1.8 Creating a Hexagonal Polygon Grid Over a Study Area

Manual of Applied Spatial Ecology

3/11/2022

1. Exercise 1.8 - 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 "GridScripts.Rmd" and run code directly from the script
4. First we need to load the packages needed for the exercise

```
library(rgdal)
library(rgeos)
library(raster)
```

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

```
crs<-"+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0"
Albers.crs <-CRS("+proj=aea +lat_0=23 +lon_0=-96 +lat_1=29.5 +lat_2=45.5 +x_0=0 +y_0=0 +datum=NAD83 +uni
```

6. Also need to import several shapefiles for mule deer from Section 1.3

```
study.counties<-readOGR(dsn=".",layer="MDcounties", verbose = FALSE)
class(study.counties)#Shows class and package used
```

```
## [1] "SpatialPolygonsDataFrame"
## attr(,"package")
## [1] "sp"
```

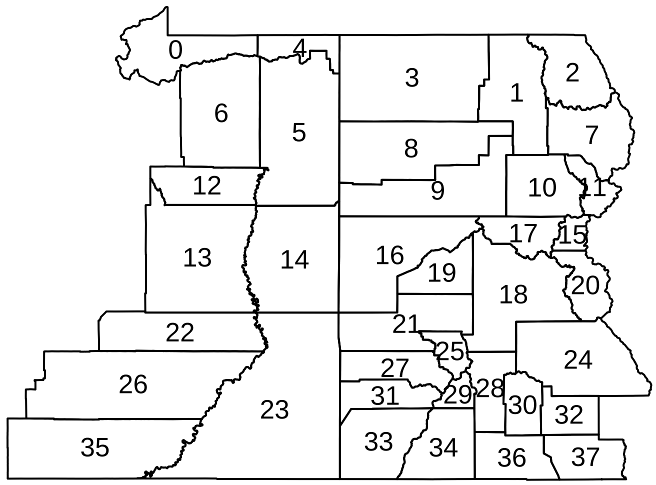
```
proj4string(study.counties)#Shows projection information
```

```
## [1] "+proj=longlat +datum=WGS84 +no_defs"
```

```
plot(study.counties)#plots study sites on map
study.counties@data$StateCO#Displays labels for counties
```

```
##  [1] "Summit County, UT"     "Routt County, CO"      "Jackson County, CO"
##  [4] "Moffat County, CO"     "Daggett County, UT"    "Uintah County, UT"
##  [7] "Duchesne County, UT"   "Grand County, CO"      "Rio Blanco County, CO"
## [10] "Garfield County, CO"   "Eagle County, CO"      "Summit County, CO"
## [13] "Carbon County, UT"     "Emery County, UT"      "Grand County, UT"
## [16] "Lake County, CO"       "Mesa County, CO"       "Pitkin County, CO"
## [19] "Gunnison County, CO"   "Delta County, CO"      "Chaffee County, CO"
## [22] "Montrose County, CO"   "Wayne County, UT"      "San Juan County, UT"
## [25] "Saguache County, CO"   "Ouray County, CO"      "Garfield County, UT"
## [28] "San Miguel County, CO" "Hinsdale County, CO"   "San Juan County, CO"
## [31] "Mineral County, CO"    "Dolores County, CO"    "Rio Grande County, CO"
## [34] "Montezuma County, CO"  "La Plata County, CO"   "Kane County, UT"
## [37] "Archuleta County, CO"  "Conejos County, CO"
```

```r
#Labels each county with @plotOrder of each polygon (i.e., county)
text(coordinates(study.counties), labels=sapply(slot(study.counties, "polygons"),
  function(i) slot(i, "ID")), cex=0.8)
```




```r
muleys <-read.csv("muleysexample.csv", header=T)

#Remove outlier locations
newmuleys <-subset(muleys, muleys$Long > -110.50 & muleys$Lat > 37.8 & muleys$Long < -107)
muleys <- newmuleys
```

7. Identify the columns with coordinates then make a spatial data frame of locations after removing outliers

```r
coords<-data.frame(x = muleys$Long, y = muleys$Lat)
head(coords)
```

```
##           x        y
## 1 -108.9784 37.86562
## 2 -108.9781 37.86521
## 3 -108.9794 37.86471
## 4 -108.9811 37.86176
## 5 -108.9803 37.86142
## 6 -108.9799 37.86124
```

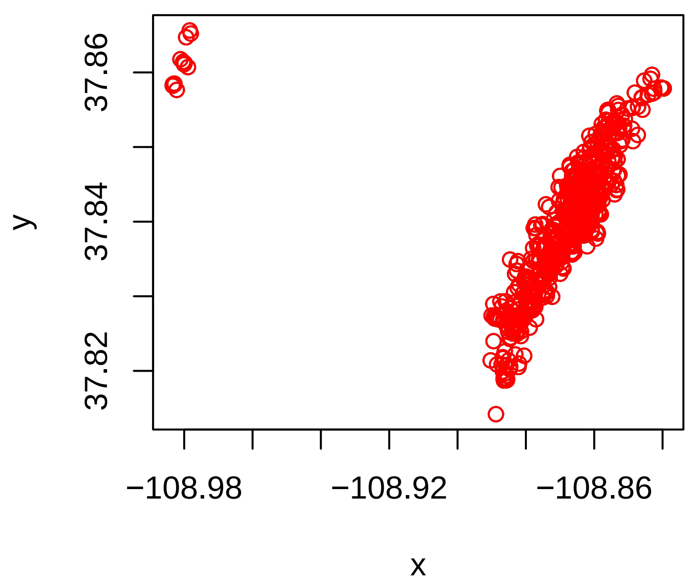
```r
plot(coords)

deer.spdf <- SpatialPointsDataFrame(coords= coords, data = muleys, proj4string = CRS(crs))
class(deer.spdf)
```

```
## [1] "SpatialPointsDataFrame"
## attr(,"package")
## [1] "sp"
```

```
proj4string(deer.spdf)
```

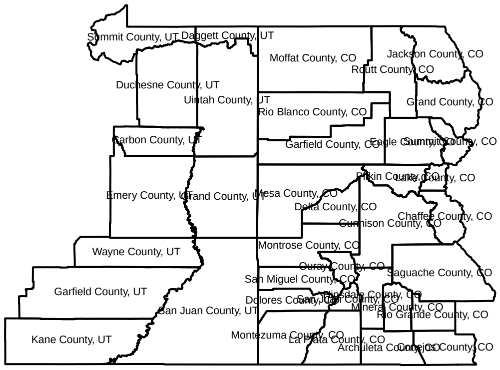
```
## [1] "+proj=longlat +datum=WGS84 +no_defs"
```

```
points(deer.spdf,col="red")
```



8. Rename labels by county name otherwise plot order would be used because duplicate counties within each state (i.e., CO, UT) occured in original shapefile from ArcMap

```
row.names(study.counties)<-as.character(study.counties$StateCO)
names.polygons<-sapply(study.counties@polygons, function(x) slot(x,"ID"))
#Now add labels of State and County to Map
plot(study.counties)
text(coordinates(study.counties), labels=sapply(slot(study.counties, "polygons"),
  function(i) slot(i, "ID")), cex=0.3)
```



9. Now lets extract counties within the extent of the mule deer locations

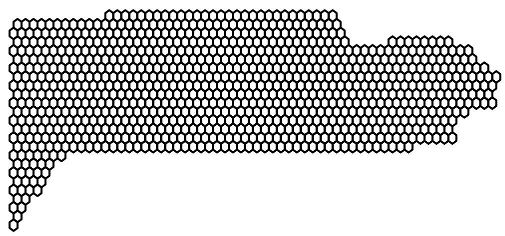
```
int <- gIntersection(study.counties,deer.spdf)#requires rgeos library
clipped <- study.counties[int,]
MDclip <- as(clipped, "SpatialPolygons")
plot(MDclip,pch=16)
#Now add labels of State and County to Map
text(coordinates(MDclip), labels=sapply(slot(MDclip, "polygons"),
  function(i) slot(i, "ID")), cex=0.8)
```


```
bbox(MDclip)
```

```
##          min       max
## x -109.04440 -107.8615
## y   37.47728   37.8964
```

10. We also can create a hexagonal grid across the study site

```
HexPts <-spsample(MDclip,type="hexagonal", n=1000, offset=c(0,0))
HexPols <- HexPoints2SpatialPolygons(HexPts)
#proj4string(HexPols) <- CRS(crs)
plot(HexPols)
```

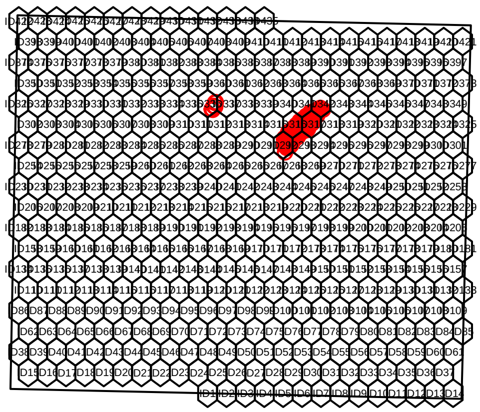


11. Create this hexagonal grid across our study site by zooming into deer locations from Section 1.3

```
#Import the study site zoomed in shapefile
study.zoom<-readOGR(dsn=".",layer="MDzoom")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "D:\OneDrive - The Pennsylvania State University\CourseExercises\Chapter1\Exercise_1.8_Hexag
## with 1 features
## It has 1 fields
```

```
plot(study.zoom,pch=16)
points(deer.spdf,col="red")

#Create new hexagonal grid
HexPts2 <-spsample(study.zoom,type="hexagonal", n=500, offset=c(0,0))
HexPols2 <- HexPoints2SpatialPolygons(HexPts2)
proj4string(HexPols2) <- CRS(crs)
plot(HexPols2, add=T)
#Now add labels to each hexagon for unique ID
text(coordinates(HexPols2), labels=sapply(slot(HexPols2, "polygons"),
  function(i) slot(i, "ID")), cex=0.3)
```



12. We can intersect the mule deer locations with the polygon shapefile (i.e., county) they occured in if needed

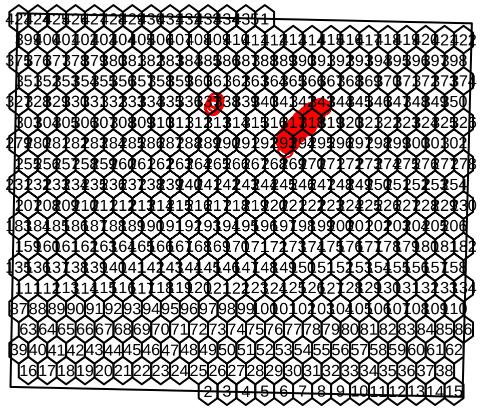
```
o = over(deer.spdf,study.counties)
new = cbind(deer.spdf@data, o)

#Used to rename labels by hexagonal grid ID only otherwise plot order with "IDxx" would be used
#and would throw an error (i.e., ID2, ID3)
row.names(HexPols2)<-as.character(HexPols2@plotOrder)
names.hex<-sapply(HexPols2@polygons, function(x) slot(x,"ID"))
```

13. As an aside, we can explore how to assign the area a location occurs in by intersecting points within the polygon shapefile.

```
o2 = over(deer.spdf,HexPols2)
new2 = cbind(deer.spdf@data,o2)
HexPols2
```

```
## class       : SpatialPolygons
## features    : 435
## extent      : -109.2448, -108.6473, 37.55422, 37.96259  (xmin, xmax, ymin, ymax)
## crs         : +proj=longlat +datum=WGS84 +no_defs
```

```
#Now plot with new grid IDs
plot(study.zoom,pch=16)
points(deer.spdf,col="red")
plot(HexPols2, add=T)
#Now add labels of State and County to Map
text(coordinates(HexPols2), labels=sapply(slot(HexPols2, "polygons"),
  function(i) slot(i, "ID")), cex=0.5)
```



14. As an alternative to importing a polygon that we created in ArcMap, we can create a polygon in R using the coordinates of the boundary box of the area of interest. In our case here, the bounding box will be the mule deer locations.

```
proj4string(deer.spdf)
```

```
## [1] "+proj=longlat +datum=WGS84 +no_defs"
```

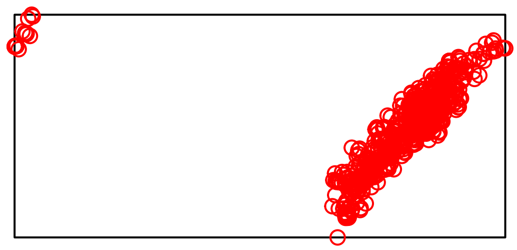
```
bbox(deer.spdf@coords)
```

```
##         min        max
## x -108.9834 -108.83966
## y   37.8142   37.86562
```

```
bb <- cbind(x=c(-108.83966,-108.83966,-108.9834,-108.9834, -108.83966),
  y=c(37.8142, 37.86562,37.86562,37.8142,37.8142))
SP <- SpatialPolygons(list(Polygons(list(Polygon(bb)),"1")), proj4string=CRS(proj4string(MDclip)))
plot(SP)
proj4string(SP)
```

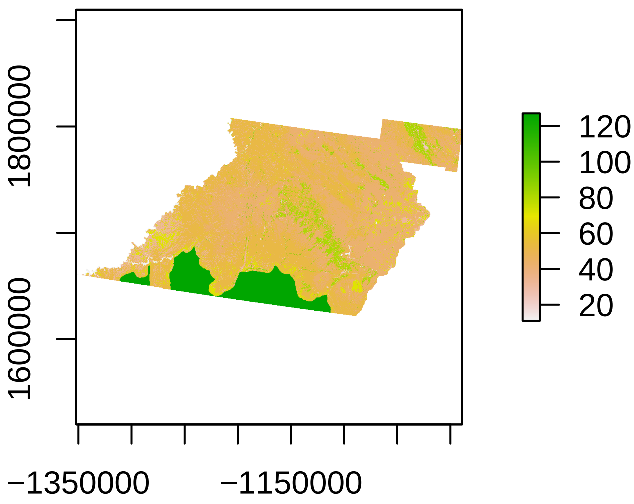
```
## [1] "+proj=longlat +datum=WGS84 +no_defs"
```

```
points(deer.spdf,col="red")
```



15. Now make practical use of the new bounding box we created by clipping a larger raster dataset. A smaller raster dataset runs analyses faster, provides a zoomed in view of mule deer locations and vegetation, and is just easier to work with.

```
#Load vegetation raster layer textfile clipped in ArcMap
veg <-raster("extentnlcd2.txt")
plot(veg)
```




```
class(veg)
```

```
## [1] "RasterLayer"
## attr(,"package")
## [1] "raster"
```

```
#Clip using the raster imported with "raster" package
bbclip <- crop(veg, SP)
veg
```

```
#WON'T WORK because projections are not the same, WHY?
#Let's check projections of layers we are working with now.
proj4string(MDclip)
```

```
## [1] "+proj=longlat +datum=WGS84 +no_defs"
```

```
proj4string(deer.spdf)
```

```
## [1] "+proj=longlat +datum=WGS84 +no_defs"
```

```
proj4string(SP)
```

```
## [1] "+proj=longlat +datum=WGS84 +no_defs"
```

```
proj4string(veg)
```

```
## [1] "+proj=aea +lat_0=23 +lon_0=-96 +lat_1=29.5 +lat_2=45.5 +x_0=0 +y_0=0 +datum=NAD83 +units=m +no_d
```

16. We need to have all layers in same projection so project the deer.spdf to Albers and then clip vegetation layer with new polygon we created in the Albers projection.

```
deer.albers <-spTransform(deer.spdf, CRS=Albers.crs)
class(deer.albers)
```

```
## [1] "SpatialPointsDataFrame"
## attr(,"package")
## [1] "sp"
```

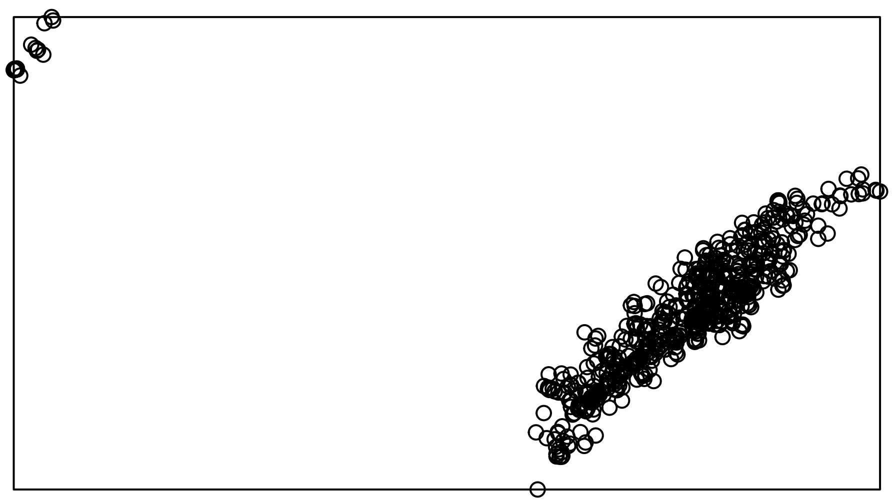
```
proj4string(deer.albers)
```

```
## [1] "+proj=aea +lat_0=23 +lon_0=-96 +lat_1=29.5 +lat_2=45.5 +x_0=0 +y_0=0 +datum=NAD83 +units=m +no_
```

```
bbox(deer.albers)
```

```
##         min      max
## x -1127964 -1115562
## y  1718097  1724867
```

```
bb1 <- cbind(x=c(-1115562,-1115562,-1127964,-1127964,-1115562),
  y=c(1718097, 1724867,1724867,1718097,1718097))
AlbersSP <- SpatialPolygons(list(Polygons(list(Polygon(bb1)),"1")),
  proj4string=CRS(proj4string(deer.albers)))
plot(AlbersSP)
points(deer.albers)
```



```
#Check to see all our layers are now in Albers projection
proj4string(veg)
```

```
## [1] "+proj=aea +lat_0=23 +lon_0=-96 +lat_1=29.5 +lat_2=45.5 +x_0=0 +y_0=0 +datum=NAD83 +units=m +no_
```

```
proj4string(deer.albers)
```

```
## [1] "+proj=aea +lat_0=23 +lon_0=-96 +lat_1=29.5 +lat_2=45.5 +x_0=0 +y_0=0 +datum=NAD83 +units=m +no_
```

```r
proj4string(AlbersSP)
```

```
## [1] "+proj=aea +lat_0=23 +lon_0=-96 +lat_1=29.5 +lat_2=45.5 +x_0=0 +y_0=0 +datum=NAD83 +units=m +no_d
```

```r
#Clip using the raster imported with "raster" package
bbclip <- crop(veg, AlbersSP)
plot(bbclip)
points(deer.albers, col="red")
plot(AlbersSP, lwd=5, add=T)
text(coordinates(AlbersSP), labels="Colorado Mule Deer")
```