



Alessandra Rocha

Olá, sou Alessandra, veterinária e trabalho com animais marinhos, em especial, cetáceos.

[exercicio1_usp.r](#)

[exerc4.r](#)

[graficosa.r](#)

[7aF](#)

[7bF](#)

[wikisimulacao.r](#)

[109wikif.r](#)

1.Trabalho final

1.1.Plano A - Cálculo da distância da costa a partir de coordenadas geográficas de interesse

Contextualização

O conhecimento da distancia onde ocorrem os animais marinhos e a atividade pesqueira, tendo como referência a linha de costa, é importante para a conservação da fauna marinha e o manejo da pesca. Por exemplo, a distância da costa pode ser uma variável preditora da ocorrência de cetáceos (do Amaral *et al.* 2018) e algumas pescarias. A sobreposição entre pequenos cetáceos e atividade pesqueira pode ocasionar interações negativas como a captura acidental de golfinhos em redes de pesca, um problema mundial de conservação que afeta várias espécies de cetáceos(Leeney *et al.* 2008). Recentemente uma Instrução Normativa (IN nº12/2012), com o intuito de ordenar a pesca de emalhe da região sudeste e sul do Brasil, estabeleceu a proibição da pesca desta modalidade à 1 milha náutica de distância da costa para todas as embarcações motorizadas e à 4 milhas náuticas para embarcações de arqueação bruta maior que 20. Neste caso, a aferição da distância da costa dos lances de pesca pode ser útil para fins de fiscalização da atividade pesqueira.

Nesse sentido, a função proposta pretende calcular as distâncias da costa de pontos de interesse, através de arquivos do tipo *shapefile*¹, criar um data frame com essas distâncias e gerar um mapa contendo a linha de costa e os pontos de interesse. Dentro da função, os objetos contendo os *shapefiles* serão transformados em unidades métricas através do EPSG 5641², que é um código que  atua em toda a América Latina (<https://epsg.io/5641>). Figura 1. EPSG 5641

¹ *shapefile* é um arquivo para dados vetoriais, desenvolvido pela empresa ESRI e é composto por três arquivos que possuem as extensões: *.shp*: armazena entidades geométricas do objeto; *.shx*: armazena o índice de entidade geométricas; *.dbf*: armazena a tabela de atributos dos objetos. Além dessas extensões, é comum encontrar a extensão *.prj* que contém as informações do sistema de coordenadas.

² Uma fonte de dados classificada como espacial deve possuir um Sistema de Coordenadas (SRC) que

está organizado em Sistema de Coordenadas Geográficas (ou Geodésicas) ou Sistema de Coordenadas Planas (ou Projetadas). Esse sistema deve ser verificado antes de executar qualquer ação no sistema de informação geográfica. Após a escolha do sistema de referência de coordenadas, o passo seguinte é a escolha do modelo da Terra ou Datum. Código EPSG é uma padronização dos Sistemas de Referência de Coordenadas do mundo formalizada pela organização *European Petroleum Survey Group* (EPSG). Uma projeção de qualquer parte do globo pode ser identificada através do padrão EPSG. Na nossa função utilizaremos o sistema de coordenadas planas (EPSG=5641) para calcularmos a distância da costa em unidades métricas. Se utilizássemos o sistema de coordenadas geográficas, calcularíamos a distância em graus.

Planejamento da função O usuário deve baixar os seguintes pacotes: “sf”, “mapview”, “tidyverse”.

Entrada: dist.costa(x,y)

- x = objeto contendo o arquivo *shapefile* da linha de costa
- y = objeto contendo o arquivo *shapefile* da camada de coordenadas geográficas de interesse

Pseudo-código

1. cria um objeto com um mapa contendo a linha de costa e as coordenadas plotadas
2. define o EPSG: transforma o *shapefile* da linha de costa em unidades métricas.
3. define o EPSG: transforma o *shapefile* da camada de pontos em unidades métricas.
4. cria um objeto com o cálculo das distâncias (em metros) das coordenadas à linha de costa em dataframe.
5. converte a camada contendo as coordenadas em dataframe
6. une o dataframe das distâncias com o dataframe da camada das coordenadas

Saída

- dataframe contendo a camada de pontos e as distancias calculadas em relação a linha de costa
- mapa de visualização contendo as coordenadas e a linha de costa.

1.2.Plano B - Conversão de coordenadas geográficas GMS em coordenadas decimais

Contextualização

A conversão de coordenadas geográficas GMS (grau, minuto, segundo) em coordenadas decimais é muito comum e fundamental para importar os pontos geográficos para softwares de SIG (Sistema de Informações Geográficas).

A conversão acontece por meio da fórmula:

Graus decimais = ¹⁾

¹⁾

segundos / 60) + minutos) + graus A função proposta pretende fazer este cálculo e criar um dataframe com as coordenadas convertidas. Para isso, a tabela de entrada deve possuir três colunas nomeadas como “grau”, “min” e “seg” contendo as respectivas informações nas linhas.

Planejamento da função **Entrada:** convert.dec(tabela) *tabela = um dataframe contendo 3 colunas nomeadas como: “graus”, “min”, “seg”. **Verificando os parâmetros**

- a tabela possui uma coluna chamada “graus”? Se não, escreve: “a tabela não possui uma

coluna chamada "graus"

- a tabela possui uma coluna chamada "min"? Se não, escreve: "a tabela não possui uma coluna chamada "min"
- a tabela possui uma coluna chamada "seg"? Se não, escreve: "a tabela não possui uma coluna chamada "seg"

Pseudo-código

1. cria um objeto que exclui os valores NA do cálculo
2. cria um objeto com a conversão da coordenada GMS em graus decimais
3. cria um dataframe com as coordenadas em GMS e as coordenadas em graus

decimais calculadas **Saída**

- dataframe com as três colunas contendo os "graus", "min" e "seg" e uma quarta coluna contendo as coordenadas em graus decimais calculadas.

2.Referências do Amaral K, Danilewicz D, Zerbini A, Di Benedetto AP, Andriolo A, Alvares DJ, Secchi E, Ferreira E, et al. 2018. Reassessment of the franciscana *Pontoporia blainvillei* (Gervais & d'Orbigny, 1844) distribution and niche characteristics in Brazil. *Journal of Experimental Marine Biology and Ecology* (508)1:12. <https://doi.org/10.1016/j.jembe.2018.07.010> Leeney RH, Amies R, Broderick AC, Witt MJ, Loveridge J, Doyle J, Godley BJ. 2008. Spatio-temporal analysis of cetacean strandings and bycatch in a UK Wsheries hotspot. *Biodivers Conserv* (2008) 17:2323–2338. DOI 10.1007/s10531-008-9377-5

Função dist.costa()

```
===== Função "dist.costa"===== Autora: Alessandra Rocha

#Carregando os pacotes necessários#
library(sf)
library(tidyverse)
library(mapview)

#criando os objetos x e y#
x<-st_read("drive/pasta do computador/arquivo.shp")
y<-st_read("drive/pasta do computador/arquivo_de_coordenadas.shp")

#####Função#####
dist.costa<-function(x, #objeto contendo a camada vetorial linear de
referência
                        y, #objeto contendo a camada de coordenadas geográficas
                        nboot=1000 #número de simulações, por padrão será 1000
valores simulados
                        )
{
  #verificando parâmetros#
  if(class(x)!= c("sf", "data.frame") && class(y)!=c("sf", "data.frame"))
  {stop("o vetor de dados não é da classe sf")} #verificando a classe dos
objetos x e y que devem ser "sf" "data.frame"

  if(nboot<=0|as.integer(nboot)!=nboot) #verifica se o número de
simulações indicado é um número inteiro maior que zero
```

```
{
  nboot=1000
  warning("nboot deve ser um número inteiro e maior que zero,
          função rodou com o nboot padrão de 1000 simulações") #se o
número em nboot não for inteiro e maior que zero, a função rodará com o
número padrão de 1000 valores e mostrará essa mensagem de 'warning'
}
sn<-readline("quer visualizar o mapa? (s/n)") #pergunta ao usuário, no
console, se quer visualizar o mapa ou não
if(sn=="s")#se o usuário quiser visualizar o mapa, deverá digitar a letra
"s" no console e a função irá prosseguir
{
  #início da função#
  w<-mapview(x, color="grey")+y} #gera o mapa
  x%>% st_transform(st_crs(5641)) #modifica o Datum do objeto x para
UTM,sistema de coordenadas planas, para realizar o cálculo das distâncias em
metros
  y%>% st_transform(st_crs(5641))#modifica o Datum do objeto y para UTM,
sistema de coordenadas planas, para realizar o cálculo das distâncias em
metros
  dist<- as.data.frame(st_distance(y,x)) #calcula as distâncias e cria um
objeto com o data.frame das distâncias
  tab<-data.frame(y) #cria um data.frame da camada vetorial y (coordenadas
geográficas)
  tab.dist<-data.frame(tab,dist) #junta o data.frame com as distâncias e o
data.frame com as coordenadas
  media.dist<-mean(tab.dist$st_distance.y..x.) #calcula a média das
distâncias calculadas
  #preparando para o ciclo for() - bootstrap#
  nboot=nboot #atribui o objeto nboot com os valor do padrão ou o valor que
o usuário escolher para a simulação do bootstrap
  boot.sim<-rep(NA, nboot) #cria um objeto com NAs, de mesmo comprimento que
o nboot
  boot.sim[1]<-media.dist #coloca na posição 1 do objeto de valores NA, a
média das distâncias calculada
  for(i in 2:nboot){ #cria um ciclo for() que começa na posição 2, pois na
posição 1 já está a média das distâncias
    com comprimento nboot
    boot.sim[i]<-mean(sample(tab.dist$st_distance.y..x., replace=TRUE))#faz
uma amostragem com reposição dos valores de distância encontrados e calcula
a média desses valores
  }
  z<-hist(boot.sim) #gera um histograma com os valores de distância média
simulados
  abline(v=boot.sim[1], lty=2, col="red") #coloca uma linha tracejada
vermelha no histograma, indicando a média da distância dos valores reais
  q<-quantile(boot.sim, prob=c(0.05, 0.95)) #calcula os quantis para o
intervalo de confiança de 95%
  #preparando os resultados da função#
  if(sn=="s"){ #se quiser o mapa
```

```

    final<-list(tab.dist,z,q, w)} #o objeto 'final' do return conterá o mapa
    if(sn=="n"){ #se não quiser o mapa
      final<-list(tab.dist,z,q)} #o objeto 'final' do return não retornará o
mapa
    return(final) #apresenta os resultados do cálculo de distâncias, o
histograma de valores simulados com a média da distância de valores
calculados e o valor de média dos dados, os quantis do intervalo de
confiança do bootstrap e um mapa caso o usuário queira visualizar o mapa.
#####fim da função#####
}

```

Help função dist.costa

dist.costa package:unkown R Documentation

Cálculo da distância da costa a partir de coordenadas geográficas de interesse

Description:

Função para calcular distâncias euclidianas entre coordenadas geográficas e um shapefile (camada vetorial linear) de referência. As distâncias são calculadas em metros.

A função fornece a opção de um mapa, representando as coordenadas e o shapefile de referência.

O mapa possui opções de 'layout' no menu do canto esquerdo. Um dataframe contendo as distâncias calculadas é fornecido, assim como a média das distâncias e um histograma contendo valores simulados e o cálculo do intervalo de confiança de 95%. A função é especialmente útil para usuários que desejam conhecer distâncias lineares, por exemplo, de pontos (coordenadas geográficas), tomando-se como referência uma linha de costa.

Usage:

```
dist.costa (x,y, nboot=1000)
```

Arguments:

x : objeto da classe "sf" "data.frame"

y : objeto da classe "sf" "data.frame"

nboot=1000: número de simulação para o bootstrap com o padrão de 1000 valores simulados.

Details:

Para usar esta função, os pacotes "sf", "tidyverse" e "mapview" devem estar instalados e carregados. Para ler o shapefile, pode-se usar a função `st_read` (e.g. Examples). A ordem das variáveis faz diferença, sendo a primeira variável(x) o shapefile de referência para o cálculo da distância e a segunda variável (y), o shapefile contendo as coordenadas geográficas

de interesse. O shapefile de referência (x) deve ser uma camada vetorial linear. A função funciona melhor na região da América Latina pois o código EPSG usado é o 5641.

É possível optar pela visualização do mapa ou não. Momentos após o usuário rodar a função, virá a pergunta sobre a visualização do mapa, que deve ser respondida com a letra 's'

se deseja visualizar o mapa e com a letra 'n' se não deseja visualizar o mapa. A função retorna a distância média, representada por uma linha tracejada, dentro de um histograma de 'nboot' valores médios simulados e o intervalo de confiança de 95%, calculados por bootstrap.

Value:

Mapa de visualização contendo a camada com as coordenadas geográficas de interesse e o shapefile de referência.

Dataframe contendo as distâncias calculadas em unidades métricas.

Histograma contendo a indicação (linha tracejada) da média dos valores das distâncias calculadas, os valores médios simulados de distância e o intervalo de confiança destes valores, calculados por bootstrap.

Warning:

Se x e y não forem arquivos da classe "sf" "data.frame" a função não é executada.

O mapa só é gerado se o usuário quiser, clicando a letra "s" depois de se executar a função, no console.

Se o valor escolhido para o nboot não for um número inteiro maior que zero, a função rodará com o número de simulações do padrão da função (1000 valores).

A função funciona melhor e tem bastante utilidade quando se usa uma camada vetorial linear como shapefile de referência.

Author(s):

Alessandra Rocha
email: emaildalessandra@gmail.com

Examples

```
linha_costa<-st_read("C:/Users/R/linha_rs.shp")  
pontos_n<-st_read("C:/Users/R/linha_rs.shp")  
  
distancias<-dist.costa(linha_costa, pontos_n, nboot=2000) #se quiser 2000  
valores simulados no bootstrap
```

```
distancias.1<-dista.costa(linha_costa, pontos_n) #se quer utilizar o número de simulações do padrão (1000 valores)
```

Link de arquivos para se utilizar como exemplo no help da função 'dist.costa':
[arquivos_exemplos_r.rar](#)

Taio

Olá, Alessandra! A **proposta B** é simplesmente a aplicação de uma fórmula. Sugiro que vc siga com a **proposta A**. Lá vai algumas comentários:

- Eu não sei o que é um **shapefile** nem um código **EPSG 5641**. Sugiro que você coloque um quadrinho explicativo sobre os conceitos específicos de sua área. Com um quadrinho explicativo, a sua função poderia ser útil para pessoas de outras áreas.
- Eu creio que sua **proposta A** pode resultar em uma função legal se ela tiver mais opções para o usuário. Por exemplo: e se o usuário quiser um mapa com paletas de cores X ou Y? Ou um mapa com as escalas Z? Ou usar um shapefile dos formatos X, Y ou Z (não sei se existe)?
- Reveja a página http://ecologia.ib.usp.br/bie5782/doku.php?id=bie5782:01_curso_atual:alunos:trabalho_final:start. Veja que umas das principais ideias é que sua função seja *uma tarefa geral ou aplicável recorrentemente* e que seja *uma tarefa factível, mas que ao mesmo tempo seja um desafio estimulante e que te tire da zona de conforto!* .

Bom tRabalho 😊 Lucas F. Nascimento (Taio)

Resposta Alessandra Rocha

Olá Lucas, Obrigada pelas sugestões. Segui com o Plano A. Acho que o cálculo da distância se encaixa em uma tarefa aplicável recorrentemente e expliquei alguns exemplos na contextualização da função. Incluí uma explicação sobre shapefile e o código EPSG. Aprimorei a função, incluindo um ciclo for() para a estimativa de intervalo de confiança da média das distâncias por bootstrap e um histograma com estes valores. Além disso, coloquei uma opção ao usuário para visualizar o mapa gerado ou não. A ideia do mapa é visualizar os pontos e o shapefile de referência. Clicando-se nos pontos, é possível resgatar toda a informação contida neles (função mapview). Eu gostei do trabalho, acho que será bastante útil. Abração! Daqui a pouco vou inserir o 'help' e a função. Obrigada pela ajuda! PS.: minha página está desconfigurando, não sei que está acontecendo. Abraços!

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

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2019:alunos:trabalho_final:emaildaalessandra:start&rev=1597223093

Last update: 2020/08/12 06:04