7.1 and 7.2 Fragstats Metrics within Polygons

Manual of Applied Spatial Ecology

3/28/2019

Some research designs may just need landscape metrics for a single area or several study areas and that is what the SDMToolsl package is able to estimate in the code that follows. While the single area can be defined by the extent of the raster we imported as in previous chapters, the ability of the SDMToolsl package to determine patch and class statistics depends on the area defined by the user from that could be study site, within polygons such as counties or townships, or within buffers around locations.

1. Exercise 7.1 and 7.2 - Download and extract zip folder into your preferred location

2. Set working directory to the extracted folder in R under Session - Set Working Directory. . .

3. Now open the script "Frag_Auto.Rmd" and run code directly from the script

4. First we need to load the packages needed for the exercise

```
library(SDMTools)
library(raster)
library(plyr)
library(maptools)
library(rgdal)
```

5. Now let's have a separate section of code to include projection information we will use throughout the exercise. In previous versions, these lines of code were within each block of code

```
new.crs <-CRS("+proj=lcc +lat_1=41.95 +lat_2=40.88333333333333 +lat_0=40.16666666666666
 +lon_0=-77.75 +x_0=600000 +y_0=0 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs")
```

6. Load vegetation raster layer clipped in ArcMap

```
crops <-raster("crop2012utm12.tif")
plot(crops)
class(crops)

# reclassify the values into 9 groups
# all values between 0 and 20 equal 1, etc.
m <- c(-Inf,0,NA,2, 7, 2, 20, 60, 3, 60, 70, 4, 110, 132, 5, 133, 150, 6, 151, 191, 7,
  192,Inf,NA)
rclmat <- matrix(m, ncol=3, byrow=TRUE)
rc <- reclassify(crops, rclmat)
plot(rc)
as.matrix(table(values(rc)))

#Now we get into Landscape Metrics with the SDTM Tool
#Calculate the Patch statistics
ps.data = PatchStat(rc)
ps.data

#Calculate the Class statistics
cl.data = ClassStat(rc)
cl.data
```

7. Some research designs may need landscape metrics for several areas that may be available as a shapefile or some other polygon layer.

```r
#Load raster file into R
raster <- raster("county_hab")

#Load PA shapefile into R
HareCounties <- readOGR(dsn=".", layer="Hare_Counties", verbose=FALSE)
image(raster)
plot(HareCounties, add=T)

#Let's project Counties to the same projection as the habitat raster
county <- spTransform(HareCounties, CRS=new.crs)
HareCounties <- county
#Matching projections successful!
row.names(HareCounties)<-as.character(HareCounties$COUNTY_NAM)
names.polygons<-sapply(HareCounties@polygons, function(x) slot(x,"ID"))
text(coordinates(HareCounties), labels=sapply(slot(HareCounties, "polygons"), function(i)
  slot(i, "ID")), cex=0.8)
```
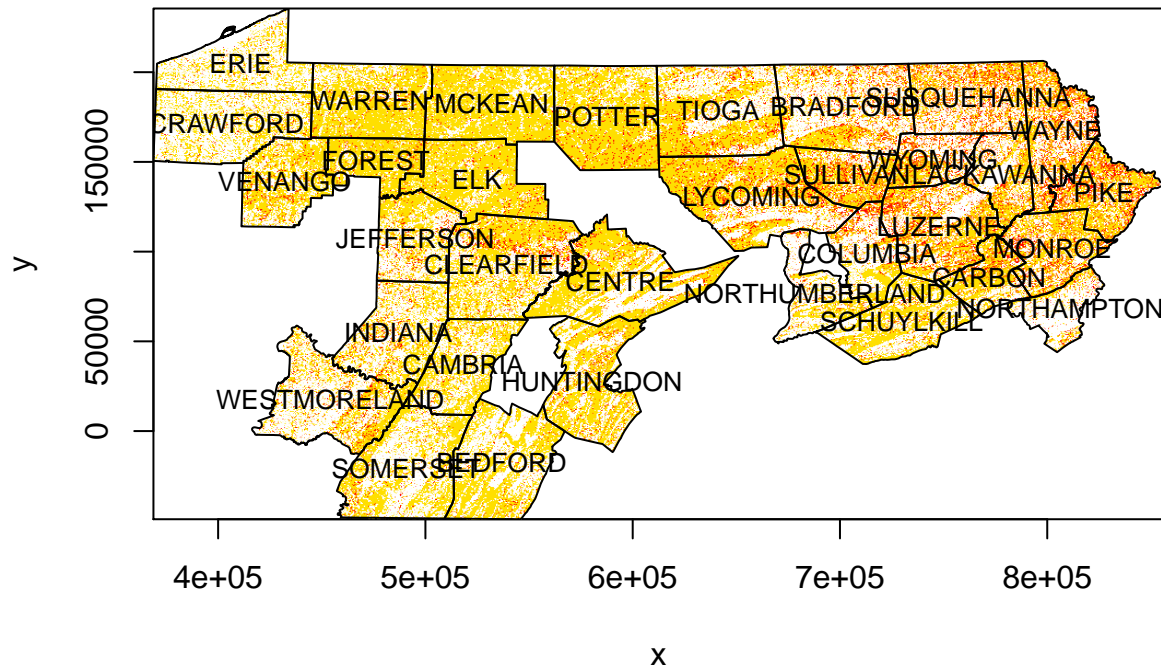

8. Now we want to export by County name (i.e., COUNTY_NAM) as individual shapefiles. We will only select the first 2 counties for processing to save time

```r
indata <- HareCounties
innames <- unique(HareCounties@data$COUNTY_NAM)
innames <- innames[1:2]#Place the number of unique polygons in your shapefile here
outnames <- innames

# set up output table
#output <- as.data.frame(matrix(0,nrow=length(innames),ncol=38))

# begin loop to create separate county shapefiles
for (i in 1:length(innames)){
  data <- indata[which(indata$COUNTY_NAM==innames[i]),]
```

```r
  if(dim(data)[1] != 0){
    #writePolyShape(data,fn=paste(outnames[i],sep="/"),factor2char=TRUE)
    writeOGR(data, dsn = ".", layer=paste(outnames[i],sep="/"), driver = "ESRI Shapefile")
    write.table(innames, "List.txt", eol=".shp\n", col.names=FALSE, quote=FALSE,
  row.names=FALSE)
}
}

#Read in a list of shapefiles files from above
Listshps<-read.table("List.txt",sep="\t",header=F)
#colnames(Listshps) <- c("id")
Listshps

shape <- function(Listshps) {
file <- as.character(Listshps[1,])
shp <- readShapeSpatial(file)
mask <- mask(raster,shp)
### Calculate the Class statistics in each county
cl.data <- ClassStat(mask)
}

results <- ddply(Listshps, 1, shape)
results
#write.table(results, "FragCounty.txt")
```