Zonal statistics for GHSL Settlement

Zonal statistics refers to the calculation of statistics on values of a raster within the zones of another dataset.

This project is a collaborative work of UXO India and IDFC.

In the following example the settlement of each state in India is calculated.

load required libraries and packages

```
library(rgdal) # To import raster data

library(maptools) # To plot the data

library(proj4)  # To reproject raster

library(xtable) # To export data to html tables

library (raster) # Required for rgdal

library (rgeos) # Required for maptools

library (spatstat) # Analysing spatial point patterns

library (tiff) # Read TIFF images and required for rgdal

library (sp) #Required for maptools

library (data.table) # Modifying columns

library (modeest) #To calculate mode value for the zone

library (foreign) # Required for maptools
```

Setting the memory limit

```
memory.limit(size = 100000)

## [1] 1e+05
```

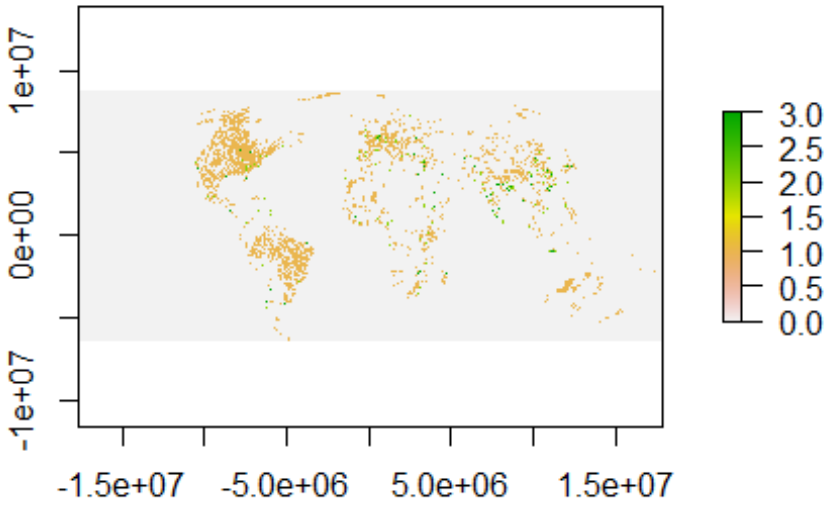
To read shapefile and assign to a variable zone

```
Zone<-readOGR("D:/K/New folder/R Markdown/Input/SHP","State") # To read shapesile (zone)

## OGR data source with driver: ESRI Shapefile
## Source: "D:/K/New folder/R Markdown/Input/SHP", layer: "State"
```

```
## with 36 features
## It has 4 fields

plot(Zone)
```

To read Raster data and assign to a variable Lumin and user can change the data as per requirement. This documentation is done by taking 2015 data as an example.

```
Lumin <- raster("D:/K/New folder/R Markdown/Input/GHSL settlement/GHS_SMOD_POP2015_GLOBE_R2016A_54009_1k_v1_0.tif")
plot(Lumin)
```



Define function for zonal statistics

```
Zonal_Stat <- function(x,y) # Function declaration and signature
{
k = x@crs #Assigning the raster projection to K
y = spTransform(y,k) #Reproject the vector to raster projection
A<-extract(x,y) # Extract raster data zone-wise
R<-array(0,dim=c(length(A),6)) # Create empty array
for (i in 1:length(A))  # Create for loop to find the zonal statistics
{
 temp=A[[i]]  #Get zone-wise data in temperary memory
 R[i,1]=length(temp) # Find number of pixel in that zone
 SRR1<-length(temp[temp==1]) #Assigning the length of class 1 of raster layer to SRR1
 SRR2<-length(temp[temp==2]) #Assigning the length of class 2 of raster layer to SRR2
 SRR3<-length(temp[temp==3]) #Assigning the length of class 3 of raster layer to SRR3
 R[i,2]= SRR1#Assign the SRR1 to column 2 of array R
 R[i,3]= SRR2 #Assign the SRR2 to column 3 of array R
 R[i,4]= SRR3 #Assign the SRR3 to column 4 of array R
 R[i,5]=min(temp) # Find minimum of that zone
 R[i,6]=max(temp) # Find maximum of that zone

 rm(temp) # remove temparary variable
}
colnames(R) <- c("Count","BAS","LDC","HDC","Min","Max") # Change column header
z<-cbind(y,R) # Bind data with shape file
```

```
return(z)
}
```

Call the zonal statistics function

```
M<-Zonal_Stat(Lumin,Zone)
M

## class       : SpatialPolygonsDataFrame
## features    : 36
## extent      : 6365419, 9372665, 834596.8, 4460278  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=moll +lon_0=0 +x_0=0 +y_0=0 +ellps=WGS84 +units=m +no_defs
## variables   : 10
## names       :                 Name, St_Csus_Cd,         Name_T,       Area,  Count,   BAS,   LDC,
HDC, Min, Max
## min values  : Andaman & Nicobar Island,          0,          State,    109.3623,    105,     2,     4,
0,   0,   2
## max values  :            West Bengal,         35, Union Territory, 342592.8684, 343863, 67215, 1
5404, 10709,   0,   3
```

Write the output to csv file

```
write.csv(M,"D:/K/New folder/R Markdown/Input/GHSL settlement/2015_ZS_Built_up.csv", na="NA")
```