1.9 Creating a Square Polygon Grid Over a Study Area

Manual of Applied Spatial Ecology

3/11/2022

1. Exercise 1.9 - 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 "GridSystem2Script.Rmd" and run code directly from the script
4. First we need to load the packages needed for the exercise

```
library(raster)
library(adehabitatMA)
```

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_1=29.5 +lat_2=45.5 +lat_0=23
  +lon_0=-96 +x_0=0 +y_0=0 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs")
crs2 <- "+proj=aea +lat_1=29.5 +lat_2=45.5 +lat_0=23 +lon_0=-96
  +x_0=0 +y_0=0 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs"
# NOTE: The difference between crs and Albers.crs is one is
# used to define projection that and the other to project a
# layer, respectively.
```

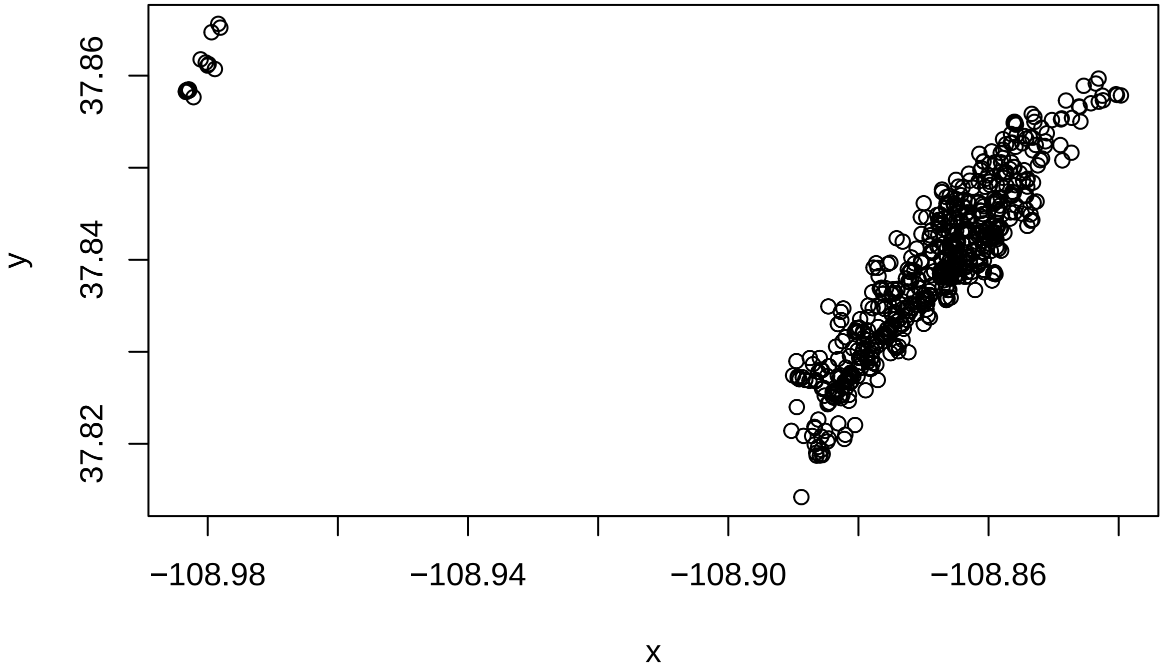
6. We need to have all layers in same projection so import, create, and remove outliers for mule deer locations then project all to the Albers projection as we did previously.

```
muleys <- read.csv("muleysexample.csv", header = T)
summary(muleys$id)
```

```
##    Length     Class      Mode
##      1091 character character
```

```
# Remove outlier locations
newmuleys <- subset(muleys, muleys$Long > -110.5 & muleys$Lat >
    37.8 & muleys$Long < -107)
muleys <- newmuleys

# Make a spatial data frame of locations after removing
# outliers
coords <- data.frame(x = muleys$Long, y = muleys$Lat)
plot(coords)
```




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

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

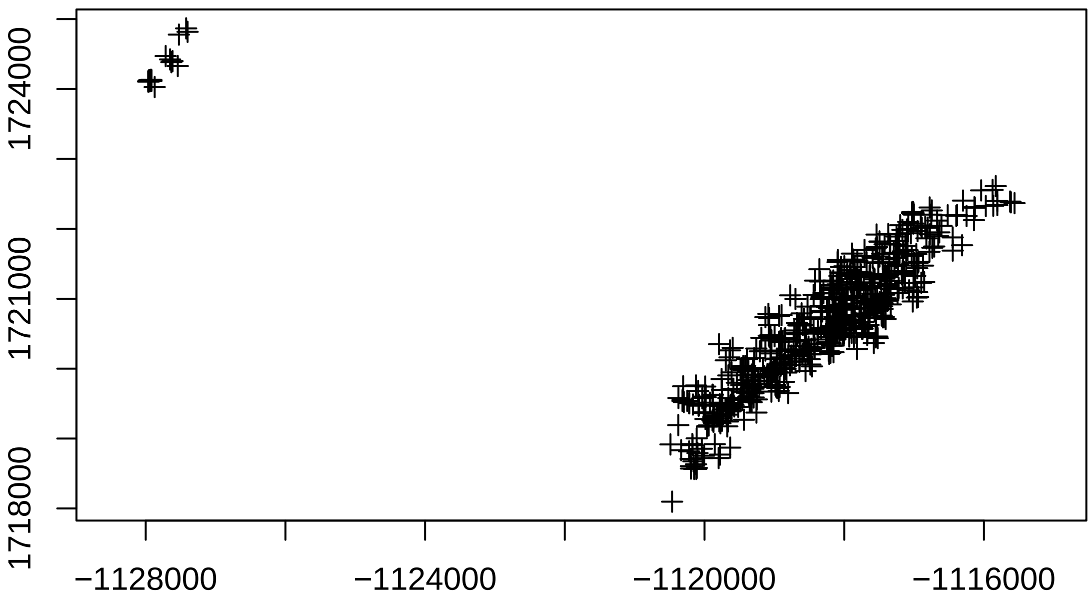
```
# Project deer.spdf to Albers as in previous exercise
deer.albers <- spTransform(deer.spdf, CRS = Albers.crs)
proj4string(deer.albers)
```

```
## [1] "+proj=aea +lat_0=23 +lon_0=-96 +lat_1=29.5 +lat_2=45.5 +x_0=0 +y_0=0 +ellps=GRS80 +units=m +no_
```

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

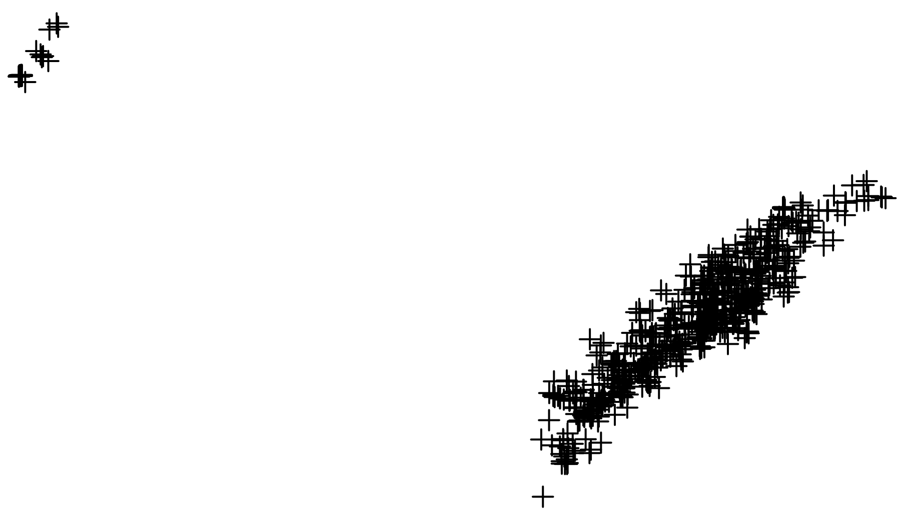
```
##         min      max
## x -1127964 -1115562
## y  1718097  1724867
```

```
plot(deer.albers, axes = T)
```



7. Create points for x and y from the bounding box of all mule deer locations with 1500 m spacing between each point.

```
plot(deer.albers)
```



```r
# Create vectors of the x and y points
x <- seq(from = -1127964, to = -1115562, by = 1500)
y <- seq(from = 1718097, to = 1724867, by = 1500)
```

8. Create a grid of all pairs of coordinates (as a data.frame) using the "expand grid" function and then make it a gridded object.

```r
xy <- expand.grid(x = x, y = y)
class(xy)
```

```
## [1] "data.frame"
```

```r
# Make grid points into a Spatial Points Data Frame
grid.pts <- SpatialPointsDataFrame(coords = xy, data = xy, proj4string = CRS(crs2))
proj4string(grid.pts)
```

```
## [1] "+proj=aea +lat_0=23 +lon_0=-96 +lat_1=29.5 +lat_2=45.5 +x_0=0 +y_0=0 +ellps=GRS80 +units=m +no_
```

```r
gridded(grid.pts)
```

```
## [1] FALSE
```

```r
class(grid.pts)
```

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

```r
# Make points a gridded object (i.e., TRUE or FALSE)
gridded(grid.pts) <- TRUE
```

```r
gridded(grid.pts)
```

```
## [1] TRUE
```

9. Make the grid of points into a Spatial Polygon then convert the spatial polygons to a SpatialPolygons-DataFrame.

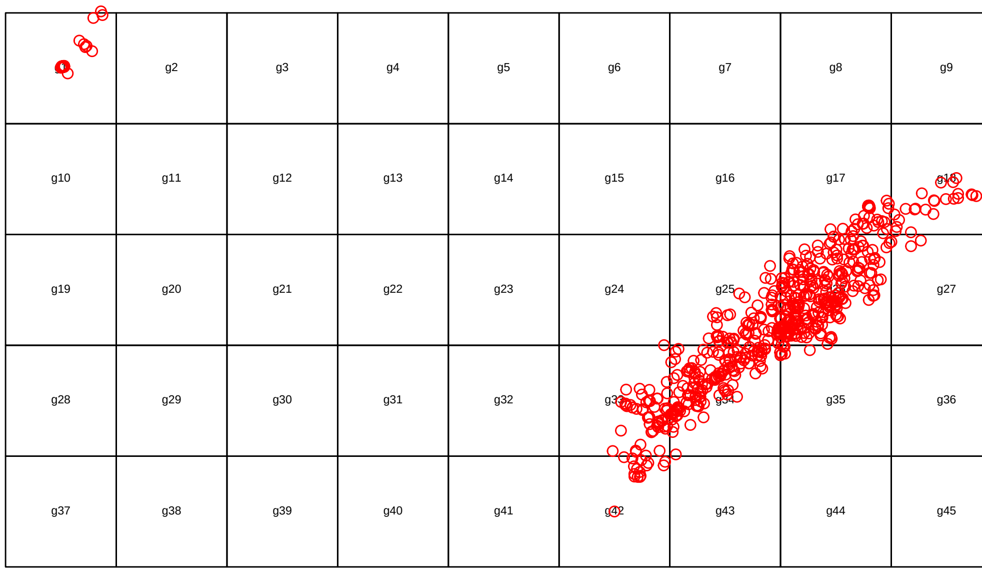
```r
grid <- as(grid.pts, "SpatialPolygons")
plot(grid)
class(grid)
```

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

```r
summary(grid)
```

```
## Object of class SpatialPolygons
## Coordinates:
##         min      max
## x -1128714 -1115214
## y  1717347  1724847
## Is projected: TRUE
## proj4string :
## [+proj=aea +lat_0=23 +lon_0=-96 +lat_1=29.5 +lat_2=45.5 +x_0=0 +y_0=0
## +ellps=GRS80 +units=m +no_defs]
```

```r
gridspdf <- SpatialPolygonsDataFrame(grid, data = data.frame(id = row.names(grid),
    row.names = row.names(grid)))
names.grd <- sapply(gridspdf@polygons, function(x) slot(x, "ID"))
text(coordinates(gridspdf), labels = sapply(slot(gridspdf, "polygons"),
    function(i) slot(i, "ID")), cex = 0.5)
points(deer.albers, col = "red")
```



g2
g3
g4
g5
g6
g7
g8
g9
g10
g11
g12
g13
g14
g15
g16
g17
g18
g19
g20
g21
g22
g23
g24
g25
g27
g28
g29
g30
g31
g32
g33
g34
g35
g36
g37
g38
g39
g40
g41
g42
g43
g44
g45

10. Similar to the hexagonal grid, identify the cell ID that contains each mule deer location.

```
o = over(deer.albers, gridspdf)
head(o)
```

```
##      id
## 20 <NA>
## 21   g1
## 22   g1
## 23   g1
## 24   g1
## 25   g1
```

```
new = cbind(deer.albers@data, o)
```

11. We get some NA errors because our grid does not encompass all mule deer locations so expand the grid then re-run the code over from xy through new2 again (i.e., Lines 62-86).

```
x <- seq(from = -1127964, to = -1115562, by = 1500)
y <- seq(from = 1718097, to = 1725867, by = 1500)
```

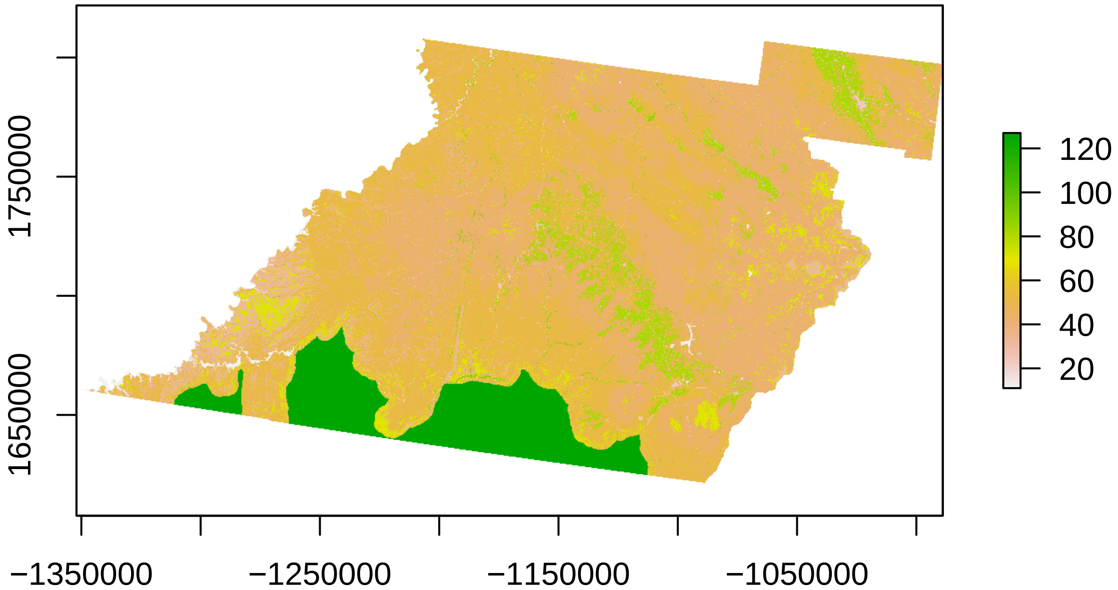
```
## BE SURE TO RUN CODE FROM XY CREATION THROUGH NEW2 AGAIN
## THEN LOOK AT DATA!!
o2 = over(deer.albers, gridspdf)
head(o2)
```

```
##      id
## 20 <NA>
## 21   g1
## 22   g1
## 23   g1
## 24   g1
## 25   g1
```

```
new2 = cbind(deer.albers@data, o2)  #No more NAs causing errors!
```

12. Now we can load a vegetation raster layer textfile clipped in ArcMap to summarize vegetation categories within each polygon grid cell.

```
veg <- raster("extentnlcd2.txt")
plot(veg)
```

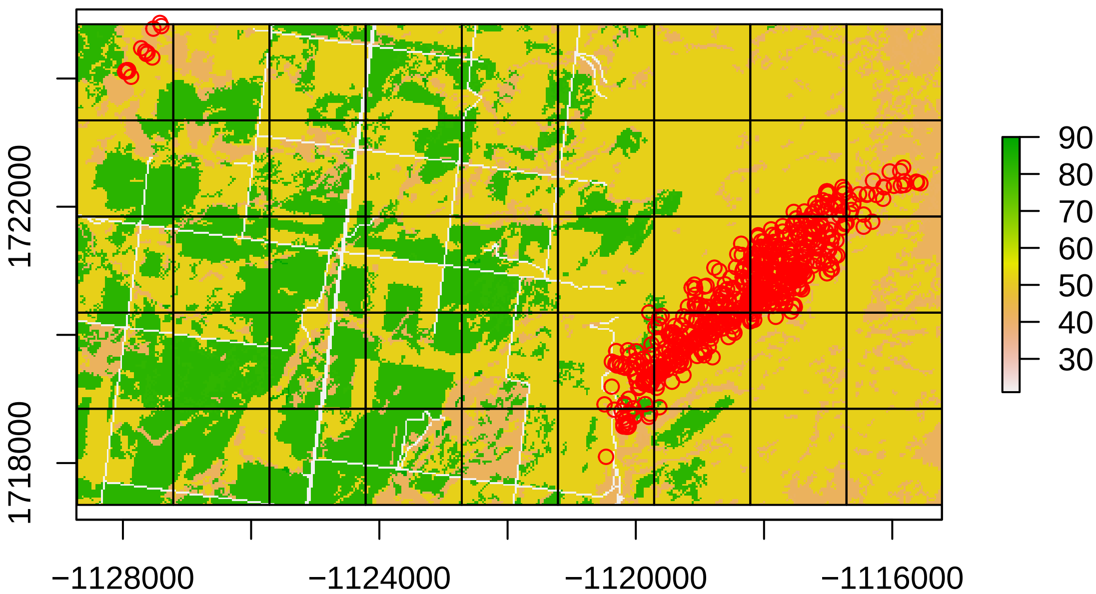



```
class(veg)
```

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

13. Clip the raster within the extent of the newly created grid

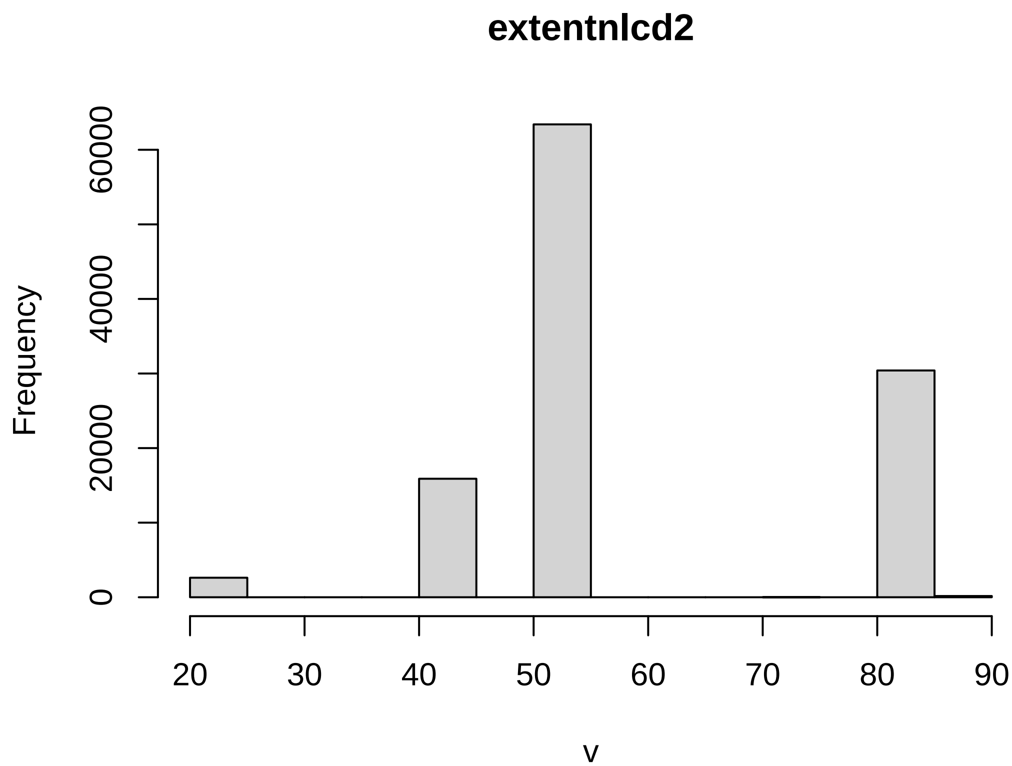
```
bbclip <- crop(veg, gridspdf)
plot(bbclip)
points(deer.albers, col = "red")
plot(gridspdf, add = T)
```




```r
# Cell size of raster layer
xres(bbclip)
```

```
## [1] 30
```

```r
# Create histogram of vegetation categories in bbclip
hist(bbclip)
```




```
# Calculate cell size in square meters
ii <- calcperimeter(gridspdf)  #requires adehabitatMA package
as.data.frame(ii[1:5, ])
```

```
##      id perimeter
## g37 g37      6000
## g38 g38      6000
## g39 g39      6000
## g40 g40      6000
## g41 g41      6000
```

14. We can extract the vegetation characteristics within each polygon of the grid.

```
table = extract(bbclip, gridspdf)
str(table[1])
```

```
## List of 1
##  $ : num [1:2500] 52 82 82 82 82 82 82 52 52 52 ...
```

15. We can then tabulate area of each vegetation category within each polygon by extracting vegetation within each polygon by ID then appending the results back to the extracted table by running it twice but with different names. Summarizing the vegetation characteristics in each cell will be used in future resource selection analysis or disease epidemiology.

```
area = extract(bbclip, gridspdf)
combine = lapply(area, table)
combine[[1]]  #Shows vegetation categories and numbers of cells in grid #1
```

```
##
```

```
##   21   42   52   81   82
##   86   75  923    1 1415
```

```
combine[[27]]
```

```
##
##   41   42   52
##  183  762 1555
```