

Carolina Siliprandi



Doutoranda em Oceanografia Biológica, pelo Instituto Oceanográfico da USP.

Título da minha tese: *"Biodiversidade e estrutura da ictiofauna da baía do Araçá, São Sebastião, litoral norte do Estado de São Paulo, a partir de pontos homólogos"*. Orientadora: Profa. Dra. Carmen L. D. B. Rossi-Wongtschowski.

Meus exercícios no Wiki [exec](#)

Proposta de Trabalho Final

Principal

A ecomorfologia ou morfologia ecológica é baseada na premissa de que a ecologia de um organismo está relacionada à sua forma, ou seja, existe forte interação entre os fatores endógenos e as características do meio em que vivem, permitindo o estudo da diversidade (morfológica + ecológica). Estes estudos avaliam a estrutura de comunidades de peixes utilizando a metodologia de pontos homólogos ou landmarks, baseados em fotografias padronizadas do lado esquerdo do corpo. O número de fotografias utilizadas é proporcional à abundância de cada espécie na amostra coletada, com diferentes artes de pesca. Para selecionar quantas e quais fotografias devem ser utilizadas nas análises, é necessário o conhecimento básico da composição em comprimento total (CT, mm), peso total (PT, g) bem como a abundância por espécie em cada estação do ano (primavera, verão, outono e inverno).

Os dados de entrada compreendem um dataframe com variáveis quantitativas (comprimentos e pesos) por estação do ano (primavera, verão, outono e inverno) nas colunas e, nas linhas, as espécies (n=33). A função terá um argumento onde é possível especificar a estação do ano desejada.

Para isso, uma função que construa:

um dataframe com media, mediana, desvio padrão, mínimo, máximo, 1º quartil, 3º quartil das variáveis comprimento (CT) e peso (PT); um dataframe composto pelas variáveis número de indivíduos, abundância, biomassa, número de fotos (5% da abundância total) que devem ser utilizadas; um dataframe com os valores dos índices clássicos (Margalef, Shannon-Wiener (H'), Pielou (J')) aplicados em ecologia de comunidades de peixes.

A etapa dos índices clássicos servirá de comparação, já que estudos sobre estrutura de comunidades a partir de pontos homólogos são realizados com estatística multivariada, utilizando as distâncias entre os 27 pontos marcados no corpo dos peixes para realizar as análises de morfologia geométrica.

Plano B

Em adição à análise morfofuncional dos peixes, alguns índices baseados na análise morfométrica de otólitos sagittae (estruturas calcárias do ouvido interno dos peixes teleósteos) também possibilitam a avaliação da diversidade das assembleias de peixes, em diferentes escalas temporais. Medidas

lineares (comprimento, altura, perímetro e área) são utilizadas para calcular os descritores de forma tridimensionais.

Os dados de entrada compõem um dataframe com as variáveis quantitativas comprimento, altura, perímetro e área dos otólitos e a variável qualitativa estação do ano nas colunas e, nas linhas, as espécies. A função terá um argumento onde é possível especificar a estação do ano desejada. Uma função que construa boxplots das variáveis quantitativas por espécie; um dataframe contendo a estatística básica descritiva das variáveis quantitativas; um dataframe com os resultados dos índices Fator de Forma, Circularidade e Razão de Aspecto; e, boxplots com os valores dos por espécie.

Carolina, achei sua proposta A um pouco cheia. Se entendi bem, para cada espécie você terá 2 histogramas, um boxplot e 2 tabelas. Se forem várias espécies, a saída da função vai ficar gigantesca e talvez difícil de analisar, será que não? Além disso, você quer o número de fotos para cada espécie e os índices para a comunidade. Sugiro que você reduza um pouco, pensando no que realmente quer que a função faça (ela não precisa ser multitarefas!). Talvez ajude definir qual nome você vai dar para essa função.

A proposta B segue a mesma linha, mas tem apenas a parte descritiva dos dados. Nos dois casos, você tem claro o que será a saída da função, só faltou definir o que serão os dados de entrada.

Reescreva as propostas, incluindo os dados de entrada. Leia as recomendações para elaboração da proposta ([Trabalho final](#)), pode te ajudar!

— *Sheina*

Comentário da proposta arrumada Carolina, você manteve bastante coisa na proposta! Não é um problema, vc só tem que se certificar de que isso não vai complicar demais. Na proposta A, você ainda fica com uma saída bem extensa (se eu entendi corretamente, você terá 104 histogramas de comprimento, um para cada espécie, e os dataframes). No final, você terá um resumo dos dados para cada estação, o que parece útil pra vc, eu só acho que a parte dos histogramas pode ficar difícil de analisar. A proposta B também, serão 104 conjuntos de boxplots. Se for manter tudo, sugiro que teste com um conjunto de dados com poucas espécies, para ir sentindo como fica. Boa sorte!

— *Sheina*

Código da Função fisheco

```
fisheco <- function(fishfauna,
estacao=c("anual","inverno","outono","primavera","verao"), rmNA=T)
{
  if(rmNA==TRUE)
  {
    fishfauna=(na.omit(dados)) ## remove os NA's para que estes
    não interfiram nos valores resultantes dos índices ecológicos.
    dif=length(dados)-length(fishfauna)
    cat("\t", dif,"valores NA excluídos\n")
  }
  else
  {
    fishfauna=x
  }
  if(estacao=="anual"||is.null(estacao))
  {
    media.CT <- tapply(X= fishfauna$CT, INDEX=fishfauna$especie,
FUN=mean,na.rm=T) ##aplica a função média para os elementos do vetor CT,
indexado pelos caracteres do vetor espécie.
    media.PT<- tapply(X=fishfauna$PT, INDEX=fishfauna$especie,
FUN=mean,na.rm=T) ##aplica a função média para os elementos do vetor PT,
indexado pelos caracteres do vetor espécie.
    mediana.CT <- tapply(X=fishfauna$CT, INDEX=fishfauna$especie,
FUN=median,na.rm=T) ##aplica a função mediana para os elementos do vetor CT,
indexados caracteres do vetor espécie.
    mediana.PT <- tapply(X=fishfauna$PT, INDEX=fishfauna$especie,
FUN=median,na.rm=T) ##aplica a função mediana para os elementos do vetor PT,
indexados caracteres do vetor espécie.
    dp.CT <- tapply(X=fishfauna$CT, INDEX=fishfauna$especie,
FUN=sd,na.rm=T) ##aplica a função desvio padrão para os elementos do vetor
CT, indexados caracteres do vetor espécie.
    dp.PT <- tapply(X=fishfauna$PT, INDEX=fishfauna$especie,
FUN=sd,na.rm=T) ##aplica a função desvio padrão para os elementos do vetor
PT, indexados caracteres do vetor espécie.
    min.CT <- tapply(X=fishfauna$CT, INDEX=fishfauna$especie, FUN=min)
## retorna o valor mínimo dos elementos do vetor CT, indexados caracteres do
vetor espécie.
    min.PT<- tapply(X=fishfauna$PT, INDEX=fishfauna$especie, FUN=min)
## retorna o valor mínimo dos elementos do vetor PT, indexados caracteres do
vetor espécie.
    max.CT <- tapply(X=fishfauna$CT, INDEX=fishfauna$especie, FUN=max)
## retorna o valor máximo dos elementos do vetor CT, indexados caracteres do
vetor espécie.
    max.PT <- tapply(X=fishfauna$PT, INDEX=fishfauna$especie, FUN=max)
## retorna o valor máximo dos elementos do vetor PT, indexados caracteres do
vetor espécie.
```

```
eb.anual <- round(data.frame(media.CT, mediana.CT, dp.CT, min.CT,
max.CT, media.PT, mediana.PT, dp.PT, min.PT, max.PT),2) ##cria um dataframe
com os valores das funções executadas.
ni.total <- length(fishfauna$CT) ##cria um objeto com o número de
elementos do vetor CT.
ni.sp <- tapply(X=fishfauna$CT, INDEX=fishfauna$especie, FUN=length)
##cria um objeto com o número de elementos do vetor CT, indexados pelos
caracteres do vetor espécie.
abun.rel <- round(((ni.sp*100)/ni.total),2) ## cria um objeto com a
proporção de cada elemento do objeto 'ni.sp', relativa ao número total de
elementos do vetor.
biomassa <- tapply(X=fishfauna$PT, INDEX=fishfauna$especie,
FUN=sum,na.rm=T) ##aplica a função soma para os elementos do vetor PT,
indexados pelos caracteres do vetor espécie.
prop<- 5 ## cria um objeto com a porcentagem de imagens determinada
pelo modelo de ecomorfologia.
n.f<- (ni.sp*prop)/100 #### cria um objeto com o numero de imagens a
serem utilizadas no modelo de ecomorfologia.
fotos<- data.frame(ni.sp, abun.rel, biomassa, n.f) ##cria um
dataframe com os valores de abundância, abundância relativa, biomassa e o
número de imagens.
ef.ano<- data.frame(eb.anual, fotos) ## cria um dataframe com os
valores dos vetores resposta, reunindo os dataframes eb.anual e fotos.
abundancia <- length(fishfauna$CT) ##cria um objeto com o número
total de elementos do vetor CT.
nS<- unique(fishfauna$especie) ## cria um objeto com o numero de
caracteres do vetor especie, sem repetições.
SM<- length(nS) ##cria um objeto com o numero de especies.
n<-log(abundancia,10) ##cria um objeto com o logaritmo do numero
total de elementos do vetor CT.
M<- (SM-1/(n)) ## cria um objeto com o valor resposta do Indice de
Margalef.
pix<- (ni.sp/sum(ni.sp)) ## cria um objeto com a porporção de
elementos do vetor CT.
l<- log(pix) ## cria um objeto com o logaritmo da proporção de
elementos do vetor CT.
pix.l<- pix*l ##cria um objeto com os valores resposta da
multiplicação da proporção de elementos pelo seu logaritmo.
H <- -(sum(pix.l)) ####cria um objeto com o valor resposta do Indice
de Shannon-Wiener.
S<- 1/sum(pix^2) ####cria um objeto com o valor resposta do Indice de
Simpson.
indices<- data.frame(M,H,S) ##cria um dataframe com os valores
dos índices calculados e guardados nos objetos M, H, S.

return(list(ef.ano,indices)) ## retorna uma lista de dataframes com
os valores requeridos na função.
}

if(estacao=="inverno")
{
```

```
inverno <- fishfauna[fishfauna$estacao=="Inverno",] ## cria um
objeto contendo os dados da estação "inverno", selecionados do dataframe
fishfauna.
media.CT <- tapply(X=inverno$CT, INDEX=inverno$especie,
FUN=mean,na.rm=T) ##aplica a função média para os elementos do vetor CT,
indexado pelos caracteres do vetor espécie.
media.PT<- tapply(X=inverno$PT, INDEX=inverno$especie,
FUN=mean,na.rm=T) ##aplica a função média para os elementos do vetor PT,
indexado pelos caracteres do vetor espécie.
mediana.CT <- tapply(X=inverno$CT, INDEX=inverno$especie,
FUN=median,na.rm=T) ##aplica a função mediana para os elementos do vetor CT,
indexados caracteres do vetor espécie.
mediana.PT <- tapply(X=inverno$PT, INDEX=inverno$especie,
FUN=median,na.rm=T) ##aplica a função mediana para os elementos do vetor PT,
indexados caracteres do vetor espécie.
dp.CT <- tapply(X=inverno$CT, INDEX=inverno$especie, FUN=sd,na.rm=T)
##aplica a função desvio padrão para os elementos do vetor CT, indexados
caracteres do vetor espécie.
dp.PT <- tapply(X=inverno$PT, INDEX=inverno$especie, FUN=sd,na.rm=T)
##aplica a função desvio padrão para os elementos do vetor PT, indexados
caracteres do vetor espécie.
min.CT <- tapply(X=inverno$CT, INDEX=inverno$especie, FUN=min) ##
retorna o valor mínimo dos elementos do vetor CT, indexados caracteres do
vetor espécie.
min.PT<- tapply(X=inverno$PT, INDEX=inverno$especie, FUN=min) ##
retorna o valor mínimo dos elementos do vetor PT, indexados caracteres do
vetor espécie.
max.CT <- tapply(X=inverno$CT, INDEX=inverno$especie, FUN=max) ##
retorna o valor máximo dos elementos do vetor CT, indexados caracteres do
vetor espécie.
max.PT <- tapply(X=inverno$PT, INDEX=inverno$especie, FUN=max) ##
retorna o valor máximo dos elementos do vetor PT, indexados caracteres do
vetor espécie.
eb.inv <- (na.omit(round(data.frame(media.CT, mediana.CT, dp.CT,
min.CT, max.CT, media.PT, mediana.PT, dp.PT, min.PT, max.PT),2))) ##cria
um dataframe com os valores das funções executadas.
ni.total <- length(inverno$CT) ##cria um objeto com o número de
elementos do vetor CT.
ni.sp <- tapply(X=inverno$CT, INDEX=inverno$especie, FUN=length)
##cria um objeto com o número de elementos do vetor CT, indexados pelos
caracteres do vetor espécie.
abun.rel <- (round(((ni.sp*100)/ni.total),2)) ## cria um objeto com
a proporção de cada elemento do objeto 'ni.sp', relativa ao número total de
elementos do vetor.
biomassa <- tapply(X=inverno$PT, INDEX=inverno$especie,
FUN=sum,na.rm=T) ##aplica a função soma para os elementos do vetor PT,
indexando pelos caracteres do vetor espécie.
prop<- 5 ## cria um objeto com a porcentagem de imagens determinada
pelo modelo de ecomorfologia.
n.f<- (ni.sp*prop)/100 ## cria um objeto com o numero de imagens a
serem utilizadas no modelo de ecomorfologia.
```

```
fotos<- (na.omit(data.frame(ni.sp, abund.rel, biomassa, n.f))) ##cria
um dataframe com os valores de abundância, abundância relativa, biomassa e o
número de imagens.
ef.inv<- (data.frame(eb.inv,fotos)) ## cria um dataframe com os
valores dos vetores resposta, reunindo os dataframes eb.inv e fotos.
abundancia <- length(inverno$CT) ##cria um objeto com o número total
de elementos do vetor CT.
nS<- unique(inverno$especie) ## cria um objeto com o numero de
caracteres do vetor especie, sem repetições.
SM<- length(nS) ##cria um objeto com o numero de especies.
n<-log(abundancia,10) ##cria um objeto com o logaritmo do numero
total de elementos do vetor CT.
M <- (SM-1/(n)) ## cria um objeto com o valor resposta do Indice de
Margalef.
pix<- (ni.sp/sum(ni.sp)) ## cria um objeto com a porporção de
elementos do vetor CT.
l<- log(pix) ## cria um objeto com o logaritmo da proporção de
elementos do vetor CT.
pix.l<- pix*l ##cria um objeto com os valores resposta da
multiplicação da proporção de elementos pelo seu logaritmo.
H <- -(sum(pix.l)) ####cria um objeto com o valor resposta do Indice
de Shannon-Wiener.
S<- 1/sum(pix^2) ####cria um objeto com o valor resposta do Indice de
Simpson.
indices<- data.frame(M,H,S) ##cria um dataframe com os valores
dos índices calculados e guardados nos objetos M, H, S.
return(list(ef.inv,indices)) ## retorna uma lista de dataframes com
os valores requeridos na função.
}

if(estacao=="outono")
{
  outono <- fishfauna[fishfauna$estacao=="Outono",] ## cria um objeto
contendo os dados da estação "outono", selecionados do dataframe fishfauna.
  media.CT <- tapply(X=outono$CT, INDEX=outono$especie, FUN=mean,
na.rm=T) ##aplica a função média para os elementos do vetor CT, indexado
pelos caracteres do vetor espécie.
  media.PT<- tapply(X=outono$PT, INDEX=outono$especie, FUN=mean,
na.rm=T) ##aplica a função média para os elementos do vetor PT, indexado
pelos caracteres do vetor espécie.
  mediana.CT <- tapply(X=outono$CT, INDEX=outono$especie, FUN=median,
na.rm=T) ##aplica a função mediana para os elementos do vetor CT, indexados
caracteres do vetor espécie.
  mediana.PT <- tapply(X=outono$PT, INDEX=outono$especie, FUN=median,
na.rm=T) ##aplica a função mediana para os elementos do vetor PT, indexados
caracteres do vetor espécie.
  dp.CT <- tapply(X=outono$CT, INDEX=outono$especie, FUN=sd, na.rm=T)
##aplica a função desvio padrão para os elementos do vetor CT, indexados
caracteres do vetor espécie.
  dp.PT <- tapply(X=outono$PT, INDEX=outono$especie, FUN=sd, na.rm=T)
##aplica a função desvio padrão para os elementos do vetor PT, indexados
```

caracteres do vetor espécie.

```
min.CT <- tapply(X=outono$CT, INDEX=outono$especie, FUN=min) ##  
retorna o valor mínimo dos elementos do vetor CT, indexados caracteres do  
vetor espécie.
```

```
min.PT<- tapply(X=outono$PT, INDEX=outono$especie, FUN=min) ##  
retorna o valor mínimo dos elementos do vetor PT, indexados caracteres do  
vetor espécie.
```

```
max.CT <- tapply(X=outono$CT, INDEX=outono$especie, FUN=max) ##  
retorna o valor máximo dos elementos do vetor CT, indexados caracteres do  
vetor espécie.
```

```
max.PT <- tapply(X=outono$PT, INDEX=outono$especie, FUN=max) ##  
retorna o valor máximo dos elementos do vetor PT, indexados caracteres do  
vetor espécie.
```

```
eb.out <- (na.omit(round(data.frame(media.CT, mediana.CT, dp.CT,  
min.CT, max.CT, media.PT, mediana.PT, dp.PT, min.PT, max.PT),2))) ##cria um  
dataframe com os valores das funções executadas.
```

```
ni.total <- length(outono$CT) ##cria um objeto com o número de  
elementos do vetor CT.
```

```
ni.sp <- tapply(X=outono$CT, INDEX=outono$especie, FUN=length)  
##cria um objeto com o número de elementos do vetor CT, indexados pelos  
caracteres do vetor espécie.
```

```
abun.rel <- round(((ni.sp*100)/ni.total),2) ## cria um objeto com a  
proporção de cada elemento do objeto 'ni.sp', relativa ao número total de  
elementos do vetor.
```

```
biomassa <- tapply(X=outono$PT, INDEX=outono$especie, FUN=sum,  
na.rm=T) ##aplica a função soma para os elementos do vetor PT, indexando  
pelos caracteres do vetor espécie.
```

```
prop<- 5 ## cria um objeto com a porcentagem de imagens determinada  
pelo modelo de ecomorfologia.
```

```
n.f<- (ni.sp*prop)/100 ## cria um objeto com o numero de imagens a  
serem utilizadas no modelo de ecomorfologia.
```

```
fotos<- (na.omit(data.frame(ni.sp, abund.rel, biomassa, n.f))) ##cria  
um dataframe com os valores de abundância, abundância relativa, biomassa e o  
número de imagens.
```

```
ef.out<- data.frame(eb.out,fotos) ## cria um dataframe com os  
valores dos vetores resposta, reunindo os dataframes eb.out e fotos.
```

```
abundancia <- length(outono$CT) ##cria um objeto com o número total  
de elementos do vetor CT.
```

```
nS<- unique(outono$especie) ## cria um objeto com o numero de  
caracteres do vetor especie, sem repetições.
```

```
SM<- length(nS) ##cria um objeto com o numero de especies.
```

```
n<-log(abundancia,10) ##cria um objeto com o logaritmo do numero  
total de elementos do vetor CT.
```

```
M<- (SM-1/(n)) ## cria um objeto com o valor resposta do Indice de  
Margalef.
```

```
pix<- (ni.sp/sum(ni.sp)) ## cria um objeto com a porporção de  
elementos do vetor CT.
```

```
l<- log(pix) ## cria um objeto com o logaritmo da proporção de  
elementos do vetor CT.
```

```
pix.l<- pix*l ##cria um objeto com os valores resposta da
```

multiplicação da proporção de elementos pelo seu logaritmo.

```
H <- -(sum(pix.l)) ###cria um objeto com o valor resposta do Índice de Shannon-Wiener.
```

```
S<- 1/sum(pix^2) ###cria um objeto com o valor resposta do Índice de Simpson.
```

```
indices<- data.frame(M,H,S) ##cria um dataframe com os valores dos índices calculados e guardados nos objetos M, H, S.
```

```
return(list(ef.out,indices)) ## retorna uma lista de dataframes com os valores requeridos na função.
```

```
}
```

```
if(estacao=="primavera")
```

```
{
```

```
    primavera <- fishfauna[fishfauna$estacao=="Primavera",] ## cria um objeto contendo os dados da estação "primavera", selecionados do dataframe fishfauna.
```

```
    media.CT <- tapply(X=primavera$CT, INDEX=primavera$especie, FUN=mean,na.rm=T) ##aplica a função média para os elementos do vetor CT, indexado pelos caracteres do vetor espécie.
```

```
    media.PT <- tapply(X=primavera$PT, INDEX=primavera$especie, FUN=mean,na.rm=T) ##aplica a função média para os elementos do vetor PT, indexado pelos caracteres do vetor espécie.
```

```
    mediana.CT <- tapply(X=primavera$CT, INDEX=primavera$especie, FUN=median,na.rm=T) ##aplica a função mediana para os elementos do vetor CT, indexados caracteres do vetor espécie.
```

```
    mediana.PT <- tapply(X=primavera$PT, INDEX=primavera$especie, FUN=median,na.rm=T) ##aplica a função mediana para os elementos do vetor PT, indexados caracteres do vetor espécie.
```

```
    dp.CT <- tapply(X=primavera$CT, INDEX=primavera$especie, FUN=sd,na.rm=T) ##aplica a função desvio padrão para os elementos do vetor CT, indexados caracteres do vetor espécie.
```

```
    dp.PT <- tapply(X=primavera$PT, INDEX=primavera$especie, FUN=sd,na.rm=T) ##aplica a função desvio padrão para os elementos do vetor PT, indexados caracteres do vetor espécie.
```

```
    min.CT <- tapply(X=primavera$CT, INDEX=primavera$especie, FUN=min) ## retorna o valor mínimo dos elementos do vetor CT, indexados caracteres do vetor espécie.
```

```
    min.PT<- tapply(X=primavera$PT, INDEX=primavera$especie, FUN=min) ## retorna o valor mínimo dos elementos do vetor PT, indexados caracteres do vetor espécie.
```

```
    max.CT <- tapply(X=primavera$CT, INDEX=primavera$especie, FUN=max) ## retorna o valor máximo dos elementos do vetor CT, indexados caracteres do vetor espécie.
```

```
    max.PT <- tapply(X=primavera$PT, INDEX=primavera$especie, FUN=max) ## retorna o valor máximo dos elementos do vetor PT, indexados caracteres do vetor espécie.
```

```
    eb.prim <- (na.omit(round(data.frame(media.CT, mediana.CT, dp.CT, min.CT, max.CT, media.PT, mediana.PT, dp.PT, min.PT, max.PT),2))) ##cria um dataframe com os valores das funções executadas.
```

```
    ni.total <- length(primavera$CT) ##cria um objeto com o número de
```

elementos do vetor CT.

```
ni.sp <- tapply(X=primavera$CT, INDEX=primavera$especie, FUN=length)
##cria um objeto com o número de elementos do vetor CT, indexados pelos
caracteres do vetor espécie.
```

```
abun.rel <- round(((ni.sp*100)/ni.total),2) ## cria um objeto com a
proporção de cada elemento do objeto 'ni.sp', relativa ao número total de
elementos do vetor.
```

```
biomassa <- tapply(X=primavera$PT, INDEX=primavera$especie,
FUN=sum,na.rm=T) ##aplica a função soma para os elementos do vetor PT,
indexando pelos caracteres do vetor espécie.
```

```
prop<- 5 ## cria um objeto com a porcentagem de imagens determinada
pelo modelo de ecomorfologia.
```

```
n.f<- (ni.sp*prop)/100 ## cria um objeto com o numero de imagens a
serem utilizadas no modelo de ecomorfologia.
```

```
fotos<- (na.omit(data.frame(ni.sp, abun.rel, biomassa, n.f))) ##cria
um dataframe com os valores de abundância, abundância relativa, biomassa e o
número de imagens.
```

```
ef.prim <- data.frame(eb.prim, fotos) ## cria um dataframe com os
valores dos vetores resposta, reunindo os dataframes eb.prim e fotos.
```

```
abundancia <- length(primavera$CT) ##cria um objeto com o número
total de elementos do vetor CT.
```

```
nS<- unique(primavera$especie) ## cria um objeto com o numero de
caracteres do vetor especie, sem repetições.
```

```
SM<- length(nS) ##cria um objeto com o numero de especies.
```

```
n<-log(abundancia,10) ##cria um objeto com o logaritmo do numero
total de elementos do vetor CT.
```

```
M<- (SM-1/(n)) ## cria um objeto com o valor resposta do Indice de
Margalef.
```

```
pix<- (ni.sp/sum(ni.sp)) ## cria um objeto com a proporção de
elementos do vetor CT.
```

```
l<- log(pix) ## cria um objeto com o logaritmo da proporção de
elementos do vetor CT.
```

```
pix.l<- pix*l ##cria um objeto com os valores resposta da
multiplicação da proporção de elementos pelo seu logaritmo.
```

```
H <- -(sum(pix.l)) ####cria um objeto com o valor resposta do Indice
de Shannon-Wiener.
```

```
S<- 1/sum(pix^2) ####cria um objeto com o valor resposta do Indice de
Simpson.
```

```
indices<- data.frame(M,H,S) ##cria um dataframe com os valores
dos índices calculados e guardados nos objetos M, H, S.
```

```
return(list(ef.prim, indices)) ## retorna uma lista de dataframes
com os valores requeridos na função.
```

```
}
```

```
if(estacao=="verao")
```

```
{
```

```
verao <- fishfauna[fishfauna$estacao== "Verão",] ## cria um objeto
contendo os dados da estação "primavera", selecionados do dataframe
fishfauna.
```

```
media.CT <- tapply(X=verao$CT, INDEX=verao$especie,
FUN=mean,na.rm=T) ##aplica a função média para os elementos do vetor CT,
```

```
indexado pelos caracteres do vetor espécie.  
media.PT<- tapply(X=verao$PT, INDEX=verao$especie, FUN=mean,na.rm=T)  
##aplica a função média para os elementos do vetor PT, indexado pelos  
caracteres do vetor espécie.  
mediana.CT <- tapply(X=verao$CT, INDEX=verao$especie,  
FUN=median,na.rm=T) ##aplica a função mediana para os elementos do vetor CT,  
indexados caracteres do vetor espécie.  
mediana.PT <- tapply(X=verao$PT, INDEX=verao$especie,  
FUN=median,na.rm=T) ##aplica a função mediana para os elementos do vetor PT,  
indexados caracteres do vetor espécie.  
dp.CT <- tapply(X=verao$CT, INDEX=verao$especie, FUN=sd,na.rm=T)  
##aplica a função desvio padrão para os elementos do vetor CT, indexados  
caracteres do vetor espécie.  
dp.PT <- tapply(X=verao$PT, INDEX=verao$especie, FUN=sd,na.rm=T)  
##aplica a função desvio padrão para os elementos do vetor PT, indexados  
caracteres do vetor espécie.  
min.CT <- tapply(X=verao$CT, INDEX=verao$especie, FUN=min) ##  
retorna o valor mínimo dos elementos do vetor CT, indexados caracteres do  
vetor espécie.  
min.PT<- tapply(X=verao$PT, INDEX=verao$especie, FUN=min) ##  
retorna o valor mínimo dos elementos do vetor PT, indexados caracteres do  
vetor espécie.  
max.CT <- tapply(X=verao$CT, INDEX=verao$especie, FUN=max) ##  
retorna o valor máximo dos elementos do vetor CT, indexados caracteres do  
vetor espécie.  
max.PT <- tapply(X=verao$PT, INDEX=verao$especie, FUN=max) ##  
retorna o valor máximo dos elementos do vetor PT, indexados caracteres do  
vetor espécie.  
eb.verao <- (na.omit(round(data.frame(media.CT, mediana.CT, dp.CT,  
min.CT, max.CT, media.PT, mediana.PT, dp.PT, min.PT, max.PT),2))) ##cria um  
dataframe com os valores das funções executadas.  
ni.total <- length(verao$CT) ##cria um objeto com o número de  
elementos do vetor CT.  
ni.sp <- tapply(X=verao$CT, INDEX=verao$especie, FUN=length) ##cria  
um objeto com o número de elementos do vetor CT, indexados pelos caracteres  
do vetor espécie.  
abun.rel <- round(((ni.sp*100)/ni.total),2) ## cria um objeto com a  
proporção de cada elemento do objeto 'ni.sp', relativa ao número total de  
elementos do vetor.  
biomassa <- tapply(X=verao$PT, INDEX=verao$especie, FUN=sum,na.rm=T)  
##aplica a função soma para os elementos do vetor PT, indexando pelos  
caracteres do vetor espécie.  
prop<- 5 ## cria um objeto com a porcentagem de imagens determinada  
pelo modelo de ecomorfologia.  
n.f<- (ni.sp*prop)/100 ## cria um objeto com o numero de imagens a  
serem utilizadas no modelo de ecomorfologia.  
fotos<- (na.omit(data.frame(ni.sp, abund.rel, biomassa, n.f))) ##cria  
um dataframe com os valores de abundância, abundância relativa, biomassa e o  
número de imagens.  
ef.ver <- data.frame(eb.verao, fotos) ## cria um dataframe com os
```

```
valores dos vetores resposta, reunindo os dataframes eb.verao e fotos.
  abundancia <- length(verao$CT) ##cria um objeto com o número total
de elementos do vetor CT.
  nS<- unique(verao$especie) ## cria um objeto com o numero de
caracteres do vetor especie, sem repetições.
  SM<- length(nS) ##cria um objeto com o numero de especies.
  n<-log(abundancia,10) ##cria um objeto com o logaritmo do numero
total de elementos do vetor CT.
  M<- (SM-1/(n)) ## cria um objeto com o valor resposta do Indice de
Margalef.
  pix<- (ni.sp/sum(ni.sp)) ## cria um objeto com a porporção de
elementos do vetor CT.
  l<- log(pix) ## cria um objeto com o logaritmo da proporção de
elementos do vetor CT.
  pix.l<- pix*l ##cria um objeto com os valores resposta da
multiplicação da proporção de elementos pelo seu logaritmo.
  H <- -(sum(pix.l)) ###cria um objeto com o valor resposta do Indice
de Shannon-Wiener.
  S<- 1/sum(pix^2) ###cria um objeto com o valor resposta do Indice de
Simpson.
  indices<- data.frame(M,H,S)    ##cria um dataframe com os valores
dos índices calculados e guardados nos objetos M, H, S.
  return(list(ef.ver, indices)) ## retorna uma lista de dataframes com
os valores requeridos na função.
}
}
```

```
fisheco(fishfauna,"verao")
fisheco(fishfauna,"outono")
fisheco(fishfauna,"primavera")
fisheco(fishfauna,"inverno")
fisheco(fishfauna)
```

Help da Função

```
**fisheco**           **pacote: nenhum**           **R documentation**
```

Determina o número de imagens a serem analisadas para cada espécie de peixe em estudos de ecomorfologia por pontos homólogos.

Descrição:

Calcula a estatística descritiva básica (média, mediana, desvio padrão, mínimo, máximo) de dois vetores de valores (CT, mm, comprimento total do

peixe;
PT, g, peso total do peixe); a abundância, a abundância relativa, a biomassa das espécies,
retornando o número de imagens a serem obtidas de cada espécie, derivada de 5% de sua
abundância na amostra; e, os índices clássicos de ecologia de comunidades Margalef, Simpson
e Shannon-Wiener. Para realizar esses cálculos por estação do ano, o argumento 'estação' deve
conter "inverno", "outono", "primavera", "verão". O default do argumento 'estação' retorna
resultados para dados agrupados (inverno, outono, primavera e verão). Caso rmNA seja verdadeiro,
os NA's serão excluídos.

Uso:

```
fisheco(x, estacao, rmNA=TRUE)
```

Argumentos:

x: dataframe com quatro vetores. O primeiro vetor contendo caracteres do nome da espécie,
o segundo vetor contendo o comprimento total (CT, mm) dos peixes, o terceiro vetor contendo o
peso total (PT,g) dos peixes, e o quarto vetor com a estação do ano na qual eles foram capturados.

Pode ter de uma a infinitas linhas.

estacao: vetor de caracteres que estabelece a estação do ano na qual os peixes foram capturados.

Este argumento pode conter "inverno", "outono", "primavera", "verão", "anual"=default.

rmNA: quando verdadeiro, exclui os NA's existentes.

Detalhes:

A função calcula o número de imagens a serem analisadas por espécie em um estudo de estrutura de comunidades a partir de pontos homólogos. Este cálculo é baseado no conhecimento da composição em comprimento e peso dos peixes desta comunidade e na abundância de cada espécie que a compõem. A função é útil quando se trata de estudos de peixes de clima tropical, em que a diversidade de espécies é alta e, conseqüentemente, a quantidade de dados disponíveis para análise se torna elevada.

Valores:

Retorna uma lista de dataframes, um com a estatística descritiva básica de

dois vetores de valores por elementos de um vetor de caracteres o qual também contém valores de abundância, abundância relativa, biomassa das espécies e o número de imagens a serem obtidas de cada uma; e, outro contendo os resultados do cálculo dos índices clássicos de ecologia de comunidades Margalef, Simpson e Shannon-Wiener.

Veja também:

```
data.frame rm.NA mean median sd min max  tapply INDEX
```

Autor:

Carolina Correia Siliprandi

(siliprandic@gmail.com)

Referencia:

Begon, M.; Harper, J.L. ; Townsend, C.R. 1996. Ecology - Individuals, Populations and Communities. Blackwell Science.

Farré, M.; Tuset, V.M.; Maynou, F.; Recasens, L.; Lombarte, A. 2013. Geometric morphology as an alternative for measuring the diversity of fish. Ecological Indicators 29, 159-166.

Magurran, A.E. 2004. Measuring biological diversity. Blackwell Science.

Exemplo:

```
dados <- read.table("Ex tfdata.csv", header=T, sep="," , as.is=T)
```

```
fisheco(dados)
```

Dados do exemplo [ex_tfdata.csv](#)

Dados da função [tfdata.csv](#)

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2015:alunos:trabalho_final:siliprandi:start



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