

# Ricardo Fernandes Campos Jr



Departamento de Genética e Biologia Evolutiva - Universidade de São Paulo

Orientadora: Cristina Yumi Miyaki

Projeto: Estrutura Populacional de *Dendrocolaptes platyrostris* (Aves: Passeriforme)

## Meus Exercícios

[Exercício 1](#) [Exercício 2](#) [Exercícios 3](#) [Exercícios 4](#) [Exercícios 5](#) [Exercícios 6](#) [Exercícios 7](#) [Exercício 8](#)  
[Exercícios 9](#)

## Proposta Final

**Plano A** Escrever uma função que selecione pontos que se distanciam em “z” quilômetros de um transecto dado por “(X1, Y1), (X2, Y2)” - em graus decimais) - formando um ângulo “w” do mesmo, a partir de uma matriz de dados que possuem os valores (“X”, “Y”, ...).

**Plano B** Escrever uma função que gere valores aleatórios para quantidades de pessoas com três (ou mais) classes de fenótipos específicos e fique aguardando resultados de Frequências de alelo e genótipos. Esta mesma ainda pode incluir parâmetros para ligação ao “X”, teste de Equilíbrio de Hardy-Weinberg, múltiplos locus, genes ligados e desvio padrão, respondendo depois se os valores foram respondido corretamente.

## Comentários

Não sei se entendi perfeitamente sua primeira proposta. Você vai ter um transecto e a função vai selecionar pontos aleatórios de amostragem, por exemplo, a partir desse transecto? Pode ser interessante. A segunda proposta pode se tornar rapidamente cabeluda... melhor ficar com a primeira.  
— [Fabio de A. Machado](#) 2011/04/06 22:59

**Resposta** Na verdade a idéia é partir do transecto para selecionar pontos de um dataframe que já possuem os valores XY, e não gerá-los.

## Página de Ajuda

selpts

package:nenhum

R Documentation

Description:

Seleciona coordenadas dos dados(xydata) distribuidos em torno de um

transecto (x1, y1, x2, y2) a partir de uma distância estabelecida (dist). Retorna um data frame com os pontos selecionados e os demais dados associados a eles. Também plota o transecto gerado, os pontos selecionados e os pontos excluídos.

#### Usage:

```
selpts = function (xydata, dist=1 , x1, y1, x2, y2, np=100)
```

#### Arguments:

xydata uma matrix ou um data frame contendo as coordenadas geográficas em graus decimais (linhas com NA são omitidas). As coordenadas devem seguir esta ordem: LONGITUDE, LATITUDE, ...

dist distancia em quilômetros dentro da qual as coordenadas de xydata devem estar para serem selecionadas.

x1 coordenada longitudinal do ponto mais a esquerda do transecto.

y1 coordenada latitudinal do ponto mais a esquerda do transecto.

x2 coordenada longitudinal do ponto mais a direita do transecto.

y2 coordenada latitudinal do ponto mais a direita do transecto.

np número de pontos usados para construir o transecto (ver detalhes).

#### Details:

selpts seleciona as coordenadas calculando a distancia euclidiana entre as coordenadas dos pontos do transecto e as coordenadas dos dados. O número de pontos que compõem o transecto é especificado em "np", então a precisão da medida de distância dos dados ao transecto depende do valor de "np" e da escala dos dados.

selpts remove todas as linhas que possuem NA. Logo, elas não serão selecionadas mesmo que estejam dentro da distancia estabelecida.

selpts plota os pontos não selecionados, os pontos selecionados, e o transecto a partir do qual a distancia foi calculada. Os pontos são, respectivamente: circulo aberto, circulo fechado, circulo vermelho.

#### Value:

Retorna um data.frame de xydata com apenas os pontos que estão dentro da distância especificada e suas variáveis.

#### Note:

Para `selpts` funcionar o pacote "`sp`" (<http://r-spatial.sourceforge.net/>) deve estar instalado.

Author(s):

Ricardo Fernandes Campos Junior  
Laboratório de Genética e Evolução Molecular de Aves - Departamento de Genética e Biologia Evolutiva, Universidade de São Paulo.  
[ricardofcj@gmail.com](mailto:ricardofcj@gmail.com)

References:

<http://r-spatial.sourceforge.net/>

See Also:

'`spDists`' do pacote "`sp`" para detalhes de como são calculadas as distâncias.

Examples:

### Exemplo 1, sem NA

```
x <- sample(runif(50, -58, -52))
y <- sample(runif(50, -30, -20))
a <- sample(seq(2, 5, length=50))
xy1 = data.frame(x,y,a)
colnames (xy1) = c("LONG", "LAT", "Variavel a")
head(xy1)

selpts (xy1, dist=150 , -56, -27, -54, -21.5, np=100)
```

Exemplo 2, com NA

```
x <- sample(runif(50, -58, -52))
y <- sample(runif(50, -30, -20))
a <- sample(seq(2, 5, length=50))
b <- sample(rep(c(NA, 2, 5, NA, 3), times=10))
xy1 = data.frame(x,y,a,b)
colnames (xy1) = c("LONG", "LAT", "Variavel a", "Variavel b")
head(xy1)

selpts (xy1, dist=150 , -56, -27, -54, -21.5, np=50)
```

## Código da Função

```
selpts = function (xydata, dist=1 , x1, y1, x2, y2, np=100)
{
  ### "x" deve ser longitude, "y" deve ser latitude. O ponto mais a esquerda
  do transecto deve ser colocado em "xy1". A distância deve ser dada em
  quilômetros, assim como o resultado sairá em quilômetros. ###
```

```
### O argumento "dados" pode ser um data frame ou uma matrix, mas a
primeira linha deve ser longitude e a segunda deve ser latitude. ###
### O argumento "np" corresponde ao número de pontos gerados dos quais
serão medidas as distâncias dos dados. Assim, quanto maior for "np", maior
será a precisão de distância. ###
### A função retira as linhas que possuem algum NA. ###
### Ricardo Fernandes Campos Junior, 2011. (ricardofcj@gmail.com) ###
require(sp)
dados = na.omit(xydata)
coords = SpatialPoints(data.frame(dados[,1], dados[,2]))
valx = seq(from=x1, to=x2, length=np)
valy = seq(from=y1, to=y2, length=np)
transecto = SpatialPoints(data.frame(valx, valy))
transectom = matrix(c(valx, valy), ncol=2)
distance= spDists(transecto, coords, longlat=TRUE)
pontos = matrix (NA, ncol=dim(dados)[2], nrow=dim(dados)[1])
names=c(colnames(dados))
for (i in 1:dim(dados)[1])
{
  if(min(distance[,i])<=dist)
  {
    pontos[i,1] = dados[i,1]
    pontos[i,2] = dados[i,2]
    if (dim(pontos)[2]>2)
    {
      for (m in 3:dim(dados)[2])
      {
        pontos[i,m] = dados[i,m]
      }
    }
  }
}
resultado = matrix(pontos[!is.na(pontos)], ncol=dim(pontos)[2])
pplot = matrix(resultado[,1:2], ncol=2)
ptsplot = SpatialPoints(data.frame(pplot))
plot(coords, pch=1)
plot(transecto, col="red", pch=1, add=TRUE)
plot(ptsplot, add=TRUE, pch=16)
colnames(resultado) = names
return(resultado)
}
```

## Arquivos da Função

[selpts help\(selpts\)](#)

From:

<http://ecor.ib.usp.br/> - **ecoR**

Permanent link:

[http://ecor.ib.usp.br/doku.php?id=05\\_curso\\_antigo:r2011:alunos:trabalho\\_final:ricardo\\_campos:start](http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2011:alunos:trabalho_final:ricardo_campos:start)

Last update: **2020/08/12 09:04**

