

LUCAS TEIXEIRA

Mestrando em Ecologia, Instituto de Biociências, USP.

[Laboratório de Diversidade e Conservação de Mamíferos](#)



Em meu mestrado, procuro entender como a cobertura florestal, associada ao contato com a floresta e ao conhecimento sobre a fauna dos moradores locais, influencia a atitude e a intenção de retaliar mamíferos que causam danos a cultivos e criações em paisagens de Mata Atlântica.

Contato: lteixeira@ib.usp.br / [Currículo Lattes](#) :)

Meus Exercícios

- [Exercício 1 - Introdução ao R](#)
- [Exercício 4.2 - Cervejas](#)
- [Exercício 4.3 - Caixetais](#)
- [Exercício 4.4 - Eucaliptos](#)
- [Exercício 4.5 - Mais caixetais](#)
- [Exercício 7.2 - O modelo mais simples](#)
- [Exercício 7b - Uma estimativa da incerteza](#)
- [Exercício 7b - Galileu estava Certo?](#)
- [Exercício 7b - Massa de recém-nascidos](#)
- [Exercício 8.1 - Reamostragem](#)
- [Exercício 8.2 - Simulando uma regressão linear](#)
- [Exercício 9.2 - Análise exploratória simultânea de duas variáveis](#)

Ou: [exec](#)

Propostas de Trabalho Final

Proposta principal

Pesquisas sociais aplicadas à conservação muitas vezes se baseiam na aplicação de questionários. A escolha das pessoas entrevistadas geralmente é feita a partir de residências dentro da área de estudo, que podem ser mapeadas a partir de imagens de satélite ou visitas em campo. Tais informações geralmente são incluídas em grandes bancos de dados, que podem servir como base para a escolha das residências cujos moradores serão entrevistados. Apesar de parecer simples, aplicar questionários é uma tarefa difícil, que requer um planejamento prévio e bem estruturado sobre os critérios que devem ser incluídos para a amostragem a fim de que, ao final, obtenha-se um quadro de unidades amostrais representativo, dada a inviabilidade logística de se entrevistar todas as pessoas de uma área. Por isso, escolher um determinado número de pessoas a partir de critérios que tais bancos de dados contenham é um método bastante prático para se escolher quais pessoas serão sorteadas.

A função

Esta função terá o objetivo de criar um subset de unidades amostrais de um banco de dados, sorteando um certo número de residências dentro de uma ou mais áreas de estudo a partir de critérios como distância mínima entre as residências, número e idade média dos moradores.

Input

Um dataframe que contenha na primeira coluna o nome da unidade amostral (residência), na segunda coluna o nome da área de estudo, nas terceira e quarta colunas a latitude e a longitude respectivamente, na quinta coluna o número de moradores e na sexta coluna a idade média dos moradores.

Output

A função deve retornar um dataframe que contenha as unidades amostrais selecionadas com suas respectivas características (área de estudo, latitude, longitude, número e idade média dos moradores).

Comentários Renata Orofino

Legal, acho que vai te dar um trabalhinho bacana.

Fiquei com algumas caramilholas:

- A idade média dos moradores de uma residência é resultado de uma conta, certo? Conta que muda em +1 todo ano. Supondo que vc não faz uma base de dados anual, seria importante vc considerar o ano de nascimento, ou adicionar a conta ano a ano...
- Eu acharia bonito se vc conseguisse retornar uma imagem com os pontos selecionados num plano cartesiano pra dar uma ideia visual da amostra final.

Lucas aqui... Oi, Rê! Obrigado pelo comentário! rs

De fato, acho que ela vai me dar um trabalhinho bastante grande... Já está dando, na real :P

Quanto a suas caramilholas (adorei essa palavra). De fato, a idade média pode variar em +1 todo ano dado que as pessoas continuam vivas, mas ela também pode variar caso haja o falecimento ou nascimento de alguém da residência, por isso a ideia seria usar esse critério dado um banco de dados já inicialmente coletado para estimativa, apenas. Agora, o que pode ser feito e que talvez seja mais válido seja a inclusão de um outro critério: adulto (homem ou mulher) maior que 18 anos, já que questionários só podem ser aplicados a maiores. O que acha? Não havia pensado em trabalhar com uma imagem gráfica para isso. Vou tentar :)

Comentários Renata Orofino

Legal essa ideia. Seria como um fator (presença de adultos? - meio obvio, né? não há residências sem maiores de 18, provavelmente), ou contínua (número de adultos)?

Talvez seja mais fácil (não sei qual a informação disponível no banco de dados) trabalhar com a data de nascimento dos indivíduos e fazer uma coluna lógica que vai estar sempre atualizada (maiores de 18==TRUE?)

Lucas respondendo... Bem, na verdade, tem casas de todo o jeito nessa vida... A gente, por

exemplo, achou algumas com menores apenas... Pessoal que casa cedo e vai seguir a vida... Mas enfim, acho que a ideia da coluna lógica é boa :) Vou trabalhar nisso! :) Obrigado, mais uma vez! _

Oi Lucas (e Rena), Estou ainda um pouco confusa sobre o que a função acessa. É o banco de dados ou é um data frame com as variáveis necessárias? Me parece ter um pouco de confusão aí. Porque o argumento lógico se tem adulto ou não seria extraído do banco de dados, certo? Ou isto seria mais uma coluna no data frame? Só tome cuidado para não ser algo muito específico para a estrutura do seu banco de dados. Mas no geral é uma boa proposta! E, por favor, faz um mapinha que vai ser lindo <3 — [Sara](#)

Lucas aqui... Ois, Rena e Sara, reformulei a proposta para dar uma maior generalização e clareza, além de ter mudado o objetivo dela para retornar também um gráfico com os pontos selecionados. :)

Proposta secundária

Pesquisas em ecologia da paisagem muitas vezes requerem a seleção aleatória de pontos amostrais dentro das paisagens de estudo para observações, instalação de câmeras ou armadilhas de capturas, como pitfalls, shermann, tomahawk etc. Em paisagens quadradas, sortear tais pontos fica mais fácil. Porém, muitas vezes, as paisagens de estudo são circulares e assim sortear tais pontos requer mais trabalho.

A função

Esta função terá o objetivo de selecionar pontos aleatórios em uma paisagem circular. Para isso, deverá ser informado a localização geográfica (latitude e longitude) do ponto central da paisagem, o raio dentro do qual estes pontos deverão ser sorteados e a distância mínima entre esses pontos.

Input

Um dataframe que contenha na primeira coluna o nome da paisagem, na segunda coluna o nome do ponto central da paisagem, nas terceira e quarta colunas a latitude e a longitude respectivamente, deste ponto central.

Output

A função deve retornar um dataframe que contenha, na primeira coluna, o nome da paisagem, na

segunda, o nome dos pontos sorteados e nas terceira e quarta colunas a latitude e a longitude respectivamente, dos pontos sorteados.

Comentários Renata Orofino

Se eu entendi direito, ela segue o mesmo princípio da anterior, não? Acho que a primeira tem mais elementos e inclui detalhes que nesta vc não precisaria pensar, então voto na proposta principal. Se precisar de qq ajuda, me avise: renata.orofino@gmail.com

Lucas respondendo... Então, na verdade, a primeira proposta seria selecionar residências para aplicações de questionários. A segunda, seria criar pontos aleatórios para amostragens diversas em paisagens. Mas de qualquer forma, também acho que a primeira vai ser mais desafiadora, então vou trabalhar na proposta principal. Obrigado, de novo! :)

Manda bala na 1! — [Sara](#)

Lucas diz: Beleza! :)

Função selecionada: proposta principal, com alterações sugeridas

Pesquisas sociais aplicadas à conservação muitas vezes se baseiam na aplicação de questionários. A escolha das unidades amostrais entrevistadas geralmente é feita a partir de residências localizadas dentro da área de estudo, que podem ser mapeadas a partir de imagens de satélite ou visitas em campo. Tais informações geralmente são incluídas em planilhas, que podem servir como base para a escolha das residências cujos moradores serão entrevistados. Apesar de parecer simples, aplicar questionários é uma tarefa difícil, que requer um planejamento prévio e bem estruturado sobre os critérios que devem ser incluídos para a amostragem a fim de que, ao final, obtenha-se um quadro de unidades amostrais representativo, dada a inviabilidade logística de se entrevistar todas as unidades amostrais de uma área. Por isso, determinar uma amostra a partir de critérios pré-estabelecidos é um método bastante prático para selecionar quais unidades poderão ser sorteadas.

A função

Esta função terá o objetivo de gerar um subset de uma amostra a partir de um dataframe, selecionando um certo número de unidades amostrais dentro de uma determinada área de estudo a partir de critérios como sexo, ano máximo de nascimento e distância mínima entre as localizações de cada unidade amostral.

Input

Um dataframe que contenha na primeira coluna o ID da unidade amostral; na segunda coluna, o sexo; na terceira coluna, o ano de nascimento da unidade amostral (para critério de idade); na quarta coluna, a área de estudo na qual a unidade amostral se encontra; na quinta coluna, o ID da localização de cada unidade amostral; na sexta e sétima colunas, a latitude e a longitude dessa localização, respectivamente.

Output

A função deve retornar um dataframe que contenha as unidades amostrais selecionadas com base nos critérios estipulados com suas respectivas características (sexo, ano de nascimento, área de estudo, localização, latitude e longitude), assim como um arquivo .txt separado por tabulação deste dataframe e uma imagem com as unidades selecionadas plotadas em um plano cartesiano.

Página de Ajuda

shantay

package: nenhum

R Documentation

Sorteio de unidades amostrais com base em critérios fornecidos pelo usuário e na distância mínima necessária entre elas.

Description:

A partir de um dataframe, sorteia um determinado número de unidades amostrais com base nos critérios sexo, ano de nascimento máximo, área de estudo e distância mínima entre suas localizações. Salva essas informações em um arquivo .txt separado por tabulação e retorna um plano cartesiano com as unidades amostrais sorteadas.

Usage:

```
shantay(dados, n.amostra, ano.nasc, sexo, area.estudo, dist.min)
```

Arguments:

dados: Dataframe; deve conter na primeira coluna o ID da unidade amostral; na segunda coluna, o sexo; na terceira coluna, o ano de nascimento da unidade amostral (para critério de idade); na quarta coluna, a área de estudo na qual a pessoa reside; na quinta coluna, o ID da localização da unidade amostral, na sexta e sétima colunas, a latitude e a longitude da localização de cada unidade amostral, respectivamente.

n.amostra: Valor numerico; numero de unidades amostrais a serem sorteadas.

ano.nasc: Valor numerico; ano de nascimento maximo de interesse (para criterio de idade minima). Se idade das unidades amostrais estiver em anos, pode-se calcular seu ano de nascimento subtraindo-se a idade em anos de cada unidade amostral do ano calendario em que a informacao foi obtida.

sexo: Caracter; sexo das unidades amostrais de interesse. Recebe: 'f' (feminino), 'm' (masculino) ou 'NA' se o sexo nao importar como criterio de selecao.

area.estudo: Caracter; recebe o nome da area de estudo de interesse.

dist.min: Valor numerico; distancia (em metros) minima a que cada unidade amostral deve estar uma da outra.

Details:

E necessario ter o pacote 'sp' instalado.

A latitude e a longitude devem estar informadas em graus decimais e, no dataframe de input, localizadas nas sexta e setima colunas, respectivamente.

Todos os argumentos da funcao devem ser fornecidos.

Value:

A funcao shantay retorna:

No console, o numero de unidades amostrais que foram removidos a cada criterio de selecao fornecido e um dataframe contendo as unidades amostrais selecionadas com suas respectivas caracteristicas (sexo, ano de nascimento, área de estudo, localizacao, latitude e longitude de cada unidade amostral).

Tambem retorna um arquivo .txt separado por tabulacao deste dataframe, salvo no diretorio de trabalho corrente do R sob o nome 'amostragem.txt'.

Um plano cartesinado em que estao plotados os pontos referentes as unidades amostrais selecionadas. Para melhor visualizacao, recomenda-se abrir uma nova janela grafica antes de executar a funcao. Para Windows, x11(), por exemplo.

Warnings:

A funcao e interrompida e retornam-se mensagens de erro quando o objeto de entrada nao e um dataframe, quando algum dos argumentos nao e fornecido, o pacote 'sp' nao esta instalado e quando o numero de unidades amostrais disponiveis para sorteio apos os criterios de selecao e menor do que o numero de unidades amostrais a serem sorteadas.

Na primeira vez que a funcao e executada, exibe uma mensagem de aviso que o pacote 'sp' foi carregado SE ele estiver instalado.

Notes:

Atencao para nao sobrescrever o arquivo .txt. Se ele for interessante, convem-se renomea-lo antes de executar a funcao novamente.

Author(s):

Lucas Teixeira
lteixeira@ib.usp.br

Sao Paulo, 16 de maio de 2016

References:

Newing, H., Eagle, C., Puri, R. & Watson, C.W. 2011. Conducting research in conservation: a social science perspective, Routledge, New York.

Pacote 'sp': <https://cran.r-project.org/web/packages/sp/sp.pdf>

See Also:

Funcoes: subset(), spDists(), sample(), plot()

Acknowledgments:

A Gabriela Marin e Vinicius Biffi,
pela disponibilidade em me explicar alguns comandos e argumentos.

Examples:

#Exemplo 1:

```
#criando vetores para incluir no dataframe  
pessoa <- c("a1", "a2", "b1", "b2", "b3", "c1", "c2", "d1", "d2", "e1")
```

```
sexo <- c("m", "f", "f", "m", "f", "m", "f", "f", "m", "m")
nascimento <- c(1973, 1977, 1980, 1979, 2000, 1950, 1949, 1995, 1994, 1965)
area_estudo <- c(rep("area2", 2), rep("area1", 8))
casa <- c("a", "a", "b", "b", "b", "c", "c", "d", "d", "e")
lat<- c(-22.92956, -22.92956, -22.93443, -22.93443, -22.93443, -22.93002,
        -22.93002, -22.95848, -22.95848, -22.93266)
long <- c(-46.30385, -46.30385, -46.30895, -46.30895, -46.30895, -46.30291,
        -46.30291, -46.30754, -46.30754, -46.28227)

#criando dataframe para exemplo
exemplo <- data.frame(pessoa, sexo, nascimento, area_estudo, casa, lat,
                      long,
                      stringsAsFactors=FALSE)

#abrindo uma nova janela grafica para melhor visualizacao do plano
cartesiano
x11()

shantay(exemplo, 3, 1997, sexo="f", "area1", 500)
shantay(exemplo, 4, 1997, sexo="NA", "area1", 500)

#Exemplo 2:
#necessario ter salvo o arquivo 'dados_exemplo_help.txt' no diretorio de
trabalho

#lendo planilha de dados para exemplo
dados <- read.table("dados_exemplo_help.txt", header=TRUE, dec=".",
                    sep="\t",
                    as.is = TRUE)

shantay(dados, 8, 1997, sexo="f", "area1", 800)
shantay(dados, 10, 1995, sexo="NA", "area1", 500)
```

Código da Função

```
shantay <- function(dados, n.amostra, ano.nasc, sexo, area.estudo, dist.min)
{
  #exibe uma mensagem de introducao:
  cat("\n Funcao shantay: uma funcao para selecionar quem fica.
      \t\t Start your engines and may the best sample... win!\n")
  #CONFERINDO SE ARGUMENTOS FORAM FORNECIDOS
  #confere se o objeto 'dados' é um dataframe. Se nao for, exibe uma
mensagem de erro e para a funcao
  if(class(dados)!="data.frame")
    stop("\n Objeto 'dados' nao e um dataframe.\n")
  #confere se esta faltando o argumento 'n.pessoas' (numero de unidades
amostrais a serem sorteadas). Se estiver faltando, exibe uma mensagem de
```

```
erro e para a funcao
  if(missing(n.amostra))
    stop("\n Numero de unidades amostrais a serem sorteadas nao
fornecido.\n")
  #confere se esta faltando o argumento 'ano.nasc' (ano de nascimento maximo
das pessoas a serem entrevistadas). Se estiver faltando, exibe uma mensagem
de erro e para a funcao
  if(missing(ano.nasc))
    stop("\n Ano de nascimento maximo das unidades amostrais a serem
sorteadas nao fornecido.\n")
  #confere se esta faltando o argumento 'sexo' (sexo das pessoas a serem
sorteadas). Se estiver faltando, exibe uma mensagem de erro e para a funcao
  if(missing(sexo))
    stop("\n Argumento 'sexo' nao fornecido. Argumento recebe 'm'
(masculino), 'f' (feminino) ou 'NA' (ambos os sexos).\n")
  #confere se esta faltando o argumento 'area.estudo' (area de estudo onde
se encontram as unidades amostrais a serem sorteadas). Se estiver faltando,
exibe uma mensagem de erro e para a funcao
  if(missing(area.estudo))
    stop("\n Nome da area de estudo nao fornecido.\n")
  #confere se esta faltando o argumento 'dist.min' (distancia minima entre
as residencias das unidades amostrais). Se estiver faltando, exibe uma
mensagem de erro e para a funcao
  if(missing(dist.min))
    stop("\n Distancia minima entre as localizacoes das unidades amostrais
nao fornecida.\n")
  #TRABALHANDO OS DADOS A PARTIR DOS CRITERIOS DE SELECAO
  #exibe uma mensagem informando o numero de unidades amostrais presentes
nos dados
  cat("\n\t Unidades amostrais presentes nos dados:", nrow(dados))
  #remove 0 (zeros) e NAs do dataframe (dado que se trata de um sorteio com
base em criterios, nao faz sentido sortear uma unidade amostral de que nao
se tem informacoes)
  dados[dados==0] <- NA #convertendo valores 0 para NA
  dados.sem.na <- na.omit(dados) #removendo unidades amostrais com NAs
  #exibe uma mensagem informando o numero de unidades amostrais removidas
por conterem 0 ou NAs
  removidas.NA <- nrow(dados) - nrow(dados.sem.na)
  cat("\n\t Unidades amostrais removidas por conterem 0 (zero) ou NAs:",
removidas.NA)
  #cria um subset apenas com as unidade amostrais que atendam ao criterio
idade minima
  subset1 <- subset(dados.sem.na, dados.sem.na[,3] <= ano.nasc)
  #exibe uma mensagem informando o numero de unidades amostrais removidas
por nao atenderem ao criterio 'ano de nascimento maximo'
  removidas.idade <- nrow(dados.sem.na) - nrow(subset1)
  cat("\n\t Unidades amostrais removidas por terem nascido apos o
requerido:", removidas.idade)
  #cria um novo subset apenas com as pessoas que atendam ao criterio de sexo
selecionado
  #se nao importar o sexo, o novo subset a ser usado se mantem igual ao
```

```
anterior
  if(sexo=="NA")
    subset2 <- subset1
  #se o sexo de interesse for o feminino, o novo subset a ser usado contem
  apenas unidades amostrais em que sexo seja 'f'
  if(sexo=="f")
    subset2 <- subset(subset1, subset1[,2]=="f")
  #se o sexo de interesse for o masculino, o novo subset a ser usado contem
  apenas unidades amostrais em que sexo seja 'm'
  if(sexo=="m")
    subset2 <- subset(subset1, subset1[,2]=="m")
  #exibe uma mensagem informando o numero de unidades amostrais removidas
  por nao atenderem ao criterio 'sexo'
  removidas.sexo <- nrow(subset1) - nrow(subset2)
  cat("\n\t Unidades amostrais removidas por nao atenderem ao criterio sexo:
", removidas.sexo)
  #depois de ter selecionado o as pessoas pelo criterio sexo, cria um novo
  subset apenas com as unidades amostrais que residem dentro da area de estudo
  subset3 <- subset2[subset2[,4]==area.estudo,]
  #exibe uma mensagem informando o numero de unidades amostrais removidas
  por nao atenderem ao criterio 'area de estudo'
  removidas.area <- nrow(subset2) - nrow(subset3)
  cat("\n\t Unidades amostrais removidas por nao atenderem ao criterio area
de estudo: ", removidas.area)
  #CALCULANDO AS DISTANCIAS ENTRE UNIDADES AMOSTRAIS
  #confere se o pacote 'sp' esta instalado e ativo. Se nao estiver, para a
  funcao e exibe uma mensagem de erro. Se sim, carrega o pacote 'sp'
  if(!require(sp))
    stop("\n\nNecessario ter o pacote 'sp' instalado.\n\n")
  else
    library("sp")
  #calcula uma matriz de distancia entre as unidades amostrais em
  quilometros e converte em metros
  dist <- (spDists(as.matrix(subset3[,6:7]),longlat=TRUE)*1000)
  #atribui o nome das colunas da matriz de distancia igual ao id das
  unidades amostrais. Importante para a indexacao posterior
  colnames(dist) <- subset3$peessoa
  #converte a matriz de distancias em dataframe
  dist.df <- as.data.frame(dist)
  #junta o dataframe das unidades amostrais com o dataframe das distancias
  conjunto <- cbind(subset3, dist.df)
  #SELECIONANDO UNIDADES AMOSTRAIS COM BASE NA DISTANCIA MINIMA ENTRE ELAS
  #cria um vetor para receber as unidades amostrais sorteadas
  pessoas.sort <- rep(NA, n.amostra)
  #sorteia a primeira unidade amostral e atribui ao vetor de unidades
  amostrais
  pessoas.sort[1] <- sample(conjunto[,1], size=1)
  #cria um loop para sortear as demais unidades amostrais respeitando o
  criterio de distancia
  for(i in 2:n.amostra)
```

```

{
  #apos uma nova unidade amostral ser sorteada, confere se a distancia
entre ela e a sorteada anteriormente e maior ou igual ao indicado pelo
usuario
  conjunto <- subset(x=conjunto, conjunto[,pessoas.sort[(i-1)]] >=
dist.min)
  #testa se o numero de unidades amostrais disponiveis para o sorteio
possuem uma quantidade suficiente para atender ao numero minimo de pessoas a
serem sorteadas. Se nao for, para a funcao e exibe uma mensagem de erro
  if(nrow(conjunto) == 0)
    stop("\n\nNumero de unidades amostrais disponiveis com base nos
criterios fornecidos nao suficiente. Tente novamente usando outros criterios
de selecao, como menos unidades amostrais ou uma distancia minima entre suas
localizacoes menor.\n\n" )
  #a medida que uma nova unidades amostral sorteada atende ao criterio de
distancia, ela e atribuida ao vetor de unidades amostrais final
  pessoas.sort[i] <- sample(conjunto[,1], size=1)
}
#PREPARANDO OUTPUTS: CONSOLE, ARQUIVO TXT E PLANO CARTESIANO
#cria um dataframe baseado no objeto de input apenas com as linhas das
unidades amostrais selecionadas
retorno <- subset3[subset3$peessoa%in%pessoas.sort,]
#calcula uma matriz de distancia entre as unidades amostrais selecionadas
em quilometros e ja converte para metros
dist2 <- (spDists(as.matrix(retorno[,6:7]),longlat=TRUE)*1000)
#atribui valor NA para todas os valores dessa matriz iguais a 0 e exibe
uma mensagem informando as distancias maxima e minima entre as unidades
amostrais selecionadas
dist2[dist2==0] <- NA
cat("\n\n\t", round(max(dist2, na.rm=T)), "m = maior distancia encontrada
entre as unidades amostrais selecionadas.\n\t", round(min(dist2, na.rm=T)),
"m = menor distancia encontrada entre as unidades amostrais
selecionadas.\n")
#plota o grafico com a latitude e a longitude
plot(retorno$long, retorno$lat, xlab = "Longitude", ylab = "Latitude",
cex.lab=1.2, cex=1.2, pch=19, col="black")
#"transformando" grafico em um plano cartesiano
#adiciona o titulo ao plano cartesiano
title(main="Localizacao das unidades amostrais selecionadas", line=+3)
axis(3) #adiciona as longitudes na parte superior do plano cartesiano
axis(4) #adiciona as latitudes na lateral direita do plano cartesiano
grid(col="black") #cria o grid no plano cartesiano
cat("\n") #adiciona o espaco de uma linha para mostrar o resultado (objeto
'retorno'). Finalidade visual apenas
#cria o arquivo .txt com as unidades amostrais selecionadas
write.table(retorno, file="amostragem.txt", sep="\t", dec=".")
#retorna as unidades amostrais selecionadas
return(retorno)
} #And that's how the story goes: the end! :)
#_____Sao Paulo, 16 de maio
de 2016

```

Arquivos da Função

[Função shantay](#)

[Help da função shantay](#)

[Dados para exemplos do help](#)

Para teste:

[Dados reais para testar a função](#)

[Script para testar a função](#)

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2016:alunos:trabalho_final:lteixeira:start 

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