**LAB 7**

**Using Geospatial Modeling Environment (GME) and R to Estimate Home Range Size in ArcMap 10.1**

**Purpose:**

* Learn how to estimate home range size of a wildlife species using R, GME, and ArcMap
* Learn about smoothing parameters (aka bandwidth estimates) in R and GME and how they relate to fixed-kernel home range estimates
* Learn how bandwidth estimates can influence animal home range size
* Learn about the effects of autocorrelation in wildlife data

**Introduction:**

**What is Home Range?**

The home range of an animal is broadly defined as the geographic area to which an organism confines its daily activities. These data are typically obtained from telemetry or Geographic Position System (GPS) locations of a particular species of interest. There are several ways to estimate the home range of an animal including minimum convex polygon and kernel-density estimation.

**Minimum Convex Polygon**

Probably the most commonly and widely used method is the minimum convex polygon (MCP). This method basically entails drawing a boundary polygon around all gathered (used) locations. While MCP home range estimates are useful for comparison purposes to past studies, they do not accurately represent where an animal is spending most of its time. In addition MCPs typically overestimate the home range size of the animal. Below is an example of a MCP home range from the US Geological Survey (USGS) website (Figure 1).

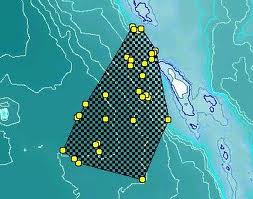


Figure 1. Home range of an unknown species

The data that we will be examining are barred owl (*Strix varia*) locations. These data are clustered around a central location (in wildlife we refer to owls as being “central place” foragers), typically a nest or a highly productive area, such as a stream. So for this lab we will be focusing on what is called a 95% utilization distribution (UD), or a raster file that shows where the owl has spent most of its time foraging (where the owl can likely be found 95% of the time). We will also generate a vector file that outlines the UD and a vector file that outlines a 50% UD (core use area or where the owl is likely to be found 50% of the time). These are also referred to as 50% or 95% kernel home ranges. Different percentages can be used for home ranges, but the most commonly used are the 50% and 95% values in wildlife studies. I recommend reading the attached Marzluff et al. (2004) to learn more about how UDs relate to wildlife species.

**Kernel-Density Estimation**

The kernel method provides an estimation for a probability density function that corresponds to an animal’s UD. To put it simply, the UD measures the intensity of use within an animal’s home range based on observed locations. However, the kernel method is very sensitive to the choice of smoothing parameters (aka bandwidth values).

Smoothing parameters or the bandwidth essentially determine how large or small the home range will be in relation to the actual locations used by the animal. For example, large bandwidth values will produce an overly smoothed UD that will overestimate the home range and vice versa.

**Methods:**

We will first play around with most commonly used type of home range estimator, the 100% minimum convex polygon (MCP) in GME and ArcMap. We will then explore fixed-kernel home range analysis using program R to determine our bandwidths/smoothing parameters. In R we will be generating two different plug-in values, one using the diagonal plug-in method and the other using the more commonly used Least Squares Cross Validation (LSCV) method to obtain two separate sets of bandwidths. We will then generate two different home range estimates from the same data set from these bandwidths. For more information on these types of bandwidth estimators, please read the provided Gitzen et al. (2006) paper.

I provided you with both a Comma Separated Value (CSV) file of the telemetry locations and a Shapefile of the locations (named “Barred\_Owl”), all referenced in North American Datum, 1983, Universal Transverse Mercator Zone 10 North. All owl locations were gathered no less than 24 hours apart to eliminate autocorrelation between points (i.e. all locations are considered separate of one another).

I have also provided you with a stream layer from Esri (clipped by perennial streams) and a current Redwood National and State Parks boundary layer (both for display purposes, if needed).

**1. Installing GME**

GME is available from the Spatial Ecology.com web site (<http://www.spatialecology.com/>). While you can use GME without ArcGIS, it does require R to be installed. Make sure you follow all the installation instructions on the “Download” web page.

Note: You may receive some errors during this process, but forge ahead and see if GME will run!

Once GME has installed, you may notice that it does not appear on the desktop. It’s in the “Start” menu under the “Spatial Ecology” folder. You may want to drag a short cut from this folder to your desktop or task menu so you can get to it later.

**2. Generating a 100% MCP home range in GME**

2.1 Open GME and in the search window type, “MCP” and click on “genmcp.”

2.2 On the “in” box navigate to where you saved the “Barred\_Owl” shapefile, and then type in where you want your mcp to be saved on the “out” box

2.3 Examine the new shape file along with the barred owl locations in ArcMap (there will be questions about this at the end of the lab).

**3. Generating kernel home ranges using Program R and GME**

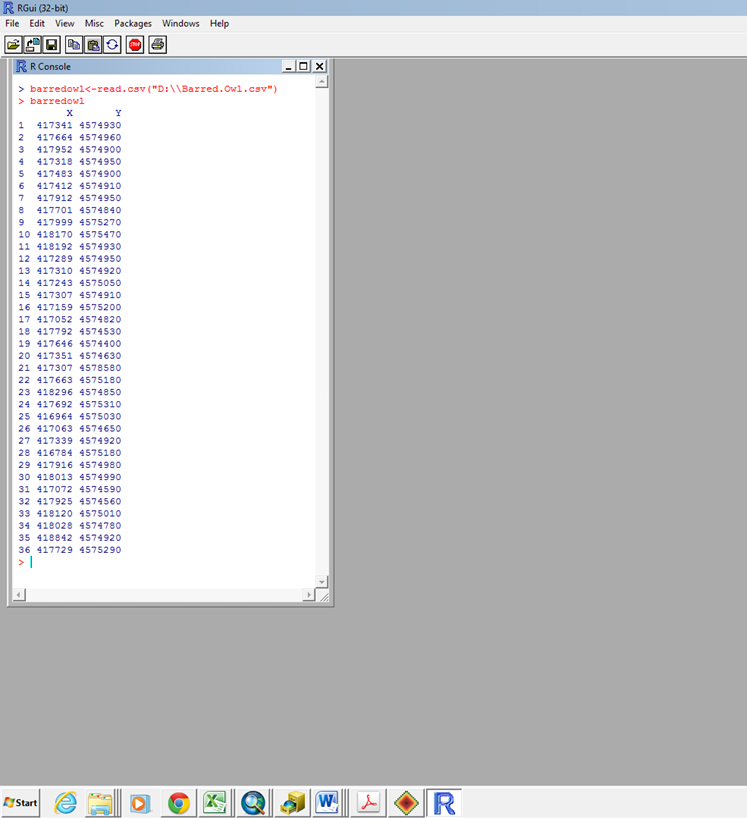
3.1 First you will have to get your bandwidth estimates using R

3.1.1 Open Program R and import the Barred.Owl.csv.

BarredOwl=read.csv(“D:/Folder/Barred\_Owl.csv”)

Note: Be careful coping text like that above and pasting it into R as the double quotes that MS-Word uses are not the same as those in “R” (MS-Word replaces them with fancy left and right double quotes) and you may need to replace them in R for your program to work.

3.1.2 The data should look like this in R:

****

3.1.3 To do the analysis, you will need to put the data into a data matrix (in this case it will be a 2 x 36 matrix).

3.1.4 First, load the package called KernSmooth into R:

#install package (must have internet access)

install.packages(pkgs=c("ks"), repos="http://cran.r-project.org")

#load required package

library (ks)

3.1.5 Next, generate bandwidth estimates using a plug-in function

#For the diagonal plug-in bandwidths

Hpi.diag(BarredOwl)

3.1.6 The output will be a 2 x 2 matrix, where the first # (top left corner) will be for the “x variance,” the second # (bottom right corner) will be for the “y variance,” and the third and fourth #’s (bottom left and top right) will be the “x/y covariance”. Keep track of these numbers in an Excel Spreadsheet

3.2 Now, we will use GME to make the kernel density estimation (KDE).

3.2.1 Open up GME.

3.2.2 Find and select “kde” from the list.

3.2.3 Make sure the “Command Builder” is tab is selected.

3.2.4. Browse to where your Barred Owl shapefile is for the “in” value.

3.2.5 Browse to where you want to save the final kde raster and give it a name that ends in “.img” for the “IMAGINE” file format which is also compatible with ArcGIS.

3.2.6 In the “bandwidth” text field, enter the following using the values that R provided in step

c(x variance,y variance,x/y corariance)

3.2.7 Enter 10 for the cell size. This will be the pixel size of the raster that is created.

3.2.8 Click “Run” at the bottom of the tab panel.

3.2.9 If an error occurred, it may be because the “bandwidth” is not properly formatted. Try just a number like 1000 and see if you are successful. Then, try building the sequence above.

3.3 You can also run commands, and later build them into scripts in GME.

3.3.1 If you click on the “Command History” tab, you’ll see the commands you have run running.

3.3.2 You’ll need to “Right-click” on the command and then select “Copy to the Clipboard” as GME does not implement the “Edit” menu or edit keystrokes. Then, you can copy these commands to the “Command Text” tab panel or paste them into a script for GME to execute.

3.3.3 Below is the command we just executed with information on the parameters. There is also help information available under the “Help” menu in GME.

kde(in="location of your shapefile.shp",out="location where you want raster to go.kde.img", bandwidth=c(the “x” # R gave you, the “y” # R gave you, the “x/y” # R gave you), cellsize=10)

**Note**: if you make a mistake with the command, notice that the command will disappear form the “Command Text” tab. You can access the command in the “Command History” tab but you’ll need to “Right-click” on the command and then select “Copy to the Clipboard” as GME does not implement the “Edit” menu or edit keystrokes.

**Note**: Most programs will interpret a forward slash (“/”) or two backslashes (“\\”) as delimiters in file paths. GME only recognizes two backslashes and will produce an error that it cannot find the files if a single forward slash is used.

3.4. You should now have a home range estimate that you can load into ArcGIS. Make sure you remove it from ArcGIS before continuing.

3.5 Try different values for the “bandwidth” and see how this effects the home range rasters.

**4. Creating 50% and 95% kernel home range shapefiles**

We can now use GME to create shapefiles that contain 50% and 95% contour lines for our home ranges.

4.1 Select “isopleth” from the “Commands”

4.2 Select one of your KDE home range rasters for the “in”

4.3 Browse to a working folder and name your “out” shapefile something appropriate.

4.4 In the “quantiles”, enter the R sequence below:

c(0.5,0.95)

**Note**: the “quantiles=c(0.5,0.95)” refers to the 50% and 95% fixed kernel home ranges.

4.5 Examine the result in ArcGIS

4.6 You can execute this same command using the code:

isopleth(in="Where you saved your kde file.kde.img”,out="where you want your vector to go.iso.shp", quantiles=c(0.5,0.95))

4.7 Using the feature to polygon tool, convert the polylines to polygons and calculate the area of each (including the 95% and 50% home ranges).

**5. Calculating a home range estimate use Least Squares Cross Validation**

You can also compute bandwidths for a KDE using Least Squares Cross Validation (LSCV) in R.

5.1 Just execute the code below in R:

Hlscv(“name of you data matrix”)

5.2 Using the same method that we used to calculate a raster file and a vector file for the diagonal plug-in (hpi.diag) estimates, now calculate the same using LSCV bandwidths.

**Final Output**

Create a short formal-style report that briefly describes the objectives and purpose of this lab as it relates to wildlife studies. Include a table with the bandwidth parameters and a map that you think most accurately represents the home range of the owl.

**In addition, answer the following questions in the appendix:**

1. Describe the basics of home-range analysis and how the kernel method compares to the traditional MCP method for generating a home range. What would be some of the advantages to using a MCP instead of a kernel home range? What are some disadvantages?
2. What differences did you observed between the two kernel bandwidth estimators (LSCV and the diagonal plug-in) generated in R? Which of the two plug-in functions do you think most accurately represents the actual home range of the owl (there’s no right or wrong answer to this question)? I recommend looking at the telemetry locations as compared with the home range outputs.
3. Can you tell which of the two kernel estimators might tend to over smooth and which one might tend to under smooth the home range (hint: read the provided Gitzen et al. 2006 paper)?
4. How could under or over smoothing bias the data?
5. The concept of autocorrelation is important when it comes to dealing with many types of data. Autocorrelation of the barred owl data was briefly mentioned in the above methods section. Below are two images of home range estimates for North American Elk (*Cervus elaphus*), collected in the Starkey Experimental Forest and Range, OR. Data represent multiple locations collected over a single month, (a) sub-sampled at n = 25 representing typical VHF-based data obtained on a daily basis, and (b) data sub-sampled at n = 603 representing GPS data collected at high sampling frequencies. Can you tell which sample has a higher rate of autocorrelation? How and why?

