Mapping with CartoDB (Tanzania Edition)

In this exercise we are going to use election data to build informative and interactive maps of Tanzania.

In order to complete this exercise you need to have a CartoDB account. This can be created at http://cartodb.com

This exercise requires a number of key datasets that are available to download from the course website.

1. The regional boundaries of Tanzania. (Source: Tanzania National Bureau of Statistics)
2. Population and region area data. (Source: Tanzania National Bureau of Statistics)
3. Education pass rate and school rank (Source: National Examinations Council of Tanzania)

Step 1 - Preparing the data (already done)

In order to prepare each dataset, several translations have been carried out to extract it from the source and translate each dataset into formats appropriate for use with CartoDB. These steps are briefly described below.

Regional boundary files

Source: Tanzania National Bureau of Statistics (NBS)
Source URL:
http://www.nbs.go.tz/nbstz/index.php/english/statistics-by-subject/population-and-housing-census/258-2012-phc-shapefiles-level-one-and-two

These files come as a rather large zip file containing a lot of shape files. Shape files (SHP) are a zip file format developed by ESRI to a “mostly” open specification. The terms of the specification are vague, particularly around usage rights, many online tools choose not to support the format for legal reasons. This means that in order to make the format usable, we need to translate it into something compatible with more tools.

Unwanted data was first removed from the source NBS zip file, leaving only the region data inside the zip file. This zip file was then uploaded to the SHP to KML translator available at http://www.mapsdata.co.uk/online-file-converter/.

The resultant KML has been made available on the course website and is the one required for the exercise.

Population and region area data

Source: Tanzania National Bureau of Statistics (NBS)
Source URL: http://www.nbs.go.tz/nbs/takwimu/references/Tanzania_in_Figures_2015.pdf, also available via Wikipedia (https://en.wikipedia.org/wiki/Regions_of_Tanzania)

For the purposes of this exercise the data has been extracted from Wikipedia, however you could use a tool like pdftables.com to extract the data easily from the PDF.

In order for the data to line up with the region data, some of the regions have had to be renamed with their local names rather than their English translations.

Education pass rate and school rank data

Source: National Examinations Council of Tanzania
Source URL: http://www.necta.go.tz/brn2015/

In order for this data to be usable, a pivot table has been created from the data to list each region only once. All values have been summed or averaged in order to work out the overall regional statistics. A pass rate percentage value has also been added by divided the number of candidates that passed by the total number of candidates.

Step 2 - Create a regions map of Tanzania

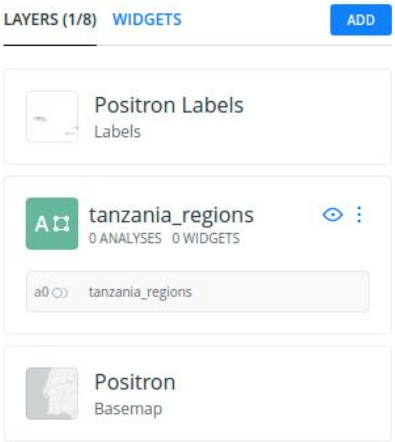
Once you are logged into CartoDB, in the top right hand corner of the screen you should be able to select **maps** or **datasets**. Click **maps** and then click the **new map** button.

On the next screen ensure you select **connect dataset** from the available options and then upload a **data file**. Here you can drag and drop or browse to upload the Tanzania_regions.kml file. Click **connect dataset** to upload this file. It may take a while to upload and process.

When complete you should get a map of Tanzania with all the regions outlined. **At the top, give it a name.**

Step 3 - Display the region names on the map

Once you have added the region names, return to the **map view** and click **edit pencil** on the right.



Under the edit area you can add up to 8 layers to your map, something we will be doing later. Currently there should be one layer available for you to edit. You can also edit the underlying map to change its style.

Click the tanzania_regions layer to enter editing mode.

Select style from the top edit menu and tick the labels box. To add region labels select the **name** column as the source column for the label names. You can then customise the font, colour and position of the label until it looks right on the map. For example click 1 on the Halo and change this to 0 to make the labels easier and cleaner to read.

You might also want to change the **Basemap** which can be done from the Layers menu which was the previous screen we looked at and can be accessed at any time by clicking the back arrow in cartodb.

Step 4 - Combine with population data

In order to combine the region data with the population data, we first need to upload the population datasets button. To do this, click the **home button** to get back to your dashboard.

From the drop down in the top left where it says **Maps**, click this and select **Your datasets**.

Click **New Dataset** and upload the population data.

Once done, click the **home button** again and select your region map from the **maps dashboard** to return to your map.

From the edit menu, select the tanzania_regions layer to enter edit mode (as we did to add labels) and this time select the **analysis** section. Finally, click **Add Analysis**.

It is from this section that we can perform lots of clever operations with our maps including enrich using openly available data from the data observatory.

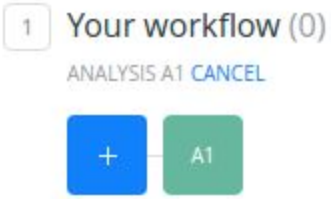
We are going to **join columns from 2nd layer** in order to join our regions dataset to a statistical dataset. In this case we are going to connect it to the population data we have just uploaded.

Select **join columns from 2nd layer** and click **add analysis**.

Now we need to construct the workflow for connecting the two datasets together. This

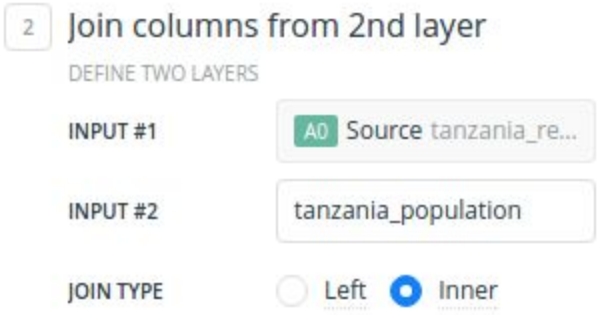
consists of 4 steps.

1) Create the workflow



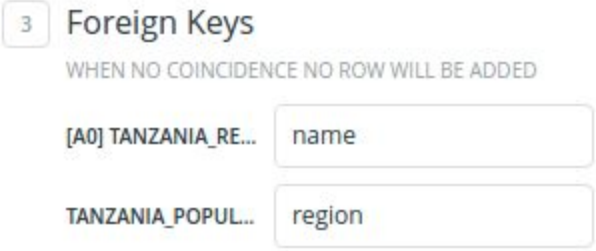
This we have already done.

2) Define the two layers



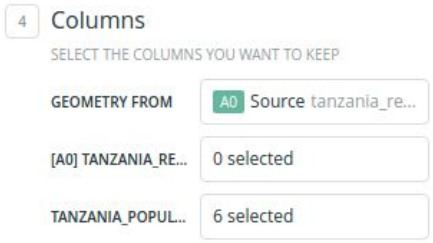
To complete this step simply select tanzania_population as our second input dataset. Ensure you are also using an inner join (as shown).

3) Define the keys which join the two layers



Here the regions dataset has a column called **name** while the population dataset refers to this same set of values in the column **region**. Select each as shown.

4) Select the data to be retained/discarded after the join



In this step we can discard all data from the original regions dataset and select **population**, **area_km2**, **zone**, **postcode**, **capital** and **region** from the population dataset.

Finally click **Apply** to add this analysis.

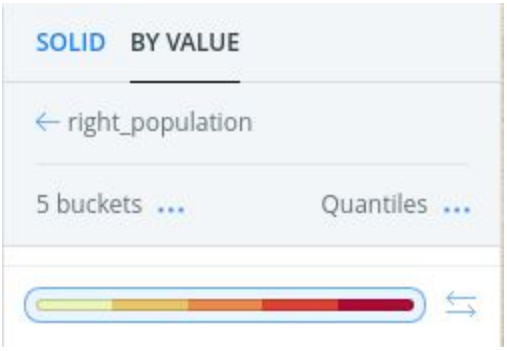
Step 5 - Restyle the map

With the analysis added, click **style** to restyle your map.

This time we are going to create a **Choropleth** map using the population data to calculate the color of each region on a continuous scale.

First add back in your region labels as per **Step 3**. Remember that the column name might have changed!

From the same style pane, now click the colour **fill** against the **polygon style** and select **By Value**. Select the **population** column as the source of values and customise your style so the map has a dark colour for a higher population and the lighter colour for a lower population.

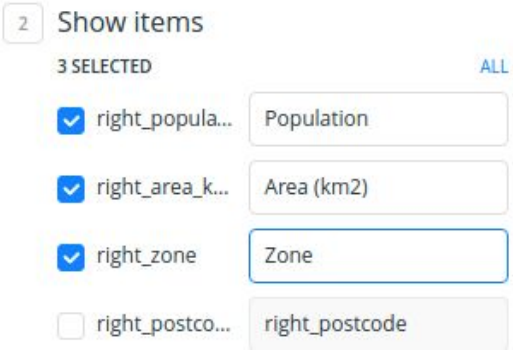


You will notice that a legend has appeared on your map. Click the **legend** section of your edit pane to customise this. Perhaps you might want to fix the title or remove it if your map name is suitable.

Step 6 - Adding labels

Click on any region on the map and you will note that the label text is not very helpful (or the box is empty).

From the edit menu, select **pop-up** and from here you can select and customise what appears when you click on a region.



Even though this is easy, it doesn't produce very clear labels. Luckily everything in CartoDB can be customised in code.

Extension exercise - Education data

Why not try repeating steps 4 onwards with the education data and create an even bigger dataset. When creating a map, this time you can use a Choropleth map with the **passed percent** column or a category map with the **category** column. Both of these columns were added when the pivot table was created.