Rosannette Quesada Hidalgo

Doutoranda em Ecologia. Area de trabalho: Seleção sexual e cuidado paternal em opiliões. Quer saber mais? Procura Opilio tracker no Facebook ou Opilio_Tracker no Instagram ;)

https://www.facebook.com/opiliotracker/

Exec

Proposta #1

Descrição da função: A função ira calcular uma matriz de coordenadas (x,y) entre pontos específicos que podem representar manchas de recursos como alimentos, plantas hospedeiras, grupos de indivíduos do sexo oposto, ninhos entre outros. Os valores obtidos serão calculados a partir da distância em diagonal entre os pontos, usando o teorema de Pitágoras para calcular os valores dos catetos restantes.

Entrada dos dados: A entrada dos dados deverá ser um data frame ou matriz contendo as distâncias em diagonal entre os pontos e alguma outra distância até uma linha de referência (por exemplo uma rua). Os argumentos da função serão: coordenadas(dados)

-A função iria incorporar os argumentos da função plot para realizar o gráfico.

Saída da função: A função ira retornar um data frame ou matriz contendo as coordenadas (x,y) para cada ponto e um gráfico representado um mapa com cada ponto.

*Segunda parte: Se der tempo e como uma segunda parte, a função poderia calcular o deslocamento de indivíduos ao longo do tempo com referência aos pontos específicos do mapa. A entrada de dados deverá conter então um segundo data frame ou matriz com as posições dos indivíduos ao longo de diferentes tempos, em referência a algum dos pontos específicos calculados acima. Os argumentos da função serão: coordenadas(dados1, dados2, deslocamento =TRUE)

A função ira retornar então um segundo data frame ou matriz contendo os valores de deslocamento de cada individuo entre os diferentes tempos.

Proposta #2

Descrição da função: A função ira calcular a média e o desvio padrão de diferentes variáveis e ira retornar um gráfico com a média e o desvio padrão de cada variável.

Entrada dos dados: A entrada dos dados deverá ser um data frame ou matriz com colunas

contendo vetores numéricos.

Os argumentos da função serão: plot.sd(dados,plot.type)

```
barplot: ira retornar o gráfico onde a média será representada por uma barra
plot: ira retornar o gráfico onde a média será representada por um ponto
```

-A função iria incorporar os argumentos da função plot e barplot para realizar o gráfico.

Saída da função: A função retornará um data frame ou matriz com a média e o desvio padrão de cada variável e um gráfico onde a média será representada no gráfico por uma barra o um ponto. No caso de que a média seja representada por uma barra, o desvio padrão será representado por uma linha vertical com uma linha horizontal pequena ou “thickmark” no inicio da linha horizontal. No caso de que a média seja representada por um ponto, o desvio padrão será representado por uma linha vertical com duas linhas horizontais pequenas ou “thickmarks” no inicio e no final da linha horizontal.

> Oi Rosa,
>
> gostei dos seus dois planos, mas achei o plano A mais interessante. Só acho que você precisa explicar um pouco melhor como funcionam esses dados de entrada, um desenho pode ajudar bastante!
>
> —-*Danilo G Muniz*

> Ok, Muito obrigada Danilo!

> *— Diogo Melo 2017/06/05 16:49*
>
> Concordo que a proposta A é bacana! Podia incluir talvez a possibilidade de um mapa?

> Obrigada Diogo. Vou incluir um mapa então. Enquanto a segunda parte, não vou incluir a parte de calcular o deslocamento dos individuos em diferentes tempos, mas vou incluir plotar os individuos no mapa também.

Trabalho final

Código da função

```
catetomap=function(tabla,individuals)#criando a funcao catetomap
{
                            ##Calculando um cateto do triangulo##
cateto=rep(NA,length(tabla$h)-1)#criando um objeto para guardar a saida do
for

#na entrada dos dados temos a hipotenusa(diagonal) e para obter um dos
catetos iremos usar um for que ira subtrair a altura de um ninho menos a
altura do ninho anterior

for(i in 2:length(tabla$h))#o for ira obter os valores do cateto a partir
das alturas(h), fazendo a diminuicao en cada linha comencando na linha #2
até a o comprimento do vetor h.Comeca na linha #2 porque sempre ira diminuir
a altura do ninho anterior

{
  a=abs(tabla$h[i]-tabla$h[i-1])#fazendo uma diminuicao entre o valor
absoluto da altura(h) de um ninho menos a altura do ninho anterior e
guardando no objeto a
  a #verificando o objeto a
  cateto[i-1]=a #dando um nome novo e posicao para o objeto a no objeto
cateto
}

cateto #verificando o objeto cateto

                         ##Calculando o outro cateto do triangulo##

b=rep(NA,length(tabla$h)-1)#criando um objeto para a saida do for

for (k in 1:(length(tabla$h)-1))#o for ira obter os valores do cateto
faltante(b) utilizando o cateto obtido no for anterior, a hipotenusa
(diagonal) do data.frame da entrada dos dados e o teorema de Pitagoras. Ira
comecar na linha #1 ate a penultima linha
{
b[k]= sqrt((tabla$diagonal.ninho[k+1]^2)-(cateto[k]^2))#formula do teorema
de Pitagoras onde o cateto faltante(b) e igual a raiz do cateto ao cuadrado
menos a hipotenusa(diagonal) oa cuadrado
}

b #verificando o objeto b
tabla1=data.frame(tabla,"b"=c(0,b)) #colocando o novo vetor b no data.frame
dos dados existentes. A primeira posicao do vetor b deverá ser igual a zero,
devido a que a primeira posicao do vetor diagonal tambem devera sem sempre
```

```
zero, pois nao existe ninho anterior ao primeiro ninho.
tabla1 #verificando o novo data.frame

                        ##Obtendo as coordenadas x,y para fazer um mapa##

#As coordenadas y sao as alturas(h) da entrada dos dados inicial
#Para obter as coordenadas x:

eje.x=rep(NA,length(tabla1$b))#criando um objeto para a saida do for
eje.x[1]=0 #no objeto eje.x o primeiro valor devera ser sempre zero de novo
eje.x #verificando o objeto eje.x


for(j in 2:(length(tabla1$b))) #o for ira obter os valores das coordenadas x
para plotar no mapa. Para o primeiro ninho a coordenada x sera sempre zero.
Para o ninho seguinte sera a coordenada x do ninho anterior mais o valor do
cateto faltante calculado no for acima.
{
   eje.x[j]=eje.x[j-1]+b[j-1]#calculando a coordenada x sumando a coordenada
x do ninho anterior mais cateto calculado no for acima
}

eje.x #verificando o objeto eje.x
tabla2=data.frame(tabla1,eje.x) #colocando o novo objeto eje.x no data.frame
com todos os dados
tabla2 #verificando o data.frame

      ##Obtendo as coordenadas x,y para plotar os pontos secundarios
(individuos no redor dos ninhos,
                                            plantas,recursos)##

#A entrada dos dados das posicoes dos individuos devem ser a distancia em
relacao ao ninho mais proximo e algum angulo a cada 45 graos do ninho

femeas.x=rep(NA,length(tabla2$angulo))#criando um objeto para a saida do for
das coordendas x
femeas.y=rep(NA,length(tabla2$angulo))#criando um objeto para a saida do for
das coordendas x
tabla2$angulo=as.factor(tabla2$angulo)#transformando os angulos em factores
para garantir o funconamento do for
levels(tabla2$angulo)=c("0","45","90","135","180","225","270","315")#indican
do todos os niveis do factor angulo

for(f in 1:length(tabla2$angulo)) #o for ira obter as coordenadas x,y para
individuos perto de um ninho, para cada nivel do fator angulo

{
  if(tabla2$angulo[f]=="0") #se o valor do angulo for "0"
  {
    femeas.x[f]=tabla2$eje.x[f] #a coordenada x sera igual a coordenada x do
```

```
ninho
    femeas.y[f]=tabla2$h[f]+tabla2$distancia.fem[f] #a coordenada y sera a
altura do ninho (ou seja a coordenada y do ninho) mais a distancia do
individuo ao ninho
  }
  if(tabla2$angulo[f]=="45") #se o valor do angulo for "45"
  {
    femeas.x[f]=tabla2$eje.x[f]+(tabla2$distancia.fem[f]/sqrt(2))#a
coordenada x sera a coordenada x do ninho mais a distancia do individuo ao
ninho
    femeas.y[f]=tabla2$h[f]+(tabla2$distancia.fem[f]/sqrt(2))#a coordenada y
sera a altura do ninho (ou seja a coordenada y do ninho) mais a distancia do
individuo ao ninho dividivo por raiz cuadrada de 2 (por teorema de
pitagoras)
  }
  if(tabla2$angulo[f]=="90") #se o valor do angulo for "90"
  {
    femeas.x[f]=tabla2$eje.x[f]+tabla2$distancia.fem[f]#a coordenada x sera
a coordenada x do ninho mais a distancia do individuo ao ninho
    femeas.y[f]=tabla2$h[f]#a coordenada y sera a altura do ninho (y do
ninho)
  }
  if(tabla2$angulo[f]=="135") #se o valor do angulo for "135"
  {
    femeas.x[f]=tabla2$eje.x[f]+(tabla2$distancia.fem[f]/sqrt(2))#a
coordenada x sera a coordenada x do ninho mais a distancia do individuo ao
ninho dividivo por raiz cuadrada de 2
    femeas.y[f]=tabla2$h[f]-(tabla2$distancia.fem[f]/sqrt(2))#a coordenada y
sera a altura do ninho (y do ninho) menos a distancia do individuo ao ninho
dividivo por raiz cuadrada de 2
  }
  if(tabla2$angulo[f]=="180") #se o valor do angulo for "180"
  {
    femeas.x[f]=tabla2$eje.x[f]#a coordenada x sera a coordenada x do ninho
    femeas.y[f]=tabla2$h[f]-tabla2$distancia.fem[f]#a coordenada y sera a
altura do ninho (y do ninho) menos a distancia do individuo ao ninho
  }
  if(tabla2$angulo[f]=="225") #se o valor do angulo for "225"
  {
    femeas.x[f]=tabla2$eje.x[f]-(tabla2$distancia.fem[f]/sqrt(2))#a
coordenada x sera a coordenada x do ninho menos a distancia do individuo ao
ninho dividivo por raiz cuadrada de 2
    femeas.y[f]=tabla2$h[f]-(tabla2$distancia.fem[f]/sqrt(2))#a coordenada y
sera a altura do ninho (y do ninho) menos a distancia do individuo ao ninho
dividivo por raiz cuadrada de 2
  }
  if(tabla2$angulo[f]=="270") #se o valor do angulo for "270"
  {
    femeas.x[f]=tabla2$eje.x[f]-tabla2$distancia.fem[f]#a coordenada x sera
a coordenada x do ninho menos a distancia do individuo ao ninho
    femeas.y[f]=tabla2$h[f]#a coordenada y sera a altura do ninho (y do
```

```
ninho)
  }
  if(tabla2$angulo[f]=="315") #se o valor do angulo for "315"
  {
    femeas.x[f]=tabla2$eje.x[f]-(tabla2$distancia.fem[f]/sqrt(2))#a
coordenada y sera a coordenada x do ninho menos a distancia do individuo ao
ninho dividivo por raiz cuadrada de 2
    femeas.y[f]=tabla2$h[f]+(tabla2$distancia.fem[f]/sqrt(2))#a coordenada y
sera a altura do ninho (y do ninho) mais a distancia do individuo ao ninho
dividivo por raiz cuadrada de 2
  }

}



femeas.x #verificando as coordenadas x dos individuos
femeas.y #verificando as coordenadas y dos individuos


tabla3=data.frame(tabla2,femeas.x,femeas.y) #colocando as coordenadas x,y
dos individuos no data.frame
tabla3 #verificando o data.frame com todos os dados

                    ##Fazendo o mapa dos ninhos,plantas,recurso e os
individuos

min.x=min(femeas.x) #fazendo um objeto como o valor minimo da coordenada x
para que seja o limite do aixo x do plot
max.x=max(femeas.x)#fazendo um objeto como o valor maximo da coordenada x
para que seja o limite do aixo x do plot
min.y=min(femeas.y)#fazendo um objeto como o valor minimo da coordenada y
para que seja o limite do aixo y do plot
max.y=max(femeas.y)+2 #fazendo um objeto como o valor maximo da coordenada y
para que seja o limite do aixo y do plot mais 2 para que fique um pouco de
espaco



plot1=plot(tabla3$eje.x,tabla3$h,tck=0,xaxt="n",yaxt="n",xlab="",ylab="",
xlim=c(min.x,max.x),ylim=c(min.y,max.y)) #fazendo o mapa dos ninhos

text(tabla3$eje.x,tabla3$h,labels = tabla3$ninho,pos=4)#colocando labels nos
ninhos

if(individuals=="TRUE")#se o argumanto individuals for = TRUE a funcao ira
plotar os pontos secundarios também
{
points(tabla3$femeas.x,tabla3$femeas.y,col="red") #colocando os individuos
ou pontos secundarios no mapa
}
```

```
return(tabla3) #o que a funcao ira retornar
}
```

Help da função help_de_catetomap.txt

```
catetomap{}             Package: unknown            R Documentation

Calcula uma matriz de coordenadas x,y e plota os pontos em um desenho (mapa)

Description:

Calcula as coordenadas x de pontos a partir da distância dos pontos até uma
linha principal (coordenadas y) e a distância diagonal entre esses pontos
(hipotenusa), utilizando o Teorema de Pitágoras. Para utilizar o teorema de
Pitágoras primeiro calcula um dos catetos a partir das diferencias entre as
distâncias entre os pontos até a linha principal. Depois calcula o cateto
faltante com o teorema de Pitágoras. Também calcula as coordenadas x,y para
pontos secundários nas proximidades dos pontos principais calculados, a
partir da distância diagonal desde o ponto secundário até o ponto principal
e o ángulo entre ponto secundário e o ponto principal, também utilizando o
teorema de Pitágoras. A função ira plotar em um desenho só os pontos
principais o tanto os pontos principais quanto os pontos secundários.

Usage:

catetomap(tabla,individuals="TRUE")

Arguments:

tabla   data frame com 6 colunas: A identidade dos pontos principais que
representam ninhos, manchas de recurso, indivíduos, etc. A distância desde
cada ponto principal até uma linha reta que pode ser por exemplo uma rua. A
distância diagonal entre os pontos principais, sempre de um ponto ao ponto
anterior. A identidade dos pontos secundários que representam indivíduos ou
qualquer outro ponto próximo aos pontos principais. A distância em diagonal
desde o ponto secundário até um ponto principal. O ângulo entre o ponto
secundário e o ponto principal (só com valores a cada 45 graus, veja o
details).

individuals lógico; se for TRUE a função irá plotar os pontos secundários no
redor dos pontos principais. Se for FALSE a função ira plotar só os pontos
principais.

Details:

Na entrada de dados, catetomap só aceita ângulos entre os pontos secundários
e um ponto principal cujos valores sejam a cada 45 graus, ou seja só podem
ser valores de: 0, 45, 90, 135, 180, 225, 270 e 315 graus.
```

```
Value:

A função catetomap retorna um data.frame com os catetos calculados a partir
do Teorema de Pitágoras, as coordenadas x para cada ponto principal, e as
coordenadas x,y para os pontos secundários nas proximidades dos pontos
principais. Também retorna um desenho ou mapa com só os pontos principais
plotados o tanto os pontos principais quanto os pontos secundários.

Notes:

Muito importante: Cada valor da distância diagonal entre os pontos
principais deve ser sempre em relação ao ponto principal anterior. Isto quer
disser que o primeiro valor do vetor contendo as distâncias diagonais deve
ser sempre zero, pois o primeiro ponto não tem ponto anterior.

Author(s):

Rosannette Quesada-Hidalgo (2017).
21rosit@gmail.com

Example:

ninho=c(1,1,2,3,4)
h=c(10,10,2,15,16)
diagonal.ninho=c(0,0,30,40,10)
femeas.id=c(1:5)
distancia.fem=c(5,5,20,7,6)
angulo=c(45,90,135,270,225)
tabla=data.frame(ninho,h,diagonal.ninho, femeas.id,distancia.fem,angulo)
head(tabla)


write.csv(tabla,"exemplo.funcao.csv")
dados=read.csv("exemplo.funcao.csv")

catetomap(dados,individuals=FALSE)
catetomap(dados,individuals=TRUE)
```

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

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2017:alunos:trabalho_final:rquesada:start

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