Dasymetric Disaggregation using Google Earth Engine

Rhorom Priyatikanto
WorldPop - University of Southampton

Pre-requisites

1. Basic skill in *Javascript* programming language
2. Basic knowledge on the Google Earth Engine
3. Basic understanding in GIS concepts, including raster and vector data processing and analysis

Table of contents

Main courses

In this tutorial, we will create a gridded population dataset using a simple dasymetric mapping that relies on a single ancillary data representing population distribution over a spatial extent.

Introduction

Gridded population data is created by taking census data, which is typically collected at the administrative unit level, and distributing that population across a grid of raster cells. Census, that becomes the primary source of population data, provides counts of people within specific administrative boundaries (e.g., counties, districts). Using a certain method, the population within each administrative unit is distributed across the grid cells that intersect with that unit. This method is known as top-down approach.

There are two major methods to distribute or disaggregate population into grid cells:

- **Areal Weighting**: This common method assumes a uniform population distribution within each administrative unit and allocates population to grid cells based on the proportion of the unit's area covered by each cell.

- **Dasymetric Mapping**: This method uses ancillary data (e.g., land cover, building density) to refine the population distribution within administrative units, resulting in a more realistic representation.

Dasymetric mapping becomes a method to create accurate and reliable gridded population datasets like WorldPop Global Project Population Data and LandScan Population Data. This method leverages additional information, called ancillary data, to more accurately reallocate the original data within these areas. The goal is to achieve a distribution that better reflects the true spatial pattern of the phenomenon being mapped. Ancillary data is the crucial ingredient that distinguishes dasymetric mapping. It consists of spatial datasets that correlate with the distribution of the source data. Common examples include:

- Land cover/land use data: Identifying areas unsuitable for human habitation (e.g., water bodies, forests) or with varying population densities (e.g., urban vs. rural).
- Building footprints/density data: Indicating areas with higher potential for population concentration.
- Nighttime lights data: Suggesting areas of human activity and potential population presence.

Based on the available ancillary data, the population count at administrative unit level is then proportionally redistributed within this zones.

Input data

In this tutorial, we aim for disaggregating population of Skaane County in Sweden using two ancillary data:

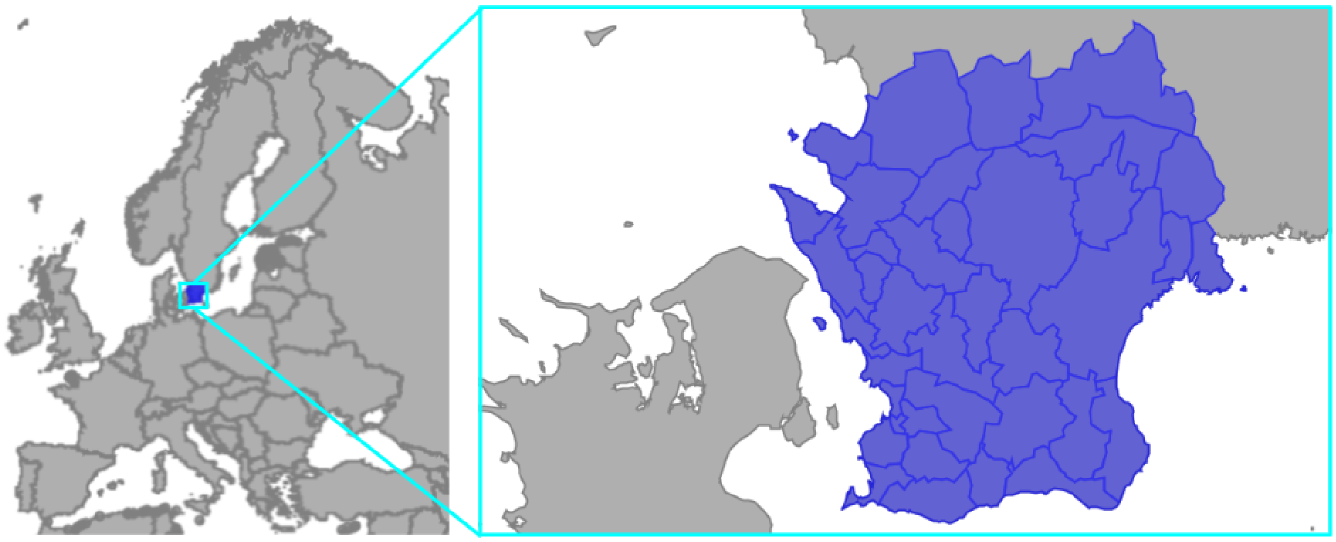
- VIIRS Nighttime Composite representing the amount of lights emitted from the Earth's surface as measured by the Visible Infrared Imaging Radiometer Suite onboard Suomi NPP satellite. The dataset contains several bands, including the average nighttime radiance to be used for dasymetric disaggregation. It has a spatial resolution of around 500 m.

- GHSL Global Building Volume depicts the global distribution of building volume, expressed in cubic metres per 100 m grid cell. Following the characteristics of the building, i.e., residential and non-residential, we can estimate the building volume allocated for residential use, which is relevant for population disaggregation.

We use WorldPop Global Project Population Data as the basis. The total population of Blekinge County (admin-1) and the population counts at every subcounty (admin-2) are obtained through zonal statistics using appropriate administrative boundaries. The Global Administrative Unit Layer contains administrative boundaries we can use to define the region of interest for this tutorial.

```
var geometry = ee.Geometry.Point(13.0,55.6); //a point near Malmoe
var adm1 = ee.FeatureCollection("FAO/GAUL_SIMPLIFIED_500m/2015/level1")
  .filterBounds(geometry)
  .select(['ADM1_NAME', 'ADM1_CODE']);

var adm2 = ee.FeatureCollection("FAO/GAUL_SIMPLIFIED_500m/2015/level2")
  .filterBounds(adm1)
  .select(['ADM1_NAME', 'ADM1_CODE', 'ADM2_NAME', 'ADM2_CODE']);
```



Then, we can import WorldPop data and the ancillary data layers.

```
var wpgp = ee.ImageCollection("WorldPop/GP/100m/pop")
  .filterDate('2020-01-01','2020-12-31')
  .filterBounds(adm1).mosaic().clip(adm1).rename('pop');

var viirs = ee.ImageCollection("NOAA/VIIRS/DNB/ANNUAL_V21")
  .filterDate('2020-01-01','2020-12-31')
  .filterBounds(adm1).first().clip(adm1).select('average');
viirs = viirs.where(viirs.lt(0.25), ee.Image(0));

var ghsv = ee.Image('JRC/GHSL/P2023A/GHS_BUILT_V/2020')
  .clip(adm1);
var ghsl = ghsv.select('built_volume_total')
  .subtract(ghsv.select('built_volume_nres'));
```

The annual composite of VIIRS data contains the average radiance in nanoWatts/sr/cm2. Typically the cell values range from 0 to 10. To obtain more realistic result where the people resides in settlement area in cities, towns, and the surroundings, we apply minimum threshold to the VIIRS data. Cell values below 0.25 nanoWatts/sr/cm2 are set to zero.

For the Building Volume dataset, we subtract the total volume with the volume of non-residential buildings. Thus, cell values represent the volume of buildings used as residence.

With the required images in hand, we are able to aggregate the data at admin-1 level. The resulting table will be used for dasymetric disaggregation. Just before aggregation, the images are stacked into an image with three bands, i.e., `pop`, `viirs`, and `ghsl`.

```
wpgp = wpgp.rename('pop')
  .addBands(viirs.rename('viirs'))
  .addBands(ghsl.rename('ghsl'));

var pop1 = wpgp.reduceRegions({
  collection: adm1,
  scale: 100,
  reducer: ee.Reducer.sum()
});
```

Printing the table gives the following information

```
FeatureCollection (1 element, 3 columns)
    type: FeatureCollection
    columns: Object (3 properties)
    features: List (1 element)
        0: Feature 0000000000000000081f (MultiPolygon, 5 properties)
            type: Feature
            id: 0000000000000000081f
            geometry: MultiPolygon, 546 vertices
            properties: Object (5 properties)
            ADM1_CODE: 2798
            ADM1_NAME: Skaane Laen
            ghsl: 1012538246.200001
            pop: 1194366.2647774743
            viirs: 4201327.650036133
```

Simple disaggregation

Suppose the total population in the selected admin-1 unit is \$pop_j\$ while the ancillary data value at each grid cell is \$x\$. Using proportional disaggregation rule, the estimated population at grid cell is \$\$\hat{pop}_i=\frac{pop_j}{\sum x_i} \times x_i=f_j \times x_i,\$\$ where \$\sum{x_i}\$ is the total value of \$x\$ at specified unit. This total acts as the normalising factor which transforms the ancillary data into a probability of getting people in a particular grid cell.

As the sum of \$x_i\$ for `viirs` and `ghsl` have been obtained through prior zonal statistics, we can compute \$f_j\$ using the following commands:

```
pop1 = pop1.map(function(feat){
  var pop = feat.getNumber('pop');
  var f1 = pop.divide(feat.getNumber('viirs'));
  var f2 = pop.divide(feat.getNumber('ghsl'));
  return feat.set('f1', f1, 'f2', f2);
});
```

Multiplication of \$f_j\$ and the ancillary data \$x\$ yields the population estimate. However, we need to convert \$f_j\$ stored in `FeatureCollection pop1` into an `Image` so that multiplication can be done at grid level. This so called rasterisation process can be performed using `reduceToImage()` method attached to `FeatureCollection` class.

```
var f1 = pop1.reduceToImage({
  properties: ['f1'],
  reducer: ee.Reducer.sum()
});

var f2 = pop1.reduceToImage({
  properties: ['f2'],
  reducer: ee.Reducer.sum()
});
```

Afterward, grid level (pixel-wise) multiplication can be performed and the estimated population using two ancillary data can be stacked for further assessments.

```
var e1 = wpgp.select('viirs').multiply(f1);
var e2 = wpgp.select('ghsl').multiply(f2);
var estimate = wpgp.select('pop')
  .addBands(e1.rename('est_viirs'))
  .addBands(e2.rename('est_ghsl'));
```

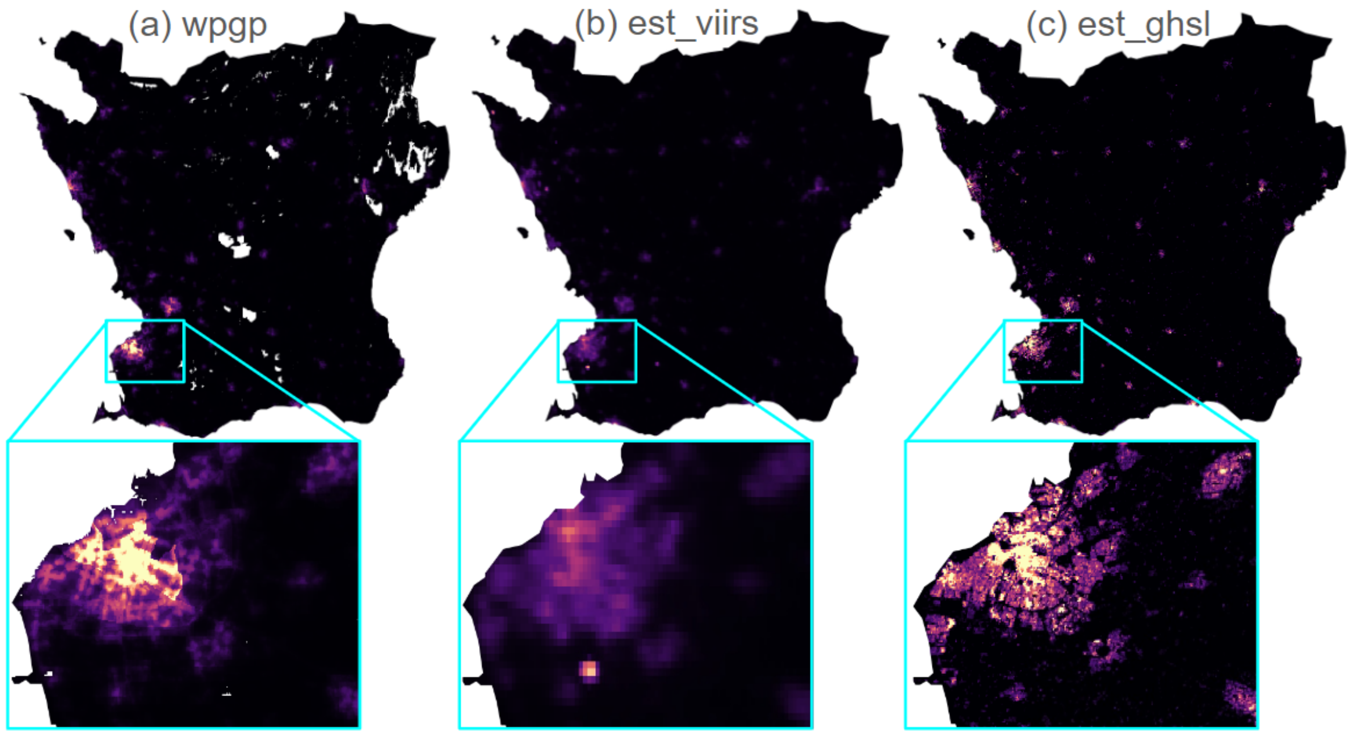
Checking the results

Let's visualise the dasymetric results on a map and compare them to the basis gridded population data.

```
var palette = ['000004', '51127c', 'b73779', 'fc8961', 'fcfdbf'];
var vis = {min:0, max:50, palette:palette};

Map.centerObject(geometry, 8);
Map.addLayer(ee.Image(1), {palette:['white']});
Map.addLayer(wpgp.select('pop'), vis, 'WPGP');
Map.addLayer(estimate.select('est_viirs'), vis, 'Dasymetric_VIIRS');
Map.addLayer(estimate.select('est_ghsl'), vis, 'Dasymetric_GHSL');
```

After some arrangements, the results look like the figure below.



Visually, the estimate from VIIRS data is fuzzier compared to the WorldPop data while the one from GHSL is more granular. As expected, VIIRS with a 500-m resolution yields a coarser gridded population estimate. In terms of resolution, GHSL data has the same granularity compared to the WorldPop data so that the resolutions of the ancillary data can't explain the difference. Variation in building volume is directly propagated to the population estimate. Meanwhile, WorldPop data is produced using Random Forest algorithm that incorporates multiple covariates to get the probability layer for dasymetric redistribution. Combination of covariates with different resolutions creates a layer that is significantly fuzzier that its grid size.

Further assessment can be performed by aggregating the population estimate to admin-2 level and compare the values with the ones from WorldPop.

```
var pop2 = estimate.reduceRegions({
  collection: adm2,
  scale: 100,
  reducer: ee.Reducer.sum()
});

pop2 = pop2.map(function(feat){
  var pop = feat.getNumber('pop');
  return feat.set('ref', pop);
});
```

The resulting scores are:

Score	est_viirs	est_ghsl
Bias	-2.16	0.08
RMSE	20017	7483

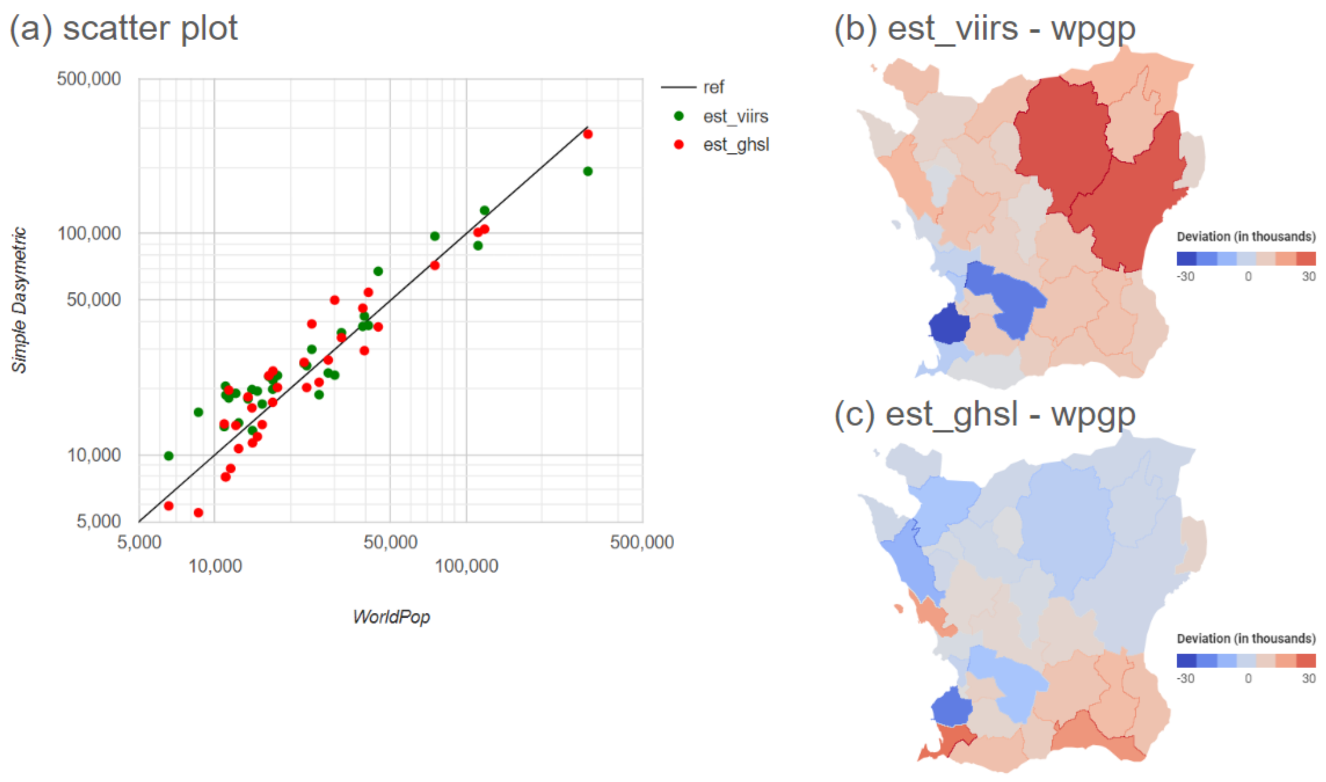
Score	`est_viirs`	`est_ghsl`
MAPE	0.80	0.36
R2	0.91	0.98

A scatter plot can also be generated as a part of the assessment.

```
var chart = ui.Chart.feature.byFeature({
  features: pop2,
  xProperty: 'pop',
  yProperties: ['ref', 'est_viirs', 'est_ghsl']
});

var series = {
  0: {pointSize:0, lineWidth:1, color:'black'},
  1: {pointSize:5, lineWidth:0, pointShape: 'circle', color:'green'},
  2: {pointSize:5, lineWidth:0, pointShape: 'circle', color:'red'},
};
var lim = {min:5e3, max:5e5};
var style = {
  hAxis: {title: 'WorldPop', scaleType:'log', viewWindow:lim},
  vAxis: {title: 'Simple Dasymetric', scaleType:'log', viewWindow:lim},
  series: series
};

chart = chart.setOptions(style);
print(chart);
```



To see how the estimated population counts deviate from the WorldPop data, we can calculate the deviations at admin-2 level and visualise the deviations on a map using a diverging color scale (see panel b and c of the figure above).

```
//Computing deviation at admin-2 level
pop2 = pop2.map(function(feat){
  var pop = feat.getNumber('pop');
  var e1 = feat.getNumber('est_viirs');
  var e2 = feat.getNumber('est_ghsl');

  return feat.set(
    'dev_viirs', e1.subtract(pop),
    'dev_ghsl', e2.subtract(pop)
    );
});

//Rasterisarion
var d1 = pop2.reduceToImage({
  properties:['dev_viirs'],
  reducer:'sum'
}).clip(adm1);

var d2 = pop2.reduceToImage({
  properties:['dev_ghsl'],
  reducer:'sum'
}).clip(adm1);

//Visualisation
var palettes = require('users/gena/packages:palettes');
var palette = palettes.misc.coolwarm[7];
vis = {min:-3e4, max:3e4, palette:palette};

Map.addLayer(ee.Image(1), {palette:['white']});
Map.addLayer(d1, vis, 'VIIRS');
Map.addLayer(d2, vis, 'GHSL');
```

Exporting the results

Lastly, the raster products of our simple dasymetric disaggregation of population in Skaane region can be exported to Google Drive. The following commands export the `est_viirs` band from the `estimate` image as `GeoTIFF` with 100-m resolution. A standard `EPSG:4326` (WGS84) projection is applied to the image.

```
Export.image.toDrive({
  image: estimate.select('est_viirs'),
  description: 'dasymetric_viirs',
  folder: 'gee',
  region: adm1,
  fileFormat: 'GeoTIFF',
  crs: 'EPSG:4326',
  scale: 100
});
```