

FUNÇÃO "biomap"

```
##### FUNCAO "biomap" - THIAGO F. RODRIGUES 2018
#####

biomap <- function(dados, grid.size=5) # Cria a funcao biomap com os
argumentos dados e grid.size com padrao de tamanho 5.
{
  ## LEITURA DO PACOTE NECESSARIO PARA RODAR A FUNCAO
  require("raster") # Inicia o pacote raster
  require("sp") # Inicia o pacote sp
  ## VERIFICANDO PARAMETROS
  ## ARGUMENTO - dados
  if(class(dados)!="data.frame") # Verifica se os dados utilizados
constitui um data.frame().
  {
    stop("\n\n ATENCAO: os dados devem estar dentro de um data.frame
\n\n") # Senao, interrompe a funcao e exibe a mensagem ao usuario.
  }
  if(length(dados)!=3) # Verifica se os dados do data.frame contem 3
colunas.
  {
    stop("\n\n ATENCAO: o data.frame deve possuir 3 colunas:
      coluna 1 (longitude)= longitute em graus decimais
      coluna 2 (latitude)= latitude em graus decimais
      coluna 3 (especies)= nome das especies. \n\n") # Senao,
interrompe a funcao e exibe a mensagem ao usuario.
  }
  if(class(dados[,1])!="numeric") # Verifica se a primeira coluna do
data.frame é da classe numeric.
  {
    stop("\n\n ATENCAO: a coluna 1 deve corresponder a LONGITUDE e ser
da classe 'numeric' \n\n") # Senao, interrompe a funcao e exibe a mensagem
ao usuario.
  }
  if(class(dados[,2])!="numeric") # Verifica se a segunda coluna do
data.frame é da classe numeric.
  {
    stop("\n\n ATENCAO: a coluna 2 deve corresponder a LATITUDE e ser da
classe 'numeric' \n\n") # Senao, interrompe a funcao e exibe a mensagem ao
usuario.
  }
  if(any(is.na(dados[,c(1:3)]))) # Verifica se existem NAs nos dados.
  {
    dados <- (na.omit(dados)) # Se positivo, as linhas com NAs serão
omitidas no data.frame.
    message("\n Valores NAs foram removido(s) do data.frame \n") # Exibe
uma mensagem avisando o usuario da exclusao de linhas.
  }
}
```

```
}  
## VERIFICANDO PARAMETROS  
## ARGUMENTO - grid.size  
if(grid.size!=round(grid.size, 0)) # Verifica se grid.size é um numero  
inteiro.  
{  
  stop("\n\n ATENCAO: grid.size deve ser um NUMERO INTEIRO. \n\n") #  
Senao, interrompe a funcao e exibe a mensagem ao usuario.  
}  
if(grid.size<1 | grid.size>10) # Verifica se grid.size esta no intervalo  
adequado para a funcao.  
{  
  stop("\n\n ATENCAO: grid.size deve estar no intervalo 1 <= grid.size  
<= 10 \n\n") # Senao, interrompe a funcao e exibe a mensagem ao usuario.  
}  
if(grid.size<=8) # Se o usuario definir um tamanho para o grid.size <=  
8, mostra a mensagem abaixo.  
{  
  message("\n ATENCAO: A FUNCAO DEMORARA UM POUCO; PROCESSAMENTO  
INTENSO PARA A CONSTRUÇÃO DE GRID!!! \n") # Mensagem para tranquilizar o  
usuario pela demora esperada do processamento da funcao.  
}  
## PARTE 1 - GRAFICO COM A DISTRIBUICAO DOS LOCAIS DE ESTUDO NO BRASIL  
brasil <- getData('GADM', country='BRA', level=1) # Faz o download de um  
shapefile do Brasil disponível online - inclui todos os Estados (level=1).  
coords <- dados[,c(1,2)] # Cria objeto da classe data.frame contendo a  
longitude e latitude.  
sp1 <- SpatialPoints(coords) # Transforma objeto coords em coordenadas  
geograficas (mostra ao R especificamente que se trata de coordenadas  
espaciais).  
sp2 <- remove.duplicates(sp1, zero = 0.0) # Remove todos os valores  
duplicados das coordenadas geograficas.  
proj4string(sp2) <- proj4string(brasil) # Atribui a mesma projecao do  
objeto Brasil as coordenadas geograficas.  
##Criacao do primeiro grafico.  
x11() # Aciona o primeiro dispositivo de tela.  
plot(brasil, col="gray97", axes = TRUE, las=1, main="Distribuição dos  
Locais de Estudo") # Plota shapefile Brasil com fundo cinza 70% e eixos  
verdadeiros.  
points(sp2, col="blue", pch=20) # Inclui as coordenadas geograficas de  
cor azul ao mapa do Brasil.  
scalebar(1500, xy=c(-45.17332,-31.15363), type="bar", divs=4,  
lonlat=TRUE, below="km") # Inclui uma escala em quilometros no grafico.  
##PARTE 2 - GRAFICO COM A QUANTIDADE DE LOCAIS DE ESTUDO POR CELULA DO  
GRID  
##Criacao do primeiro grid.  
g1 <- raster(xmn=-73.98971, ymn=-33.74708, xmx=-34.04263, ymx=5.264878)  
# Cria raster vazio com dimensoes especificas.  
res(g1) <- grid.size # Define a resolucao, isto e, o tamanho do grid.  
g1[] <- 0 # Preenche celulas vazias do raster com zeros.
```

```
proj4string(g1) <- proj4string(brasil) # Atribui a mesma projecao do
objeto Brasil ao grid.
tab <- table(cellFromXY(g1, sp2)) # Conta quantas coordenadas existem em
cada celula do grid.
g1[as.numeric(names(tab))] <- tab # Adiciona os valores contados como
número ao grid.
##Corte do grid apenas para a regioa do Brasil.
g1.ras <- rasterize(brasil, g1, getCover=TRUE) # Transforma somente o
poligono do Brasil em um raster.
g1.ras[g1.ras==0] <- NA # Substitui 0 por NAs (area sem sobreposicao ao
mapa do Brasil).
values(g1)[is.na(values(g1.ras))] <- NA # Adiciona NAs ao grid (objeto
g1).
##Criacao do segundo grafico.
x11() # Aciona o segundo dispositivo de tela.
plot(g1, col=grey(16.5:4/17, alpha = 0.85), xaxt="n", yaxt="n",
main="Quantidade de Estudos") # Plota grid numa escala de cinza com um pouco
de transparencia sem informacao dos eixos.
plot(brasil, border="gray70", axes=TRUE, add=TRUE, las = 1,
xlim=c(-73.98971,-34.04263), ylim=c(-33.74708,5.264878)) # Adiciona mapa do
Brasil com linhas cinzas, eixos verdadeiros e legenda do eixo y na
horizontal com limites especificos.
plot(rasterToPolygons(g1), add=TRUE, border="gray80", lwd=1) #
Transforma o raster em poligono(shapefile) para adicionar borda em cada
celula.
if(grid.size==1) # Se tamanho do grid.size for igual a 1. Adiciona um
tamanho menor de letra.
{
  text(g1, col="black", cex=0.4, font=2, fun=function(x){x!=0}) #
Adiciona texto no mapa com a quantidade de locais por celula com tamanho
reduzido de letra (apenas celulas que possuem informacao).
}else # Senao, mantem um tamanho de letra padrao para os demais tamanhos
de grid.size.
{
  text(g1, col="black", cex=0.7, font=2, fun=function(x){x!=0}) #
Adiciona texto no mapa com a quantidade de locais por celula com tamanho
padrao de letra (apenas celulas que possuem informacao).
}
axis(1, at=c(-70,-60,-50,-40), labels = c("70°W", "60°W", "50°W",
"40°W")) # Adiciona o eixo x no grafico.
axis(2, at=c(0,-10,-20,-30), labels = c("0°S", "10°S", "20°S", "30°S"))
# Adiciona o eixo y no grafico.
##PARTE 3 - GRAFICO COM A QUANTIDADE DE ESPECIES (RIQUEZA) POR CELULA DO
GRID
##Criacao do segundo grid.
g2 <- raster(xmn=-73.98971, ymn=-33.74708, xmx=-34.04263, ymx=5.264878)
# Cria raster vazio com dimensoes especificas.
res(g2) <- grid.size # Define a resolucao, isto e, o tamanho do grid.
g2[] <- 0 # Preenche as celulas vazias do raster com zeros.
proj4string(g2) <- proj4string(brasil) # Atribui a mesma projecao do
objeto Brasil ao grid.
```

```
sp3 <- SpatialPointsDataFrame(dados[,c(1,2)], data.frame(spp=dados[,3]))
# Cria data.frame espacial com as coordenadas(sp3) seguido de um data.frame
com as respectivas especies por coordenada.
rich <- rasterize(sp3, g2, field='spp', fun=function(x, ...)
length(unique(na.omit(x)))) # Quantifica o total de especies unicas em cada
celula do grid, considerando todas as coordenadas em cada celula.
##Corte do grid apenas para a regioao do Brasil.
g2.ras <- rasterize(brasil, g2, getCover=TRUE) # Transforma somente o
poligono do Brasil em um raster.
g2.ras[g2.ras==0] <- NA # Substitui 0 por NAs (area sem sobreposicao ao
mapa do Brasil).
values(g2)[is.na(values(g2.ras))] <- NA # Adiciona NAs ao grid (objeto
g2).
values(g2)[!is.na(values(rich))] <- values(rich)[!is.na(values(rich))] #
Adiciona valores calculados de riqueza ao grid (objeto g2).
##Criacao do terceiro grafico.
x11() # Aciona o segundo dispositivo de tela.
plot(g2, col=grey(16.5:4/17, alpha = 0.85), xaxt="n", yaxt="n",
main="Quantidade de Espécies (Riqueza)") # Plota grid numa escala de cinza
com um pouco de transparencia sem informacao dos eixos.
plot(brasil, border="gray70", axes=TRUE, add=TRUE, las = 1,
xlim=c(-73.98971,-34.04263), ylim=c(-33.74708,5.264878)) # Adiciona mapa do
Brasil com linhas cinzas, eixos verdadeiros e legenda do eixo y na
horizontal com limites especificos.
plot(rasterToPolygons(g2), add=TRUE, border="gray80", lwd=1) #
Transforma o raster em poligono(shapefile) para adicionar borda em cada
celula.
if(grid.size==1) # Se tamanho do grid.size for igual a 1. Adiciona um
tamanho menor de letra.
{
  text(g2, col="black", cex=0.4, font=2, fun=function(x){x!=0}) #
Adiciona texto no mapa com a quantidade de especies por celula com tamanho
reduzido de letra (apenas celulas que possuem informacao).
}else # Senao, mantem um tamanho de letra padrao para os demais tamanhos
de grid.size.
{
  text(g2, col="black", cex=0.7, font=2, fun=function(x){x!=0}) #
Adiciona texto no mapa com a quantidade de especies por celula com tamanho
padrao de letra (apenas celulas que possuem informacao).
}
axis(1, at=c(-70,-60,-50,-40), labels = c("70°W", "60°W", "50°W",
"40°W")) # Adiciona o eixo x no grafico.
axis(2, at=c(0,-10,-20,-30), labels = c("0°S", "10°S", "20°S", "30°S"))
# Adiciona o eixo y no grafico.
##PARTE 4 - INFORMACOES QUANTIFICADAS NOS MAPAS E DISPONIVEIS AO USUARIO
d1 <- data.frame(grid_id=paste("id", sep="_", 1:length(g1)),
coordinates(g1), n_estudos=g1[,], n_especies=g2[,]) # Cria data.frame com 4
colunas resultante do que é mostrado nos graficos.
names(d1)[2:3] <- c("longitude", "latitude") # Renomeia as colunas 2 e 3
para longitude e latitude.
```

```
lc <- na.omit(d1) # Retira linhas com NAs
d2 <- data.frame(
  cel_com_estudo=length(lc$n_estudos[lc$n_estudos!=0]), #
  cel_sem_estudo=length(lc$n_estudos[lc$n_estudos==0]), #
  prob_estudo=round(length(lc$n_estudos[lc$n_estudos!=0])/length(lc$n_estudos)
,2), # Quantidade de celulas com estudo/total de celulas do grid.
  cel_max_estudo=max(lc$n_estudos), # Celula com o maximo
  cel_max_especie=max(lc$n_especies)) # Celula com o
maximo valor encontrado de numero de especies(riqueza).
resultado <- list(info_mapa=d1,lacuna_de_conhecimento=d2) # Cria uma
lista com os dois data.frames acima, além de deixar disponivel esse objeto
no "Global Environmet".
return(resultado) # Mostra no console o objeto resultado.
}
```

FIM - FUNCAO "biomap" - THIAGO F. RODRIGUES 2018
#####

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

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2018:alunos:trabalho_final:thiagorodrigues:biomap

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