Practical 2: Assessing Health Facilities Coverage

GRID3

22 November 2019

This exercise provides an example of how we can use GRID3 data within an application.

The problem: We want to assess health facility coverage for maternal healthcare in Kaduna state. We are interested in finding out which areas are over-stretched with a high number of women of child-bearing age (WOCBA) per health-facility. By identifying a target number of people per health facility, we can begin to highlight locations that may need further invention.

Exercise Overview

Using geospatial analysis in R, this demo will show an assessment of health facility coverage for maternal health using the GRID3 population data for Kaduna State.

In this exercise we are going to:

1. Load some spatial data
2. Subset the data to focus on points of interest
3. Aggregate dataset using some basic geospatial techniques
4. Try and identify connections between datasets to set targets

Loading Packages

```r
require(sp)
require(sf)
require(raster)
require(dismo)
require(spatialEco)
require(tmap)
require(dplyr)
require(DT)
```

Step 1: Loading Datasets

We have four datasets used in this example:

> These datasets can be obtained from the GRID3 Nigeria Data portal at https://grid3.gov.ng/

1. **Health facility locations**:
2. **Ward boundaries**
3. **Gridded population (women age 15-49) data for Kaduna State**

4. **State boundary**

```
health_facilities <- read.csv("data/kaduna_health_sub.csv", fileEncoding="latin1")
population <- raster("data/nga_pop_wocba.tif")
ward_boundaries <- st_read("data/kaduna_wards.shp")
state_boundaries <- st_read("data/kaduna_state.shp")
```

We will view our health facilty data below. In total there are health centres recorded within the dataset. This includes data such as their location, whether the type of health centre *(primary, secondary, tertiary)*, and whether it is private or publicly owned. An example of some of the data is shown below:

```
datatable(head(health_facilities[8:18]),
          options = list(scrollX = TRUE)) # printing first rows of dataset
```

Show 10 entries

Search:

	latitude	longitude	ri_service	timestamp	cce_availa	category	global_id	name	functional	ty
1	10.707223	6.688533	Unknown	2018-09-25T14:24:37Z	Unknown	Primary Health Center	965e847e-a149-4edf-90d3-30c9461959fa	Pole Wire's Health Center	Unknown	Pri
2	10.725047	6.745297	Unknown	2018-09-25T14:34:39Z	Unknown	Primary Health Center	1938268f-426c-4221-a264-ee7bcfd6b2e0	Health Center Gayam	Functional	Pri
3	10.648297	7.041757	Unknown	2018-09-25T14:24:39Z	Unknown	Primary Health Clinic	ad534918-f452-4f81-b313-61ee4cbcbf36	Health Clinic Labi	Unknown	Pri
4	10.769058	6.642158	Unknown	2018-09-25T14:24:37Z	Unknown	Primary Health Clinic	4bbad2db-1bc5-4e28-9842-4abb8b2c5119	Girezin Primary Health Carep	Functional	Pri
5	10.777357	6.180544	Unknown	2018-09-25T14:24:37Z	Unknown	Primary Health Center	73c0922c-b6aa-427f-aba9-8614b26ecc6b	Kankangi Model Primary Health Center	Unknown	Pri
6	10.64994	6.35428	Yes	2018-09-25T14:24:32Z	Unknown	Primary Health Center	9700a9a6-aac3-469b-966b-afa1c9b19492	Sabon Layi Primary Health Center	Functional	Pri

Showing 1 to 6 of 6 entries

Previous 1 Next

Step 2: Filtering Public Health Facilities

For our example, we are only interested in **public** health centers. We will therefore filter the dataset below:

```
public_hf <- health_facilities %>% filter(ownership == "Public")
```

Step 3: Create Point Data

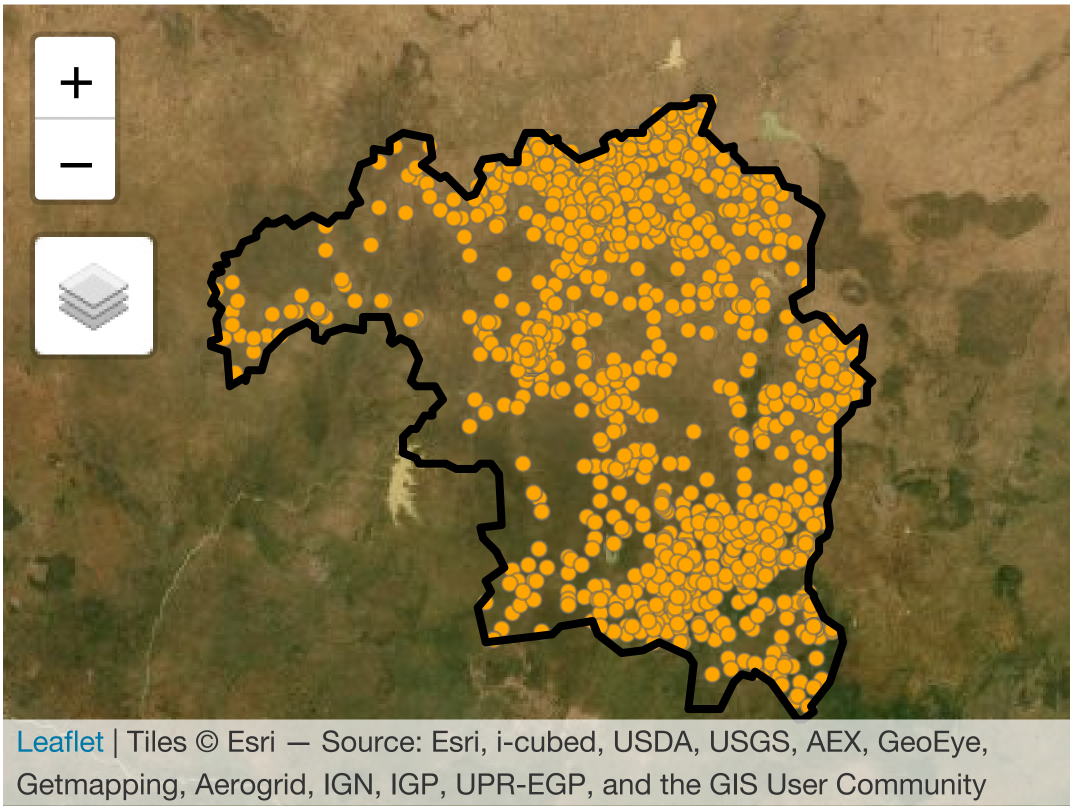
In order to conduct spatial analysis on the data, we must convert these into a spatial object:

```r
public_hf_pt <- st_as_sf(public_hf, coords = c("longitude", "latitude"))
```

We will quickly visualise this data below:

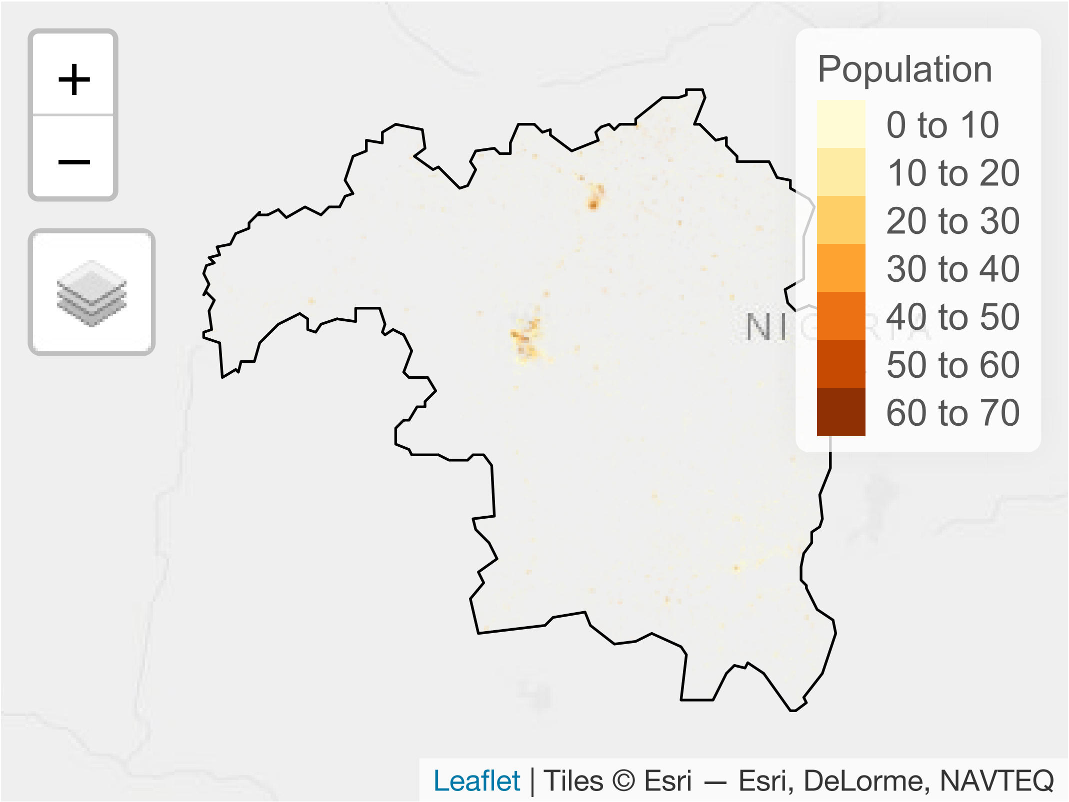
```r
tmap_mode("view") # to make map interactive

tm_shape(public_hf_pt) +
  tm_dots(id = "name", col = "orange") +
  tm_shape(state_boundaries) +
  tm_borders(lwd = 3, col = "black") +
  tm_basemap("Esri.WorldImagery")
```



Visualise the gridded population data:

```r
tm_shape(state_boundaries) +
  tm_borders(col = "black") +
  tm_shape(population) +
  tm_raster(title = "Population")
```



Step 4: Computing Voronoi Polygons

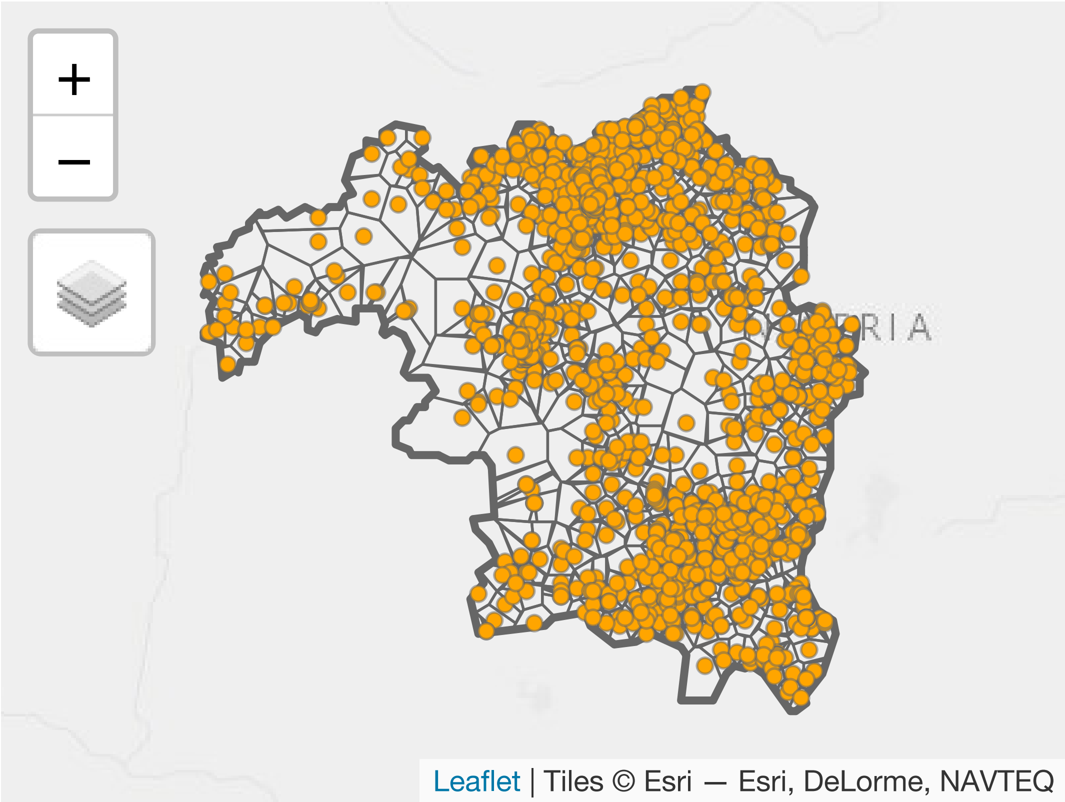
We are interested in finding out the health facility coverage across space.

- Idea: optimize the partitioning of the area into polygons such that each polygon contains one health facility.
- Method: Voronoi Polygon

```
voronoi <- dismo::voronoi(st_coordinates(public_hf_pt)) # compute voronoi from points coordinates
voronoi <- st_as_sf(voronoi) # convert into a "sf" object
voronoi <- cbind(voronoi, public_hf_pt)
st_crs(voronoi) <- st_crs(state_boundaries) # setting geographical coordinate system
voronoi <- voronoi %>% st_intersection(state_boundaries) # constrain polygon to state boundaries
```

Again, we will visualise these below:

```
tm_shape(voronoi) +
  tm_borders() +
  tm_shape(state_boundaries) +
  tm_borders(lwd = 3) +
  tm_shape(public_hf_pt) +
  tm_dots(col = "orange")
```



Step 5: Identifying Population Coverage Target with Zonal Statistics

Idea: find outliers in population per health facility to set a target.

Using Zonal statistics

Zonal statistic: summmary statistics of raster values at polygon level.

The Zonal statistic function requires more time than we have! Let's load the precalculation.

```
# z.stat <- zonal.stats(v, r, stat = sum, trace = FALSE, plot = FALSE)
pop_per_voronoi <- readRDS("data/pop_per_voronoi")
datatable(head(pop_per_voronoi))
```

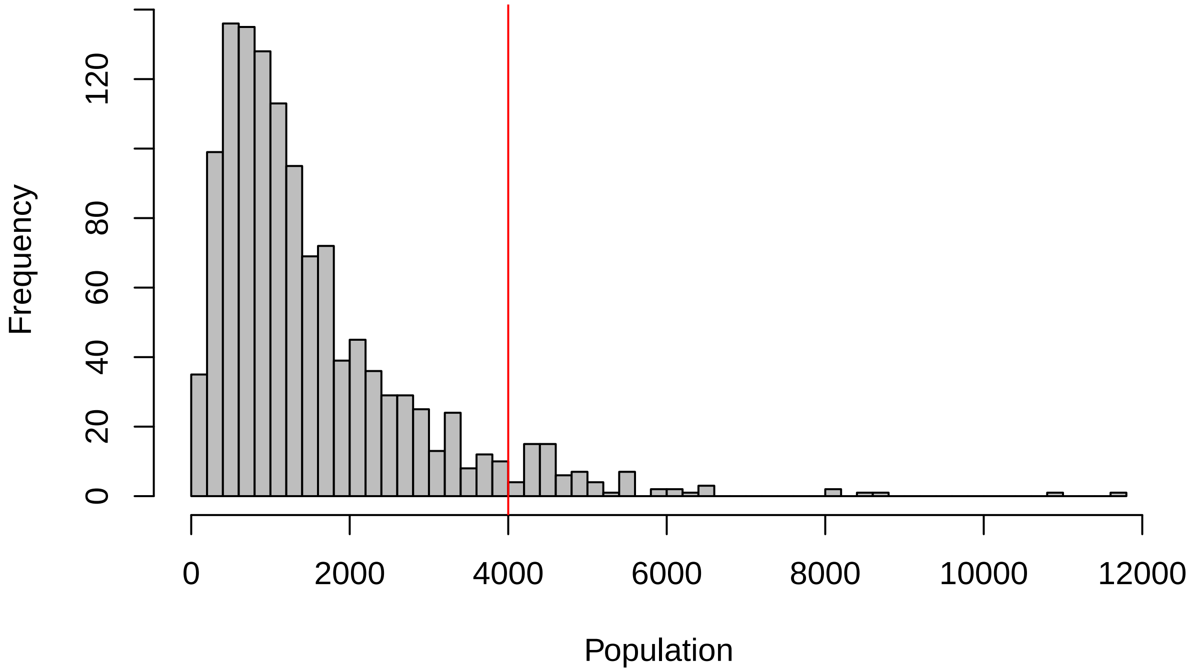
Show 10 entries Search:

	health_facility	population
1	Pole Wire's Health Center	1900.7906203568
2	Girezin Primary Health Carep	1530.22517833114
3	Kankangi Model Primary Health Center	2410.49551776052
4	Sabon Layi Primary Health Center	2004.24397283792
5	Ikon Allah Nursing And Maternity	2385.30325949192
6	Kakangi Primary Health Care	826.777569681406

Showing 1 to 6 of 6 entries Previous 1 Next

Setting a target

```
hist(pop_per_voronoi$population,
  breaks = 50,
  main = "",
  xlab = "Population",
  col = "grey"
)
abline(v = 4000, col = "red")
```



Step 6: Identifying areas above target

We are subsetting the data to only include those voronois above the 'target' that can be seen in orange. These would be recommended for further intervention, such as the expansion of existing health facilities or placement of new facilities.

```
voronoi <- voronoi %>% left_join(pop_per_voronoi, by = c("name" = "health_facility"))

above_target <- voronoi %>% filter(population > 4000)

tmap_mode("plot")
tm_shape(voronoi) +
  tm_borders() +
  tm_fill(col = "white") +
  tm_shape(above_target) +
  tm_borders() +
  tm_fill(col = "orange")
```