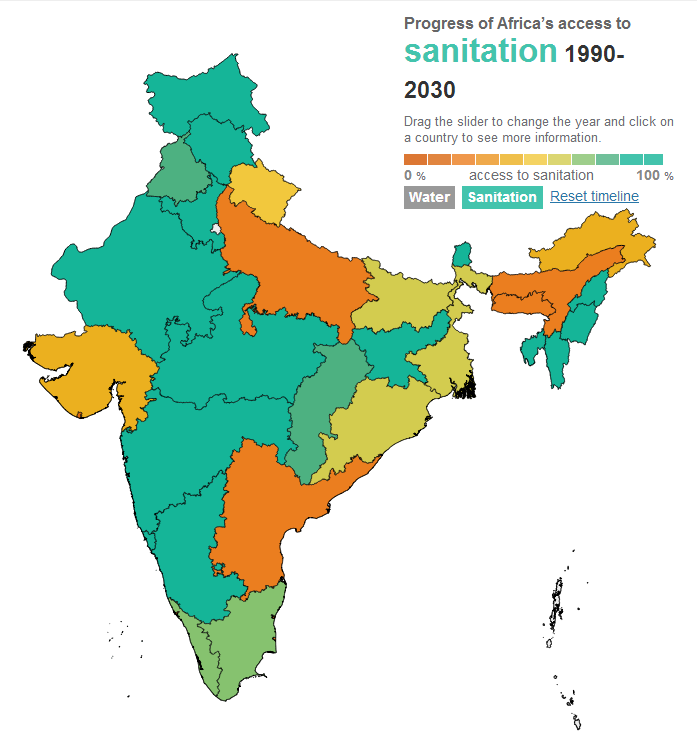
**Steps used to create a chloropleth map for India**



To create a straightforward chloropleth map of a given area we can use the Aptivate Africa map template (<https://github.com/aptivate/africawatermap>) by creating an appropriate topojson-file of the map and a json file of corresponding data we want to show.

**Creating the map:**

We can convert shapefiles to topojson in two stages, first with a program called ogr2ogr and then using the topojson-program. Shapefiles are a standard format for maps and are freely available. Shapefiles can be found for example here: <http://www.diva-gis.org/gdata>.

Ogr2ogr can be used to filter only the parts of a shapefile map we are interested in.

Install ogr2ogr, run the shell terminal it contains and run the following:

**Ogr2ogr –f GeoJSON subunits.json [shapefile.shp]**

If you want to filter some data use:

Ogr2ogr –f GeoJSON –where “[country\_name] IN (‘India’, ‘Bangladesh’,’Sri Lanka’)” subunits.json [shapefile.shp]

Where country\_name can be any property within the shapefile.

You can use Node.js to install topojson and then run the following command to convert the shapefile into a topojson-file with a command like the below:

**Topojson --id-property=country\_name –p name=NAME –p name –o outputmapfile.json subunits.json**

Where the --id-property flag is necessary as it elevates one property to geometry id. This should be a property which we can map similarly in the .json data file. For example, if we are mapping districts, we could set –id-property to the district\_id property and then we need the corresponding ids in the data file so that the program recognises which area of the map links to which data.

The –p flag sets any properties that we want to retain. Other properties will not be included in the end file.

The map should now be complete. Be sure to change the target map file in the d3js/index.html file to the newly created file.

The project files contain a tool for converting Excel files into a suitable json file. We need to adjust the Excel files appropriately, and also set the mapping of names within the xls2json.py file under the data-folder.

Take the existing mapdata1 and mapdata3 files as your starting point and populate them with the data you want. If you change the structure of the sample Excel files, you will need to also adjust the NAME\_CODE dictionary in the python file. This maps the names in the Excel file to IDs that match the geometry id of the topojson file just created.

Once you’ve got the Excel files you need, open the xls2json.py file and Make sure you have Python 2.7 installed and then you can run the command:

Python xls2json.py [percentage data.xlsx] [numerical data.xlsx]

This will create a file called output.json which needs to be then referred to within the index.html file in the d3js folder.