

## Zonal statistics for HDC Population



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 population of HDC [High Density Cluster] is calculated.

**Load required libraries and packages [Packages are installed to use the required functions in the library]**

```
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
```

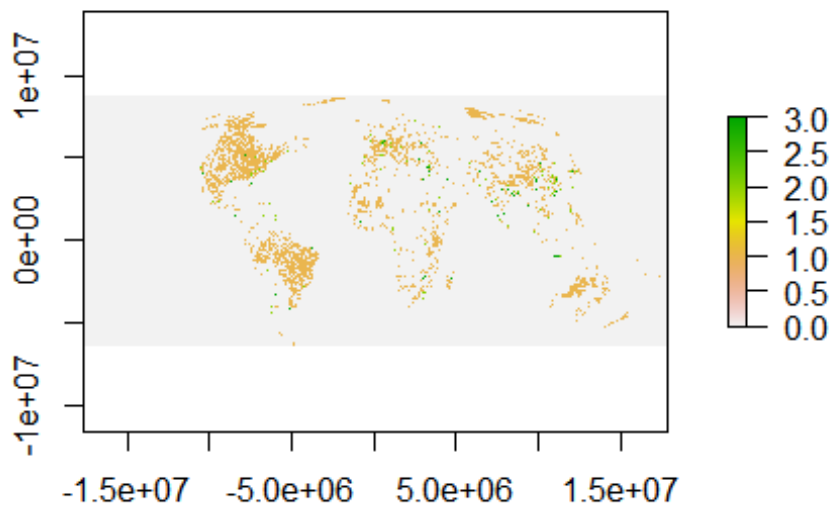
### Memory Limiting

```
memory.limit(size=100000)

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

### Reading the settlement raster layer

```
Lumin1 <- raster("D:/K/world HDC area population
zs/2000/Raster_Settlement/GHS_SMOD_POP2000_GLOBE_R2016A_54009_1k_v1_0.tif") #
To read Raster data
plot(Lumin1)
```



## Assigning the

raster to a variable R

```
R <- Lumin1
```

## Selecting the HDC (High Density crystal) layer from the settlement layer

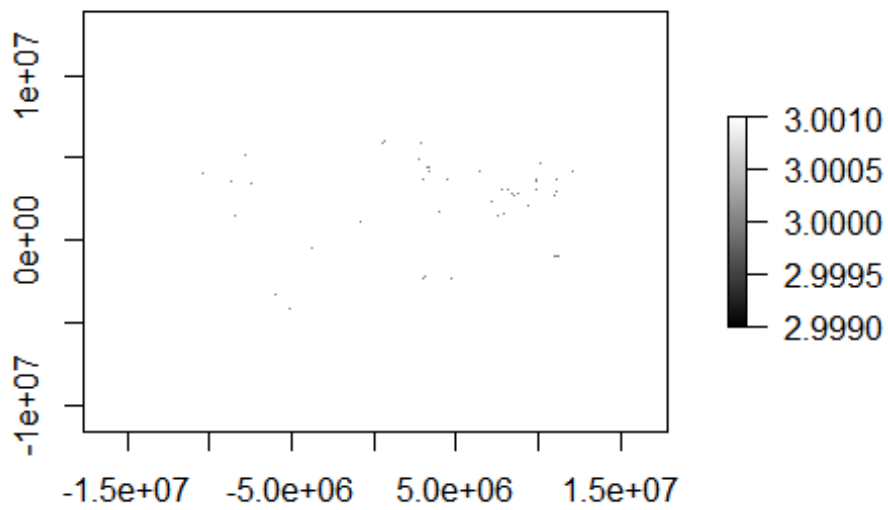
```
R[R<3] <- NA
```

## Writing the HDC layer to a new raster

```
writeRaster(R,
filename="D:/K/output/world_settlement_raster_HDC_c3_2000.tif",
format="GTiff", overwrite=TRUE)
```

## Read the HDC raster

```
Lumin2 <- raster("D:/K/output/world_settlement_raster_HDC_c3_2000.tif") # To
read Raster data
plot(Lumin2,col=grey(1:100/100))
```



to Polygons

## Convert Raster

```
pp <- rasterToPolygons(Lumin2,dissolve=TRUE) #Raster to vector function  
plot(pp)
```



## Write HDC

polygons to shapefile

```
writeOGR(pp, dsn = 'D:/K/output', layer
='world_settlement_raster_to_vector_c3_2000', driver = 'ESRI
Shapefile', check_exists=T, overwrite_layer=T) # To write vector data

## Warning in writeOGR(pp, dsn = "D:/K/output", layer =
## "world_settlement_raster_to_vector_c3_2000", : Field names abbreviated for
## ESRI Shapefile driver
```

### Read the HDC settlement in vector format

```
map <- readOGR("D:/K/output",
layer="world_settlement_raster_to_vector_c3_2000")

## OGR data source with driver: ESRI Shapefile
## Source: "D:/K/output", layer: "world_settlement_raster_to_vector_c3_2000"
## with 1 features
## It has 1 fields

plot(map)
```



HDC shapefile to world mercator

## Reproject the

```
map <- spTransform(map, CRS("+init=epsg:3395"))  
plot(map)
```



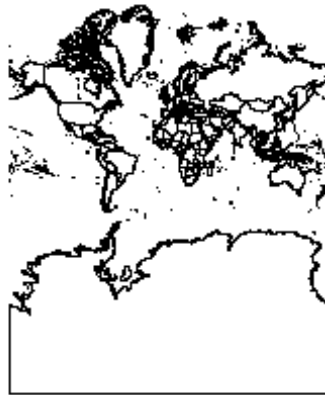
## Write the

reprojected HDC shapefile to a new shapefile

```
writeOGR(map, dsn = 'D:/K/output', layer  
='world_settlement_raster_to_vector_c3_2000_world_meracator', driver = 'ESRI  
Shapefile', check_exists=T, overwrite_layer=T)
```

## Read the all countries data

```
Country_zone <- readOGR("D:/K/world HDC area population  
zs/2000/shp", "All_country_world_mercator_r_area_calculation")  
  
## OGR data source with driver: ESRI Shapefile  
## Source: "D:/K/world HDC area population zs/2000/shp", layer:  
"All_country_world_mercator_r_area_calculation"  
## with 256 features  
## It has 73 fields  
  
plot(Country_zone)
```



```
## Read the  
reprojected HDC vector layer for merging function  
  
HDc2000_zone <-  
readOGR("D:/K/output", "world_settlement_raster_to_vector_c3_2000_transfer_mer  
acator_new")  
  
## OGR data source with driver: ESRI Shapefile  
## Source: "D:/K/output", layer:  
"world_settlement_raster_to_vector_c3_2000_transfer_meracator_new"  
## with 1 features  
## It has 1 fields  
  
plot(HDc2000_zone)
```



```
country shapefile with HDC settlement layer                                ## Merging the all  
crop_merge_hdc2000 <- crop (Country_zone,Hdc2000_zone)  
## Warning in crop(Country_zone, Hdc2000_zone): non identical CRS  
plot(crop_merge_hdc2000)
```





merged data to shapefile ## Write the

```
writeOGR(crop_merge_hdc2000, dsn = 'D:/K/output', layer
='clip_countrywise_hdc_shapefile_2000_world_mercator_new', driver = 'ESRI
Shapefile',check_exists=T, overwrite_layer=T)
```

## Reading the merged data shapefile of HDC and country file

```
Zone2<-
readOGR("D:/K/output","clip_countrywise_hdc_shapefile_2000_world_mercator_new
") # To read shapessile (zone)

## OGR data source with driver: ESRI Shapefile
## Source: "D:/K/output", layer:
"clip_countrywise_hdc_shapefile_2000_world_mercator_new"
## with 186 features
## It has 73 fields
```

## Calculating area of zones and add that to the attribute table

```
Zone2@data$area_km2_1 <- gArea(Zone2, byid = TRUE) / 1000000
```

## write the edited shapefile to a new vector layer

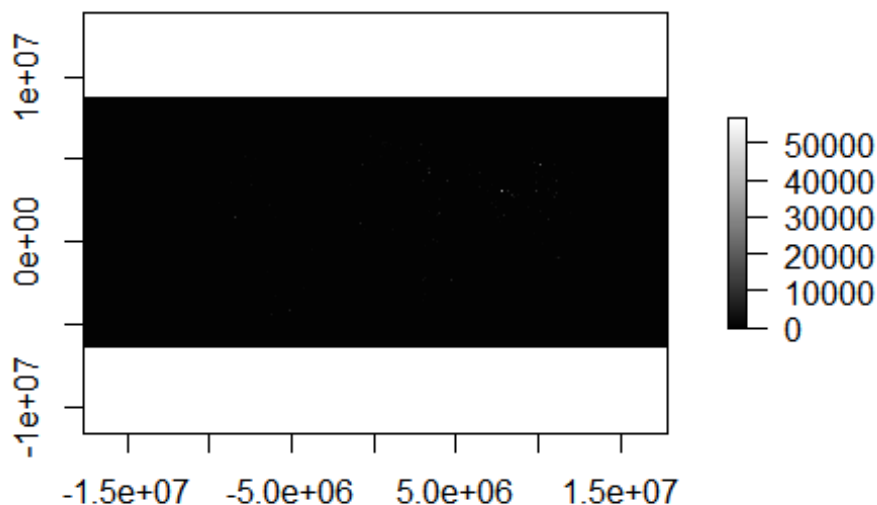
```
writeOGR(Zone2, dsn = 'D:/K/output', layer
='clip_countrywise_hdc_shapefile_2000_cal_area_world_mercator', driver =
'ESRI Shapefile',check_exists=T, overwrite_layer=T)
```

## Read the edited shapefile

```
Zone<-  
readOGR("D:/K/output","clip_countrywise_hdc_shapefile_2000_cal_area_world_mer  
cator")  
  
## OGR data source with driver: ESRI Shapefile  
## Source: "D:/K/output", layer:  
"clip_countrywise_hdc_shapefile_2000_cal_area_world_mercator"  
## with 186 features  
## It has 73 fields
```

## Read the population data

```
Lumin <- raster("D:/K/world HDC area population  
zs/2000/Raster_Population/2000_pop_1km.tif") # To read Raster data  
plot(Lumin,col=grey(1:100/100))
```



## Extract the  
values from the population data from the merged file, sum them and add it to the attribute  
table

```
out <- extract(Lumin, Zone, fun = sum, na.rm = T, small = T, df = T)
```

## Get the attribute data of merged shapefile

```
z <- Zone@data
```

## Combine the extract values(out) to z

```
M <- cbind(z,out)
```

## Write the bind data to csv file

```
write.csv(M,"D:/k/output/world_countrywise_HDc_area_population_2000_world_mer  
cator_final.csv", na="NA") # Enter Output csv file name and path
```