3.7 Movement Trajectory Animation

Manual of Applied Spatial Ecology

2/25/2021

1. Exercise 3.7 - 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 "TrajDynScript.Rmd" and run code directly from the script
4. First we need to load the packages needed for the exercise

```
library(adehabitatLT)
library(chron)
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

```
utm.crs <- "+proj=utm +zone=12 +ellps=WGS84"
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")
```

```
## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO", prefer_proj
## = prefer_proj): Discarded datum Unknown based on GRS80 ellipsoid in Proj4
## definition
```

6. We will be using the mule deer dataset for this exercise

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

#CODE FOR AN INDIVIDUAL ANIMAL
muley16 <- subset(muleys, id=="D16")
muley16$id <- factor(muley16$id)
summary <- table(muley16$UTM_Zone,muley16$id)

#Sort data to address error in code and then look at first 10 records of data to confirm
muley16 <- muley16[order(muley16$GPSFixTime),]

######################################################
## Example of a trajectory of type II (time recorded)
### Conversion of the date to the format POSIX
#Needs to be done to get proper digits of date into R then POSIXct
#uses library(chron)
da <- as.character(muley16$GPSFixTime)
da <- as.POSIXct(strptime(muley16$GPSFixTime,format="%Y.%m.%d %H:%M:%S"))
#Attach da to muley15
muley16$da <- da

timediff <- diff(muley16$da)
```

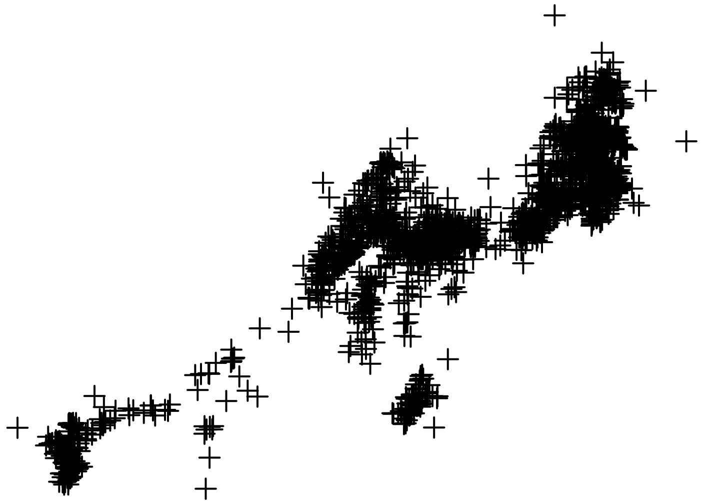
```r
muley16 <-muley16[-1,]
muley16$timediff <-as.numeric(abs(timediff))

#Clean up muley15 for outliers
newmuleys <-subset(muley16, muley16$X > 599000 & muley16$X < 705000 & muley16$Y > 4167000
    & muley16$timediff < 14401)
muley16 <- newmuleys

data.xy = muley16[c("X","Y")]
#Creates class Spatial Points for all locations
xysp <- SpatialPoints(data.xy)
proj4string(xysp) <- CRS("+proj=utm +zone=12 +ellps=WGS84")
```

```
## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO", prefer_proj
## = prefer_proj): Discarded datum Unknown based on WGS84 ellipsoid in Proj4
## definition
```

```r
#Creates a Spatial Data Frame from
sppt<-data.frame(xysp)
#Creates a spatial data frame of ID
idsp<-data.frame(muley16[2])
#Creates a spatial data frame of dt
dtsp<-data.frame(muley16[24])
#Creates a spatial data frame of Burst
busp<-data.frame(muley16[23])
#Merges ID and Date into the same spatial data frame
merge<-data.frame(idsp,dtsp,busp)
#Adds ID and Date data frame with locations data frame
coordinates(merge)<-sppt
plot(merge)
```



```
#Give dataset projection information then project to Albers
proj4string(merge) <- CRS(utm.crs)
deer.albers <-spTransform(merge, CRS=Albers.crs)
```

7. Need to create a movement trajectory as we did in previous exercises

```
ltr.albers <- as.ltraj(coordinates(deer.albers),merge$da,id=merge$id)
```

8. Now let's have a little fun with these mule deer locations and explore

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

9. Code below is used to just zoom in on all of our locations and crop within it so just select a study area around your locations using the drawExtent function below

```
plot(deer.albers)
e <- drawExtent()#click on top left of crop box and bottom right of crop box to create
#a polygon around all locations
newclip <- crop(veg,e)
plot(newclip)
points(deer.albers, col="red")
vegspdf <- as(newclip,"SpatialPixelsDataFrame")
plot(ltr.albers, spixdf=vegspdf)
```

10. #Or zoom in even closer on a few areas by repeating the drawExtent function above to a specific area

```
e2 <- drawExtent()
newclip2 <- crop(newclip,e2)
```

```
plot(newclip2)
points(deer.albers)
zoomspdf <- as(newclip2,"SpatialPixelsDataFrame")
zoom.ltr <- crop(deer.albers,zoomspdf)
ltr.zoom <- as.ltraj(coordinates(zoom.ltr),zoom.ltr$da,id=zoom.ltr$id)
plot(ltr.zoom, spixdf=zoomspdf)
```

11. We are first going to randomly select one location day for the calendar year our deer is monitored. This will result in fewer locations to plot overall. Then create an ltraj of the subset locations or all locations if you skipped lines 112-120 below

```
deer.albers$Year <- format(deer.albers$da, "%Y")
deer.albers <- subset(deer.albers,deer.albers$Year != "NA")
deer.albers$YearBurst <- c(paste(deer.albers$id,deer.albers$Year,sep="_"))
deer.albers$YearBurst <- as.factor(deer.albers$YearBurst)
range(deer.albers$da)

deer.albers$subDate <-  as.POSIXct(as.factor(deer.albers$da), format="%Y-%m-%d", tz="EST")
deer.albers$Oneperday <- paste(deer.albers$YearBurst,deer.albers$subDate,sep="_")
deer.albers2 <- do.call(rbind, lapply(split(deer.albers,deer.albers$Oneperday) ,
function(deer.albers) deer.albers[sample(nrow(deer.albers), 1) , ] ))
ltr.year <- as.ltraj(coordinates(deer.albers2),deer.albers2$da,id=deer.albers2$id)
```

12. Now we can use the function to create movements of our deer over the landscape

```
windows() #NOTE: a new window is needed in Rstudio
#Line of code below plots trajectory one location at a time
trajdyn(ltr.year,spixdf=vegspdf)
```