to do it is to drag your *Sales* Master Item to

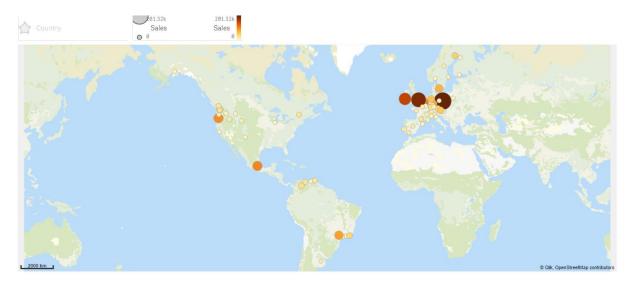
Replace "City&; "&Country"	A	.dd "Sales
I IT WIII	R	eplace
Color by: Sales	с	olor by: S
Cancel by:	С	ancel

Add "Sales"	
Replace	•
Color by: Sales	
Cancel	

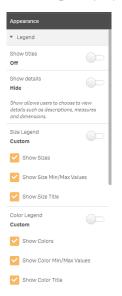
GeoAnalytics Bubble Layer
extension. If you select Add Sales affect the size. If you select Color Sales it will affect the colors. Drag times to have the desired effect.

+

Notice that the extension is autoamtically adding some legend to help understand both the bubble size and color meanings.



You may customize this at will on Appearance, Legend properties of the extension.



You can also add a **label next to the bubble**, so it can display the corresponding City. Select the **Appearance**, **Label** properties pane of the extension and **type in an expression** of your choosing, such as *City* to represent the value of the field. You may also affect the behavior so this label only appears upon a specific level of **zoom**. Try to select something in the range of 3000 and scroll over your map.

If you feel that the map is getting too confusing due to its underneath layering representing too much information such as City names, roads, etc. you can change the **level of detail of representation of the map background.** Click on the map to get its **properties pane** available, go to **Map Settings** and change the **Base Map** representation to **Plain Map**.



#### **Exercise 4: Drill from one layer to another based on user interactions**

Now that we both have the Country areas representing the number of customers and the City bubbles representing the overall sales we can define some behavior such as: display by default the Countries areas and once a selection is made, zoom in and display all the City bubbles for that selection.

This can easily be done by defining a **Drill Dimension** and setting up a behavior in each extension to **only display a specific Drill Level.** 

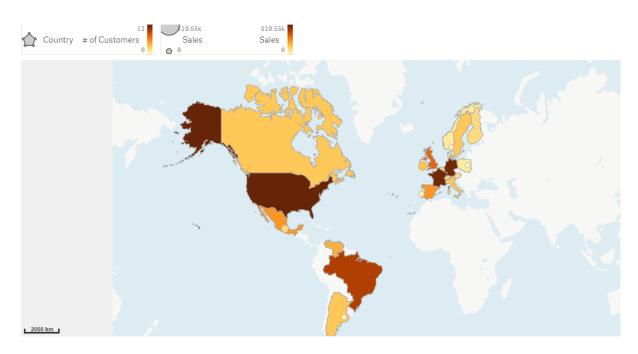
Let's start by creating our Drill Dimension on the Master Items with Country and <City,Country>.

Create new dimensions		
🔵 Single ( Drill-down		
Filter by table	Field:	
All tables 🔹	= Country	fx
Q Search	= =City&','&Country	fx fx
Address	Name:	
CategoryID	GeoDrill	
😵 City	Description:	
ContactName		
COS		
Country	Tags:	•
Customer		
CustomerID		
Description	Add dimension	
		Done

And now we **replace the dimensions** of our GeoAnalytics Area Layer and GeoAnalytics Bubble Layer with our newly created Drill Dimension.

Replace "Country"	27	Replace "City"
Color by: GeoDrill	ale	Cancel
Cancel	+	

Make sure both layers are now with the *Visible* mode activated and you should have the following temporary representation.



We will now setup which drill level should each extension react to:

- Go to the GeoAnalytics Area Layer properties, select Layer Options and set Restrict Drill • Down to 0
- Go to the GeoAnalytics Bubble Layer properties, select Layer Options and set Restrict Drill . Down to 1

You will now see that depending on the drill level, one of the extensions will be grayed out to let you know it's inactive. Select a Country, for example, and you will see that the bubbles of that Country will now be displayed.



### Exercise 5: Chart Layer to display sales by product category on the map

As an alternative to the bubbles representation, we can add mini-charts right on top of the map, that can bring an extra level of information. In this specific case, we will display the amount of sales per product category, affected by the drill level the user has selected at the time.

Add the GeoAnalytics Chart Layer to your sheet and drag the GeoDrill Master Item Dimension to it.



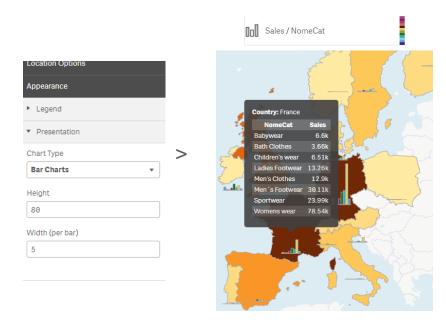
Go to the extension properties and select the **ID**, **Chart Dimension** pane. The GeoDrill will take care of positioning the charts, but we still need to indicate what is the dimension of the charts we want to display. Add **a second dimension** with the NomeCat field.

ID, Chart Dimension				
8				
8				

As a second configuration step, we now need to indicate the measure for our charts. Just drag the Sales Master Item Measure or manually add the *sum(Sales)* expression on the **Location, Chart Value** pane.

Chart Layer				
ID, Chart Dimension				
Location, Chart Value				
Add measure				
▼ Sales	8			
Measure				
Sales	6 <sup>D</sup>			
Number formatting				
Auto	•			

Finally, you may setup the type of chart to display: **Pie or Bar**. Go to the **Appearance** pane and select **Presentation**. Fine tune your charts at your liking.



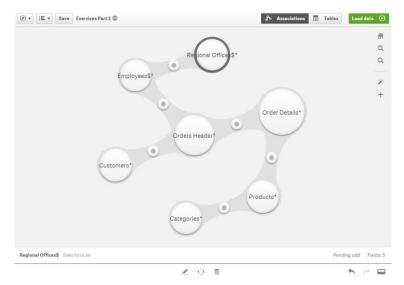
#### Exercise 6: Line Layer to shipping from warehouses to customers

As the last exercise for this first part we will import the location of our warehouses and look into where their shipping is made to and display the lines width affected by the sales representation of each trajectory. For practical reasons of this exercise, we will assume that the sales offices and the warehouses are the same.

Let's start by **importing an additional excel file**, named Sales force.xls, to our datamodel by dragging it to our application. Import both of the available tables: Employees\$ and Regional Offices\$. Use the Prepare Data functionality to speed up the process as there are some field names that don't match with our existing datamodel.

Tables			
<b>Q</b> , Filter tables			
Employees\$	8		(Prepare data
Regional Offices\$	3	+	

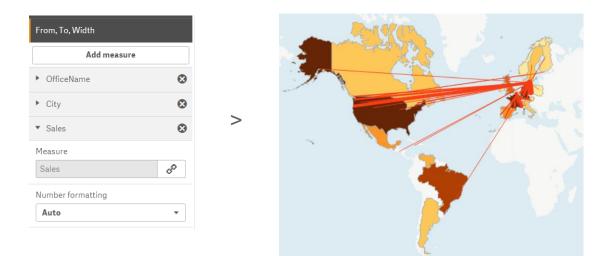
Click on the *Magic Wand* so the data preparation functionality suggests the datamodel. You should get the same result as indicated in the image below:



Now we will go back to our sheet and add the **GeoAnalytics Line Layer** extension and drag City as our relationship field to be used as ID/Dimension.



This object needs us to identify the origin and the destination of the line to display. For that we will go to the **From, To, Width** pane to configure that information as well as affect the line's width with the overall of sales. Add *OfficeName* as Origin, *City* as Destination and the *Sales Master Item* (or sum(Sales)) as Width.



At last, but not least, we can fine tune our lines representations. For that let's go to the **Appearance** pane and select the **Shape and Size** pane. Add some **max width**, i.e. 16.

<ul> <li>Shape and Size</li> </ul>	
Width Min 1 - Max 16	
Min Width Value Auto	
Max Width Value Auto	

Next, we will affect the lines **colors**. Our customer would like to understand flows between US and Europe.

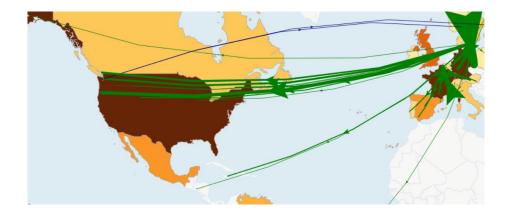
Go to the **Colors** pane, deactivate the *Auto* option and select **By expression** in the dropdown and add the following expression

if(Region='	US'.blue	().areen()
		(),g. ee. (),

For a last touch, we will add some **line curviness** and **arrows** to show the direction of the flow. Go to the **Line** pane and select **Forward** for the **Arrow Style** and add some **Line Curviness** as well as define the **Arrow Position** of your liking.

▼ Line	
Arrow Style	
Forward	•
Arrow Position	
Line Curviness	

Here is an example of the final result. As we can quickly see, there are a lot of shipments from Europe to US and a few from US to Europe, which certainly can be looked at.



## Exercises Part 2 – Using the GeoAnalytics Connector

In this section, we will cover a few exercises to let you know the GeoAnalytics Connector a little bit. We will connect to external sources that have geographic information and we will build an application that takes advantage of a few functions available from the GeoAnalytics Connector, such as TravelArea, Binning and Within.

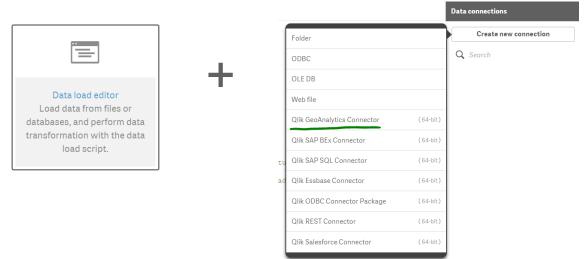
#### Exercise 1: Load geographical data from an external source (GeoJSON)

We will create an application from scratch, so you can get familiar with the data. This time we will load data that is in the public domain in a GeoJSON format. The source is Lisboa Aberta and it this belongs to the Open Knowledge International initiative.

We will use the Data Load Editor combined with a GeoAnalytics Connector connection.

Note: always check the GeoAnalytics reference page for full details using the Load Operation: <u>https://bi.idevio.com/wp-content/qlik/geoanalytics/releases/IdevioGeoAnalyticsConnector-5.13.0/doc/geoanalytics\_reference-September\_2018.html#Operation.Load</u>

Start by entering the **Data Load Editor** and **Create a new connection** to your **Qlik GeoAnalytics Connector**.



Accept the **default URL** for the GeoAnalytics server and add a **Connection Name**. You may also click on **Test Connection** to check if your connection to our hosted server is working.

Create Connection - GeoAnalytics Connector					
Connector version: 1.7.0					
GeoAnalytics server:					
https://ga.qlikcloud.com					
Connection name:					
GeoAnalytics					
Test Connection Cancel Save					

We will now load a GeoJSON file available at <u>Lisboa Aberta's</u> site that contains all the water spots of the city (drinkeable, fountains, decorative, etc.). The direct access to the GeoJSON is via the following URL:

### https://services.arcgis.com/1dSrzEWVQn5kHHyK/ArcGIS/rest/services/Ambiente\_DMEVAE/FeatureS erver/1/query?where=1%3D1&outFields=\*&f=pgeojson

On your Qlik Sense **Data Load Editor** click on the **Select data** icon ( $\blacksquare$ ) of your GeoAnalytics Connection and select the **Operation: Load**. Give *WaterRaw* as **Name** for your *Dataset*, select *File based geo dataset* for **Type** and paste the link in **URL**. You may leave all the other options as suggested by default and click on **Next**.

Operation:	Load •	?
Dataset		
Name:	WaterRaw	
Type:	File based geo dataset	?
URL*:	https://services.arcgis.com/1dSrzEWVQn5	
Key Field:		
File Type:	Auto •	
Expected Geometry Type:	Auto •	
Character Encoding:		
CRS:	Auto	

The Load operation of the GeoAnalytics connector is a function that simply loads datasets, where you can easily specify the source and quickly add geographic data to your application. As you can see there is a question mark next to some areas of the wizard. These question marks allow you to jump directly to the help page so you can understand what is expected to do for each operation configuration possibilities.

Once you click on **Next** the connector will check if the parameters are ok and give you access to the **Select data to load** window you are familiar with. Depending on the operation used, more tables can be presented and by clicking on **Insert Script** the required script will automatically be generated for you.

For our application, we don't need all the fields available from the GeoJSON. You may **rename and select** *Dataset\_Geometry* to *WaterGeometry*, **rename and select** *DESIGNACAO* to *WaterDesignation*, **rename and select** *MORADA* to *WaterAddress* and **rename and select** *\_autoIndex\_* to *WaterId.* Once you've selected and renamed the fields you may click on **Insert Script**.

Select data to load Tables Q. Filter tables		Hide script
Dataset 4		
	Fields Data preview Metadata	<b>Q</b> Filter fields
	WaterGeometry OBJECTID WaterDesignation COD_SIG WaterAddress	GlobalID WaterId
	/* Generated by GeoAnalytics for operation Load	:], [_autoIndex_] AS [WaterId] FROM Load( FeatureServer/1/query?where=1%301&outFie](
		Cancel Back Insert script

#### The following script is expected to have been generated for you:

LIB CONNECT TO 'GeoAnalytics';

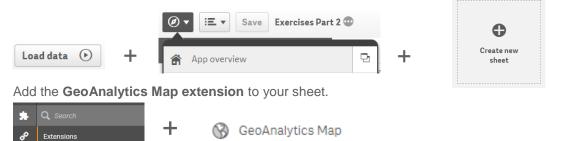
/\* Generated by GeoAnalytics for operation Load ------ \*/

[WaterRaw]: SQL SELECT [Dataset\_Geometry] AS [WaterGeometry], [DESIGNACAO] AS [WaterDesignation], [MORADA] AS [WaterAddress], [\_autoIndex\_] AS [WaterId] FROM Load(dataset='Dataset')

DATASOURCE Dataset GEOFILE url='https://services.arcgis.com/1dSrzEWVQn5kHHyK/ArcGIS/rest/services/Ambiente\_DMEVAE/FeatureServer/1/query?where=1%3D1&out Fields=\*&f=pgeojson', keyField='', type='auto', expectedGeomType='auto', encoding='', crs='Auto'

/\* End GeoAnalytics operation Load ------ \*/

#### Load your data and test it out by creating a new sheet.



Add the GeoAnalytics Bubble Layer to your sheet and drag the WaterGeometry field to it.

*	Q Search		~	GeoAnalytics Bubble La		Add "WaterGeometry"	
в	Extensions	Ŧ	0	GeoAnalytics Bubble La	Ŧ	Add as measure	•
						Cancel	

You are now able to see all the water spots available in Lisbon, Portugal.



We need to reduce this number of spots to the information that is really relevant to us: the drinking water spots. Those points are easily identifiable from the *WaterDesignation* field. They are all the descriptions that start with *Bebedouro*.

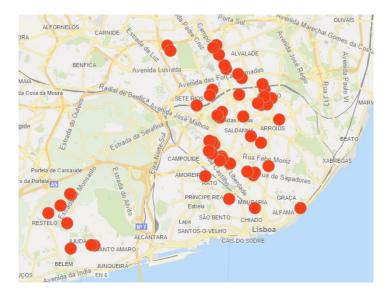


Let's go back to our **Data load editor** and fine tune our script to restrict the data we want in our data model by creating a new table based on our **resident** and restrict to *WaterDesignation* **like** *"Bebedouro\*"*. You may copy the following script snippet to speed up your script creation.

DrinkingWater: NoConcatenate Load WaterId, WaterAddress, WaterDesignation, WaterGeometry resident WaterRaw where WaterDesignation like 'Bebedouro\*';

drop table WaterRaw;

You should now have loaded only 52 lines of data and your map will look like this:



Now let's make these spots more user friendly by giving them a **Symbol** and adding some information in their **Info Bubble**.

Starting with the **Info Bubble** configuration, go to the GeoAnalytics Bubble Layer extension properties, select **Appearance**, expand **Info Bubble** and **deactivate** the **Auto** option. Note: If you get the Add measure to use expression warning message go to the **Location ID** pane, add **WaterGeometry** as a measure (only the field, no aggregations) and return to the previous Info Bubble configuration area. Click on the **fx** option to open the full expression editor and **add a custom expression** at your choosing to display at least the *WaterDesignation*. **The Info Bubble is affected by HTML tags** to customize its layout and you can even add external images. Add an expression like:

ppearance
Legend
Shape and Size
Colors
Label
Info Bubble
Show Info Bubble
nfo Bubble
' <h1>'&amp;WaterDesignation&amp;'<!--</td--></h1>

'<h1>'&WaterDesignation&'</h1><hr />'&WaterAddress

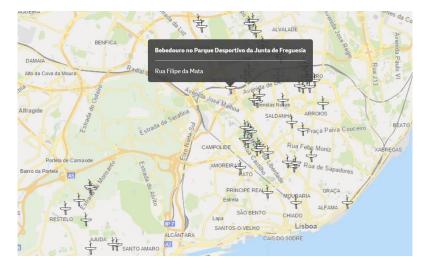
Finishing with the **Symbol** configuration, go to the GeoAnalytics Bubble Layer extension properties, select **Appearance**, expand **Shape and Size**, select **Symbols** in the **Shape** dropdown and add the following URL:

'http://3.1m.yt/3B-cdU.png'

Resize it by setting the **Scale** property to 0.6.

Appearance	
▶ Legend	
<ul> <li>Shape and Size</li> </ul>	
Shape	
Symbols	•
Image URL	
'http://3.1m.yt/3B-cdU.png'	fx
Scale	
0.6	fx
Rotation (Degrees)	
	fx

Here is an example of how your work may look like:



### **Exercise 2: Using TravelAreas**

Now that we know where we could stop for a little of water, let's give some more information to our map. An interesting way to look at this information could be to find out what would be the potential travel area available for each of these water spots for a maximum distance of five minutes walking and five minutes on bicycle.

To do so we will use another function available on the GeoAnalytics Connector named TravelAreas.

Note: always check the GeoAnalytics reference page for full details using the TravelAreas Operation: <u>https://bi.idevio.com/wp-content/qlik/geoanalytics/releases/IdevioGeoAnalyticsConnector-</u> <u>5.13.0/doc/geoanalytics\_reference-September\_2018.html#Operation.TravelAreas</u>

Let's start by going to the **Data load editor**, make sure your cursor is positioned at the end of the script and click on the **Select data** icon ( **D**) of your GeoAnalytics Connection.

Select the **Operation: TravelAreas** and we will now set it up so it gets the necessary information from our *DrinkingWater* table. Remember the importance of the little question marks on the wizard to help you in these configuration windows.

Regarding the **Operation parameters** input *5* as **Cost Value**, change **Cost Unit** to *Minutes* and select *Pedestrian* as **Transportation**.

In the **Origins** parameters setup the **Name** to *WaterFiveMinWalk* (quick note: this is the name of the temporary dataset sent via the connector and not the final table name), select *Loaded table* in the **Type** dropdown, input *DrinkingWater* as **Table Name** (this will let the connector know what is the name of our resident table) and input *WaterId*, *WaterGeometry* in **Table Fields (key,geometry...)**. You may let the other fields all with the default values. A quick note: it is important to define the **Table Fields** (**key,geometry...**) correctly since this is the parameter that will match what the connector needs to look at as well as how the resulting table will be associated with your data model via the "key" field.

Once all the parameters are introduced click on **Next**. Here is a screenshot to let you double check how the parameters should be introduced:

Operation:     TravelAreas     ?       Operation parameters	eration parameters bet Value: 5 bet Field:	Step 1 - Select Operation		
Cost Value:       5         Cost Field:	bat Value:     5       bat Field:     Image: Station:       bat Vinit:     Minutes       bat Vinit:     Pedestrian       gins     Image: Station:       gins     Image: Station:       ble Name:     Loaded table       ble Fields (key,geometry)*:     WaterId,WaterGeometry       ble Fields (key,geometry)*:     WaterId,WaterGeometry       hly load distinct:     No       cation ID Suffix:     Image: Station       KS:     Auto	Operation:	TravelAreas	• ?
Cost Field: Cost Unit: Minutes • Transportation: Pedestrian • Origins Name: WaterFiveMinWalk Type: Loaded table • ? Table Name: DrinkingWater Table Fields (key,geometry)*: WaterId.WaterGeometry Geometry Type: Point • Only load distinct: No •	Dist Field:	Operation parameters		
Cost Unit: Minutes  Transportation: Pedestrian  Grigins Type: Loaded table  Table Name: DrinkingWater Table Fields (key,geometry)*: WaterId,WaterGeometry Geometry Type: Point  Only load distinct: No	bit Unit:     Minutes       ansportation:     Pedestrian       ansportation:     Pedestrian       gins     Image:	Cost Value:	5	
Transportation: Pedestrian	anaportation:     Pedestrian       gins       ume:     WaterFiveMinWalk       pe:     Loaded table       ble Name:     DrinkingWater       ble Fields (key,geometry)*:     WaterId,WaterGeometry       eometry Type:     Point       nly load distinct:     No       cation ID Suffix:     Image: Cation Comparison       RS:     Auto	Cost Field:		
Origins       Name:     WaterFiveMinWalk       Type:     Loaded table       Table Name:     DrinkingWater       Table Fields (key,geometry)*:     WaterId,WaterGeometry       Geometry Type:     Point       Only load distinct:     No	gins       ume:     WaterFiveMinWalk       pe:     Loaded table       ble Name:     DrinkingWater       ble Fields (key,geometry)*:     WaterId,WaterGeometry       ormetry Type:     Point       nly load distinct:     No       cation ID Suffix:	Cost Unit:	Minutes	Ŧ
Name:     WaterFiveMinWalk       Type:     Loaded table     ?       Table Name:     DrinkingWater     ?       Table Fields (key,geometry)*:     WaterId,WaterGeometry     ?       Geometry Type:     Point     ?       Only load distinct:     No     ?	waterFiveMinWalk       pe:     Loaded table       ble Name:     DrinkingWater       ble Fields (key,geometry)*:     WaterId,WaterGeometry       eometry Type:     Point       nly load distinct:     No       cation ID Suffix:	Transportation:	Pedestrian	•
Type:     Loaded table     ?       Table Name:     DrinkingWater     ?       Table Fields (key,geometry)*:     WaterId,WaterGeometry     ?       Geometry Type:     Point     •       Only load distinct:     No     •	Laaded table     ?       ble Name:     DrinkingWater       ble Fields (key,geometry,)*:     WaterId,WaterGeometry       sometry Type:     Point       nly load distinct:     No       cation ID Suffix:	Origins		
Table Name:     DrinkingWater       Table Fields (key,geometry)*:     WaterId,WaterGeometry       Geometry Type:     Point       Only load distinct:     No	.     DrinkingWater       ble Name:     DrinkingWater       ble Fields (key,geometry)*:     WaterId,WaterGeometry       sometry Type:     Point       nly load distinct:     No       cation ID Suffix:	Name:	WaterFiveMinWalk	
Table Fields (key,geometry)*:     Waterld,WaterGeometry       Geometry Type:     Point       Only load distinct:     No	ble Fields (key,geometry)*: WaterId,WaterGeometry cometry Type: Point • nly load distinct: No • ccation ID Suffix: 2S: Auto	Туре:	Loaded table	• ?
Geometry Type: Point   Only load distinct: No	cometry Type:     Point       nly load distinct:     No       acation ID Suffix:	Table Name:	DrinkingWater	
Only load distinct: No	No     No	Table Fields (key,geometry,)*:	WaterId,WaterGeometry	
	Ccation ID Suffix:	Geometry Type:	Point	¥
	Auto	Only load distinct:	No	•
Location ID Suffix:		Location ID Suffix:		
CRS: Auto	Required field	CRS:	Auto	
*=Required field		*=Required field		
				Cancel Next

Once you get the **Select data to load** window, mark the **TravelAreas** table checkbox to load all the fields and click on **Insert script**.

<pre>     Prior tooles     Fileds     Fields     Fields     Fields         VaterFiveMinWalk</pre>	Select data to load Tables Q. Filter tables		Hide script
WaterFiveMinWalk       Filds         Image: Star preview Metadat       Image: Star preview Metadat         Image: Star preview Metadat       Image: Sta			
Fields       Q Filter fields            Water, Wa			
<pre>/* Generated by GeoAnalytics for operation TravelAreas*/ Let [WaterFiveHinWalkInlineTable] = 'WaterTd' &amp; Chr(9) &amp; 'WaterGeometry'; Let numBous = NeoTRow('OrinkingBater'); Let chunkSize = NeoTRow('OrinkingBater'); Let chunksize; For i = 0 To chunkSize; For i = 0 To chunkSize:1 Let row = '; Let row = '; Let row = rowIr&gt;&gt; numBous; For Let funk therEnd', 'WaterGeometry'</pre>			<b>Q</b> Filter fields
<pre>Let [WaterFiveWinWalkInlineTable] = 'WaterId' &amp; Chr(9) &amp; 'WaterGeometry'; Let numKous = NotFouringWater'); Let chunKs = numRous(chunKsIze; For n = 0 to chunKs; Let chunK = numRous(chunKsIze; Let chunK = n*chunKsIze; Let row = ''; Let row = ''; Let row = rowIr &gt;= numRous; For Let for when rowIr &gt;= numRous; For face fin (WaterId', 'WaterGeometry'</pre>		Vate VaterFiveMinWalk_Trav VaterFiveMinWalk_Ori VaterFiveMinWalk_Cost	WaterFiveMinWalk_Sta
1		<pre>Let [WaterfiveHinNalkInlineTable] = 'WaterId' &amp; Chr(9) &amp; 'WaterGeometry'; Let chunkSize = 10006; Let chunkSize = 10006; Let chunks = numRows/chunkSize; For n = 0 to chunks Let chunk = n*chunkSize; For i = 0 To chunkSize; For i = 0 To chunkSize; Let row = ''; Let row = ''; Let row = chunk+i; Exit for when rowNr &gt;= numRows; For Each In 'WaterId', 'WaterGeometry'</pre>	4*\ FL=(20\ 1\.00771\ ;

GeoAnalytics Enablement | 26

Finally (and optionally), to help us making a clear and readable datamodel, **rename** the [TravelAreas] table name to [WaterFiveMinWalk].

#### Your generated script should look like the following:

LIB CONNECT TO 'GeoAnalytics';

```
/* Generated by GeoAnalytics for operation TravelAreas ------ */
Let [WaterFiveMinWalkInlineTable] = 'WaterId' & Chr(9) & 'WaterGeometry';
Let numRows = NoOfRows('DrinkingWater');
Let chunkSize = 1000;
Let chunks = numRows/chunkSize;
For n = 0 to chunks
          Let chunkText = ";
          Let chunk = n*chunkSize:
          For i = 0 To chunkSize-1
                    let row = ".
                    Let rowNr = chunk+i;
                    Exit for when rowNr \geq= numRows:
                    For Each f In 'WaterId', 'WaterGeometry'
                              row = row & Chr(9) & Replace(Replace(Replace(Replace(Replace(Replace(Replace(Peek('$(f)', $(rowNr),
'DrinkingWater'), Chr(39), '\u0027'), Chr(34), '\u0022'), Chr(91), '\u005b'), Chr(47), '\u002f'), Chr(42), '\u002a'), Chr(59), '\u003b');
                    Next
                    chunkText = chunkText & Chr(10) & Mid('$(row)', 2);
          Next
          [WaterFiveMinWalkInlineTable] = [WaterFiveMinWalkInlineTable] & chunkText;
Next
chunkText="
[WaterFiveMinWalk]:
SQL
       SELECT
                    [WaterId],
                                   [WaterFiveMinWalk_TravelArea],
                                                                    [WaterFiveMinWalk_Origin],
                                                                                                      [WaterFiveMinWalk_Cost],
[WaterFiveMinWalk_CostUnit], [WaterFiveMinWalk_Status] FROM TravelAreas(costValue='5', costField='', costUnit='Minutes',
transportation='pedestrian', dataset='WaterFiveMinWalk')
DATASOURCE WaterFiveMinWalk INLINE tableName='DrinkingWater', tableFields='WaterId,WaterGeometry', geometryType='POINT',
loadDistinct='NO', suffix='', crs='Auto' {$(WaterFiveMinWalkInlineTable)}
[WaterFiveMinWalkInlineTable] = ";
/* End GeoAnalytics operation TravelAreas ------ */
```

Load your data and test it out returning to your previously created sheet.

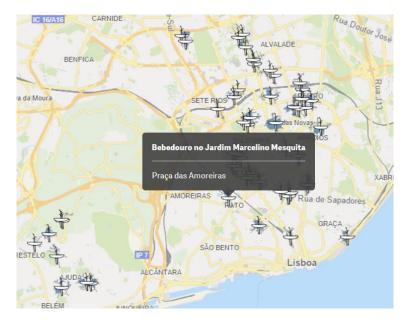


Add the **GeoAnalytics Area Layer** to your sheet and drag the **WaterFiveMinWalk\_TravelArea** field to it.

	~		+ 🍐	GeoAnalytics Area Layer 🕂		Add "WaterFiveMinWalk_TravelArea"		
	Q Search	+			+	Add as measure	۲	
e <sup>g</sup>	Extensions	-	<del>مس</del> ە		•	Cancel		

You will now see on the map the corresponding areas of reach for each of the water spots in a 5 minutes walking distance.

Optionally fine tune the look of the areas by changing their color and transparency as you see fit going to the GeoAnalytics Area Layer extension properties, select **Appearance**, open the **Color** pane and change its properties.



Now we will repeat the process of adding a TravelArea table, this time specifying it for bicycle as the transportation.

Go back to the **Data load editor**, make sure your cursor is positioned at the end of the script and click on the **Select data** icon ( ) of your GeoAnalytics Connection.

Select the **Operation: TravelAreas** set it to load the necessary information from our *DrinkingWater* table.

In the **Operation parameters** section input *5* as **Cost Value**, change **Cost Unit** to *Minutes* and select <u>*Bike*</u> as **Transportation**.

In the **Origins** parameters section setup the **Name** to *WaterFiveMinBike*, select *Loaded table* in the **Type** dropdown, input *DrinkingWater* as **Table Name** (this will let the connector know what is the name of our resident table) and input *WaterId*, *WaterGeometry* in **Table Fields (key,geometry,...)**. You may let the other fields all with the default values.

Once all the parameters are introduced click on **Next**. Here is a screenshot to let you double check how the parameters should be introduced:

peration parameters	TravelAreas	• ?
P		
Cost Value:	5	
Cost Field:		
Cost Unit:	Minutes	T
Fransportation:	Bike	¥
rigins		
Name:	WaterFiveMinBike	
Гуре:	Loaded table	• ?
Fable Name:	DrinkingWater	
Table Fields (key,geometry,)*:	WaterId,WaterGeometry	
Geometry Type:	Point	•
Only load distinct:	No	¥
ocation ID Suffix:		
CRS:	Auto	
=Required field		

Once you get the **Select data to load** window, mark the **TravelAreas** table checkbox to load all the fields and click on **Insert script**.

Y TravelAreas       6         WaterFiveMinBike       Filds         Data preview       Metadata         WaterFiveMinBike_Cost       VaterFiveMinBike_Cost         WaterFiveMinBike_TraveL       WaterFiveMinBike_Cost         /* Generated by GeoAnalytics for operation TravelAreas         /* Generated by GeoAnalytics for operation TravelAreas         /* Generated by GeoAnalytics for operation TravelAreas         /* Let (WaterFiveMinBike_TraveL         /* Generated by GeoAnalytics for operation TravelAreas         /* Generated by GeoAnalytics for operation TravelAreas         /* Let (WaterFiveMinBike_TraveL         /* Generated by GeoAnalytics for operation TravelAreas         /* Generated by GeoAnalytics for operation T	Select data to load Tables Q. Filter tables		Hide script
Fields       Q Filter fields            Wate         WaterFiveMinBike_TraveL.         WaterFiveMinBike_Ori         WaterFiveMinBike_Cost         WaterFiveMinBike_Cost         WaterFiveMinBike_Stat             /* Generated by Goodnalytics for operation TravelAreas	TravelAreas 6		
<pre>/* Generated by GeoMnalytics for operation TravelAreas</pre>	WaterFiveMinBike		Q Filter fields
Let [WaterfiveHinBikeInlineTable] = 'WaterId' & Chr(9) & 'WaterGeometry'; Let numKous = NofNos('DrinkingHater'); Let chunKs = numRous(chunKSize; For n = 0 to chunKs; Let chunKText = ''; Let chunK = n*chunKsize-1 Let row = ''; Let row = ''; Let row = routh = chunk+1; Exit for when routh >= numRows; For Each f In 'WaterId', 'WaterGeometry'		Vate VaterFiveMinBike_Travel VaterFiveMinBike_Ori VaterFiveMinBike_C VaterFiveMinBike_Cost	WaterFiveMinBike_Stat
		<pre>Let [WaterFiveHinBikeInlineTable] = "MaterId" &amp; Chr(9) &amp; 'WaterGeometry'; Let numRous = NoofRoux('OrinkingWater'); Let chunkSize = 1000; Let chunks = numRoux/chunkSize; For n = 0 to chunkSize; For a = 0 to chunkSize; Let chunktext = "'; Let chunktext = "i; Let row = "; Let row = "; Let row = chunkSi; For fach f n "MaterGeometry"</pre>	Muture 1) (122(20) 132-00171) (

Once again (and optionally), to help us making a clear and readable datamodel, **rename** the [TravelAreas] table name to [WaterFiveMinBike].

Your generated script should look like the following:

LIB CONNECT TO 'GeoAnalytics';

/* Generated by GeoAnalytics for operation TravelAreas */
Let [WaterFiveMinBikeInlineTable] = 'WaterId' & Chr(9) & 'WaterGeometry';
Let numRows = NoOfRows('DrinkingWater');
Let chunkSize = 1000;
Let chunks = numRows/chunkSize;
For n = 0 to chunks
Let chunkText = ";
Let chunk = n*chunkSize;

For i = 0 To chunkSize-1 Let row = ''; Let rowNr = chunk+i; Exit for when rowNr >= numRows; For Each f In 'WaterId', 'WaterGeometry'
row = row & Chr(9) & Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Peek('\$(f)', \$(rowNr), 'DrinkingWater'), Chr(39), '\u0027'), Chr(34), '\u0022'), Chr(91), '\u005b'), Chr(47), '\u002f'), Chr(42), '\u002a'), Chr(59), '\u003b'); Next chunkText = chunkText & Chr(10) & Mid('\$(row)', 2); Next [WaterFiveMinBikeInlineTable] = [WaterFiveMinBikeInlineTable] & chunkText; Next
chunkText="
[WaterFiveMinBike]: SQL SELECT [WaterId], [WaterFiveMinBike_TravelArea], [WaterFiveMinBike_Origin], [WaterFiveMinBike_Cost], [WaterFiveMinBike_CostUnit], [WaterFiveMinBike_Status] FROM TravelAreas(costValue='5', costField='', costUnit='Minutes', transportation='bike', dataset='WaterFiveMinBike') DATASOURCE WaterFiveMinBike INLINE tableName='DrinkingWater', tableFields='WaterId,WaterGeometry', geometryType='POINT', loadDistinct='NO', suffix='', crs='Auto' {\$(WaterFiveMinBikeInlineTable)} ;
[WaterFiveMinBikeInlineTable] = ";
/* End GeoAnalytics operation TravelAreas */
Load your data and test it out returning to your previously created sheet.
Load data       H     App overview     App overview
it.



You will now see on the map the corresponding areas of reach for each of the water spots in a 5 minutes cycling distance.

Optionally fine tune the look of the areas by changing their color and transparency as you see fit going to the GeoAnalytics Area Layer extension properties, select **Appearance**, open the **Color** pane and change its properties.



As our multi-layer work evolves, we might need to take ownership about the drawing order of the layers. In this case we can ensure that the bicycle travel area is in the bottom, the walking travel area is in the middle and the water spot icon is on top of both the other layers.

This can be achieved by setting up the **Draw Order** in each of the extensions' properties. This is located within the **Layer Options** pane. Deactivate the *Auto* option for **Draw Order** and manually edit the layering level. Increasing this value makes the layer draw later, i.e. be more visible in the category. Allowed values are in the range -10 to 10. If you want the layer to be on top of other layers in the same category increase this value, if you want it to be overdrawn by other layers in the category decrease it.

Recommended settings for this exercise are:

- Water spot Bubble Layer: *Bubble Layer* for **Draw Order Category** and 0 for **Draw Order** Adjustment
- Five Minutes Walk Area Layer: Area Layer for Draw Order Category and 1 for Draw Order Adjustment
- Five Minutes Bicycle Area Layer: Area Layer for Draw Order Category and 0 for Draw Order Adjustment

### **Exercise 3: Using Binning**

To bring up our active lifestyle up a notch we will go back to our data source <u>Lisboa Aberta</u> and this time load all the public and private sport centers, gyms, etc. available in the city and check if they have water spots nearby in case we go out for a run as well.

Let's start by going to the **Data load editor**, make sure your cursor is positioned at the end of the script and click on the **Select data** icon ( ) of your GeoAnalytics Connection. Select the **Operation: Load**. Give *SportCenters* as **Name** for your dataset, select *File based geo dataset* for **Type** and paste the link bellow in **URL**.

### https://services.arcgis.com/1dSrzEWVQn5kHHyK/arcgis/rest/services/Desporto\_EntidadesDesportiva s/FeatureServer/0/query?where=1%3D1&outFields=\*&f=pgeojson

You may leave all the other options as suggested by default and click on **Next**.

Step 1 - Select Operation		
Operation:	Load	• ?
Dataset		
Name:	SportCenters	
Type:	File based geo dataset	• ?
URL*:	https://services.arcgis.com/1dSrzEWVC	նո5
Key Field:		
File Type:	Auto	•
Expected Geometry Type:	Auto	•
Character Encoding:		
CRS:	Auto	
	C	ancel Next

For our application, we don't need all the fields available from the GeoJSON. You may **rename and select** *SportCenters\_Geometry* to *SportCentersGeometry*, **rename and select** *DESIGNACAO* to *SportCentersDesignation*, **rename and select** *NATUREZA* to *SportCentersType*, rename and select *MORADASEDESOCIAL* to *SportCentersAddress* and **rename and select** *\_autoIndex\_* to *SportCentersId*. Once you've selected and renamed the fields you may click on **Insert Script**.

SportCenters       5         Fields       Data preview Metadata       ♀ Filter fields         SportCentersGeom       OBJEC       ♀ SportCentersDesignat       UIT_MIC       ♀ SportCentersT       COD_SIG_E       FREGUE         /* Generated by GeoAnalytics for operation Load	e script
Data preview       Metadata               Filter fields                  SportCentersGeom                OBJEC               SportCentersT               COD_SIG_E               FREGUE                 /* Generated by GeoAnalytics for operation Load               */               SportCentersGeometry] AS             [SportCentersGeometry]             [SportCentersGeometry] AS             [SportCentersGeometry]             [SportCentersGeometry]	
/* Generated by GeoAnalytics for operation Load */ [SportCenters]: SQL SELECT [SportCenters.Geometry] AS [SportCentersGeometry], [DESIGNACA0] AS [SportCentersType], [M	
[SportCenters]: SQL SELECT [SportCenters_Geometry] AS [SportCentersGeometry], [DESIGNACAD] AS [SportCentersDesignation], [NATUREZA] AS [SportCentersType], [M	IDTI
SQL SELECT [SportCenters_Geometry] AS [SportCentersGeometry], [DESIGNALAU] AS [SportCentersDesignation], [NATUREZA] AS [SportCentersType], [M	
DDTASOURCE SportCenters GEOFILE url="https://services.arcgis.com/14SrzEWVQnSkH+yK/arcgis/rest/services/Desporto_EntidadesDesportivas/FeatureServ ; /* End GeoAnalytics operation Load */	(ADASEDESO( >/0/query?

#### The following script is expected to have been generated for you:

LIB CONNECT TO 'GeoAnalytics';

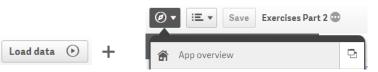
/\* Generated by GeoAnalytics for operation Load ------\*/ [SportCenters]:

SQL SELECT [SportCenters\_Geometry] AS [SportCentersGeometry], [DESIGNACAO] AS [SportCentersDesignation], [NATUREZA] AS [SportCentersType], [MORADASEDESOCIAL] AS [SportCentersAddress], [\_autoIndex\_] AS [SportCentersId] FROM Load(dataset='SportCenters')

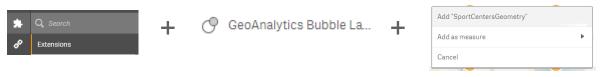
DATASOURCE SportCenters GEOFILE url='https://services.arcgis.com/1dSrzEWVQn5kHHyK/arcgis/rest/services/Desporto\_EntidadesDesportivas/FeatureServer/0/query?where= 1%3D1&outFields=\*&f=pgeojson', keyField='', type='auto', expectedGeomType='auto', encoding='', crs='Auto'

<sup>/\*</sup> End GeoAnalytics operation Load ------\*/

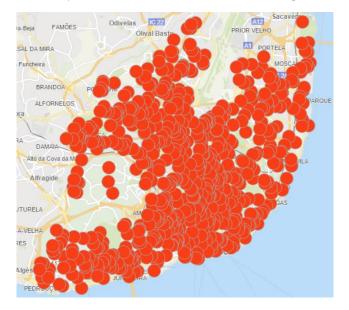
Load your data and test it out returning to your previously created sheet.



Add the GeoAnalytics Bubble Layer to your sheet and drag the SportCentersGeometry field to it.



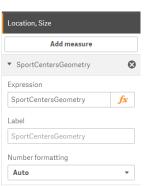
You are now able to see all the sport centers available in Lisbon, Portugal



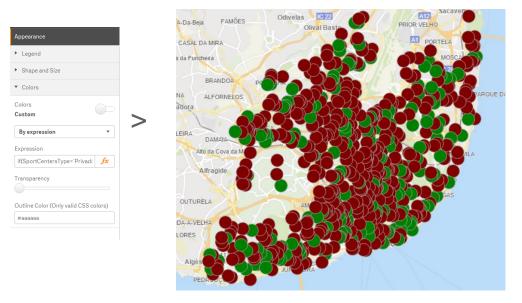
Before going into simplifying the representation of this information by creating bins where the corresponding points can be contained, we will only add a quick color coding to the bubbles so we can quickly identify which of these sport centers are public or private.

Go to the GeoAnalytics Bubble Layer extension properties, select **Appearance** and expand the **Colors** pane. Deactivate the *Auto* option and search in the dropdown for the **By expression** option. You'll notice it is not available. To make it available you must have the location specified via **Location**, **Size** pane instead of the default **ID** Pane.

Go to the **Location**, **Size** pane and **add** the **measure** *SportCentersGeometry* without any kind of aggregation.



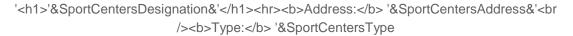
Go back to the **Appearance** pane and check if the **By Expression** option is now available within the **Colors** pane. Click on **fx** to add an expression with a simple if just like this one:

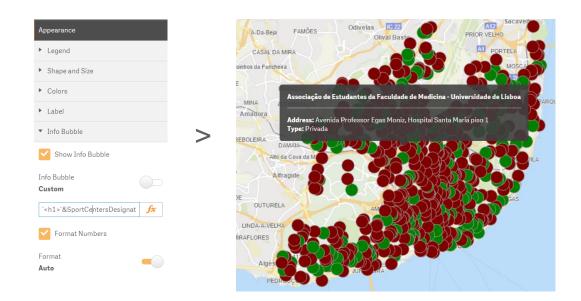


if(SportCentersType='Privada', red(),green())

Optionally, you can also add some contextual information in the **Info Bubble** such as the Sport Center name, type and address.

Go to the **Info Bubble** pane and deactivate the *Auto* option to access the **fx** expression window and add the information you would like to display, such as:





Now we will clean up the map a little bit using the **Binning** function. This function will allow us to create bins/blocks that cover the corresponding points with their area. This way we can display a few blocks and once a user selects the block we can drill into detail and show the actual points.

Note: always check the GeoAnalytics reference page for full details using the Binning Operation: <u>https://bi.idevio.com/wp-content/qlik/geoanalytics/releases/IdevioGeoAnalyticsConnector-</u> <u>5.13.0/doc/geoanalytics\_reference-September\_2018.html#Operation.Binning</u>

Let's start by going back to the **Data load editor**, make sure your cursor is positioned at the end of the script and click on the **Select data** icon ( ) of your GeoAnalytics Connection.

Select the **Operation: Binning** and we will now set it up so it gets the necessary information from our *SportCenters* table. Remember the importance of the little question marks on the wizard to help you in these configuration windows.

Regarding the **Operation parameters** select *Hexagonal* for the Shape of bins, input 0.0075 for the **Side length of bins (deg)** - 1 corresponds to approximately 10 000m, don't forget to refer the help - and set 1.5 to **Bin width-height-ratio**.

In the **Point Dataset** parameters input *SportCentersBinning* as **Name**, select *Loaded Table* for **Type**, input *SportCenters* for **Table Name** and input *SportCentersId*, *SportCentersGeometry* for **Table Fields** (key,geometry,...).

Once all the parameters are introduced click on **Next**. Here is a screenshot to let you double check how the parameters should be introduced:

Operation:	Binning	•	?
Operation parameters			
Shape of bins:	Hexagonal	۲	
Side length of bins (deg)*:	0.0075		
Bin width-height-ratio:	1.5		
Point Dataset			
Name:	SportCentersBinning		
Type:	Loaded table	۲	?
Table Name:	SportCenters		
Table Fields (key,geometry,)*:	SportCentersId,SportCentersGeometry	k	
Geometry Type:	Point	۲	
Only load distinct:	No	٠	
Location ID Suffix:			
CRS:	Auto		
*=Required field			

Once you get the **Select data to load** window, notice that there won't be any selectable table available. Don't worry as this is an expected behavior since it is something that is actually created and calculated via script and not via the receiving of an external table from a source. Click on **Insert Script** to continue.

Select data to load Tables Q	Hide script
	Fields Data preview Metadata Q
	/* Generated by GeoAnalytics for operation Binning
	SportCentersId, sqrt(3)/3 * subfield(TextBetween(SportCentersGeometry, '[', ']'), ', ',1)/0.006495 - 1/3 * subfield(TextBetween(SportCentersGeometry, '[', ']'), ', ',2). 2/3 * subfield(TextBetween(SportCentersGeometry, '[', ']'), ', ',2)/0.004330 as rr Resident 'SportCenters' ; [HexRoundTable1]: Load SportCentersId, 99. 
	Cancel Back Insert script

#### Your generated script should look like the following:

```
LIB CONNECT TO 'GeoAnalytics';
```

```
/* Generated by GeoAnalytics for operation Binning ------ */
[HexTemp1]:
Load
 SportCentersId,
 sqrt(3)/3 * subfield(TextBetween(SportCentersGeometry, '[', ']'), ', ',1)/0.006495 - 1/3 * subfield(TextBetween(SportCentersGeometry, '[',
']'), ',' ,2)/0.004330 as qq,
2/3 * subfield(TextBetween(SportCentersGeometry, '[', ']'), ', ', 2)/0.004330 as rr
Resident 'SportCenters'
[HexRoundTable1]:
Load
 SportCentersId,
 qq,
rr,
 -qq - rr as ss
Resident [HexTemp1]
drop table [HexTemp1];
[HexRoundTable2]:
Load
SportCentersId,
 qq,
rr,
 SS,
 fabs(Round(qq) - qq) as qDiff,
 fabs(Round(rr) - rr) as rDiff,
 fabs(Round(ss) - ss) as sDiff
Resident [HexRoundTable1]
drop table [HexRoundTable1];
[HexRoundTable3]:
Load
 SportCentersId,
 If(qDiff > rDiff and qDiff > sDiff, -Round(rr) - Round(ss), Round(qq)) as rq,
If(rDiff > sDiff and (qDiff <= rDiff or qDiff <= sDiff), -Round(qq) - Round(ss), Round(rr)) as rr,
If((qDiff <= rDiff or qDiff <= sDiff) and (rDiff <= sDiff), -Round(qq) - Round(rr), Round(ss)) as rs
Resident [HexRoundTable2]
;
```

drop table [HexRoundTable2];
[HexToCoordTable]: Load SportCentersId, (sqrt(3) * rq + sqrt(3)/2 * rr) * 0.006495 as lon, (3/2 * rr) * 0.004330 as lat Resident [HexRoundTable3]
; drop table [HexRoundTable3];
<pre>[HH0_0075W0_01125GridTable]: Load SportCentersId, '[[[[' &amp; Num(lon + 0.005625,'#########','.') &amp; ',' &amp; Num(lat + 0.002165,'##########','.') &amp; '],[' &amp; Num(lon + 0.005625,'#########','.') &amp; ',' &amp; Num(lat + 0.002165,'##########','.') &amp; '],[' &amp; Num(lon + -0.005625,'##########','.') &amp; ',' &amp; Num(lat + -0.002165,'##############','.') &amp; '],[' &amp; Num(lon + -0.005625,'#########','.') &amp; ',' &amp; Num(lat + -0.002165,'###############','.') &amp; '],[' &amp; Num(lon + 0.005625,'#########','.') &amp; ',' &amp; Num(lat + -0.002165,'##############','.') &amp; '],[' &amp; Num(lon + 0.005625,'#########','.') &amp; ',' &amp; Num(lat + -0.002165,'################','.') &amp; '],[' &amp; Num(lon + 0.005625,'#########','.') &amp; ',' &amp; Num(lat + -0.002165,'####################################</pre>

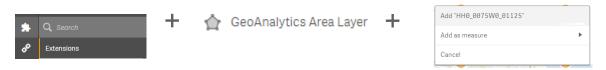
```
drop table [HexToCoordTable];
```

/\* End GeoAnalytics operation Binning ------ \*/

Load your data and test it out returning to your previously created sheet.



Add the GeoAnalytics Area Layer to your sheet and drag the HH0\_0075W0\_01125 field to it.



As a suggestion, make all the other layers invisible while we work in the look of our bins.

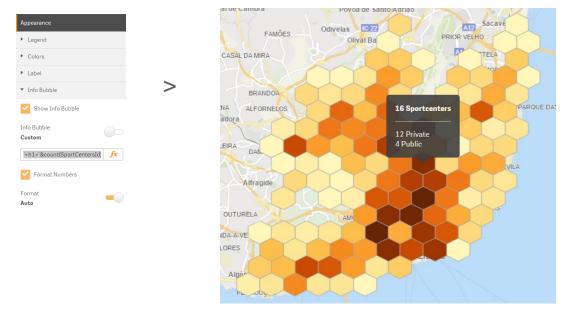


To bring some additional information we can color the bins based on the number of Sport Centers they have. Drag the field *SportCentersID* over your **GeoAnalytics Area Layer** extension, select **Color by**, select **By measure** and choose **Count(SportCentersId)**.



Represent also the overall resume of each bin by adding a custom **Info Bubble** with an expression such as

# '<h1>'&count(SportCentersId) &' SportCenters</h1><hr />'&count({<SportCentersType={"Privada"}>}SportCentersId)&' Private<br />'&count({<SportCentersType={"Pública"}>}SportCentersId)&' Public<br />'



Finally, we will add a drill down interaction between our Area Layer with the bins and our Bubble layer that contains the sport centers locations. To achieve this we will generate an activation/deactivation of the layer based on the map zoom. Make sure both layers are visible before starting the configuration.

Go to the **GeoAnalytics Area Layer** extensions properties for the Binning areas and select the **Layer Options** pane. Set the **Zoom Limits In** to range starting in 7 and **Out** ending in 160 000.

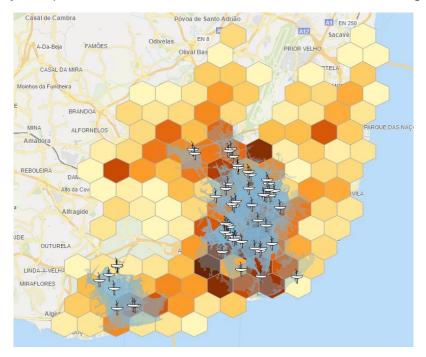
Next, go to the **GeoAnalytics Bubble Layer** extensions properties for the sport centers locations and select the **Layer Options** pane. Set the **Zoom Limits In** to range starting in 0 and **Out** ending in 7.

Now if you zoom in the map or select a bin you should be able to see only the correct information based on your zoom level.

# **Exercise 4: Using Within**

We have now information from water spots as well as sport centers. If we try and make visible all the layers, we can have 2 issues.

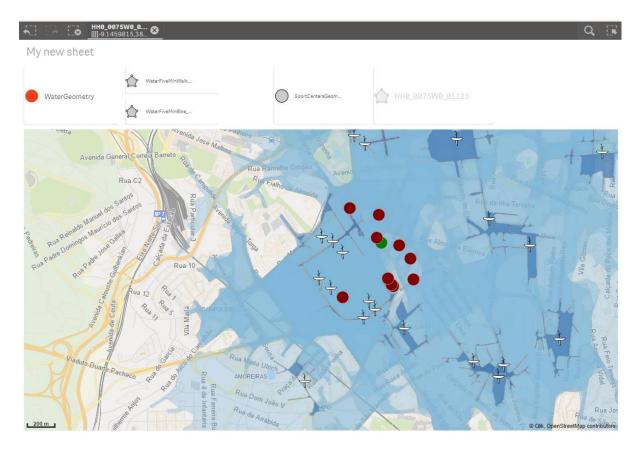
**The first issue**, and simpler one to solve, is the layering order so we can make sure representations are correct. In my example below the travel areas from exercise 2 are over the binning areas.



Here is a suggestion of how each extension Layer Options, Draw Order could be setup:

Extension	Draw Order Category	Draw Order Adjustment
GeoAnalytics Area Layer: Binning areas	Area Layer	5
GeoAnalytics Bubble Layer: Sport Centers	Area Layer	4
GeoAnalytics Bubble Layer: Water Spots	Area Layer	3
GeoAnalytics Area Layer: Five Minues Walking	Area Layer	2
GeoAnalytics Area Layer Five Minutes Cycling	Area Layer	1

The second issue is the one you get when you select a binning area and this has no association with the water spots.



To solve this issue you can use another of the functions available in the Qlik GeoAnalytics Connector: **Within**. The **Within** function allows you to indicate two datasets and find out which points are within/correlated between them.

Note: always check the GeoAnalytics reference page for full details using the Within Operation: <u>https://bi.idevio.com/wp-content/qlik/geoanalytics/releases/IdevioGeoAnalyticsConnector-</u> <u>5.13.0/doc/geoanalytics\_reference-September\_2018.html#Operation.Within</u>

In our case, we have some areas, the binning areas, that will certainly have water spots within them. Let's start by going to the **Data load editor**, make sure your cursor is positioned at the end of the script and click on the **Select data** icon (<sup>III</sup>) of your GeoAnalytics Connection and select the **Operation:** Within.

Let's start by indicating which is the **Dataset to test within on**: the water spots. Select *Loaded Table* for **Type**, insert *DrinkingWater* as **Table Name** and insert *WaterId*,*WaterGeo*metry as **Table Fields** (key,geometry,...).

Now we will indicate which is the **Area dataset:** the sport centers. Select *Loaded Table* for **Type**, insert *HH0\_0075W0\_01125GridTable* as **Table Name**, insert *SportCentersId*,*HH0\_0075W0\_01125* as **Table Fields (key,geometry...)**, select *Polygon* as **Geometry Type** and click on **Next**.

Step 1 - Select Operation		
Operation:	Within	• ?
Dataset to test within on		
Name:	Enclosed	
Туре:	Loaded table	▲ 5
Table Name:	DrinkingWater	
Table Fields (key,geometry,)*:	WaterId,WaterGeometry	
Geometry Type:	Point	T
Only load distinct:	No	T
Location ID Suffix:		
CRS:	Auto	
Area dataset		
Name:	Enclosing	
Type:	Loaded table	• ?
Table Name:	HH0_0075W0_01125GridTable	
Table Fields (key,geometry,)*:	SportCentersId,HH0_0075W0_01125	
Geometry Type:	Polygon	•
Only load distinct:	No	T
Location ID Suffix:		
CRS:	Auto	
*=Required field		
		Cancel Next

The resulting **Select data to load** window will give you 3 resulting tables. Of our interest is only the table that has the relations between both our Waterld and SportCentersId fields. Check the **WithinAssociations** table and click on **Insert script**.

Select data to load Tables Q. Filter tables			Hide script
WithinAssociations 2			
Enclosed	Fields		
Enclosing	Data preview Metadata		<b>Q</b> Filter fields
	₩aterId	SportCentersId	
	<pre>/* Generated by GeoAnalytics for operation Within Let [EnclosedInlineTable] = 'Materid' &amp; Chr(9) &amp; 'I Let numRous = NoOfRour('DrinkingMater'); Let chunkSize = 1000; Let chunks = numRous;ChunkSize; For n = 0 to chunks Let chunk = n*ChunkSize; For i = 0 To chunkSize; For i = 0 To chunkSize; Let roukr = chunks; Let roukr = chunks; For i = 0 To chunkSize; For fach f In WaterId', 'WaterGeo For Each f In WaterId', 'WaterGeo For Each f In VaterId', 'WaterId', 'WaterGeo For Each f In VaterId', 'WaterId', 'Wate</pre>	WaterGeometry';	6/===36=1 -10==1=1=10 =11=0000011 -
			Cancel Back Insert script

#### The following script is expected to have been generated for you:

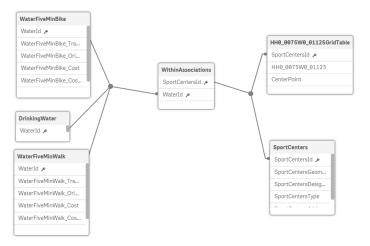
LIB CONNECT TO 'GeoAnalytics';

/\* Generated by GeoAnalytics for operation Within ------\*/
Let [EnclosedInlineTable] = 'WaterId' & Chr(9) & 'WaterGeometry';
Let numRows = NoOfRows('DrinkingWater');
Let chunkSize = 1000;
Let chunkSize = 1000;
Let chunks = numRows/chunkSize;
For n = 0 to chunks
Let chunkText = ";
Let chunkText = ";
Let chunk = n\*chunkSize;

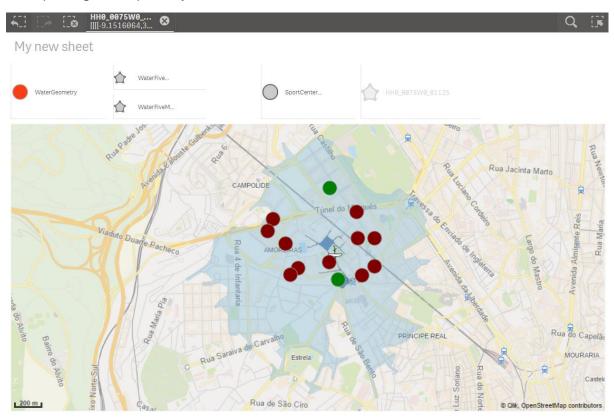
```
For i = 0 To chunkSize-1
                                           Let row = ";
                                           Let rowNr = chunk+i;
                                           Exit for when rowNr \geq numRows:
                                           For Each f In 'WaterId', 'WaterGeometry'
                                                                row = row & Chr(9) & Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(R
'DrinkingWater'), Chr(39), '\u0027'), Chr(34), '\u0022'), Chr(91), '\u005b'), Chr(47), '\u002f'), Chr(42), '\u002a'), Chr(59), '\u003b');
                                           Next
                                           chunkText = chunkText & Chr(10) & Mid('$(row)', 2);
                     Next
                     [EnclosedInlineTable] = [EnclosedInlineTable] & chunkText;
Next
chunkText=''
Let [EnclosingInlineTable] = 'SportCentersId' & Chr(9) & 'HH0_0075W0_01125';
Let numRows = NoOfRows('HH0_0075W0_01125GridTable');
Let chunkSize = 1000;
Let chunks = numRows/chunkSize;
For n = 0 to chunks
                    Let chunkText = ";
                     Let chunk = n*chunkSize;
                     For i = 0 To chunkSize-1
                                           Let row = ";
                                           Let rowNr = chunk+i;
                                           Exit for when rowNr >= numRows;
                                           For Each f In 'SportCentersId', 'HH0_0075W0_01125'
                                                                row = row & Chr(9) & Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace)), $(rowNr),
'HH0_0075W0_01125GridTable'), Chr(39), '\u0027'), Chr(34), '\u0022'), Chr(91), '\u005b'), Chr(47), '\u002f'), Chr(42), '\u002a'),
                     Chr(59), '\u003b');
                                           Next
                                           chunkText = chunkText & Chr(10) & Mid('$(row)', 2);
                     Next
                     [EnclosingInlineTable] = [EnclosingInlineTable] & chunkText;
Next
chunkText="
[WithinAssociations]:
SQL SELECT [WaterId], [SportCentersId] FROM Within(enclosed='Enclosed', enclosing='Enclosing')
DATASOURCE Enclosed INLINE tableName='DrinkingWater', tableFields='WaterId,WaterGeometry', geometryType='POINT',
loadDistinct='NO', suffix='', crs='Auto' {$(EnclosedInlineTable)}
DATASOURCE Enclosing INLINE tableName='HH0_0075W0_01125GridTable', tableFields='SportCentersId,HH0_0075W0_01125',
geometryType='POLYGON', loadDistinct='NO', suffix='', crs='Auto' {$(EnclosingInlineTable)}
;
[EnclosedInlineTable] = ";
[EnclosingInlineTable] = ";
```

/\* End GeoAnalytics operation Within ------ \*/

You may now **load** your data and check for the results. As you can see, all the tables of our data model are associated.



And if you go your application sheet and select a binning area you will be able to see only the corresponding water spots of your selection.



# **Exercises Part 3 – Additional GeoAnalytics Connector exercises**

# **Exercise 1: Cluster**

Let's start by loading some Pharmacies location information and cluster them by proximity. In our use case, we will cluster them by (a max of) 2000m.

Similar to the previous exercises, load the information from:

## https://services.arcgis.com/1dSrzEWVQn5kHHyK/arcgis/rest/services/POISaude/FeatureServer/1/que ry?where=1%3D1&outFields=\*&f=pgeojson

Note: always check the GeoAnalytics reference page for full details using the Load Operation: <u>https://bi.idevio.com/wp-content/qlik/geoanalytics/releases/IdevioGeoAnalyticsConnector-5.13.0/doc/geoanalytics\_reference-September\_2018.html#Operation.Load</u>

On your Qlik Sense **Data Load Editor** click on the **Select data** icon (<sup>III</sup>) of your GeoAnalytics Connection and select the **Operation: Load**. Give *PharmaciesRaw* as **Name** for your *Dataset*, select *File based geo dataset* for **Type** and paste the link in **URL**. You may leave all the other options as suggested by default and click on **Next**.

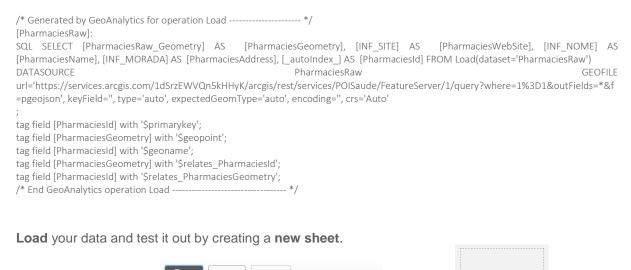
Operation:	Load	▼ 2
Jataset		
Name:	PharmaciesRaw	
Type:	File based geo dataset	<b>▼</b> ?
URL*:	https://services.arcgis.com/1dS	rzEWVQn5
Key Field:		
File Type:	Auto	•
Expected Geometry Type:	Auto	v
Character Encoding:		
CRS:	Auto	

We will not need all the fields available. You may **select and rename** *PharmaciesRaw\_Geometry* to *PharmaciesGeometry*, **select and rename** *INF\_SITE* to *PharmaciesWebSite*, **select and rename** *INF\_NOME* to *PharmaciesName*, **select and rename** *INF\_MORADA* to *PharmaciesAddress* and **select and rename** *\_autoIndex\_* to *PharmaciesId*. Once you've selected and renamed the fields you may click on **Insert Script**.

Select data to load		Hide s	cript
Tables $\rightarrow$			
Q, Filter tables			
PharmaciesRaw 5			
	Fields Data preview Metadata	Q Filter fields	
	PharmaclesGeom INF_DESCRIC Globa OBJEC COD PharmaclesWeb INF_TELEFO INF_FO IDTL. INF_F	PharmaciesNa	F 🖌
	/* Generated by GeoMalytics for operation Load	inder.] AS [Phareacia leider'', type='auto',	sId] FRC expects
	Can	el Back Ins	ert script

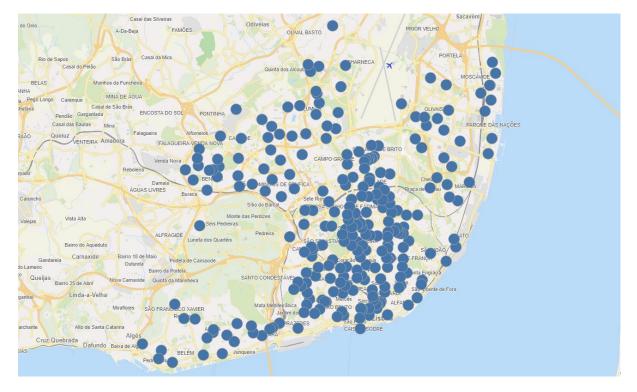
#### The following script is expected to have been generated for you:

LIB CONNECT TO 'GeoAnalytics';



		<i>(e)</i> <b>v</b>		Save 🚭	Exercises Part 3		<b>+</b>	
Loa	d data 🕑 🕂	😭 Арр	overview	V	5	+	Create new sheet	
Add	the GeoAnalytics	Map ext	ensic	on to your she	et.			
* 8	Q Search Extensions	+	8	GeoAnalytics	Мар			
Add	the GeoAnalytics	Bubble	Laye	<b>r</b> to your shee	t and drag	the Pharma	aciesGeometry f	ield to it.
*	Q Search	+	0	GeoAnalytics I	Bubble La.		Add "PharmaciesGeometry"	
в	Extensions		0	-			Add as measure	
							Cancel	

You are now able to see all the pharmacies available in Lisbon, Portugal.



We can see that in fact there are a lot of Pharmacies available within the city. To simplify the reading, one way to aggregate all these points can be using clusters. As mentioned previously, we will do the clustering aggregation based on a maximum distance 2000m from the center point of the cluster.

Return to the **Data load editor**, make sure your cursor is positioned at the end of the script and click on the **Select data** icon ( ) of your GeoAnalytics Connection.

Select the **Operation: Cluster** and we will now set it up so it gets the necessary information from our *PharmaciesRaw* table. Remember the importance of the little question marks on the wizard to help you in these configuration windows.

Note: always check the GeoAnalytics reference page for full details using the Cluster Operation: <u>https://bi.idevio.com/wp-content/qlik/geoanalytics/releases/IdevioGeoAnalyticsConnector-5.13.0/doc/geoanalytics\_reference-September\_2018.html#Operation.Cluster</u>

Regarding the **Operation parameters** input 2000 as **Distance (m)**.

In the **Point Dataset** parameters input *PharmaciesCluster* as **Name**, select *Loaded Table* for **Type**, input *PharmaciesRaw* for **Table Name** and input *PharmaciesId*, *PharmaciesGeometry* for **Table Fields** (key,geometry,...).

Once all the parameters are introduced click on **Next**. Here is a screenshot to let you double check how the parameters should be introduced:

Step 1 - Select Operation				
Operation:	Cluster	٣	?	
Operation parameters				
Distance (m)*:	2000			
Point Dataset				
Name:	PharmaciesCluster			
Type:	Loaded table	٣	?	
Table Name:	PharmaciesRaw			
Table Fields (key,geometry,)*:	PharmaciesId, PharmaciesGeometry			
Geometry Type:	Point	۳		
Only load distinct:	No	۳		
Location ID Suffix:				
CRS:	Auto			
	]	Cane	cel	Next

Once you get the **Select data to load** window, mark the **ClusterAssociations** and **Clusters** tables checkboxes to load all the fields of both tables and click on **Insert script**.

Select data to load Tables → Q. Filter tobles	Hide serijit
ClusterAssociations 2	
Clusters 3	Fields
PharmaciesCluster	Data preview Metadata Q Filter fields
	Clusters_ClusterID Clusters_ClusterCenter Clusters_Clusters_ClusterCenter
	<pre>/* Generated by GeoMnalytics for operation Cluster*/ Let [PharmacLesCluster:nilmeTable] = 'PharmacLesId' &amp; Chr(0) &amp; 'PharmacLesGeometry'; Let numburs - NoofRow('PharmacLesId' &amp; Chr(0) &amp; 'PharmacLesGeometry'; Let numburs/chunkles; for n= 0 to chunks Let chunks = '';</pre>
	Let chunk = n'chunkSie; For 1 = 0 = ChunkSie-1 Let roue = ": Let roue = chunk := nombrus; Let roue = chunk := nombrus; For Esch f In "Pharmaciesies", "Pharmaciesiesetry" rou = cours. Chrich L Belancobenlancobenlancobenlanoobenlanoobenlanoobenk/15/61: Sconder, "Pharmaciestaw", rho/10; "Ludott", rho/10; "Ludott, rho/10; "Ludott", rho/10; "Ludott, rho/10; "Ludot
	Cancel Back Insert script

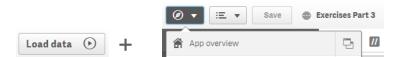
The following script is expected to have been generated for you:

LIB CONNECT TO 'GeoAnalytics';

/\* Generated by GeoAnalytics for operation Cluster ------ \*/ Let [PharmaciesClusterInlineTable] = 'PharmaciesId' & Chr(9) & 'PharmaciesGeometry'; Let numRows = NoOfRows('PharmaciesRaw'); Let chunkSize = 1000;

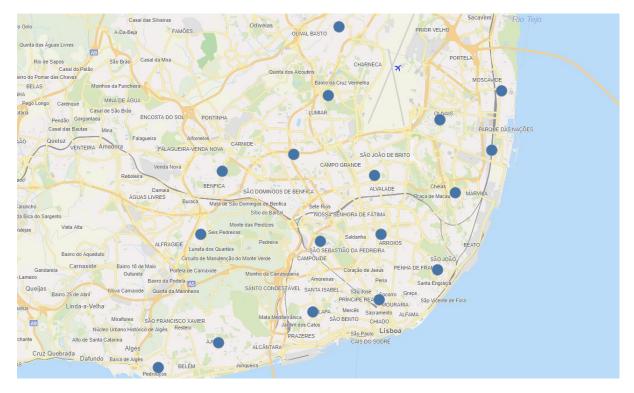
```
Let chunks = numRows/chunkSize;
For n = 0 to chunks
          Let chunkText = ";
          Let chunk = n*chunkSize;
          For i = 0 To chunkSize-1
                     Let row = ";
                     Let rowNr = chunk+i;
                     Exit for when rowNr >= numRows;
                     For Each f In 'PharmaciesId', 'PharmaciesGeometry'
                                row = row & Chr(9) & Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Peek('$(f)',
$(rowNr), 'PharmaciesRaw'), Chr(39), '\u0027'), Chr(34), '\u0022'), Chr(91), '\u005b'), Chr(47), '\u002f'), Chr(42), '\u002a'), Chr(59), '\u003b'),
Chr(125), '\u007d'), Chr(123), '\u007b');
                     Next
                     chunkText = chunkText & Chr(10) & Mid('$(row)', 2);
          Next
          [PharmaciesClusterInlineTable] = [PharmaciesClusterInlineTable] & chunkText;
Next
chunkText="
[ClusterAssociations]:
SQL SELECT [PharmaciesId], [Clusters_ClusterID] FROM Cluster(distance='2000', points='PharmaciesCluster')
                                        INLINE
                                                     tableName='PharmaciesRaw',
                                                                                       tableFields='PharmaciesId,PharmaciesGeometry',
DATASOURCE
                  PharmaciesCluster
geometryType='POINT', loadDistinct='NO', suffix='', crs='Auto' {$(PharmaciesClusterInlineTable)}
SELECT [Clusters_ClusterID], [Clusters_ClusterCenter], [Clusters_PointCount] FROM Clusters;
[Clusters]:
SQL LOAD * FROM Clusters;
tag field [PharmaciesId] with '$primarykey';
tag field [Clusters_ClusterID] with '$primarykey';
tag field [Clusters_ClusterCenter] with '$geopoint';
tag field [Clusters_ClusterID] with '$geoname';
tag field [Clusters_ClusterCenter] with '$relates_Clusters_ClusterID';
tag field [Clusters_ClusterID] with '$relates_Clusters_ClusterCenter';
[PharmaciesClusterInlineTable] = ";
/* End GeoAnalytics operation Cluster ------ */
```

Load your data and return to your previously created sheet to test your results.



Add a new GeoAnalytics Bubble Layer to your sheet and drag the ClustersCenter field to it.

Here is an example of the expected result. For clarity's sake at this point the Pharmacies Bubble Layer is deactivated.



To bring some clarity into our map with our new developments, let's give some layout configurations to both the Bubble Layers representing the Clusters as well as the Pharmacies.

As a simple starting point, we will give different coloring and sizing to both Bubble Layers.

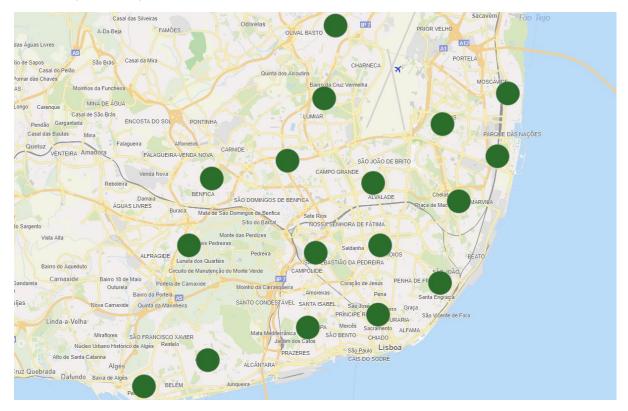
Start with the Clusters GeoAnalytics Bubble Layer properties, select **Appearance** and expand the **Shape and Size** pane. Select a larger radius, for example minimum of 24 (we can also vary the size based on an expression, which is covered further in this exercise).

Appearance	
▶ Legend	
<ul> <li>Shape and Size</li> </ul>	
Shape	
Bubbles	•
Radius Min 24 - Max 24	
Min Radius Value <b>Auto</b>	-
Max Radius Value <b>Auto</b>	-

Now in the same **Appearance** properties pane expand **Colors.** Turn off the **Auto** selector to enter in **Custom** mode. Select **Single Color** from the dropdown and a color of your choosing (I will select green).

Appearance
▶ Legend
<ul> <li>Shape and Size</li> </ul>
▼ Colors
Colors
Custom
Single color 🔹
Color 📃 🔻
Transparency
Outline Color (Only valid CSS colors)
#aaaaaa

### Here is a quick example of the outcome:



Now configure the Pharmacies Bubble Layer, but making the bubbles smaller. Don't forget to make this layer visible again if you've hidden it before.

Select the Pharmacies GeoAnalytics Bubble Layer properties, select **Appearance** and expand the **Shape and Size** pane. Select a smaller radius, for example minimum of 7.

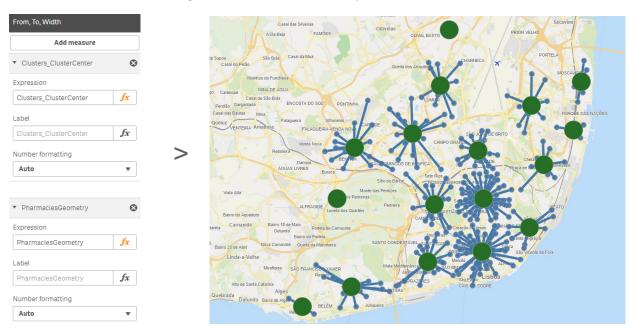


Now that we can see both clusters and pharmacies in the same geographical representation we can take it a step further. We can show visually how actually each pharmacy relates with it's cluster. For that we can use the **GeoAnalytics Line Layer** to draw connection line between the Clusters and the Pharmacies. Finally, we could also add a quick hover context at the Cluster bubble to identify how many Pharmacies belong to that Cluster.

Starting with the lines, add a **GeoAnalytics Line Layer** to your sheet and drag *PharmaciesId* as our relationship field to be used as ID/Dimension.



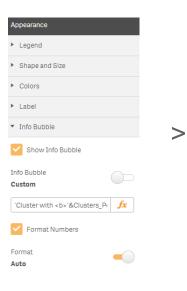
Go to the **From**, **To**, **Width** pane to configure that information needed to draw the line. Add *Clusters\_ClusterCenter* as Origin and *PharmaciesGeometry* as Destination.



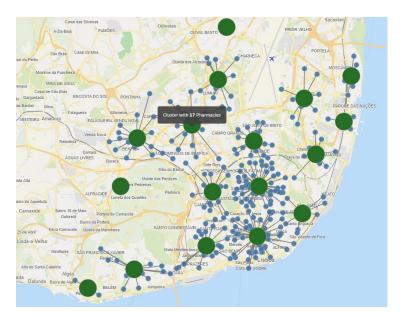
You can also reduce the line width and give it a different color to improve readability. Select **Appearance**, go to the **Shape and Size** pane and set a smaller width, such as 2. For the color, go to the **Colors** pane and select grey (#7b7a78).



As last steps, we can now configure the hover behavior for the Cluster related GeoAnalytics Bubble Layer. Jump to the Clusters' **GeoAnalytics Bubble Layer** properties and select the **Appearance** pane. Under the **Info Bubble** pane deactivate the **Auto** feature and enter the following HTML in the expression's text area:



'Cluster with <b>'&Clusters\_PointCount&'</b> Pharmacies'

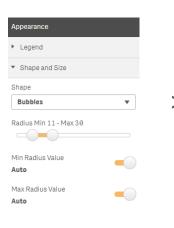


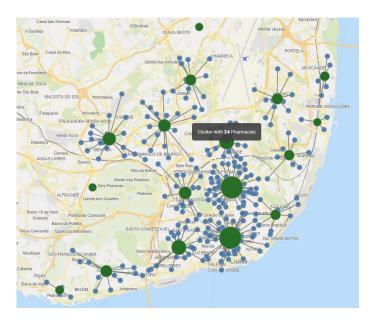
And we can also make the cluster size vary based on the number of points (pharmacies) associated with the cluster. Once more, go to the Clusters' **GeoAnalytics Bubble Layer** properties and select the **Location, Size** pane and add a new expression:

Location, Size	
Add measure	
Clusters_ClusterCenter	8
<ul> <li>Sum(Clusters_PointCount)</li> </ul>	8
Expression	
Sum(Clusters_PointCount)	fx
Label	
Sum(Clusters_PointCount)	fx
Number formatting	
Auto	•

#### Sum(Clusters\_PointCount)

Now go to the **Appearance** pane and (re)configure the **Shape and** Size, giving a minimum value of 11 and a maximum value of 30.





# **Exercise 2: Routes**

Another representation that we could leverage based on our Clusters' calculated data is the route. For example, which would be the fastest path by car from a Cluster to a Pharmacy.

Go to the **Data load editor**, make sure your cursor is positioned at the end of the script. We will need to create a new table that has the required information so the function can calculate correctly. This will

be the route's origin and destiny as well as the corresponding Pharmacies and Clusters IDs. We must make sure it is a correct "one to many" identification instead of a "many to many".

To do so, add the following snippet to your script:

RoutesRaw: load PharmaciesId, PharmaciesGeometry as Origin Resident [Pharmacies]; join (RoutesRaw) load PharmaciesId, Clusters\_ClusterID as DestinationId Resident [ClusterAssociations]; join (RoutesRaw) load Clusters\_ClusterID as DestinationId, Clusters\_ClusterCenter as Destination Resident Clusters;

Once again, make sure your cursor is positioned at the end of the script and click on the **Select data** icon ( <sup>III</sup>) of your GeoAnalytics Connection. Select the **Operation: Routes**.

Note: always check the GeoAnalytics reference page for full details using the Routes Operation: https://bi.idevio.com/wp-content/qlik/geoanalytics/releases/IdevioGeoAnalyticsConnector-5.13.0/doc/geoanalytics\_reference-September\_2018.html#Operation.Routes

In the **Operation parameters** section for the **Criteria** choose *Fastest*, for **Transportation** select *Car*, as the **Destination Field** type *Destination* and for the **Destination Geometry Type** leave selected *Point*.

In the **End points** section input the **Name** as *RoutesCalculated*, select the **Type** as *Loaded table*, **Table Name** as *RoutesRaw*, **Table Fields (key,geometry,...)** with *PharmaciesId*, *Origin*, *Destination*. Select the **Geometry Type** as *Point* and leave the rest of the fields as default.

When all the options are filed, click on Next. Here is an example of the configurations above:

Operation:	Routes	٣	?
peration parameters			
Criteria:	Fastest	•	
ransportation:	Car	*	
Destination Field*:	Destination		
Destination Geometry Type:	Point	•	
Destination Location ID Suffix:			
nd points			
lame:	RoutesCalculated		
ype:	Loaded table	•	?
able Name:	RoutesRaw		
able Fields (key,geometry,)*:	PharmaciesId,Origin,Destination		
Seometry Type:	Point	٣	
Only load distinct:	No	*	
ocation ID Suffix:			
CRS:	Auto		

As a result, the connector will show the tables selection wizard. Check the **Routes** table, make sure all the fields are selected and click in **Insert Script**.

lect data to load	÷					Hide sc
Filter tobles	5					
RoutesCalculated	Fields	weview Metadata	ta			Q Filter fields
		armaciesId	RoutesCalculated_Distance	RoutesCalculated_Duration	RoutesCalculated_Status	RoutesCalculated_PathLowRes
		erated by GeoAnal	ytics for operation Routes	"/" "0'eie' & chr/9) & 'Destination':		
	Let   Let r Let c	mRows = NoOfRows(' unkSize = 1000) unks = numRows/chu = 0 to chunks Let chunkText = Let chunk = n*ch For i = 0 To chu Let row Let row Exit for	unkSize;  hunkSize; unkSize-1			

#### Here is an example of the expected script generated by the connector:

```
LIB CONNECT TO 'GeoAnalytics';
```

```
/* Generated by GeoAnalytics for operation Routes ------ */
Let [RoutesCalculatedInlineTable] = 'PharmaciesId' & Chr(9) & 'Origin' & Chr(9) & 'Destination';
Let numRows = NoOfRows('RoutesRaw');
Let chunkSize = 1000;
Let chunks = numRows/chunkSize;
For n = 0 to chunks
          Let chunkText = ";
          Let chunk = n*chunkSize;
          For i = 0 To chunkSize-1
                     Let row = ";
                     Let rowNr = chunk+i;
                     Exit for when rowNr >= numRows;
                     For Each f In 'PharmaciesId', 'Origin', 'Destination'
                               row = row & Chr(9) & Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Peek('$(f)',
$(rowNr), 'RoutesRaw'), Chr(39), '\u0027'), Chr(34), '\u0022'), Chr(91), '\u005b'), Chr(47), '\u002f'), Chr(42), '\u002a'), Chr(59), '\u003b'),
Chr(125), '\u007d'), Chr(123), '\u007b');
                     Next
                     chunkText = chunkText & Chr(10) & Mid('$(row)', 2);
          Next
          [RoutesCalculatedInlineTable] = [RoutesCalculatedInlineTable] & chunkText;
Next
chunkText=''
[Routes]:
        SELECT
                    [PharmaciesId],
                                        [RoutesCalculated_Distance],
                                                                         [RoutesCalculated_Duration],
                                                                                                            [RoutesCalculated_Status],
SQL
[RoutesCalculated_PathLowRes] FROM Routes(criteria='fastest', transportation='car', dest='Destination', destType='Point', suffix='',
dataset='RoutesCalculated')
DATASOURCE RoutesCalculated INLINE tableName='RoutesRaw', tableFields='PharmaciesId,Origin,Destination', geometryType='POINT',
loadDistinct='NO', suffix=", crs='Auto' {$(RoutesCalculatedInlineTable)}
tag field [PharmaciesId] with '$primarykey';
tag field [RoutesCalculated_PathLowRes] with '$geoline';
tag field [PharmaciesId] with '$geoname';
tag field [RoutesCalculated_PathLowRes] with '$relates_PharmaciesId';
tag field [PharmaciesId] with '$relates_RoutesCalculated_PathLowRes';
[RoutesCalculatedInlineTable] = ";
/* End GeoAnalytics operation Routes ------ */
```

Load your data and return to your previously created sheet to test your results.

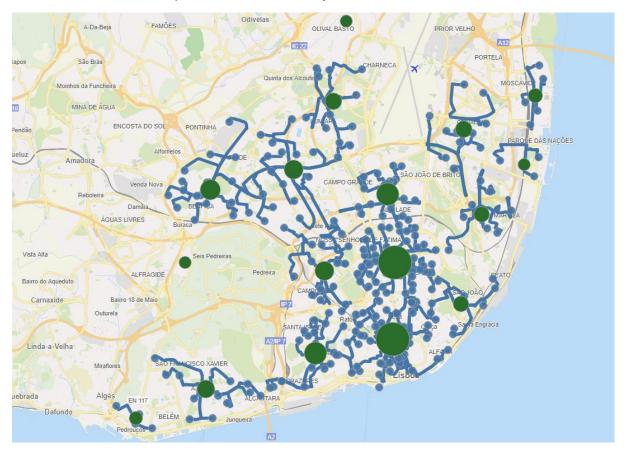
	🕖 🔻 🖃 🖉 Exerc	cises Pa	rt 3
Load data 🕑 🕂	App overview	Ð	//

Add a new **GeoAnalytics Line Layer** to your sheet and drag the **RoutesCalculated\_PathLowRes** field to it..



- I - I - I - I - I - I - I - I - I - I	
Add "RoutesCalculated_PathLowRes"	
Add as measure	•
Cancel	

You can now see represented on your map the Routes for the fastest path by car between the Pharmacies and the center points of the Clusters they are allocated to.



You can add some more visual information by fine tuning some appearance properties, such as reduce the path width and color it based on the *expected* travel time. We will force a color coding of our own instead of using the default **Colors** by Measure available in the **Appearance** properties pane.

If you go now to the **GeoAnalytics Line Layer**'s properties, under the **Appearance** pane and select **Colors**, when indicating you want to use custom colors, you should only be able to select either *Single Color, by dimension* or *by measure* as color source options. To enable the *by expression* option you must add a measure to the extension to "boot up" the extensions' hypercube.

Drag a field, such as *RoutesCalculated\_Duration* to the extension and add it as an expression, such as *sum(RoutesCalculated\_Duration)*. This will make sure the extension has an hypercube.



You will now see that under the **Appearance** pane **Colors**, the option *By expression* is available. The problem is that now the extension is also trying to use the expression we've specified as a location. To make sure it keeps using the routes provided before go to the **Location Options** and under **Location Source** select *Dimension*.

٢.	
ID	
From, To, Width	
Layer Options	
Location Options	
Location Source	
Dimension	•
Location Service URL	
i:///pointgeom/default	fx
Load via Server	
Location Service Type	
Geometry Service	•
Advanced Location Off	
Country	
Any	•
Туре	
Any	•
Manual Remap	

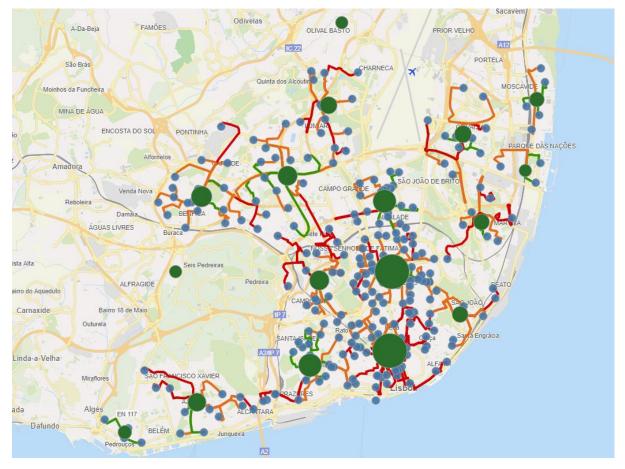
As a final step, we can now specify our color expression for the routes. Go once more to the **Appearance** pane, select **Colors** and the option *By expression*. Add the following expression:

=if(sum(RoutesCalculated\_Duration)<180,rgb(48,150,0), if(sum(RoutesCalculated\_Duration)>=180 and sum(RoutesCalculated\_Duration)<300,rgb(229,110,20),rgb(210,0,1)))

The expression translates the sum of seconds to the following criteria:

Time	Color
< 3 minutes	Green
>= 3 minutes and < 5 minutes	Orange
>= 5 minutes	Red





# **Exercise 3: Closest**

We were able to calculate all the routes between the Clusters and the Pharmacies related with them. Another use case can be to only show the X closest. For example, which are the 10 closest Pharmacies to each Cluster center point at a maximum of 5 minutes of travel time?

To answer this use case, the GeoAnalytics Connector has a function, named Closest, that does this calculation.

Go to the **Data load editor**, make sure your cursor is positioned at the end of the script. **Select data** icon ( <sup>III</sup>) of your GeoAnalytics Connection and select the **Operation: Closest**.

Note: always check the GeoAnalytics reference page for full details using the Closest Operation: https://bi.idevio.com/wp-content/qlik/geoanalytics/releases/IdevioGeoAnalyticsConnector-5.13.0/doc/geoanalytics\_reference-September\_2018.html#Operation.Closest

In the **Operation parameters** section for the **Distance unit** choose *Minutes*. In the **Maximum distance** (cost) between geometries field type 5. For the **Distance type** select *Car* and **Closest count** type in *10*.

For the section **Dataset containing geometries to measure from,** under **Name** type *Cluster*. The **Type** is *Loaded table* and the **Table name** is *Clusters*. The **Table fields (key,geometry,...)** are

*Clusters\_ClusterID, Clusters\_ClusterCenter.* The **Geometry type** is *Point* and you can leave the rest of the fields of this section as default.

For the section **Dataset containing geometries to measure to** under **Name** type *Pharma*. The **Type** is *Loaded table* and the **Table name** is *PharmaciesRaw*. The **Table fields (key,geometry...)** are *PharmaciesId,PharmaciesGeometry*. The **Geometry type** is *Point* and you can leave the rest of the fields of this section as default.

When all the options are filed, click on **Next**. Here is an example of the configurations above:

Step 1 - Select Operation			
Operation:	Closest	۳	?
Operation parameters			
Distance unit:	Minutes	۳	
Maximum distance (cost) between geometries*:	5		
Distance type:	Car	٣	
Closest count:	10	_	
Dataset containing geometries to m	easure from		
Name:	Cluster		
Type:	Loaded table	۳	?
Table Name:	Clusters		
Table Fields (key,geometry,)*:	Clusters_ClusterID,Clusters_ClusterCen	ter	
Geometry Type:	Point	۳	
Only load distinct:	No	۳	
Location ID Suffix:			
CRS:	Auto		
Dataset containing geometries to m	easure to		
Name:	Pharma		
Type:	Loaded table	۳	?
Table Name:	PharmaciesRaw		
Table Fields (key,geometry,)*:	PharmaciesId, PharmaciesGeometry		
Geometry Type:	Point	۳	
Only load distinct:	No	۳	
Location ID Suffix:			
CRS:	Auto		
*=Required field			
		Cane	cel Next

As a result, the connector will show the tables selection wizard. Check the **ClosestAssociations** table, make sure all the fields are selected and click in **Insert Script**.

Select data to load				Hide script
Tables -				
ClosestAssociations				
Cluster	Fields Data preview Metadata			Q. Filter fields
Pharma	Date preview Metadata			Q Filter pelas
	Cluster_Pharma_RelationId	D PharmaciesId	Cluster_Pharma_Distance	Cluster_Pharma_Route
	/* Generated by GeoAnalytics for operation Closest	•/		
	Let [PharmaInlineTable] = 'PharmaciesRaw'); Let numRows = NoOfRows('PharmaciesRaw');	metry';		
	Let chunkSize = 1000; Let chunks = numRows/chunkSize;			_
	<pre>For n = 0 to chunks Let chunkText = ''; Let chunk = n*chunkSize;</pre>			
	For i = 0 To chunkSize-1 Let row = '';			
	Let rowNr = chunk+i; Exit for when rowNr >= numRows;			
	For Each f In 'PharmaciesId', 'PharmaciesGeometry'	e/Renlace/Renlace/Renlace/Renlace/	Desk('\$(f)' \$(roudir) 'DharmariasRaw')	rhe(30) '\u8827') rhe(34) '\u8822') rhe(01) '\u88
				Cancel Back Insert script

GeoAnalytics Enablement | 59

Important note, the resulting table is a new associations table between the Pharmacies and the Clusters. When reloading you will have a synthetic key. Don't forget to optimize your datamodel accordingly, especially when your application is going to production.

Here is an example of the expected script generated by the connector:

LIB CONNECT TO 'GeoAnalytics'; /\* Generated by GeoAnalytics for operation Closest ------ \*/ Let [PharmaInlineTable] = 'PharmaciesId' & Chr(9) & 'PharmaciesGeometry'; Let numRows = NoOfRows('PharmaciesRaw'); Let chunkSize = 1000;Let chunks = numRows/chunkSize; For n = 0 to chunks Let chunkText = "; Let chunk = n\*chunkSize; For i = 0 To chunkSize-1 Let row = ": Let rowNr = chunk+i; Fxit for when rowNr  $\geq$ = numRows: For Each f In 'PharmaciesId', 'PharmaciesGeometry' row = row & Chr(9) & Replace(R \$(rowNr), 'PharmaciesRaw'), Chr(39), '\u0027'), Chr(34), '\u0022'), Chr(91), '\u005b'), Chr(47), '\u002f'), Chr(42), '\u002a'), Chr(59), '\u003b'), Chr(125), '\u007d'), Chr(123), '\u007b'); Next chunkText = chunkText & Chr(10) & Mid('\$(row)', 2); Next [PharmaInlineTable] = [PharmaInlineTable] & chunkText; Next chunkText=" Let [ClusterInlineTable] = 'Clusters\_ClusterID' & Chr(9) & 'Clusters\_ClusterCenter'; Let numRows = NoOfRows('Clusters'); Let chunkSize = 1000; Let chunks = numRows/chunkSize; For n = 0 to chunks Let chunkText = "; Let chunk = n\*chunkSize; For i = 0 To chunkSize-1 Let row = "; Let rowNr = chunk+i; Exit for when rowNr  $\geq$  numRows; For Each f In 'Clusters\_ClusterID', 'Clusters\_ClusterCenter' row = row & Chr(9) & Replace(R \$(rowNr), 'Clusters'), Chr(39), '\u0027'), Chr(34), '\u0022'), Chr(91), '\u005b'), Chr(47), '\u002f'), Chr(42), '\u002a'), Chr(59), '\u003b'), Chr(125), '\u007d'), Chr(123), '\u007b'); Next chunkText = chunkText & Chr(10) & Mid('\$(row)', 2); Next

Next [ClusterInlineTable] = [ClusterInlineTable] & chunkText;

```
chunkText=''
```

Next

[ClosestAssociations]:

SQL SELECT [Cluster\_Pharma\_RelationId], [Clusters\_ClusterID], [PharmaciesId], [Cluster\_Pharma\_Distance], [Cluster\_Pharma\_Route] FROM Closest(costUnit='Minutes', distance='5', distanceType='car', closestCount='10', dataset1='Cluster', dataset2='Pharma')

DATASOURCE Cluster INLINE tableName='Clusters', tableFields='Clusters\_ClusterID,Clusters\_ClusterCenter', geometryType='POINT', loadDistinct='NO', suffix='', crs='Auto' {\$(ClusterInlineTable)}

DATASOURCE Pharma INLINE tableName='PharmaciesRaw', tableFields='PharmaciesId,PharmaciesGeometry', geometryType='POINT', loadDistinct='NO', suffix='', crs='Auto' {\$(PharmaInlineTable)}

tag field [Cluster\_Pharma\_RelationId] with '\$primarykey';

tag field [Cluster\_Pharma\_Route] with '\$geoline';

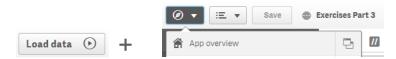
tag field [Cluster\_Pharma\_RelationId] with '\$geoname';

tag field [Cluster\_Pharma\_Route] with '\$relates\_Cluster\_Pharma\_RelationId';

tag field [Cluster\_Pharma\_RelationId] with '\$relates\_Cluster\_Pharma\_Route';

[ClusterInlineTable] = "; [PharmaInlineTable] = "; /\* End GeoAnalytics operation Closest ------ \*/

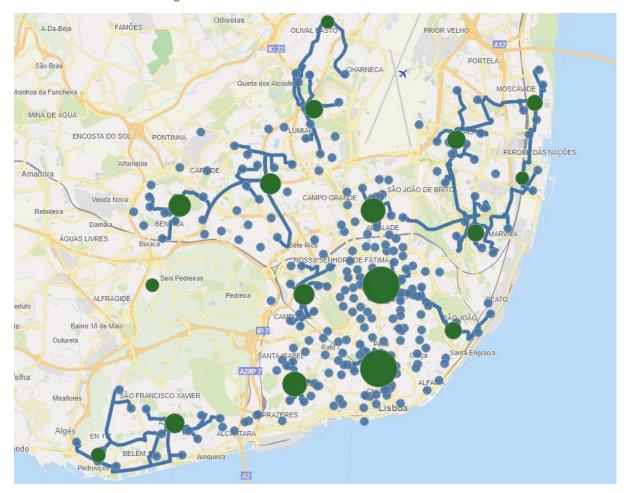
Load your data and return to your previously created sheet to test your results.



Add a new GeoAnalytics Line Layer to your sheet and drag the Cluster\_Pharma\_Route field.



You can now see represented on your map the Routes for the 10 closest Pharmacies to each Cluster. Notice that albeit Pharmacies can be perceived close to a Cluster on map it doesn't mean they have the fastest travel time, being in fact associated to a further Cluster that has faster access roads.



## **Exercise 4: Dissolve**

The next exercise will show you how to dissolve multiple polygons into a single larger polygon. To achieve this, we will use the GeoAnalytics Connector function Dissolve.

Jump to the **Data load editor**, make sure your cursor is positioned at the end of the script. We will first load some polygons. These can be found here:

# https://services.arcgis.com/1dSrzEWVQn5kHHyK/ArcGIS/rest/services/Limites\_Cartografia/FeatureS erver/0/query?where=1%3D1&outFields=\*&f=pgeojson

Select the data icon ( E ) of your GeoAnalytics Connection and select the Operation: Load.

Note: always check the GeoAnalytics reference page for full details using the Load Operation: <u>https://bi.idevio.com/wp-content/qlik/geoanalytics/releases/IdevioGeoAnalyticsConnector-5.13.0/doc/geoanalytics\_reference-September\_2018.html#Operation.Load</u>

Give *Parish* as **Name** for your *Dataset*, select *File based geo dataset* for **Type** and paste the link in **URL.** You may leave all the other options as suggested by default and click on **Next**.

Operation:			
operation.	Load	₹ ?	
Dataset			
Name:	Parish		
Type:	File based geo dataset	♥ ?	
URL*:	https://services.arcgis.com/1dSrzEWVQ	n5	
Key Field:			
File Type:	Auto	٣	
Expected Geometry Type:	Auto	•	
Character Encoding:			
CRS:	Auto		

We will not need all the fields available. You may **select and rename** *Parish\_Geometry* to *ParishGeometry*, **select and rename** *Shape\_Length* to *ParishLength*, **select and rename** *Shape\_Area* to *ParishArea*, **select and rename** *AREA\_M2* to *ParishAreaM2*, **select and rename** *NOME* to *ParishName* and **select and rename** *\_autoIndex\_* to *ParishId*. Once you've selected and renamed the fields you may click on **Insert Script**.

Select data to load		Hide script
Tables		
<b>Q</b> , Filter tables		
Parish		
	Fields	
	Data preview Metadata	Q, Filter fields
	☑ ParishGeome       OBJECT     COO_S     ☑ ParishLen     FREGUESIAS     ☑ ParishA     IDTL     ☑ ParishArea     ☑ ParishArea	Global 🔽 Paris
	/* Generated by Geoknalytics for operation Load	utoIndex ] AS [ParishId] FF
	DATASOURCE Parish GEOFILE url='https://services.arcgis.com/ldSrzEWQmSkHHyK/ArcGIS/rest/services/Limites_Cartografia/FeatureServer/l/query?where=1%3D1BoutFields=*8f=pgeojson', ke	eyField="', type='auto', exp
	tag fiald [Parishd] uith 'Sprianyrkey': tag field [ParishdSometry] with 'Sgeopolygon'; tag field [Parishd] uith 'Sgeopolygone';	
	tag field [Parishdeometry] with 'Srelates_Parishld'; tag field (Parishld) with 'Srelates_ParishGeometry'; /* End Geshualytics operation Load	
	Cancel	Back Insert script

### The following script is expected to have been generated for you:

LIB CONNECT TO 'GeoAnalytics';

/* Generated by GeoAnalytics for operation Load [Parish]:	*/	
SQL SELECT [Parish Geometry] AS [ParishGeometry],	[Shape Length] AS [ParishLength], [Shape	Areal AS [ParishArea], [ARFA M2] AS
[ParishAreaM2], [NOME] AS [ParishName], [_autoInde		
DATASOURCE	Parish	GEOFILE
url='https://services.arcgis.com/1dSrzEWVQn5kHHyK/	'ArcGIS/rest/services/Limites_Cartografia/Featu	reServer/1/query?where=1%3D1&out
Fields=*&f=pgeojson', keyField=", type='auto', expected	edGeomType='auto', encoding='', crs='Auto'	
· · · · · · · · · · · · · · · · · · ·		
tag field [ParishId] with '\$primarykey';		
tag field [ParishGeometry] with '\$geopolygon';		
tag field [ParishId] with '\$geoname';		
tag field [ParishGeometry] with '\$relates_ParishId';		
· · · · · · · · · · · · · · · · · · ·		

tag field [Parishld] with '\$relates\_ParishGeometry'; /\* End GeoAnalytics operation Load ------ \*/

Load your data and return to your previously created sheet to test your results.

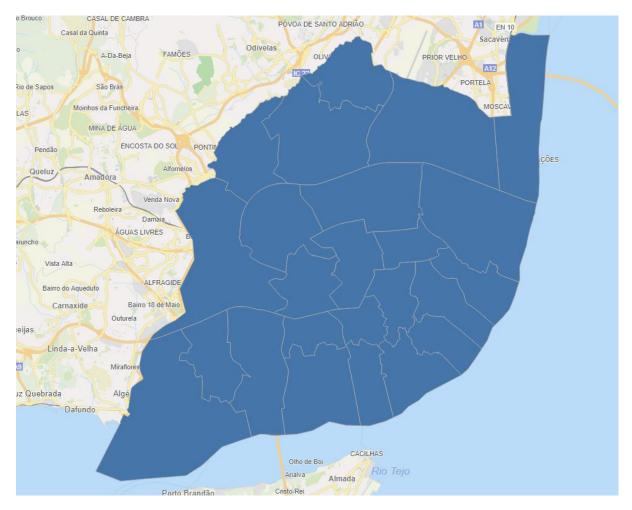
	🖉 🔻 📃 💌 Save 💮 Exer	cises Part 3
Load data 🕟 🕂	App overview	Ę <b>//</b>

#### Add the GeoAnalytics Area Layer to your sheet and drag the ParishGeometry field to it.

•	Q Search					Add "ParishGeor
ð		+	Ŷ	GeoAnalytics Area Layer	+	Add as measure
G.	Extensions					Cancel

Add "ParishGeometry"	4
Add as measure	•
Cancel	

You should now to be able to see all the Parishes of Lisbon city.



Now, to dissolve these polygons into a <del>blob</del> single polygon we will use the GeoAnalytics Connector Dissolve function.

Return to the **Data load editor**, make sure your cursor is positioned at the end of the script. As we did for the Routes exercise, we need to do a little pre-work to ensure we give the connector the right information about the relation between the polygons to dissolve together.

Add the following script:

ParishDissolvedRaw: Load 'Lisboa' as DissolveField, ParishId, [ParishGeometry] as ParishGeo Resident Parish;

This will create a resulting table associating all the *Parishld* values to the *Lisboa* value of the newly created field *DissolveField*. This way we will have a common factor that will result creating a single giant polygon by merging all the smaller ones when we use the GeoAnalytics Connector Dissolve function.

Once added the script **select the data** icon (<sup>E</sup>) of your GeoAnalytics Connection and select the **Operation: Dissolve**.

Note: always check the GeoAnalytics reference page for full details using the Dissolve Operation: https://bi.idevio.com/wp-content/qlik/geoanalytics/releases/IdevioGeoAnalyticsConnector-5.13.0/doc/geoanalytics\_reference-September\_2018.html#Operation.Dissolve

In the **Operation parameters** section for the **Dissolve field** type *DissolveField* and for **Resolution** leave it in *Auto*.

For the section **Dissolve definition dataset**, under **Name** type *DissolveDef*. The **Type** is *Loaded table* and the **Table name** is *ParishDissolvedRaw*. The **Table fields (key,geometry,...)** are *ParishId,ParishGeo,DissolveField*. The **Geometry type** is *Polygon* and you can leave the rest of the fields of this section as default.

For the section Geometries to dissolve select Type as None.

When all the options are filed, click on Next. Here is an example of the configurations above:

Step 1 - Select Operation			
Operation:	Dissolve	٣	?
Operation parameters			
Dissolve field*:	DissolveField		
Resolution:	Auto	۳	
Dissolve definition dataset			
Name:	DissolveDef		
Type:	Loaded table	Ŧ	?
Table Name:	ParishDissolvedRaw		
Table Fields (key,geometry,)*:	ParishId,ParishGeo,DissolveField		
Geometry Type:	Polygon	٣	
Only load distinct:	No	٣	
Location ID Suffix:			
CRS:	Auto		
Geometries to dissolve			
Name:	Dataset		
Type:	None	۳	?
*=Required field			
		Cance	I Next

As a result, the connector will show the tables selection wizard. Check the **DissolveTable** table, make sure all the fields are selected and click in **Insert Script**.

Select data to load Tables Q. Filter tobles	÷	Höde serlyt.
DissolveTable	2	
DissolveDef	Fields	
	Data preview Metadata	Q Filter fields
	SissolveField	C DissolvField_Geometry
	/* Generated by Geoknalytics for operation Dissolve	lssolvefield'; mlace(Banlace(Banlace(Banlace(Bank/15/61) ≤(condor) '9a=(ch0(conlumBan') Ch=/16) 'lumB07') Ch=/163 'lumB07') (h=/81) Cancel Back Insert Scipt

#### Here is an example of the expected script generated by the connector:

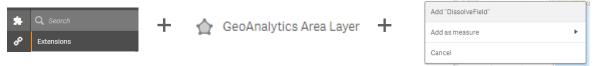
LIB CONNECT TO 'GeoAnalytics';

```
/* Generated by GeoAnalytics for operation Dissolve ------ */
Let [DissolveDefInlineTable] = 'ParishId' & Chr(9) & 'ParishGeo' & Chr(9) & 'DissolveField';
Let numRows = NoOfRows('ParishDissolvedRaw');
Let chunkSize = 1000;
Let chunks = numRows/chunkSize;
For n = 0 to chunks
          Let chunkText = ";
          Let chunk = n*chunkSize;
          For i = 0 To chunkSize-1
                     Let row = ";
                     Let rowNr = chunk+i;
                     Exit for when rowNr >= numRows;
                     For Each f In 'ParishId', 'ParishGeo', 'DissolveField'
                               row = row & Chr(9) & Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Peek('$(f)',
$(rowNr), 'ParishDissolvedRaw'), Chr(39), '\u0027'), Chr(34), '\u0022'), Chr(91), '\u005b'), Chr(47), '\u002f'), Chr(42), '\u002a'), Chr(59),
'\u003b'), Chr(125), '\u007d'), Chr(123), '\u007b');
                     Next
                     chunkText = chunkText & Chr(10) & Mid('$(row)', 2);
          Next
          [DissolveDefInlineTable] = [DissolveDefInlineTable] & chunkText;
Next
chunkText=''
[DissolveTable]:
                  [DissolveField],
                                   [DissolveField_Geometry]
                                                                FROM
                                                                             Dissolve(dissolveField='DissolveField',
SOL SELECT
                                                                                                                     resolution='auto'.
dissolveDef='DissolveDef')
                                                   tableName='ParishDissolvedRaw',
                                    INLINE
DATASOURCE
                DissolveDef
                                                                                          tableFields='ParishId,ParishGeo,DissolveField',
geometryType='POLYGON', loadDistinct='NO', suffix='', crs='Auto' {$(DissolveDefInlineTable)}
tag field [DissolveField] with '$primarykey';
tag field [DissolveField_Geometry] with '$geopolygon';
tag field [DissolveField] with '$geoname';
tag field [DissolveField_Geometry] with '$relates_DissolveField';
tag field [DissolveField] with '$relates_DissolveField_Geometry';
[DissolveDefInlineTable] = ";
/* End GeoAnalytics operation Dissolve ------ */
```

Load your data and return to your previously created sheet to test your results.



Add the GeoAnalytics Area Layer to your sheet and drag the DissolveField field to it.



You should now to be able to see a single polygon with the merge of all Parishes of Lisbon city.



# **Exercise 5: Intersect**

As an alternative to the Within function, you can also calculate the intersection between components using the GeoAnalytics Connector. In this exercise we will calculate the intersection between the Pharmacies and the Parishes.

Go to the **Data load editor**, make sure your cursor is positioned at the end of the script. **Select data** icon ( <sup>III</sup>) of your GeoAnalytics Connection and select the **Operation: Intersects**.

Note: always check the GeoAnalytics reference page for full details using the Intersects Operation: <u>https://bi.idevio.com/wp-content/qlik/geoanalytics/releases/IdevioGeoAnalyticsConnector-5.13.0/doc/geoanalytics\_reference-September\_2018.html#Operation.Intersects</u> In the **Dataset containing geometries** section for the **Name** type *ParishGeometries* and select **Type** as *Loaded table*. For the **Table Name** type *Parish*, in **Table Fields (key,geometry,...)** type *ParishId,ParishGeometry* and set the **Geometry Type** as *Polygon*.

In the second **Dataset containing geometries** section for the **Name** type *PharmaciesPoints* and select **Type** as *Loaded table*. For the **Table Name** type *PharmaciesRaw*, in **Table Fields (key,geometry,...)** type *PharmaciesId,PharmaciesGeometry* and set the **Geometry Type** as *Point*.

Step 1 - Select Operation			
Operation:	Intersects	۳	?
Dataset containing geometries.			
Name:	ParishGeometries		
Type:	Loaded table	۳	?
Table Name:	Parish		
Table Fields (key,geometry,)*:	ParishId,ParishGeometry		
Geometry Type:	Polygon	٣	
Only load distinct:	No	۳	
Location ID Suffix:			
CRS:	Auto		
Dataset containing geometries.			
Name:	PharmaciesPoints		
Type:	Loaded table		?
Table Name:	PharmaciesRaw		
Table Fields (key,geometry,)*:	PharmaciesId, PharmaciesGeometry		
Geometry Type:	Point	٣	
Only load distinct:	No	۳	
Location ID Suffix:			
CRS:	Auto		
*=Required field			

You may leave all the other options as suggested by default and click on Next.

As a result, the connector will show the tables selection wizard. Check the **IntersectsTable** table, make sure all the fields are selected and click in **Insert Script**.

Select data to load			Hide script
Tables Q Filter tobles	$\rightarrow$		
IntersectsTable	3		
ParishGeometries	Fields		
PharmaclesPoints	Data preview Metadata		Q. Filter fields
	Z ParishGeometries_PharmaclesPoints_RelationKey	Parishid	PharmaciesId
	<pre>/* Generated by GeoAnalytics for operation Intersects</pre>		I
	For n = 0 to chunkis to chunkista = "; Let chunk = n*chunkista: for i = Let row = "i;		
	Let rouhr = chunki; Exit for when rouhr >= numRous; For Each f In 'Parishid', 'Parishideometry' For Each f In 'Parishid', Parishideometry'	nlace/Qenlace/Deek/'S(f)' S(multe) Derish')	che/30) "\u0027"\ che/30) "\u0022"\ che/31\ "\u005h"\ (
			Cancel Back Insert script

```
Here is an example of the expected script generated by the connector:
```

```
LIB CONNECT TO 'GeoAnalytics';
```

```
/* Generated by GeoAnalytics for operation Intersects ------ */
Let [ParishGeometriesInlineTable] = 'ParishId' & Chr(9) & 'ParishGeometry';
Let numRows = NoOfRows('Parish');
Let chunkSize = 1000;
Let chunks = numRows/chunkSize;
For n = 0 to chunks
          Let chunkText = ";
          Let chunk = n*chunkSize;
          For i = 0 To chunkSize-1
                     Let row = ";
                     Let rowNr = chunk+i;
                     Exit for when rowNr >= numRows;
                     For Each f In 'ParishId', 'ParishGeometry'
                                row = row & Chr(9) & Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Peek('$(f)',
$(rowNr), 'Parish'), Chr(39), '\u0027'), Chr(34), '\u0022'), Chr(91), '\u005b'), Chr(47), '\u002f'), Chr(42), '\u002a'), Chr(59), '\u003b'), Chr(125),
'\u007d'), Chr(123), '\u007b');
                     Next
                     chunkText = chunkText & Chr(10) & Mid('$(row)', 2);
          Next
          [ParishGeometriesInlineTable] = [ParishGeometriesInlineTable] & chunkText;
Next
chunkText=''
Let [PharmaciesPointsInlineTable] = 'PharmaciesId' & Chr(9) & 'PharmaciesGeometry';
Let numRows = NoOfRows('PharmaciesRaw');
Let chunkSize = 1000;
Let chunks = numRows/chunkSize;
For n = 0 to chunks
          Let chunkText = ";
          Let chunk = n*chunkSize;
          For i = 0 To chunkSize-1
                     Let row = ";
                     Let rowNr = chunk+i;
                     Exit for when rowNr >= numRows;
                     For Each f In 'PharmaciesId', 'PharmaciesGeometry'
                                row = row & Chr(9) & Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Peek('$(f)',
$(rowNr), 'PharmaciesRaw'), Chr(39), '\u0027'), Chr(34), '\u0022'), Chr(91), '\u005b'), Chr(47), '\u002f'), Chr(42), '\u002a'), Chr(59), '\u003b'),
Chr(125), '\u007d'), Chr(123), '\u007b');
                     Next
                     chunkText = chunkText & Chr(10) & Mid('$(row)', 2);
          Next
          [PharmaciesPointsInlineTable] = [PharmaciesPointsInlineTable] & chunkText;
```

chunkText=''

#### [IntersectsTable]:

SQL SELECT [ParishGeometries\_PharmaciesPoints\_RelationKey], [ParishId], [PharmaciesId] FROM Intersects(dataset1='ParishGeometries', dataset2='PharmaciesPoints')

DATASOURCE ParishGeometries INLINE tableName='Parish', tableFields='ParishId,ParishGeometry', geometryType='POLYGON', loadDistinct='NO', suffix='', crs='Auto' {\$(ParishGeometriesInlineTable)}

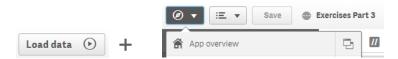
DATASOURCE PharmaciesPoints INLINE tableName='PharmaciesRaw', tableFields='PharmaciesId,PharmaciesGeometry', geometryType='POINT', loadDistinct='NO', suffix='', crs='Auto' {\$(PharmaciesPointsInlineTable)}

tag field [ParishId] with '\$primarykey';

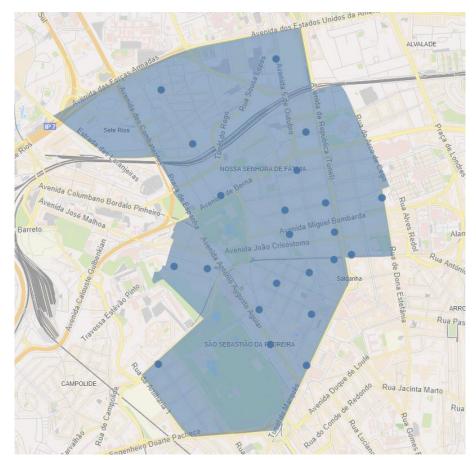
[ParishGeometriesInlineTable] = "; [PharmaciesPointsInlineTable] = ";

/\* End GeoAnalytics operation Intersects ------ \*/

Load your data and return to your previously created sheet to test your results.



Try to make a selection on a Parish and check if the corresponding Pharmacies are displayed on your map.



# Mobile "Near me" Location

Currently, Qlik Sense is not leveraging a web browser's location capability. This can be extremely handy for users on the field requiring "near my" geo location information. You can offer this feature leveraging an open source extension available at Qlik Branch: **My Location Finder for Qlik Sense**.

### https://developer.qlik.com/garden/56fc07ae509a5ad18831bef8

Note: please refer to Qlik's help site for documentation about how to install extensions

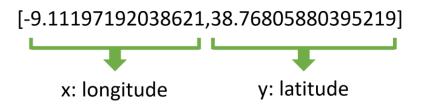
 Qlik Sense Server: <u>https://help.qlik.com/en-</u> <u>US/sense/September2018/Subsystems/ManagementConsole/Content/Sense\_QMC/import-</u> <u>extensions.htm</u>

To make the extension work you need to have within your datamodel two fields corresponding to Latitude and Longitude. It will ask to the user's web browser location and perform an associative search on the fields based on a radius interval you also define in the extension's properties.

On our examples, we could try to find the nearest Pharmacy. You are probably not located in Lisbon, but the idea of this exercise is that you can have a train of thought to use as base for your own developments.

If we look into the values in the **PharmaciesGeometry** field from exercise 1 in part 3, we can see they have a Point format ready to use, such as [-9.11197192038621,38.76805880395219].

We need to break this information to create two separate fields: one for latitude and another for longitude. And... of course for some reason, <u>breaking this up right away will not get us the correct values</u>. Here is why: the format actually represents [x,y] and on a map, Latitude is y whilst Longitude is x.



So, in our script all we need to do is to add a preceding load to our *PharmaciesRaw* and ensure we load the Latitudes and Longitudes correctly. Here is an example of the script you should add:

LOAD \*,

subfield(replace(PharmaciesGeometry,'[',"),',1) as PharmaLongitude, subfield(replace(PharmaciesGeometry,']',"),',',2) as PharmaLatitude;

And for your convenience, if you prefer, you can copy the fully updated load script for the *PharmaciesRaw* table:

LIB CONNECT TO 'GeoAnalytics';

/\* Generated by GeoAnalytics for operation Load ------ \*/ [PharmaciesRaw]:

LOAD \*,

subfield(replace(PharmaciesGeometry,'[',''),',',1) as PharmaLongitude,

subfield(replace(PharmaciesGeometry,']',''),',',2) as PharmaLatitude;

SQL SELECT [PharmaciesRaw\_Geometry] AS [PharmaciesGeometry], [INF\_SITE] AS [PharmaciesWebSite], [INF\_NOME] AS [PharmaciesName], [INF\_MORADA] AS [PharmaciesAddress], [autoIndex\_] AS [PharmaciesId] FROM Load(dataset='PharmaciesRaw')

DATASOURCE PharmaciesRaw GEOFILE url='https://services.arcgis.com/1dSrzEWVQn5kHHyK/arcgis/rest/services/POISaude/FeatureServer/1/query?where=1%3D1&outFields=\*&f =pgeojson', keyField=", type='auto', expectedGeomType='auto', encoding=", crs='Auto'

tag field [PharmaciesId] with '\$primarykey';

tag field [PharmaciesGeometry] with '\$geopoint';

tag field [PharmaciesId] with '\$geoname';

tag field [PharmaciesGeometry] with '\$relates\_PharmaciesId';

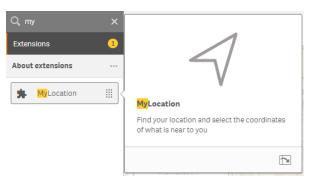
tag field [PharmaciesId] with '\$relates\_PharmaciesGeometry';

/\* End GeoAnalytics operation Load ------\*/

Load your data and return to your previously created sheet to test your results.



Drag the Extension to your sheet.

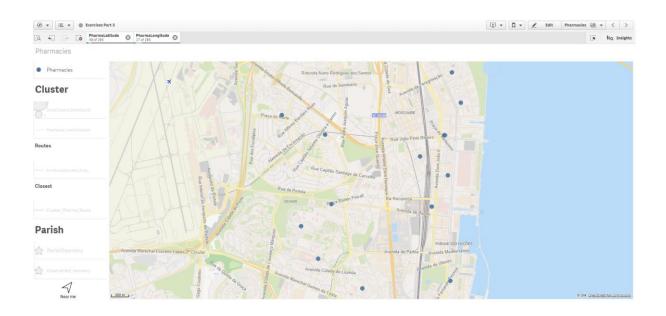


On the extension's properties, under the **Fields** pane, for the **Latitude Field** type *PharmaLatitude* and for the **Longitude Field** type *PharmaLongitude*.

Configure the coordinates reach radius the extension should search by going to the **Search Reach** pane and for the **Reach distance in km** type 2.

My Location Finder Configuration
▼ Fields
Latitude Field
PharmaLatitude
Longitude Field
PharmaLongitude
<ul> <li>Variable to store my position</li> </ul>
<ul> <li>Search Reach</li> </ul>
Reach distance in km
2

The extension is now configured. If you exit from the sheet edit mode and hit the icon, the extension will perform a search in the specified fields and select them so the map zooms to show which Pharmacies are nearby.



As a final extra, you can also add a new **GeoAnalytics Bubble Layers** to simulate the user's position in the map.

The extension can also update a variable that you specify. This way you can use this information to render a bubble using the **GeoAnalytics Bubble Layer.** 

Go to your application's **Variable Editor** and create a new variable *vMyPosition*.

Variables		Create new
Name	Definition	
New variable		Î 🖉
Name:		
vMyPosition		
Definition:		
		fx
Description:		
Tags:		
		0
chunk	0	
chunks	0.267	
		Close

Add a first **GeoAnalytics Bubble Layer** to your sheet to display the user's received position.



+

GeoAnalytics Bubble La...

Jump to the **Properties** pane and under **ID** add text of your choosing in the **Field** identification, such as ='*My Position*'.

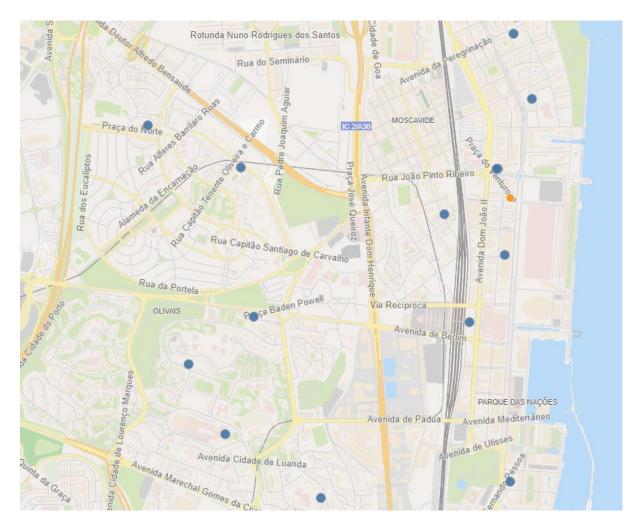
ID	
Add dimension	
<ul> <li>"My Position"</li> </ul>	⊗
Field	
='My Position'	fx
Label	
'My Position'	fx
Include null values	
Limitation	
No limitation	•

Now go to the **Location**, **Size** pane and this is where we configure the **GeoAnalytics Bubble Layer** to use our previously created variable. Under **Expression** type =vMyPosition.

⊗ fx
fx
fx
<i>fx</i>

Go to the **Appearance** pane, and under **Shape and Size** fine tune the bubble size to your liking. This should be the smallest center bubble representing the user's location. A value around 5 to 8 should be ok. Under the **Color** pane you can also choose a color if the default doesn't suit you. In my example I am using orange and outlining with white (#ffffff).

		ID
cation, Size		Location, Size
yer Options		Layer Options
cation Options		Location Options
pearance		Appearance
		▶ Legend
Legend	1. A. A.	Shape and Size
Shape and Size	T	<ul> <li>Colors</li> </ul>
ape		Colors
Bubbles 🔻		Custom
dius Min 5 - Max 8		Single color
		Calar
n Radius Value		Color
to		Transparency
ax Radius Value		
to		Outline Color (Only
		#ffffff

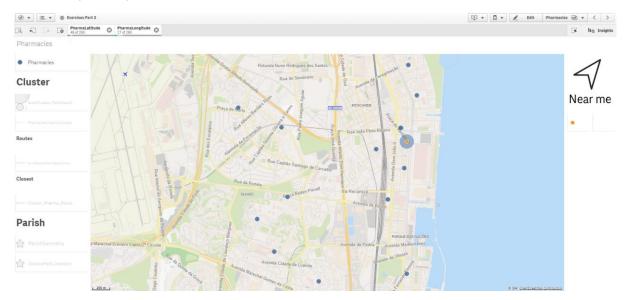


To finish the user's location representation, we can add a second **GeoAnalytics Bubble Layer** that can have a larger radius and some transparency. Copy and paste the **GeoAnalytics Bubble Layer** you've just created and go to its properties.

Under the **Appearance** pane, go to **Shape and Size** and select a larger radius value, i.e. 22. Under **Colors** (update the color if you prefer and) give some T*ransparency* to your bubble, around 50%. You can also update the outline color to match the bubble color. Also, if the bubble is over the previously created, make sure the drawing order is the expected. Go to the **Layer Options** pane and edit the **Draw Order**, for example attributing *-1* to this bubble as **Draw Order Adjustment**.

ID	ID	ID
Location, Size	Location, Size	Location, Size
Layer Options	Layer Options	Layer Options
Location Options	Location Options	Map ID
Appearance	Appearance	GeoAnalytics Map (1537786531
Legend	► Legend	Maximum Number of Objects
<ul> <li>Shape and Size</li> </ul>	Shape and Size	Calculation Condition
Shape	▼ Colors	<b>f</b> .
Bubbles v	Colors Custom	Zoom Limits In 0   Out 160000
Radius Min 22 - Max 22	Single color 🔹	Restric Drill Down 0-16
Min Radius Value	Color	Include in Auto Zoom
Max Radius Value	Transparency	Draw Order Custom
Auto	Outline Color (Only valid CSS colors)	Draw Order Category
	#4477aa	Bubble Layer
		Draw Order Adjustment
		-1

You've done it! Your user's can now search for nearby points within your Qlik Sense application. Here is an example of a possible final result:



# **Additional Guidelines**

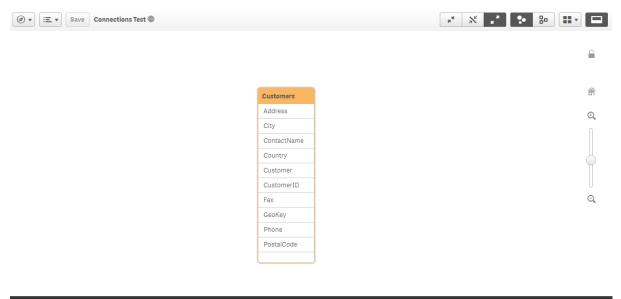
## How to perform GeoCoding

This section only explains a general configuration example to perform GeoCoding: the process to retrieve Latitude and Longitude coordinates from an address.

This is not an exercise covered in the session and the Qlik internal license does not include this functionality. You must explicitly request it as this is a service provided by a third party to Qlik.

The configuration is straight forward as you only have to match your data to the parameters required for the **AddressPointLookup** operation of the GeoAnalytics Connector.

In a table such as the one representing the Customers and used for the Exercises Part 1



Customers		Preview of data						
Rows	92	Address	City	ContactName	Country	Customer		
Fields	10	Obere Str. 57	Berlin	Albert von Einstein	Germany	Eintrach GS		
Keys	0	Avda. de la Constitución 2222	México Distrito Federal	Paco el Maco	Mexico	La Tienda de la Esquina		
Tags	\$text \$ascii \$numeric \$integer	C/ Ritual de lo Habitual 2312	México Distrito Federal	Sancho Panza	Mexico	La Ropa Vieja		
		120 Hanover Sq.	London	Carl Montgomery	UK	Dr Jims Trousers		
		Berguvsvägen 8	Luleå	Urra Gurra Aktersnurra	Sweden	Urras Shop		
		Forsterstr. 57	Mannheim	Herman Hinschler	Germany	Man Kleider		

The configuration would be:

Step 1 - Select Operation			
	Address Datable alway	<b>T</b>	
Operation: Operation parameters	AddressPointLookup	•	5
Search Text:			
Country:	Country		
-	Country		
State:			
City:	City		
Postal Code:			
Street:	Address		
House Number:			
Match Threshold:	0.5		
Service to use for lookup:	default		
Address Table			
Name:	Dataset		
Type:	Loaded table	٣	?
Table Name:	Customers		
Table Fields (key,geometry,)*:	CustomerID,Address,City,Country		
Geometry Type:	None	٣	
Only load distinct:	No	٣	
Location ID Suffix:			
CRS:	Auto		
*=Required field			
		Cane	cel Next
		Call	

# How to perform Reverse GeoCoding

This section only explains a general configuration example to perform Reverse GeoCoding: the process to retrieve Addresses from Latitude and Longitude coordinates.

This is not an exercise covered in the session and the Qlik internal license does not include this functionality. You must explicitly request it as this is a service provided by a third party to Qlik.

The configuration is straight forward as you only have to match your data to the parameters required for the **PointToAddressPointLookup** operation of the GeoAnalytics Connector.

In this example I am using a table that has the following structure:

CoordinatesTable: LOAD RowNo() as location\_id, CoordX as Longitude, CoordY as Latitude, '['&CoordX&','&CoordY&']' as Coordinates FROM [lib://QVDs/CoordsXY.qvd] (qvd);



#### ▼ Preview

CoordinatesTable		Preview of data				
Rows	5	location_id	Longitude	Latitude	Coordinates	
Fields	4	1	41.579347780111	1.6189809067536	[41.579347780111,1.6189809067536]	
Keys	0	2	41.982869321354	2.822778783606	[41.982869321354,2.822778783606]	
Tags	\$numeric \$integer \$ascii \$text	3	41.64640087989	1.1395293245344	[41.64640087989,1.1395293245344]	
		4	41.616902172986	0.6282436697822	[41.616902172986,0.6282436697822]	
		5	41.451039222264	2.2486787266201	[41.451039222264,2.2486787266201]	

Quick note: if you look at a map in a X,Y perspective, X is the Longitude and Y is the Latitude. Check <u>https://en.wikipedia.org/wiki/Longitude</u> and <u>https://en.wikipedia.org/wiki/Latitude</u> for more information.

The configuration for **PointToAddressPointLookup** would be:

Step 1 - Select Operation			
Operation:	PointToAddressLookup	٠ ?	
Operation parameters			
Service to use for lookup:	default		
Point table			
Name:	Dataset		
Type:	Loaded table	¥ ?	
Table Name:	CoordinatesTable		
Table Fields (key,geometry,)*:	location_id,Coordinates		
Geometry Type:	Point	Ŧ	
Only load distinct:	No	٣	
Location ID Suffix:			
CRS:	Auto		

# **Enablement Materials**

You can access to Qlik GeoAnalytics full documentation on Qlik's help site:

#### • <u>https://help.qlik.com/en-US/geoanalytics/Content/Home.htm</u>

For more Qlik Sense examples check the Qlik Sense Guides & Examples micro site:

• https://bi.idevio.com/products/idevio-maps-for-qlik-sense/qlik-sense-2

For QlikView examples check the IM5QV Guides and Examples micro site:

• https://bi.idevio.com/products/idevio-maps-5-for-qlikview/guides-and-examples