

Cartes dans qliksense

<https://community.qlik.com/message/924588#924588>

Mon projet :

Tableau de bord socio-économique de mon territoire à partir de données annuelles compilées et organisées dans des fichiers exel.

Sources d'infos :

[https://help.qlik.com/sense/2.0/fr-](https://help.qlik.com/sense/2.0/fr-FR/online/#../Subsystems/Hub/Content/Visualizations/Map/load-map-data.htm)

[FR/online/#../Subsystems/Hub/Content/Visualizations/Map/load-map-data.htm](https://help.qlik.com/sense/2.0/fr-FR/online/#../Subsystems/Hub/Content/Visualizations/Map/load-map-data.htm)

<http://blogbi.asi.fr/2014/11/14/creation-dune-analyse-geographique-dans-qlik-sense/>

Faire une carte avec des cercles proportionnels

Il faut d'abord charger une liste des communes avec le centroïde en XY.

- Ces données sont disponibles sur le site de l'INSEE mais pas avec la bonne projection. J'ai essayé les explications de cet article : <http://blogbi.asi.fr/2014/11/14/creation-dune-analyse-geographique-dans-qlik-sense/> mais je ne suis pas arrivée à insérer ce bout de script. Surement un problème de parenthèses en trop ou pas.
- Pour ma part j'ai finalement récupéré un fichier des communes avec leur code INSEE sur le fil de discussion qlikview (merci Florent).

Dans tous les cas il faut récupérer des données comportant une colonne latitude (Y) et longitude (X), en WGS84 (la projection de google), et en données décimales.

- Le problème de la base INSEE c'est que ces données sont en degré minuté seconde d'où le bout de code pour les convertir.
- Si vous récupérez ces données à partir d'un fichier shape, il faudra convertir le dbf en xls car le dbf est un des rares format que qlikview/qliksense ne charge pas (en tous cas il me semble mais je ne suis encore qu'une utilisatrice novice).

Dans le script il faut insérer, avant de charger les données proprement dites, la fonction "geomakepoint". Chez moi ça donne ça :

Ville:

```
LOAD *, GeoMakePoint(Y,X);
```

```
LOAD "Nom Ville",
```

```
    MAJ,
```

```
if(len([Code INSEE])=4,0&[Code INSEE],[Code INSEE]) as codgeo,
```

```
"Code Région",
```

```
    Latitude as Y,
```

```
    Longitude as X,
```

```
Eloignement
```

```
FROM [lib://RGC/villecp_geoloc.xls] (biff, embedded labels, table is ville$);
```

- Si le code INSEE se charge mal en ne reconnaissant pas le format texte et en supprimant les zéros de départ rajouter ça dans le script :
if(len([Code INSEE])=4,0&[Code INSEE],[Code INSEE]) as codgeo,

```

1 Ville :
2 LOAD *,
3   GeoMakePoint(Y,X) as XY;
4
5 LOAD
6   "Nom Ville",
7   MAJ,
8   if(len([Code INSEE])=4,0,[Code INSEE],[Code INSEE]) as codegeo,
9   "Code Région",
10  Latitude as Y,
11  Longitude as X,
12  Eloignement
13 FROM [lib://RGC/villecp_geoloc.xls]
14 (biff, embedded labels, table is ville$);
15

```

Dans le mode présentation, sélectionner le mode carte.

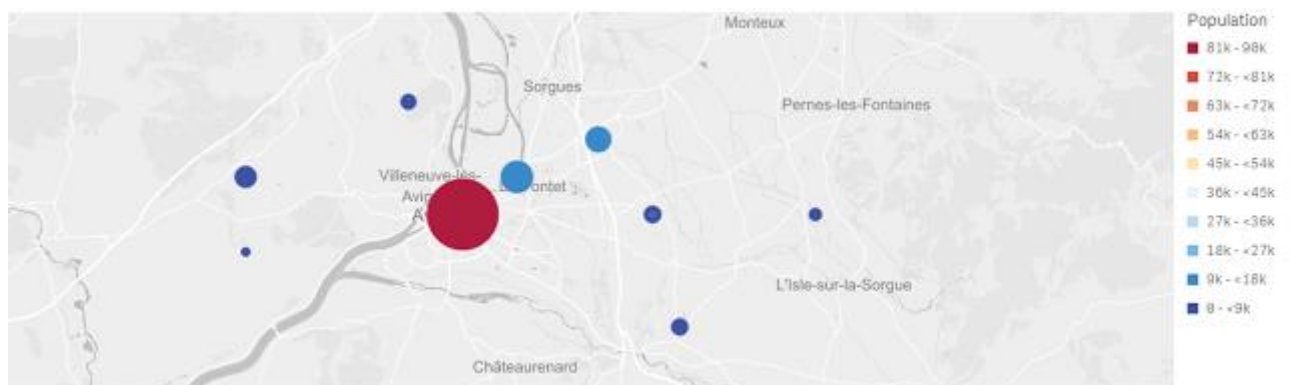
Choisissez en dimension n'importe quelle donnée rattachée à vos données XY.

- Pour moi, comme toutes mes données sont reliées par le codegeo j'utilise NOM_COMM. Il s'agit du nom de mes communes qui sont dans un table que j'ai appelé "échelles" et qui me permet de trier mes données statistiques enregistrées avec 3 départements, et aussi d'organiser des regroupements statistiques : ville-centre, côté Gard, côté Vaucluse.

Ensuite je choisis de représenter la population de mes communes avec un cercle proportionnel avec la formule :

$Sum(\{<NOM_COMM=\{*\}, annee_pops=\{ "\$ (=Max(annee_pops))" \} >\} pops_com)$

- Ma formule max me permet de retenir la dernière année de mes statistiques, soit la population 2012 à ce jour.
- NOM_COMM={*} me permet de ne pas tenir compte de ma sélection et de bien représenter mes 15 communes



Faire ensuite une carte avec des aplats

Il faut ici charger un fichier kml. Pour ma part je suis partie d'une couche géofla communes. On trouve aujourd'hui ces données sur data.gouv.fr en format shape.

- Il faut les exporter en kml. J'ai pas mal tâtonné pour exporter mon kml avec mon fameux codegeo. J'y suis arrivée en mettant le code INSEE (codegeo) en étiquette dans arcgis et en exportant ensuite en kml.
- L'export peut se faire aussi sous qgis et là il faut préciser l'option fieldname = codegeo.

Le fichier kml se charge ensuite tout simplement comme n'importe quelle donnée. Décocher le champ Communes.point qui n'aura aucune utilité.

Voilà ce que ça donne chez moi :

```
GDA_commune: LOAD GDA_communes5.Name as codegeo,  
GDA_communes5.Area  
FROM  
[lib://SIG/Grandavignon_donneesvalidees\Grandavignon_perimetres\GDA_communes5.kml] (kml, Table is [COMMUNES_GdAvignon/COMMUNES_GdAvignon]);
```



Pour faire la carte c'est comme pour les données points. Je mets comme dimension codegeo ou NOM_COMM.



Une carte avec simplement le nom de mes communes, la légende est construite à partir de la dimension "commune"

Attention : la couche de points doit être chargée dans le script avant la couche kml.