Rafael dos Santos Henrique

Mestrando em Zoologia pelo Laboratório de Anfíbios - Instituto de Biociências - USP

Projeto: Estudo do movimento da Rã-Touro (*Lithobates catesbeianus*) em ambientes invadidos no Brasil através do uso de radiotelemetria.

Orientador: Prof. Dr. Taran Grant

rafa.henrique.biologia@gmail.com

---

exec

Trabalho Final

Proposta A

Durante meu mestrado, trabalharei com dados resultantes do rastreamento de animais na natureza. Serão coletadas coordenadas geográficas de cada indivíduo de acordo com o seu movimento. Essas coordenadas deverão ser analisadas e para isso, anteriormente transformadas. Um tipo de informação a ser extraída desses dados é a distância (em unidade métrica) entre dois pontos amostrados. Assim, a função terá como objetivo calcular a distância entre duas coordenadas geográficas, ou seja, a distância entre uma observação de um indivíduo até o próximo ponto no qual o mesmo foi encontro, representando assim, o seu movimento entre os pontos. Para isso, o usuário deverá entrar com um data frame ou tabela contendo as coordenadas latitudinais na primeira coluna e as longitudinais na segunda coluna. Nas linhas, estarão os subsequentes encontros com o indivíduo. O usuário poderá especificar se entrar com coordenadas do tipo UTM ou Graus, Minutos e Segundos. Ainda, haverá a opção de se extrair as distâncias mínima, máxima, média e a distância total entre o primeiro e último encontro.

Proposta A Corrigida

Durante meu mestrado, trabalharei com dados resultantes do rastreamento de animais na natureza. Serão coletadas coordenadas geográficas de cada indivíduo de acordo com o seu movimento. Essas coordenadas deverão ser analisadas e para isso, anteriormente transformadas. Um tipo de informação a ser extraída desses dados é a distância (em unidade métrica) entre dois pontos amostrados. Assim, a função terá como objetivo calcular a distância entre duas coordenadas geográficas, ou seja, a distância entre uma observação de um indivíduo até o próximo ponto no qual o mesmo foi encontro, representando assim, o seu movimento entre os pontos. Para isso, o usuário deverá entrar com um data frame ou tabela contendo as coordenadas latitudinais na primeira coluna e as longitudinais na segunda coluna. Nas linhas, estarão os subsequentes encontros com o indivíduo.

O usuário poderá especificar se entrar com coordenadas do tipo UTM ou Graus, Minutos e Segundos. Ainda, haverá a opção de se extrair as distâncias mínima, máxima, média e a distância total entre o primeiro e último encontro. Como saída, a função nos dará os valores das distâncias entre os pontos, as distâncias mínimas e máximas e a distância total percorrida pelo animal.

Proposta B

Alguns animais apresentam a tendência de restringir sua atividade para uma área específica e então defender essa área contra indivíduos de sua espécie. A esse comportamento damos o nome de territorialidade (Emlen, 1968). Sabendo que determinada espécie apresenta comportamento de territorialidade, e conhecendo sua extensão, é possível calcular quantos indivíduos provavelmente serão encontrados em uma determinada área. Por exemplo, machos de *Lithobates catesbeianus*, a espécie de estudo do meu mestrado, apresenta uma média de raio de territorialidade de 2,6 metros em períodos reprodutivos (Currie & Bellis, 1969), assim, a área de defesa de cada macho de *L. catesbeianus* em um sítio de reprodução é de aproximadamente 21,2 m^2 (Área = pi.r^2). Em um corpo d'água hipotético de 2 mil metros quadrados, teríamos então, cerca de 94 machos defendendo os seus territórios. Com isso, a função a ser desenvolvida terá como objetivo calcular a quantidade de indivíduos, que apresentam comportamento de territorialidade, encontrada em uma determinada área. Para isso, deverá ser informado para a função, um vetor contendo a área defendida por cada indivíduo e o tamanho da área de estudo e o output seria o provável número de indivíduos encontrados naquela área.

Currie, W. & Bellis, E. D. 1969. Home Range and Movements of the Bullfrog, Rana catesbeiana Shaw, in an Ontario Pond. *Copeia* 1969(4): 688-692.

Emlen, S. T. 1968. Territoriality in the Bullfrog, Rana catesbeiana. *Copeia* 1968(2): 240-243.

Comentários

Rafael, achei a sua proposta A interessante e acho que vai te ajudar bastante na sua pesquisa. Acredito que calcular as distâncias seja algo tranquilo, mas talvez você possa incluir aspectos do terreno, como altitude, para tornar a estimativa mais precisa. A Joana Araujo, por exemplo, propôs algo muito parecido. Se sobrar tempo, você também pode plotar as distâncias em um mapa (não sei qual o grau de dificuldade). Não sei quais funções utilizar para isso, mas fazendo uma pesquisa você pode encontrar boas opções. Você precisa especificar qual será o objeto de saída da sua função!

Achei a sua proposta B simples demais. É basicamente uma conta de divisão. Você poderia elaborar mais esta proposta, mas eu sugiro que você foque na proposta A.

— *Lucas Medeiros*

Função track.distances (Proposta A)

```
########## TRACK.DISTANCES ##########

#Função que calcula a distância entre duas coordenadas geográficas amostras
#Os objetos de entrada são: 1 vetor contendo as longitudes e outro contendo
as latitudes
#As coordenadas podem ser inseridas no sistema UTM ou em Graus Decimais
#Os objetos de saída são: as distâncias entre os pontos, distância mínima,
máxima, média e total além de um gráfico evidenciando os pontos conectados

#Função: x(coordendas longitudinais), y(coordenadas latitudinais), type=(se
"UTM" ou Graus("G"))
track.distances<- function(x,y,type)
{
    ##########MENSAGENS DE AVISO##########
    if(length(x)>length(y))#Se o número de valores de x for maior que os de
y.
    {
        cat("Há mais valores longitudinais que valores
latitudinais")#fornece a mensagem e trava a função
    }
    if(length(x)<length(y))#Se o número de valores de y for maior que os de
x
    {
        cat("Há mais valores latitudinais que valores
longitudinais")#fornece a mensagem e trava a função
    }
    ##########FIM MENSAGEM##########
    n<-length(x)#salva o número de valores de x no objeto n
    distancias = rep(NA, n-1)#cria um vetor(distancias) com o número de
valores igual ao número de valores de x-1(n-1), preenchidos por NA.

    ##########UTM##########
    if(type=="UTM")#se type for igual a UTM(coordenas em UTM)
    {
        ##########GRÁFICO UTM##########
        par(pch=16, font.lab=2)#Função que estabele padrões gráficas.
        #Argumento pch=16, estabelece que o símbolo a ser plotado será o de
número 16(circulo preenchido).
        #Argumento font.lab=2, estabelece que a fonte a ser usada nos nomes
dos eixos x e y será a de número 2.
        plot(y~x, type="o", xlab="Longitude", ylab="Latitude",
main="Coordenadas UTM")#Plota um gráfico de pontos com as coordenadas
latitudinais(y) em função(~) das longitudinais(x).
        #Argumento type="o", estabelece que os pontos plotados serão
conectados por linhas e essas passarão por cima dos pontos ("o").
        #Argumentos xlab e ylab, estabele que os nomes dos eixos de x e y
serão os fornecidos entre parênteses.
        #Argumento main, estabelece que o título do gráfico será o fornecido
```

```
entre parênteses
        text(x, y, labels=1:length(x), pos=3)#função que adiciona textos ao
longo de um gráfico, dadas as coordenadas x e y
        #Argumento labels, vetor que especifica que o texto a ser escrito
vai do número 1 ao número de valores de x
        #Argumento pos=3, estabelece que a posição do texto será acima do
ponto estabelecido por (x,y)
        ##########FIM GRÁFICO UTM##########
        ##########LOOP UTM##########
        for(i in 1:n-1)#Loop com contador i, que vai de 1 à n-1
        {
            for(j in (i+1):n)#Loop com contador j, que vai de i+1 até n
            {
                distancias[i]<- sqrt(((y[i]-x[i])^2)+((y[i+1]-
x[i+1])^2))#fórmula que calcula as distâncias entre os pontos e salva esses
valores dentro do objeto distancias, que foi previamente criado e indexado
com o contador i. Assim, a fórmula rodará(loop) i vezes e os loops são
indexados com j, que faz com que a cada novo loop é adicionado (+1) aos
valores das posições dos vetores x e y. Com isso, cada loop gera os valores
entre os respectivos pontos dos vetores x e y, e vai substituindo os valores
já criados de NA dentro do objeto distancias.
            }
        }
        ##########FIM LOOP UTM##########
        ##########OBJETOS DE SAÍDA UTM##########
        distancias<- as.matrix(distancias)#Transforma o vetor distancia em
uma matriz
        lista<-list(distancias, min(distancias), max(distancias),
mean(distancias), sum(distancias))#Cria um obejto(lista) que contém:
(1)Matriz das distâncias entre os pontos, (2)Distância Mínima encontrada,
(3)Distância Máxima encontrada, (4)Distância Média encontrada, (5)Distância
Total encontrada
        names(lista)<- c('Distância entre Pontos', 'Distância Mínima',
'Distância Máxima', 'Distância Média', 'Distância Total')#Dá os nomes entre
aspas aos respectivos objetos do objeto lista
        return(lista)#Retorna o objeto lista no console do R
        ##########FIM OBJETOS DE SAÍDA UTM##########
    }
    ##########FIM UTM##########
    ##########G##########
    if(type=="G")#se type for igual a G(coordenas geográficas em Graus
Decimais)
    {
        ##########GRÁFICO G##########
        par(pch=16, font.lab=2)#Função que estabele padrões gráficas.
        #Argumento pch=16, estabelece que o símbolo a ser plotado será o de
número 16(circulo preenchido).
        #Argumento font.lab=2, estabelece que a fonte a ser usada nos nomes
dos eixos x e y será a de número 2.
        plot(y~x, type="o", xlab="Longitude", ylab="Latitude",
```

```
main="Coordenadas Geográficas (Graus Decimais)")#Plota um gráfico de pontos
com as coordenadas latitudinais(y) em função(~) das longitudinais(x).
        #Argumento type="o", estabelece que os pontos plotados serão
conectados por linhas e essas passarão por cima dos pontos ("o").
        #Argumentos xlab e ylab, estabele que os nomes dos eixos de x e y
serão os fornecidos entre parênteses.
        #Argumento main, estabelece que o título do gráfico será o fornecido
entre parênteses
        text(x, y, labels=1:length(x), pos=3)#função que adiciona textos ao
longo de um gráfico, dadas as coordenadas x e y
        #Argumento labels, vetor que especifica que o texto a ser escrito
vai do número 1 ao número de valores de x
        #Argumento pos=3, estabelece que a posição do texto será acima do
ponto estabelecido por (x,y)
        ##########FIM GRÁFICO G##########
        ##########CONVERSÃO DE GRAUS DECIMAIS PARA UTM##########
        library(rgdal)#Carrega o pacote rgdal

        d <- data.frame(lon=x, lat=y)#transforma os vetores x e y em um data
frame, com os nomes de "lon" e "lat" respectivamente
        coordinates(d) <- c("lon", "lat")#Estabelece que os valores do data
frame d são coordenadas para criar um objeto espacial
        proj4string(d) <- CRS("+init=epsg:4326")#Informa que o datum a ser
utilizado será o WGS 84 como um atributo da projeção e torna essa projeção
uma classe de interface entre os pacotes sp e rgdal. Essa classe refere-se a
um objeto preenchido por NA's que será posteriormente substituido por
informações espaciais associadas com um sistema de coordenadas geográficas.
        atrib.conversao <- CRS("+proj=somerc +lat_0=46.9524056
+lon_0=7.43958333 +ellps=bessel             +x_0=2600000 +y_0=1200000
+towgs84=674.374,15.056,405.346 +units=m +k_0=1 +no_defs")#Infere atributos
específicos para a conversão entre Graus Decimais e UTM, e joga todos esses
atributos no objeto atrib.conversao
        d.utm <- spTransform(d, atrib.conversao)#Transforma os dados de d
segundo os atributos de atrib.conversao
        x2<- d.utm$lon#Extrai os valores convertidos de longitude do objeto
d.utm e os coloca no objeto x2
        y2<- d.utm$lat#Extrai os valores convertidos de latitude do objeto
d.utm e os coloca no objeto y2
        ##########FIM CONVERSÃO DE GRAUS DECIMAIS PARA UTM##########
        ##########LOOP G##########
        for(i in 1:n-1)#Loop com contador i, que vai de 1 à n-1
        {
            for(j in (i+1):n)#Loop com contador j, que vai de i+1 até n
            {
                distancias[i]<- sqrt(((y2[i]-x2[i])^2)+((y2[i+1]-
x2[i+1])^2))#fórmula que calcula as distâncias entre os pontos e salva esses
valores dentro do objeto distancias, que foi previamente criado e indexado
com o contador i. Assim, a fórmula rodará(loop) i vezes e os loops são
indexados com j, que faz com que a cada novo loop é adicionado (+1) aos
valores das posições dos vetores x2 e y2. Com isso, cada loop gera os
valores entre os respectivos pontos dos vetores x2 e y2, e vai substituindo
```

```
os valores já criados de NA dentro do objeto distancias.
            }
        }
        ##########FIM LOOP G##########
        ##########OBJETOS DE SAÍDA G##########
        distancias<- as.matrix(distancias)#Tranforma o vetor distancias em
matriz
        lista<-list(distancias, min(distancias), max(distancias),
mean(distancias), sum(distancias))#Cria um obejto(lista) que contém:
(1)Matriz das distâncias entre os pontos, (2)Distância Mínima encontrada,
(3)Distância Máxima encontrada, (4)Distância Média encontrada, (5)Distância
Total encontrada
        names(lista)<- c('Distância entre Pontos', 'Distância Mínima',
'Distância Máxima', 'Distância Média', 'Distância Total')#Dá os nomes entre
aspas aos respectivos objetos do objeto lista
        return(lista)#Retorna o objeto lista no console do R
        ##########FIM OBJETOS DE SAÍDA G##########
    }
    ##########FIM G##########
}
########## FIM TRACK.DISTANCES ##########
#UFA XD
```

Help track.distances

```
track.distances          package{unknown}                   R Documentation

            Calculate Geographic Coordinates Distances
Description


Calculates the distance between one sampled geographic coordinate point (UTM
or Decimals Degrees)  to the subsequent one and from the last one to the
next and so on. The function also returns the Minimum, Maximum, Mean and
Total Distances.


Usage


track.distances(x, y, type="UTM")


Arguments


x , y   Vector of numeric type containing the longitude/latitude geographic
coordinates. x and y must have the same length.
```

```
type    A character string which determined the type of Geographic
Coordinates. If type is set as "UTM", the
        geographic coordinate data will be read as Universal Transverse
Mercator (UTM), if set as "G" it will be read as Decimals Degrees.


Details


The position of one value on the argument x represents the longitude
geographic coordinates of one point while the same position on argument y
represents the latitude geographic coordinates of the same point.
If the "UTM" value was chosen on the argument type, the distances could be
in meters or kilometers, depending on the values of the vectors x and y,
however, both MUST be in the same units of measurement. If the "G" value was
chosen, the results will be always in meters. The values of the argument
type must be inside parentheses.


Value


track.distances returns a plot graphic of the x and y coordinates and a list
of distances
    comp1 : list of the distances between the points
    comp2 : the minimum distance found
    comp3 : the maximum distance found
    comp4 : the mean distance of the distances between the points
    comp5 : the total distance of all the points


Warning


Package rgdal is required


Author


Henrique, Rafael dos Santos.


References


DISTÂNCIA EUCLIDIANA. In: WIKIPÉDIA, a enciclopédia livre. Flórida:
Wikimedia Foundation, 2013. Available at:
<http://pt.wikipedia.org/w/index.php?title=Dist%C3%A2ncia_euclidiana&oldid=3
6898368> Accessed in 05/05/2014
Falconi, C. E. Como calcular a precisão de uma coordenada geográfica. In:
```

```
Piloto Policial, Portal da Aviação de Segurança Pública e Defesa Civil,
2009. Available at:
<http://www.pilotopolicial.com.br/como-calcular-a-precisao-de-uma-coordenada
-geografica/> Accessed in 05/13/2014
Perna, M. A. O Sistema UTM. In: Engenharia Cartográfica. UERJ, 2010.
Available at: <http://www.carto.eng.uerj.br/cgi/index.cgi?x=utm.htm>
Accessed in 05/06/2014
Short, T. R Reference Card. EPRI PEAC, 2004.
Whuber. Converting geographic coordinate system in R. In: Geographic
Information Systems, 2013. Available at:
<http://gis.stackexchange.com/questions/45263/converting-geographic-coordina
te-system-in-r> Accessed in 05/15/2014
```


```
Exemple

a<-rnorm(10, 50, 0.9)
b<-rnorm(10, 20, 0.9)

track.distances(a,b,type="UTM")
```

track.distances r_documentation2.r

From:
http://ecor.ib.usp.br/ - **ecoR**

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2014:alunos:trabalho_final:raf4rick:start

Last update: **2020/08/12 06:04**