

# Fernando Silvério Ribeiro

Mestrando em Ecologia, Laboratório de Diversidade e Conservação de Mamíferos, IB, USP

[exec](#)

## Trabalho Final

### **Proposta A** - Seleção aleatória de pontos de amostragem com restrições de distâncias

A partir de uma matriz contendo todos os possíveis pontos de amostragem de uma área com suas respectivas coordenadas geográficas, a função pretende selecionar aleatoriamente um determinado número de pontos que estejam localizados a, no mínimo, uma determinada distância entre eles. Além disso, caso o usuário acrescente informações à matriz, como o estrato a que os pontos pertencem e um valor para este estrato, como por exemplo sua área, a função será capaz de realizar uma seleção alocando a quantidade de pontos proporcionalmente ao valor de cada estrato. Os argumentos da função serão o número de pontos a ser sorteado, a distância mínima entre pontos e se a seleção será estratificada ou não.

### **Proposta B** - Padrões temporais de atividade

Em trabalhos com armadilhas fotográficas, além de dados de presença/ausência por sítio, também é possível obter dados sobre o padrão de atividade temporal das espécies. A partir de uma matriz contendo os sítios das câmeras, as espécies e o horário de registro de cada fotografia, a função pretende construir gráficos que demonstrem esses padrões e que sejam facilmente comparáveis entre espécies, ou seja, os gráficos devem apresentar o padrão de atividade a partir de valores proporcionais. Nos argumentos, além da escolha da espécie, o usuário terá a opção de que a função retorne o padrão geral da espécie (incluindo todos os registros) ou o padrão para diferentes sítios (ou conjuntos de sítios).

Fernando, achei sua proposta A bem interessante, mas vale notar que o usuário tem que fornecer as coordenadas dos pontos amostrais de alguma forma certo? Lembre-se de especificar no HELP o formato em que essas coordenadas tem que ser passadas pra função.

A proposta B achei um pouco mais fraca, se você for perseguir esse plano acho que vale a pena permitir que o usuário escolha mais de uma espécie de cada vez e talvez incluir algum tipo de comparação "não gráfica" entre diferentes espécies. Se o horário em que a espécie tem atividade for importante, será que vale a pena partir pra estatística circular? (Não quero complicar sua vida, é só uma sugestão, caso você ache isso emocionante dê uma olhada no pacote circular.).

— [Danilo G. Muniz](#)

## Final: proposta A

```
#Cria uma função que faz sorteios aleatórios de pontos de amostragem com
distâncias mínimas entre esses pontos.
#O sorteio pode ser estratificado ou não
sortptos <- function (x, n=10, dist=10, estratificado=FALSE)
{
#cria uma função semelhante a anterior, com a finalidade de aplicá-la
diretamente ao data frame (caso o sorteio
#não seja estratificado, ou a subsets do data frame (em caso de sorteio
estratificado)
func <- function (x, m=n, d=dist)
    {
#carrega o pacote sp, necessário para o cálculo de distância a partir de
coordenadas
#pensei em criar uma função para este cálculo (por trigonometria), mas os
resultados tinham um pequeno erro
#(devido a curvatura da terra) que não consegui corrigir
    library("sp")
#cria uma matriz de distâncias (em kms) entre todos os pontos
    mat.dist=spDists(as.matrix(x[,2:3]),longlat=TRUE)
#cria um vetor para guardar os pontos sorteados que respeitem a distância
mínima. O vetor é grande de modo a
#garantir que muitos pontos sejam testados.
    resultado=rep(NA,nrow(mat.dist))
#cria um vetor de tamanho igual ao acima, para guardar todos os pontos
sorteados (incluindo os que não
#respeitam a distância mínima)
#necessário criar dois vetores para facilitar o teste de distância
    pre.sort=rep(NA,nrow(mat.dist))
#sorteia o primeiro ponto entre todos os possíveis e o guarda no vetor de
resultados
    resultado[1]=sample(1:nrow(mat.dist),1)
#guarda o primeiro ponto no vetor pre.sort
    pre.sort[1]=resultado[1]
#cria um loop que gera os próximos sorteios
    for (i in 2:nrow(mat.dist))
    {
#guarda no vetor pre.sort todos os outros pontos sorteados
    pre.sort[i]=sample(1:nrow(mat.dist),1)
#um teste lógico que compara as distâncias entre os pontos sorteados (vetor
pre.sort) e os que foram realmente
#escolhidos (vetor resultado)
#caso o ponto fique igual ou maior que a distância escolhida...
    if (sum(mat.dist[pre.sort[i],c(resultado[1:(i-1)])]<d)==0)
    {
#o ponto é guardado no vetor resultado
    resultado[i]=pre.sort[i]
    }
}
#caso o ponto fique mais próximo que a distância definida de pelo menos um
```

```
dos pontos do vetor resultado...
  else
  {
#o elemento correspondente no vetor resultado recebe o valor 0
  resultado[i]=0
  }
}
#a função retorna o data frame de dados indexado pelos m primeiro valores do
vetor resultado que não são 0
  return(x[resultado[resultado!=0][1:m],])
}
#se o usuário não escolher um sorteio estratificado...
  if (estratificado==FALSE)
  {
#aplica-se a função nos dados selecionados
  return(func(x))
}
#se o usuário escolher um sorteio estratificado...
  if (estratificado==TRUE)
  {
#iniciam-se os passos para calcular a proporção de pontos por estrato
#cria um vetor com os estratos existentes
  estratos=c(unique(x[,4]))
#cria um vetor com o valor de cada estrato
  v.val.est=c(unique(x[,5]))
#cria um loop para aplicar a função uma vez em cada estrato
  for (j in 1:length(estratos))
  {
#cada estrato é representado por um subset dos pontos totais
#o número de pontos por estrato é definido pela proporção do valor do
estrato
#em relação a soma dos valores de todos estratos
result=func(subset(x,x[,4]==j),m=round(v.val.est[j]/sum(v.val.est)*n))
#como o resultado do sorteio em cada subset é um passo do loop, usa-se a
função print para
#aparecer no console todos os resultados
  print(result)
  }
}
}
```

## Help

sortptos

package:nenhum

R Documentation

Sorteia pontos amostrais aleatoriamente, de maneira estratificada ou não.

### Description:

A partir de um data frame com coordenadas de pontos, a função sorteia aleatoriamente pontos amostrais,

com distâncias mínimas entre eles. Caso os pontos pertençam a diferentes estratos, a função sorteia pontos em cada estrato proporcionais ao seu valor. A função utiliza o pacote `sp`.

#### Usage:

```
sortptos (x, n=10, dist=10, estratificado=FALSE)
```

#### Arguments:

`x` Data.frame com todos os pontos para sorteio e suas respectivas coordenadas. Deve ter pelo menos 3 colunas, sendo a primeira a identificação de cada ponto, a segunda a longitude e a terceira a latitude. Em caso de sorteio estratificado o data.frame deve ter mais 2 colunas, a quarta sendo o número do estrato ao qual o ponto pertence e a quinta o valor deste estrato

`n` Valor numérico: número de pontos que será sorteado

`dist` Valor numérico: distância mínima (em kms) entre os pontos a serem sorteados

`estratificado` Lógico. Se "TRUE" a função realizará o sorteio estratificado. Se "FALSE" o sorteio não levará em conta os estratos.

#### Details:

Necessário ter o pacote `sp`.

#### Value:

A função retornará um subconjunto do data.frame (`x`) contendo os `n` pontos escolhidos. Caso não seja possível completar os `n` pontos, os faltantes serão preenchidos com NAs.

#### Author(s):

Fernando Silvério Ribeiro  
fernando.silverio.ribeiro@usp.br

**Examples:**

```
sortptos(tabela.pontos)
```

```
sortptos(tabela.pontos, x=20, dist=5, estratificado=TRUE)
```

**Arquivos:**

arquivo csv com 500 pontos na Paraíba (retirado da internet) [pontos\\_interesse.csv](#)

script adicionando mais 2 colunas ao arquivo anterior (estratos e valores) para uma seleção estratificada [criar\\_dataframe\\_est.r](#)

script com código da função [script\\_funcao.r](#)

From:

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

Permanent link:

[http://ecor.ib.usp.br/doku.php?id=05\\_curso\\_antigo:r2015:alunos:trabalho\\_final:fernando.silverio.ribeiro:start](http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2015:alunos:trabalho_final:fernando.silverio.ribeiro:start) 

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