4.6 Dynamic Brownian Bridge Movement Models (dBBMM)

Manual of Applied Spatial Ecology

3/11/2022

With the wide-spread use of GPS technology to track animals in near real time, estimators of home range and movement have developed concurrently. Unlike the traditional point-based estimators (i.e., MCP, KDE with href/hplug-in) that only incorporate density of locations into home range estimation, newer estimators incorporate more data provided by GPS technology. While BBMM incorporates a temporal component and GPS error into estimates, dynamic Brownian Bridge Movement Models (dBBMM) incorporate temporal and behavioral characteristics of movement paths into estimation of home range (Kranstauber et al. 2012). Estimating a movement path over the entire trajectory of data, however, should be separated into behavorial movement patterns (i.e., resting, feeding) prior to estimating the variance of the Brownian motion. Overestimating the variance will cause an imprecision in estimation of the utilization distribution that dBBMM seeks to address (Kranstauber et al. 2012).

1. Exercise 4.6 - 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 "CatdBBMM.Rmd" and run code directly from the script

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

```r
library(adehabitatLT)
library(move)
library(rgdal)
library(adehabitatHR)
library(maptools)
library(PBSmapping)
library(stringr)
```

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

```r
utm.crs <- CRS("+proj=utm +zone=17N +ellps=WGS84")
```

6. Read in panther dataset we have used previously

```r
#Creates a Spatial Points Data Frame for 2 animals by ID
panther<-read.csv("pantherjitter.csv",header=T)
panther$CatID <- as.factor(panther$CatID)
#To run BBMM we first need to use the original dataset to calculate time between locations
panther$NewTime <- str_pad(panther$TIMEET2,4, pad= "0")
panther$NewDate <- paste(panther$DateET2,panther$NewTime)
#Used to sort data in code below for all deer
panther$DT <- as.POSIXct(strptime(panther$NewDate, format='%Y %m %d %H%M'))
#Sort Data
panther <- panther[order(panther$CatID, panther$DT),]
#TIME DIFF NECESSARY IN BBMM CODE
timediff <- diff(panther$DT)*60
```

```
# remove first entry without any difference
panther <- panther[-1,]
panther$timelag <-as.numeric(abs(timediff))

cat143<-subset(panther, panther$CatID == "143")
cat143 <- cat143[-1,] #Remove first record with wrong timelag
cat143$CatID <- droplevels(cat143$CatID)
cat143.xy <- data.frame(x=cat143$X, y=cat143$Y)
cat143.spdf <- SpatialPointsDataFrame(coords=cat143.xy, data = cat143, proj4string = utm.crs)
```

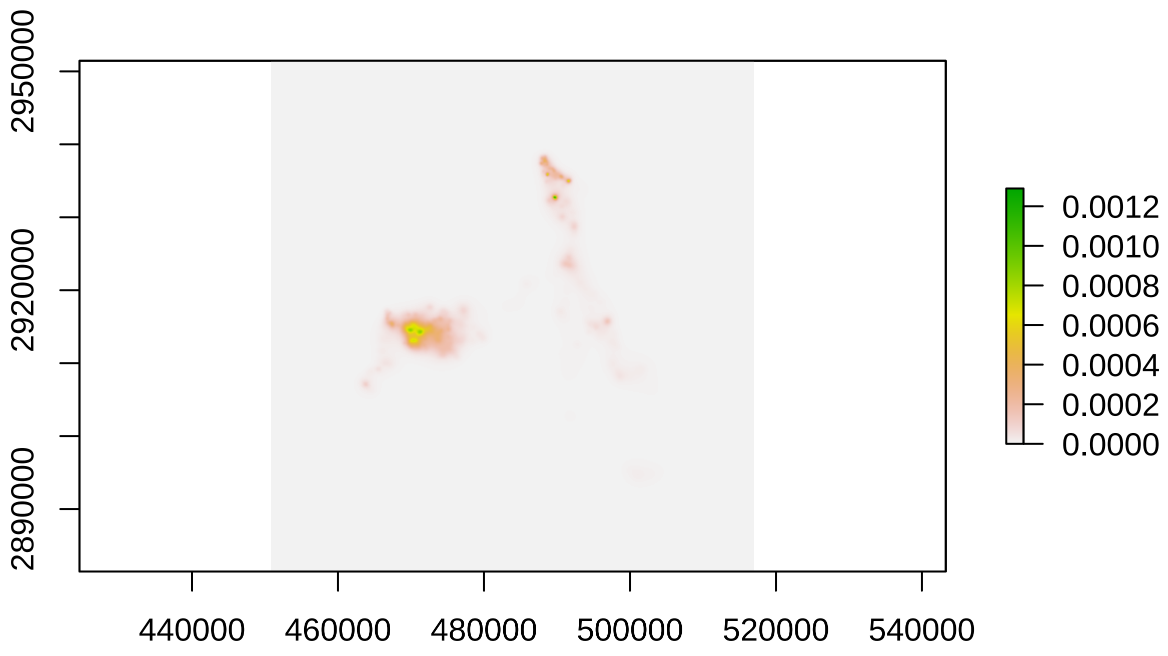
7. Create a move object for all deer using the Move package

```
loc <- move(x=cat143$X, y=cat143$Y, time=as.POSIXct(cat143$DT,
     format="%Y %m %d %H%M"), proj=utm.crs,data=cat143,
     animal=cat143$CatID)
min(timeLag(x=loc,units="mins"))
```

```
## [1] 418
```

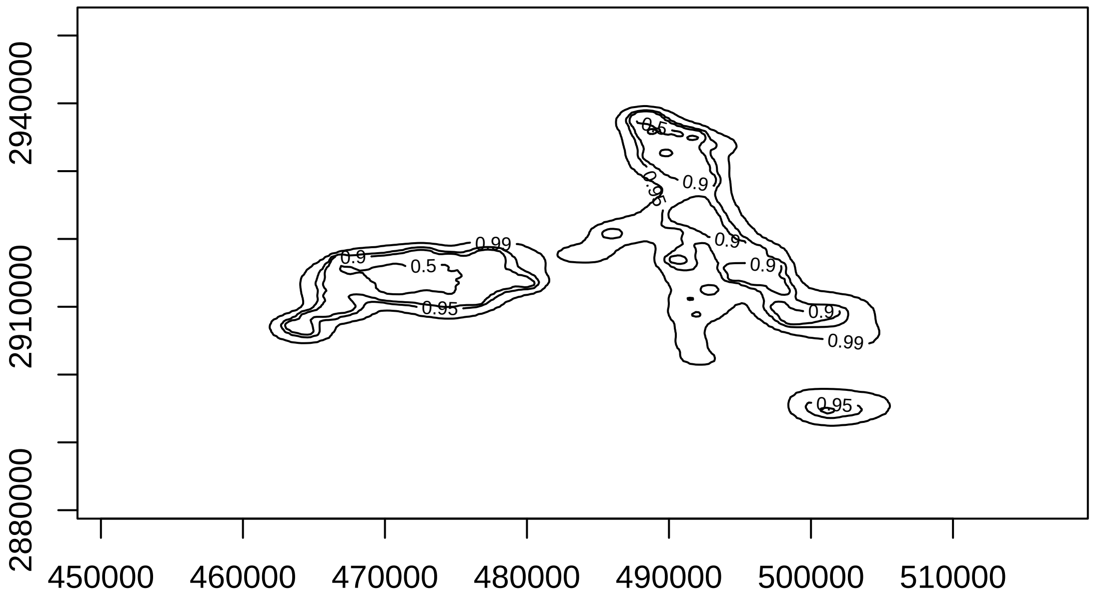
8. Now create a dBBMM object

```
cat_dbbmm <- brownian.bridge.dyn(object=loc, location.error=34, window.size=19, margin=7,
dimSize=500,time.step=180)
plot(cat_dbbmm)
```



9. We can then explore the isopleth sizes and manipulate package move output to plot isopleths like previous exercises and write them out as shapefiles if needed

```
contour(cat_dbbmm, levels=c(.5,.9,.95,.99))
```

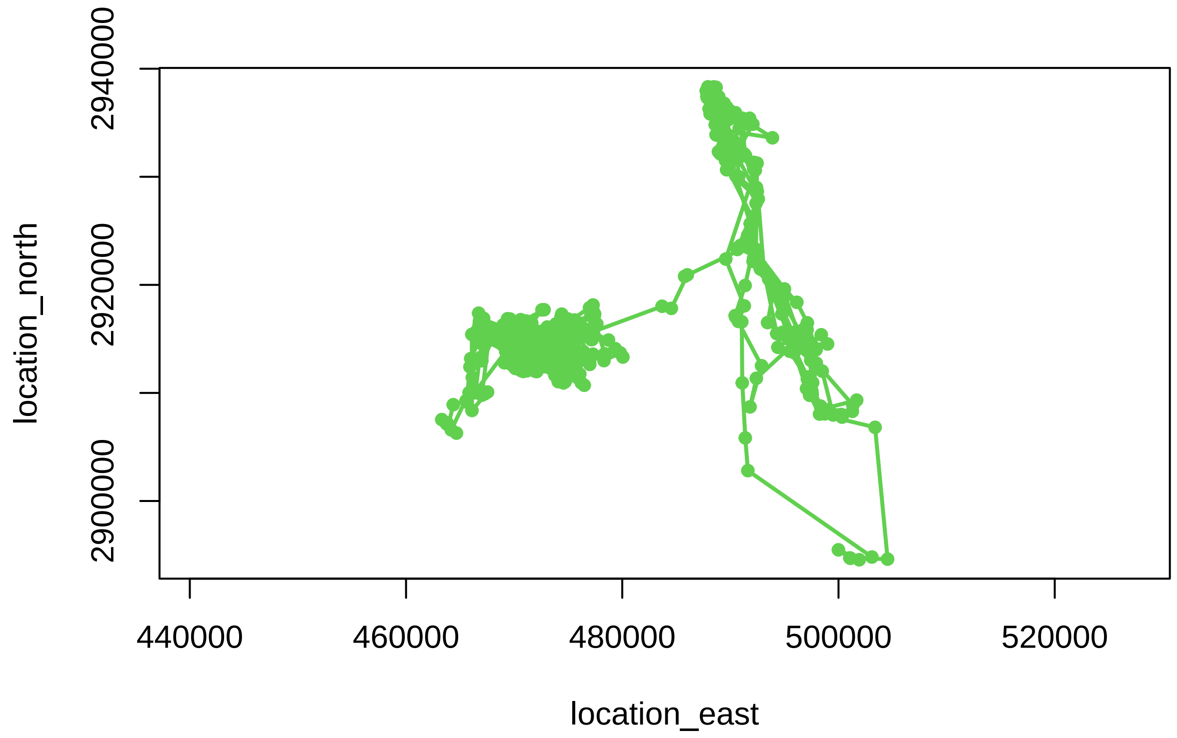


```
## NULL
```

```
show(cat_dbbmm)
```

```
## class       : DBBMM
## dimensions  : 500, 472, 236000  (nrow, ncol, ncell)
## resolution  : 140.026, 140.026  (x, y)
## extent      : 450876.3, 516968.6, 2881437, 2951450  (xmin, xmax, ymin, ymax)
## crs         : +proj=utm +zone=17 +ellps=WGS84 +units=m +no_defs
## source      : memory
## names       : layer
## values      : 0, 0.001289832  (min, max)
```

```
#Plot the movement of the animal
plot(loc, type="o", col=3, lwd=2, pch=20, xlab="location_east",ylab="location_north")
```



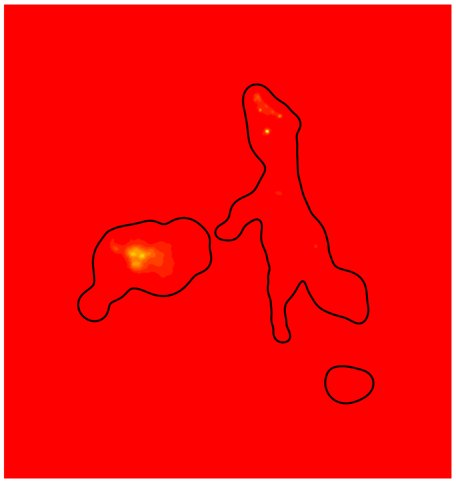
```r
#Code below will get area of each isopleth
cat_cont <- getVolumeUD(cat_dbbmm)
cat_cont50 <- cat_cont<=.50
cat_cont95 <- cat_cont<=.95
area50 <- sum(values(cat_cont50))
area50
```

```
## [1] 1447
```

```r
area95 <- sum(values(cat_cont95))
area95
```

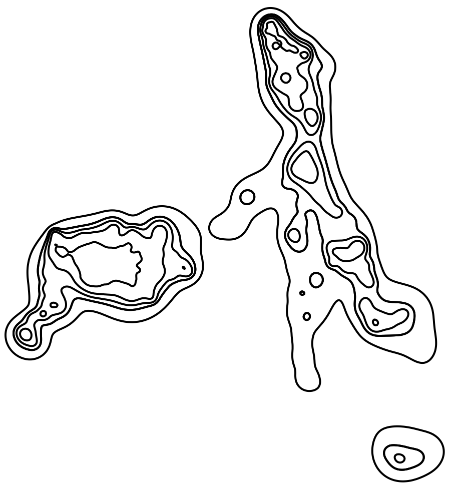
```
## [1] 14064
```

```r
##Cast the data over to an adehabitatHR estUD
dbbmm.px <- as(cat_dbbmm, "SpatialPixelsDataFrame")
image(dbbmm.px)
dbbmm.ud <- new("estUD",dbbmm.px)
dbbmm.ud@vol = FALSE
dbbmm.ud@h$meth = "dBBMM"

shp99 <- getverticeshr(dbbmm.ud, percent=99, standardize=TRUE)
plot(shp99, add=TRUE)
```



```
map.ps99 <- SpatialPolygons2PolySet(shp99)
diss.map.99 <- as.PolySet(map.ps99, projection = 'UTM', zone = '17')
diss.map.p99 <- PolySet2SpatialPolygons(diss.map.99, close_polys = TRUE)
data99 <- data.frame(PID = 1)
diss.map.p99 <- SpatialPolygonsDataFrame(diss.map.p99, data = data99)
plot(diss.map.p99)
# writeOGR(diss.map.p99, dsn = ".", layer="catcontour99", driver = "ESRI Shapefile")
# catmap.99 <- readOGR(dsn=".", layer="catcontour99")

shp95 <- getverticeshr(dbbmm.ud, percent=95, standardize=TRUE)
plot(shp95, add=TRUE)
map.ps95 <- SpatialPolygons2PolySet(shp95)
diss.map.95 <- as.PolySet(map.ps95, projection = 'UTM', zone = '17')
diss.map.p95 <- PolySet2SpatialPolygons(diss.map.95, close_polys = TRUE)
data95 <- data.frame(PID = 1)
diss.map.p95 <- SpatialPolygonsDataFrame(diss.map.p95, data = data95)
plot(diss.map.p95, add=T)
#writeOGR(diss.map.p95, dsn = ".", layer="catcontour95", driver = "ESRI Shapefile")
#catmap.95 <- readOGR(dsn=".", layer="catcontour95")

shp90 <- getverticeshr(dbbmm.ud, percent=90, standardize=TRUE)
map.ps90 <- SpatialPolygons2PolySet(shp90)
diss.map.90 <- as.PolySet(map.ps90, projection = 'UTM', zone = '17')
diss.map.p90 <- PolySet2SpatialPolygons(diss.map.90, close_polys = TRUE)
data90 <- data.frame(PID = 1)
diss.map.p90 <- SpatialPolygonsDataFrame(diss.map.p90, data = data90)
```

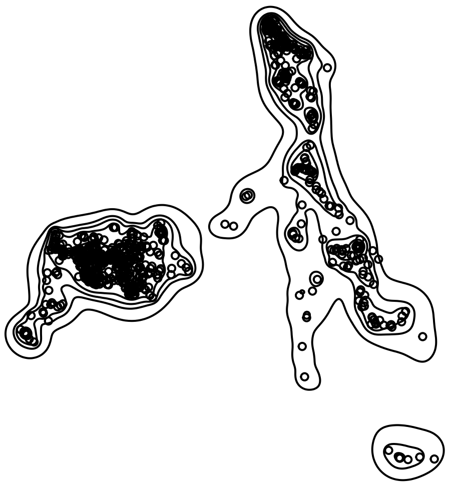
```r
plot(diss.map.p90,add=T)
# writeOGR(diss.map.p90, dsn = ".", layer="catcontour90", driver = "ESRI Shapefile")
# catmap.90 <- readOGR(dsn=".", layer="catcontour90")

shp80 <- getverticeshr(dbbmm.ud, percent=80, standardize=TRUE)
map.ps80 <- SpatialPolygons2PolySet(shp80)
diss.map.80 <- as.PolySet(map.ps80, projection = 'UTM', zone = '17')
diss.map.p80 <- PolySet2SpatialPolygons(diss.map.80, close_polys = TRUE)
data80 <- data.frame(PID = 1)
diss.map.p80 <- SpatialPolygonsDataFrame(diss.map.p80, data = data80)
plot(diss.map.p80,add=T)
# writeOGR(diss.map.p80, dsn = ".", layer="catcontour80", driver = "ESRI Shapefile")
# catmap.80 <- readOGR(dsn=".", layer="catcontour80")

shp50 <- getverticeshr(dbbmm.ud, percent=50, standardize=TRUE)
map.ps50 <- SpatialPolygons2PolySet(shp50)
diss.map.50 <- as.PolySet(map.ps50, projection = 'UTM', zone = '17')
diss.map.p50 <- PolySet2SpatialPolygons(diss.map.50, close_polys = TRUE)
data50 <- data.frame(PID = 1)
diss.map.p50 <- SpatialPolygonsDataFrame(diss.map.p50, data = data50)
plot(diss.map.p50,add=T)
```



```r
# writeOGR(diss.map.p50, dsn = ".", layer="catcontour50", driver = "ESRI Shapefile")
# catmap.50 <- readOGR(dsn=".", layer="catcontour50")
```

10. We can plot out the 50-99% isopleths to compare to previous estimators in size and shape around locations

```
plot(diss.map.p99)
plot(diss.map.p95, add=T)
plot(diss.map.p90,add=T)
plot(diss.map.p80,add=T)
plot(diss.map.p50,add=T)
points(cat143.spdf,pch=1, cex=0.5)
```



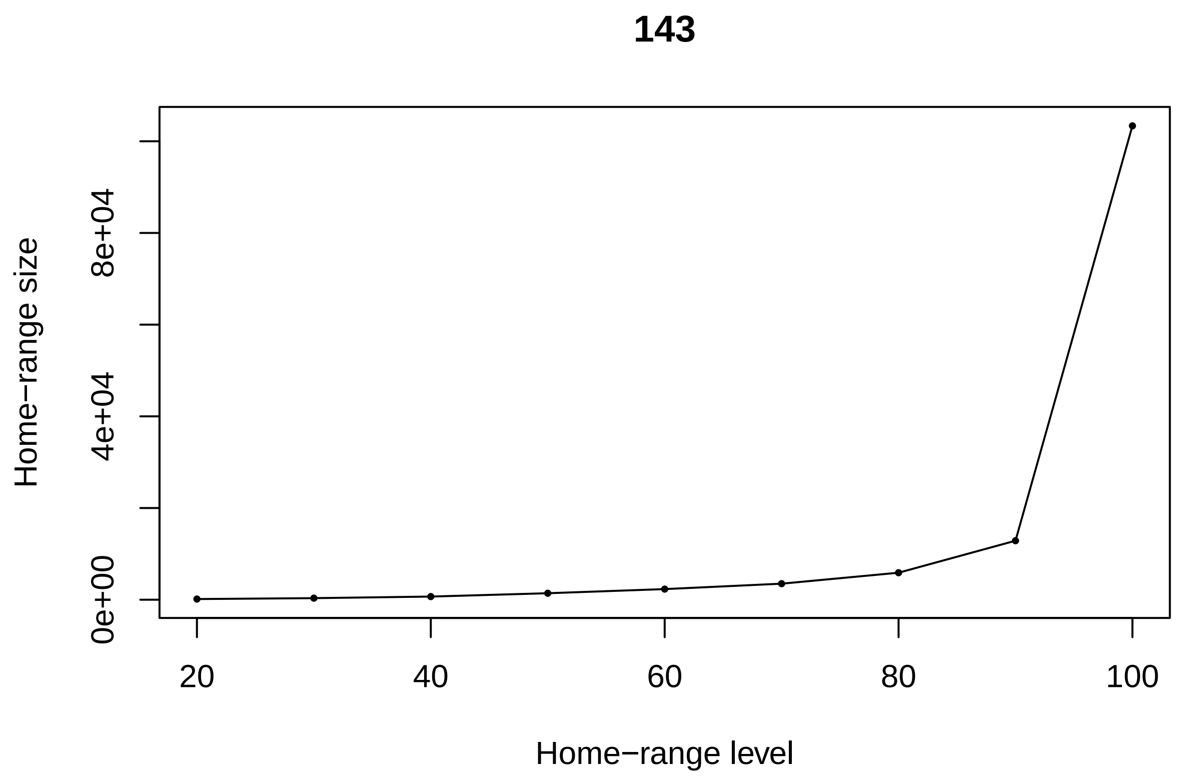
11. Now we will shift towards polgyon-based estimators of home range to compare them to dBBMM. We will start with Characteristic Hull Polygons (CHP) in adehabitatHR package using the CharHull function.

```
data.xy = cat143[c("X","Y")]

#Creates class Spatial Points for all locations
xysp <- SpatialPoints(data.xy)
proj4string(xysp) <- utm.crs

#Creates a Spatial Data Frame from
sppt<-data.frame(xysp)

#Creates a spatial data frame of ID
idsp<-data.frame(cat143[2])

#Merges ID data frame with GPS locations data frame
#Data frame is called "idsp" comparable to the "relocs" from puechabon dataset
coordinates(idsp)<-sppt

#Home Range estimation
```

```
res <- CharHull(idsp[,1])
class("res")
```

```
## [1] "character"
```

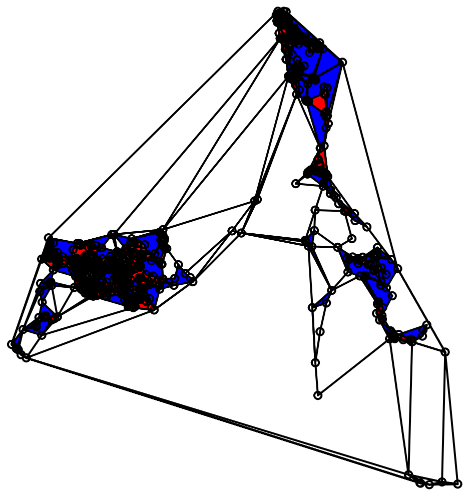
```
#Computes the home range size for 20-100 percent
MCHu2hrsize(res)
```



```
##             143
## 20     149.2545
## 30     342.9901
## 40     686.9159
## 50    1415.5709
## 60    2318.0083
## 70    3498.8652
## 80    5889.5274
## 90   12867.4591
## 100 103361.1331
```

```
#OR use

res_ver99 <- getverticeshr(res, percent=99)
res_ver95 <- getverticeshr(res, percent=95)
res_ver90 <- getverticeshr(res, percent=90)
res_ver80 <- getverticeshr(res, percent=80)
res_ver50 <- getverticeshr(res, percent=50)
```

```r
plot(res_ver99)
plot(res_ver95,add=T)
plot(res_ver90, add=TRUE, col="blue")
plot(res_ver80, add=TRUE, col="red")
plot(res_ver50, add=T, col="green")
points(cat143.spdf,pch=1, cex=0.5)
```

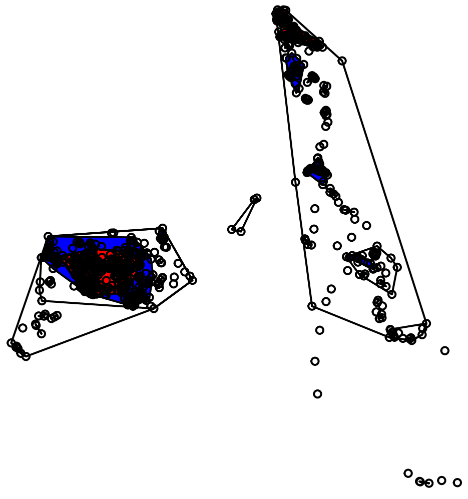


11. Next we will estimate home range with the Single-linkage Cluster (SLCA) using the clusthr function

```r
uu <- clusthr(idsp)
class(uu)
```

```
## [1] "MCHu"
```

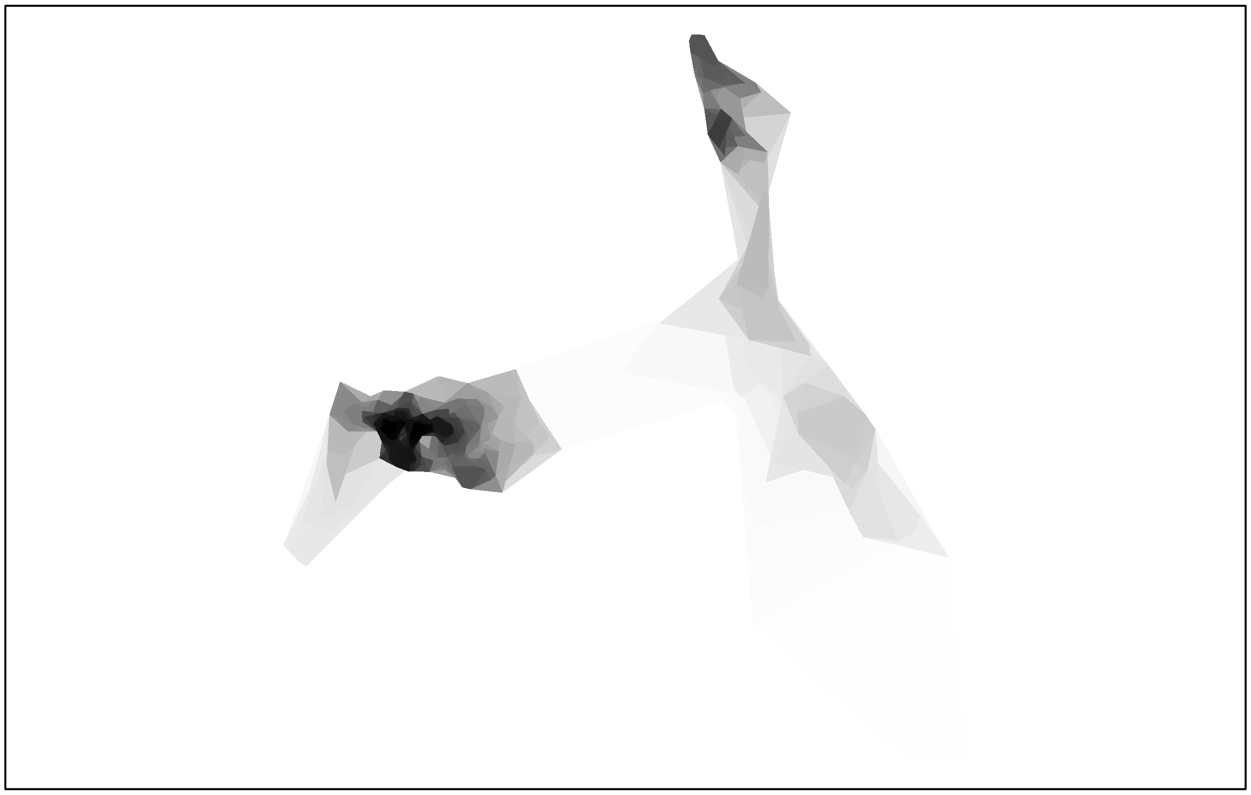
```r
uu_ver99 <- getverticeshr(uu, percent=99)
uu_ver95 <- getverticeshr(uu, percent=95)
uu_ver90 <- getverticeshr(uu, percent=90)
uu_ver80 <- getverticeshr(uu, percent=80)
uu_ver50 <- getverticeshr(uu, percent=50)

plot(uu_ver99)
plot(uu_ver95,add=T)
plot(uu_ver90, add=TRUE, col="blue")
plot(uu_ver80, add=TRUE, col="red")
plot(uu_ver50, add=T, col="green")
points(cat143.spdf,pch=1, cex=0.5)
```



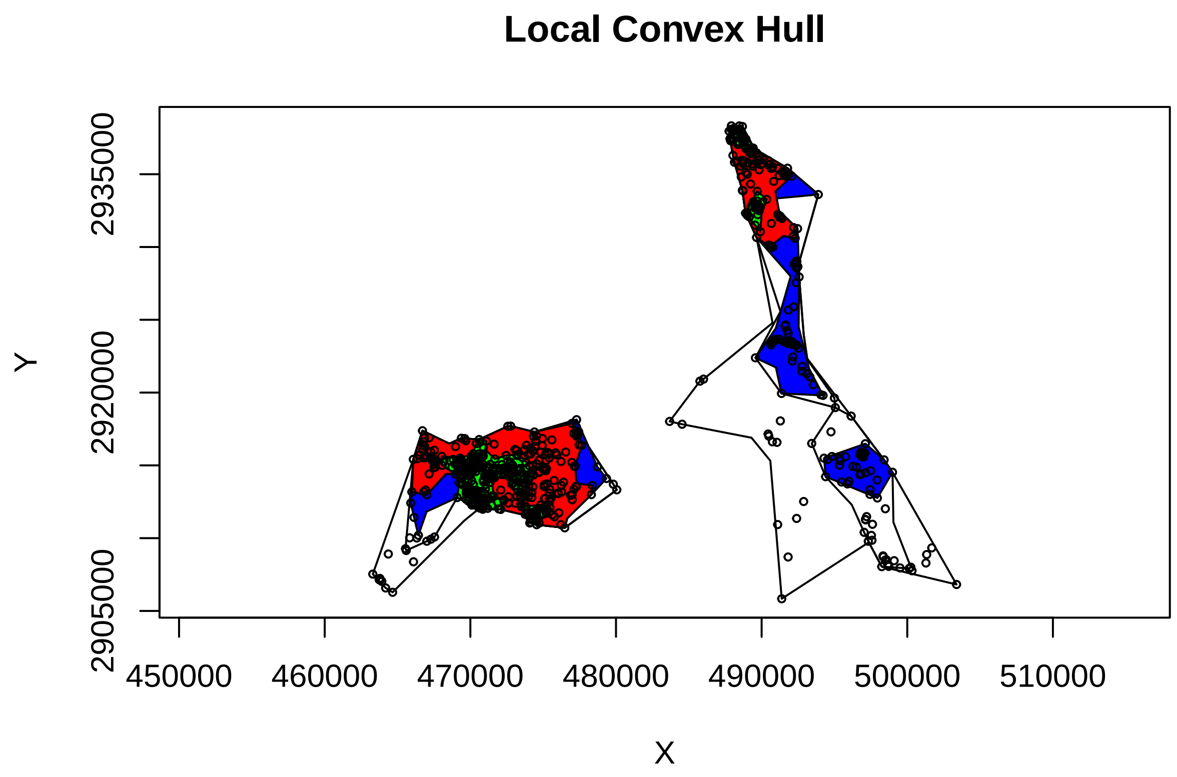
12./ Next we will explore Local Convex Hull (LoCoH)

```
## Exams the changes in home-range size for various values of k
## Be patient! the algorithm can be very long
#LoC.area <- LoCoH.k.area(idsp, k=c(5:40))
#NOTE: The line of code above does not run for this animal

## the k-LoCoH method:
nn <- LoCoH.k(idsp[,1], k=30)
## Graphical display of the results
plot(nn, border=NA)
```

143



```
## the object nn is a list of objects of class

#Save shapefiles of resulting home range
ver <- getverticeshr(nn)

#writeOGR(ver,dsn="FixedK",layer="FixedK24", driver = "ESRI Shapefile", overwrite=TRUE)
##Overwrite will not work so must edit path so "FixedK" folder is created with code below.
nn_ver50 <-getverticeshr(nn, percent=50)
#writeOGR(ver50,dsn="FixedK",layer="50FixedK24", driver = "ESRI Shapefile",overwrite=TRUE)
nn_ver80 <-getverticeshr(nn, percent=80)
#writeOGR(ver80,dsn="FixedK",layer="80FixedK24", driver = "ESRI Shapefile",overwrite=TRUE)
nn_ver90 <-getverticeshr(nn, percent=90)
#writeOGR(ver90,dsn="FixedK",layer="90FixedK24", driver = "ESRI Shapefile",overwrite=TRUE)
nn_ver95 <-getverticeshr(nn, percent=95)
#writeOGR(ver95,dsn="FixedK",layer="95FixedK24", driver = "ESRI Shapefile",overwrite=TRUE)
nn_ver99 <-getverticeshr(nn, percent=99)
#writeOGR(ver99,dsn="FixedK",layer="99FixedK24", driver = "ESRI Shapefile",overwrite=TRUE)

plot(nn_ver99,main="Local Convex Hull",xlab="X", ylab="Y", font=1, cex=0.8, axes=T)
plot(nn_ver95,add=T)
plot(nn_ver90, add=TRUE, col="blue")
plot(nn_ver80, add=TRUE, col="red")
plot(nn_ver50, add=T, col="green")
points(cat143.spdf,pch=1, cex=0.5)
```



13. We can add 4 estimators to the plot window to compare across estimators

```
par(mfrow=c(2,2))

plot(diss.map.p99,main="dynamic BBMM",xlab="X", ylab="Y", font=1, cex=0.8, axes=T)
plot(diss.map.p95, add=T)
plot(diss.map.p90,add=T)
plot(diss.map.p80,add=T)
plot(diss.map.p50,add=T)
points(cat143.spdf,pch=1, cex=0.5)

plot(res_ver99,main="Characteristic Hull Polygons",xlab="X", ylab="Y", font=1, cex=0.8, axes=T)
plot(res_ver95,add=T)
plot(res_ver90, add=TRUE, col="blue")
plot(res_ver80, add=TRUE, col="red")
plot(res_ver50, add=T, col="green")
points(loc, pch=1, cex=0.5)
points(cat143.spdf,pch=1, cex=0.5)

plot(uu_ver99,main="Single-linkage Cluster",xlab="X", ylab="Y", font=1, cex=0.8, axes=T)
plot(uu_ver95,add=T)
plot(uu_ver90, add=TRUE, col="blue")
plot(uu_ver80, add=TRUE, col="red")
plot(uu_ver50, add=T, col="green")
points(cat143.spdf,pch=1, cex=0.5)

plot(nn_ver99,main="Local Convex Hull",xlab="X", ylab="Y", font=1, cex=0.8, axes=T)
```

```r
plot(nn_ver95,add=T)
plot(nn_ver90, add=TRUE, col="blue")
plot(nn_ver80, add=TRUE, col="red")
plot(nn_ver50, add=T, col="green")
points(cat143.spdf,pch=1, cex=0.5)
```