

4.2 Kernel Density Estimation (KDE) with least squares cross validation (lscv)

Manual of Applied Spatial Ecology

3/11/2022

1. Exercise 4.2 - 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 “HrefScript.Rmd” and run code directly from the script
4. First we need to load the packages needed for the exercise

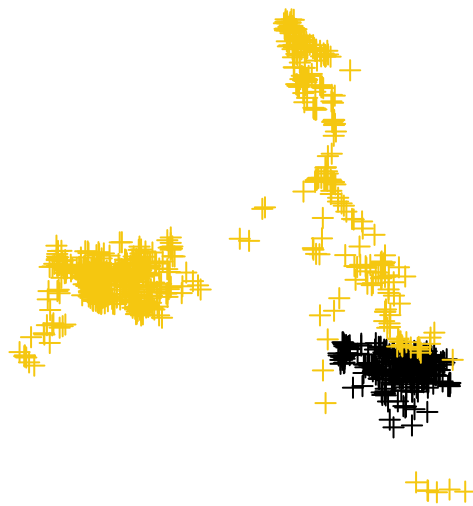
```
library(adehabitatHR)
```

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

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

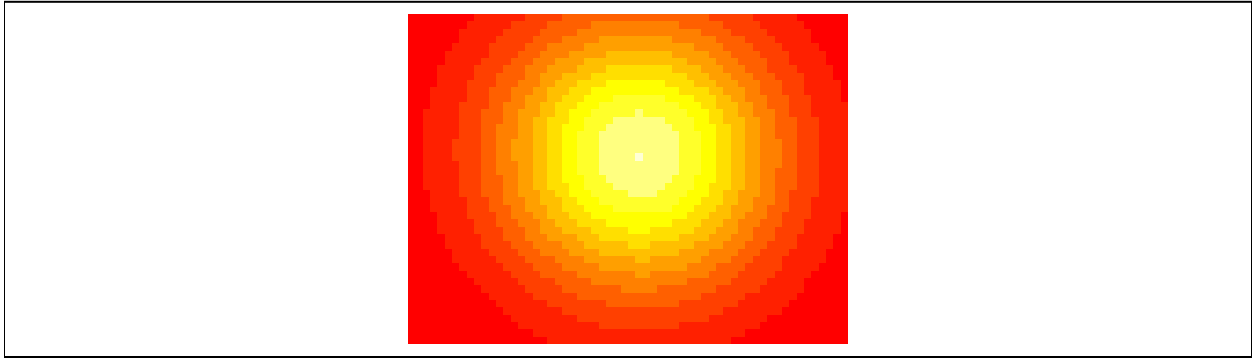
6. Now we can run fixed kernel home range with hlscv

```
panther <- read.csv("pantherjitter.csv", header=T)
loc <- data.frame("x"=panther$X, "y"=panther$Y)
pantherspdf <- SpatialPointsDataFrame(loc, panther, proj4string = utm.crs)
plot(pantherspdf, col=pantherspdf$CatID)
```

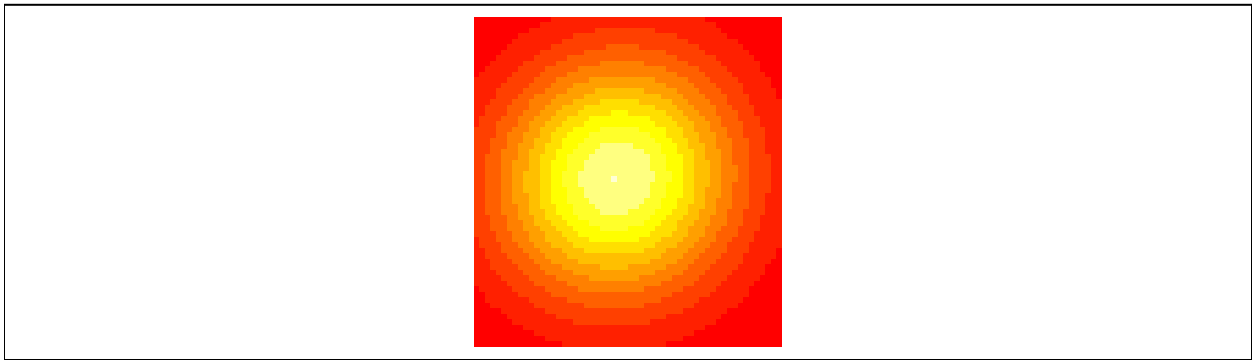


```
## Example of estimation using LSCV
udbis2 <- kernelUD(pantherspdf[,2], h = "LSCV", hlim = c(10,50),extent=1)
image(udbis2)
```

121

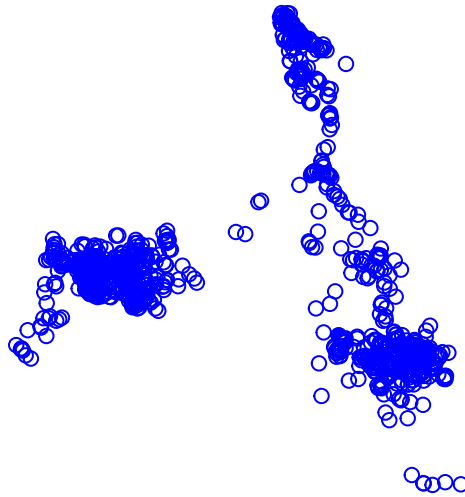


143



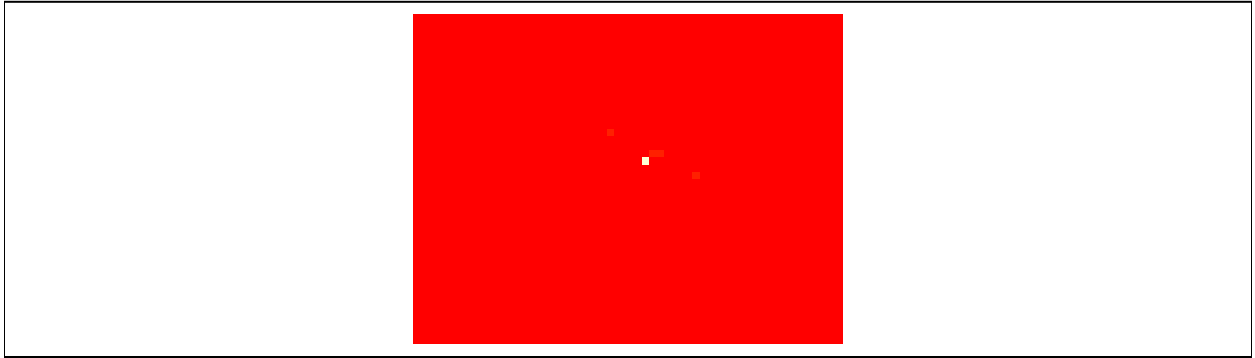
7. Note that regardless of change `hlim` or extent, LSCV will not converge for these animals so we can try a trick here. I believe LSCV is a poor estimator with GPS locations being too numerous and very close together compared to traditional VHF datasets which LSCV were originally evaluated. So we will jitter locations 50 meters from their original location and try again.

```
panther$jitterX <- jitter(panther$X, factor=50)
panther$jitterY <- jitter(panther$Y, factor=50)
locjitter <- data.frame("x"=panther$jitterX,"y"=panther$jitterY)
proj4string <- CRS("+proj=utm +zone=17N +ellps=WGS84")
jitterspdf <- SpatialPointsDataFrame(locjitter,panther, proj4string = proj4string)
plot(jitterspdf, col=pantherspdf$id)
points(pantherspdf, col="blue")
```

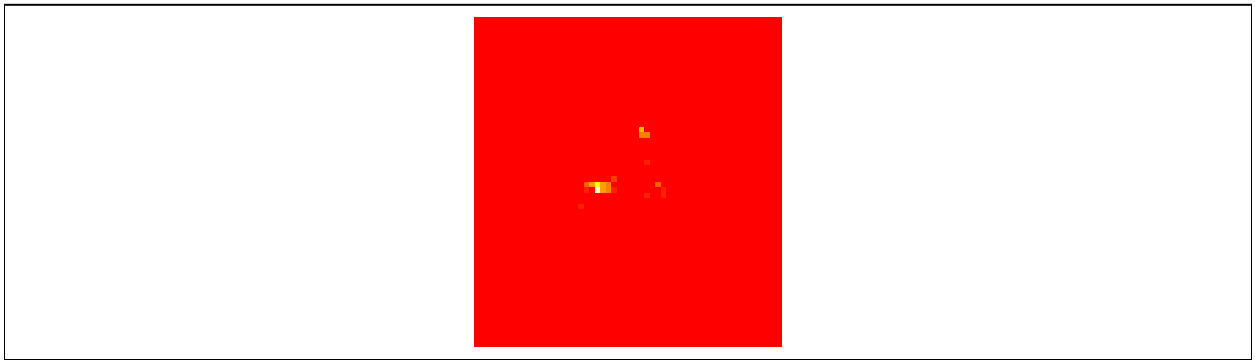


```
udbis3 <- kernelUD(jitterspdf[,2], h = "LSCV")#, hlim = c(1, 5), extent=1)
image(udbis3)
```

121



143



7. Now rerun with jitter factor = 100 instead of 50 and see what happens? Then rerun with jitter factor = 500 instead of 100 and see what happens?