

Livia de Almeida Rodrigues



Sou formada em Biologia pela UFRJ e Analista Ambiental do ICMBio/CENAP. O uso da linguagem S (Software R) se tornou importante no CENAP, auxiliando na geração de melhores informações para elaboração de estratégias de conservação dos grupos sob nossa responsabilidade (Carnívoros, Ungulados, Marsupiais e Roedores).

Exercícios

Página de exercícios resolvidos: [exercícios](#)

Trabalho Final

Proposta A

Função para análise dos dados do Programa Monitora do ICMBIO

O Programa de Monitoramento da Biodiversidade (Programa Monitora) coordenado pelo ICMBio em parceria com diversas instituições nacionais e internacionais tem como objetivos gerar informação qualificada para apoio à gestão das unidades de conservação federal (UCs) e estabelecer parâmetros ecológicos para avaliação da efetividade das UCs; Um dos protocolos estabelecidos segue o protocolo TEAM utilizando armadilhas fotográficas (camera trap). O Centro Nacional de Pesquisa e Conservação de Mamíferos Carnívoros (CENAP) coordena o programa para os mamíferos de médio e grande porte. A função irá permitir uma maior agilidade nas análises com excelente qualidade técnica, permitindo que o usuário escolha se análise será por ano de amostragem ou análise conjunta dos anos.

Os dados de entrada na função serão do dataframe de saída da identificação dos registros das armadilhas fotográficas, onde cada linha é um registro e as colunas, as variáveis. Esse dataframe é padrão para a grande maioria dos projetos com armadilhamento fotográfico ([dados_monitora.csv](#) - dados não publicados). Para a função os argumentos são referentes aos vetores das colunas: `dados_monitora$bin = ID.spp`, `dados_monitora$Camera.Trap.Name = sampling.units` e `dados_monitora$Photo.Date = date`.

Planejamento da função

Entrada: `monitora(ID.spp, sampling.units, date, period="all", na.omit=T)`

- `ID.spp` = vetor referente as identificações das espécies (classe=factor).
- `sampling.units` = vetor referente a unidade amostral (classe=factor).
- `date` = vetor referente a data do registro (classe=factor)
- `period` = o usuário poderá escolher o período da análise ("annual" ou "all"). Annual - análise por ano; All - análise conjunta de todos os anos de monitoramento.
- `NA.omit` = lógico, exclui os registros sem informação (NA).

Verificando os parâmetros:

- Todos os vetores de entrada devem ter o mesmo tamanho. Caso não tenha, a função não rodará e uma mensagem de erro aparecerá com a mensagem “os vetores não tem os mesmos comprimentos”.
- date = no formato %Y-%m-%d, se não, escreve mensagem de erro “data em formato incorreto”.
- period = “annual” ou “all”. se não, escreve “o period indicado não foi encontrado, o resultado apresentado será usando o padrão “all”.

Pseudo-código:

- #Controle de fluxo encaminha para period= “all”:
- #Cria um vetor com a contagem por ID.spp;
- #Cria objeto com contagem do número acumulado de ID.spp observado por sampling.units;
- #Cria objeto com o número total de ID.spp estimado por sampling.units;
- #Plota a curva acumulada de ID.spp observado e estimado por sampling.units, incluindo o intervalo de confiança;
- #Cria objeto com a estimativa de riqueza de ID.spp da area de estudo;
- #Plota a estimativa da riqueza de ID.spp para área de estudo.
- se period=“annual”
- Cria um dataframe com a contagem por ID.spp por ano;
- Cria um array[ID.spp, sampling.units, ano] com contagem do número acumulado de ID.spp observado por sampling.units por ano;
- Cria um array [ID.spp, sampling.units, ano] com o número total de ID.spp estimado por sampling.units por ano (vegan::specaccum())
- Plota a curva acumulada de ID.spp observado e estimado por ano (uma linha para cada ano) por sampling.units, incluindo o intervalo de confiança em cada linha;
- Cria objeto com a estimativa de riqueza de ID.spp por ano;
- Plota a estimativa da riqueza de ID.spp por ano para area de estudo.

Saída:

- se period=“all” a função irá resultar em uma lista contendo:
- vetor com contagem de ID.spp por sampling.units e um vetor com as 10 ID.spp mais registradas na área de estudo;
- Gráfico de acumulação de ID.spp observado e estimado;
- Gráfico da estimativa de riqueza de ID.spp para a area de estudo.
- se period=“annual” a função irá resultar:
- um array [ID.spp, sampling.units, ano] com a contagem por ID.spp por ano e por sampling.units;
- Gráfico de acumulação de ID.spp observado e estimado por ano com o intervalo de confiança;
- Gráfico da estimativa de riqueza de ID.spp para a area de estudo por ano.

— [Alexandre Adalardo de Oliveira](#) 2019/06/04 14:24

Oi Livia,

A proposta parece interessante, mas senti falta de uma explicação da estrutura dos dados. No início fala em "*Os dados precisarão ser previamente trabalhados a fim de excluir as observações das espécies sem identificação e demais NAs em outras variáveis. As observações devem ser agrupadas em eventos independentes*". Como a ideia da função é justamente extrair informações estruturadas de um conjunto de dados maiores, precisa de uma explicação da estrutura dos dados brutos, até pq não sei se entendo o que pretende com "*observações devem ser agrupadas em eventos independentes*". Vc. explica os métodos que são utilizados, o que é bom, mas não se refere a estrutura dos dados e como eles estarão armazenados, se dentro da função ou lidos junto com outros argumentos. Outra coisa que precisa explicar melhor é o argumento `data` que recebe um vetor taxonômico, não entendi o que define o nível taxonômico e como, por exemplo, o usuário leigo como eu, tem acesso aos grupos taxonômicos que estão disponíveis para serem consultados. O mesmo para `sampling-units`, fico sem entender como diferencia dia de coleta de camera-trap, e como, o usuário sabe qual as datas disponíveis e cameras... O pseudo-código ajuda, mas sem saber a estrutura dos dados fonte fica difícil avaliar a viabilidade e grau de dificuldade da função. Pode postar uma explicação sucinta sobre a estrutura dos dados, ou mesmo um exemplo de alguns registros, e como pretende que a função acesse esses dados?

— [Livia de Almeida Rodrigues](#) 2019/06/05 10:42

oi Alexandre,

fiz um ajustes na proposta baseado no seu comentário. Os dados de entrada não são de nenhum banco de dados, e sim a planilha de saída da identificação dos registros das armadilhas fotográficas, inclui o arquivo no texto para que consiga ter acesso, pode facilitar a entender. A ideia é que essa função seja divulgada e utilizada por todos os parceiros e pesquisadores que realizam projetos de pesquisa com armadilhas fotográficas.

— [Alexandre Adalardo de Oliveira](#) 2019/06/05 13:45 Oi Livia, com os dados ficou melhor para entender a função. Entendi que os dados dos registros da armadilha fotográfica são sempre iguais (padronizados), portanto, não parece fazer

muito sentido chamar os vetores separadamente, já que estão na mesma planilha e podem ser lidos como um data frame. O melhor é a função ler diretamente a planilha do arquivo csv, indicado pelo usuário. Faz sentido? Caso os nomes das variáveis na planilha, possa variar, indique argumentos onde o usuário possa mudar o padrão do nome. Nos seus argumentos não há um “arquivo”, ou seja, indicando onde está o arquivo de dados. Do jeito que colocou os dados precisam ser lidos antes de rodar a função e as variáveis separadas em vetores, que depois serão lidas pela função, não parece ser muito amigável. Além disso, as sínteses de dados que irá fazer estão parecendo muito simples, podem ser feitas com um `tapply`, portanto, fiquei em dúvida do grau de dificuldade da função. De qq forma, vou deixar que siga com esse plano e faço as seguintes sugestões:

- 1. use um argumento para indicar onde o arquivo de dados está, para ser lido diretamente pela função a partir do csv padronizado no computador. Algo como `filedata = "dados_monitoria.csv"`
- 2. ao ler o arquivo de dados, faça a correta transformação da variável `date` para que seja da classe `date` e não um factor como coloca.
- 3. permitir que o argumento de agrupamento `period` possa ser definido por `dia`, `'mes'`, `'ano'` ou `'all'`
- 4. permitir a definição de agrupamento taxonômico com um argumento que tenha `taxlevel`, podendo ser, por exemplo: `"Order"`, `"Family"`, `"Genus"`, `"Species"`¹⁾
- 5. permita que sua saída seja para os períodos definidos pelo argumento `period`, não apenas `ano`. O período `all` eu não sei se entendi ainda muito, seria o período todo, portanto não há curva cumulativa de espécies certo? Acho que poderia permitir que a curva fosse construída por `dia`, `mes` ou `ano` e pelo grupo taxonomico definido em `taxlevel`, ou seja se os argumentos forem definidos como `genus` e `semana`, quantos novos generos acumulados por semana. Não sei se faz sentido, vc deve saber bem o que os usuários fazem com esses dados e em que variáveis é importante fazer algum agrupamento.

Bom, siga em frente, to tempo é curto e podemos fazer algum outro ajuste se achar necessário.

Proposta B

A função permitirá avaliar o grau de sobreposição de habitat entre duas espécies simpátricas.

Entrada: `sobrepor(sp1, sp2, dist)`

- `sp1` = dataframe com as coordenadas geográficas (graus decimais) da primeira espécie (dataframe com as colunas espécie, longitude, latitude)
- `sp2` = dataframe com as coordenadas geográficas (graus decimais) da segunda espécie (dataframe com as colunas espécie, longitude, latitude)
- `dist` = definição da maior distancia para agrupar os vizinhos mais proximos (baseado na área de vida das espécies)

Verificação de parametros:

- `sp1` e `sp2` deverão estar em graus decimais(numericos). Se não, a função para e aparece a mensagem de erro “coordenadas geograficas em sistema diferente”

Pseudo-código:

- #criar objeto tabela com as informações de long e lat das espécies, retirando os NA caso tenha
- #criar objeto que é uma matriz de distancia euclidiana, função `spDist` do pacote `sp`, para cada espécie
- # atribuir NA a diagonal para evitar os zeros que seria computados como menor distancia `# diag(dist.mat) ← NA`
- #criar vetor com minima distancia do vizinho mais proximo por espécie
- #Criar novas colunas de long e lat para cada espécie referente a posição do vizinho mais proximo de cada ponto
- #Criar poligono (menor poligono convexo) com os vizinhos mais proximos a uma distancia maxima definida em “dist”.
- #Avaliar a porcentagem de sobreposição entre os poligonos de cada espécie.

Saída:

- matriz com a informação da porcentagem de sobreposição referente a cada poligono da espécie.

Não sei se entendi o que a função faz. Talvez tenha faltado uma contextualização ou me falta conhecimento da área para entender o que pretende. Entendi que para cada espécie e suas ocorrências geográficas irá tomar o vizinho mais próximo e pegar as coordenadas dele. Como essa informação se relaciona ao polígono de uma distância pré-definida com seus vizinhos e como isso se transforma em sobreposição entre espécies? Talvez um esquema ou uma referência ajude.

— *Livia de Almeida Rodrigues* 2019/06/05 11:18

oi Alexandre,

Essa função seria pensando em espécies que vivem em grupo. Então, cada polígono seria criado com base nos vizinhos mais próximos tendo como limite a distancia pré-definida, pensando na area de vida de cada grupo. Isso seria feito para a sp1 e para a sp2, a comparação dos poligonos mostraria se ocorre sobreposição entre os grupos da sp1 e sp2.

Função monitora

```
monitora<- function(filedata=NULL, #dataframe com os dados
                     ID.spp=NULL, #colunas da variavel ID.spp, caso o nome da
coluna esteja diferente do default, o usuário poderá informar
                     sampling.units=NULL, #coluna variavel da unidade
amostral, caso o nome da coluna esteja diferente do default, o usuário
poderá informar
                     date=NULL, #coluna variavel da data, caso o nome da
coluna esteja diferente do default, o usuário poderá informar
                     period="all") #period (o usuário poderá configurara
análise para o período todo ou por ano)
{
  if("lubridate" %in% rownames(installed.packages()) == FALSE)
{install.packages("lubridate")} #verifica se os pacotes necessarios estão
instalados, se não os instala.

  if("vegan" %in% rownames(installed.packages()) == FALSE)
{install.packages("vegan")}#verifica se os pacotes necessarios estão
instalados, se não os instala.
  if("ggplot2" %in% rownames(installed.packages()) == FALSE)
{install.packages("ggplot2")}#verifica se os pacotes necessarios estão
instalados, se não os instala.
  require("lubridate") #pacote que precisa ser chamado
  require("vegan")     #pacote que precisa ser chamado
  require("ggplot2")   #pacote que precisa ser chamado
  if(is.null(ID.spp)) #se o argumento não for informado pelo usuário
  {
    ID.spp=filedata[7] #associa o ID.spp a coluna do filedata na posição
correta
  }
  if(is.null(sampling.units))#se o argumento não for informado pelo usuário
  {
    sampling.units=filedata[2] #associa o sampling.units a coluna do
filedata na posição correta
  }
}
```

```
if(is.null(date)) #se o argumento não for informado pelo usuário
{
  date=filedata[3] #associa o date a coluna do filedata na posição
correta
}

dados<-data.frame(ID.spp,sampling.units,date) #cria o dataframe com as
variaveis para analise
colnames(dados)<-c("ID.spp","sampling.units","date")
dados[dados==""] <-NA #incluir NA nos campos vazios
dados[,3]<-as.Date(dados$date, format="%d/%m/%Y") #converte o vetor
referente ao argumento date em class date
dados$ano<- year(dados[,3]) #extrair os anos da data
years<- unique(dados$ano) #vetor com os valores unicos

if(ifelse(sum(is.na(dados))!=0, TRUE,FALSE)) #verifica a ocorrencia de NA
no dataframe e e avisa ao usuario que será excluido da analise
{
  warning("Sua planilha contem celulas vazias cujos os registros serao
omitidos da analise")
  dados<-na.omit(dados) #omissao dos NA dos dados
}

if(period!="all"& period!="annual") #conferindo se o usuario indicou o
periodo corretamente, senão seguirá o default
{
  warning("periodo informado não encontrado, analise seguirá o default
period=all")
  period="all"
}
if(period=="all") #analise para o period=all
{
  n.spp<-data.frame(sort(table(dados[,1]), decreasing = T)) #Cria um
dataframe com a frequencia de registro por ID.spp
  colnames(n.spp)<- c("Especies", "Frequencia") #Nome das colunas
  #n.spp.10<-n.spp[1:10,] #extraindo as 10 especies mais frequentes
  sp.all <- with(dados, table(dados[,3],dados[,1])) #Cria objeto com
contagem do numero acumulado de ID.spp observado por dia
  accum.sp<-specaccum(sp.all) # riqueza de especie e desvio padrao ao
longo do esforco amostral
  accum.sp2 <- specaccum(sp.all, "random") #riqueza de especie estimada e
desvio padrao ao longo do esforco amostral
  x11()
  plot(accum.sp, ci.type="poly", col="red", lwd=2, ci.lty=0,
ci.col=rgb(1,0,0,0.2), ylab="Riqueza de Especies", xlab="Esforco amostral
(dias)", main="Curva de Acumulacao de Especies") #grafico da curva de
acumulacao observada
  par(new=TRUE) #permite que os gráficos sejam plotados uma sobre o outro
para comparacao
  plot(accum.sp2, ci.type="poly", col="black", lwd=2, ci.lty=0,
ci.col=rgb(0,0,0,0.2), xaxt="n", yaxt="n", xlab="", ylab="") # grafico da
```

```
curva de acumulação estimada
  legend(x=100, y=20, legend= c("Observada", "Estimada"), col=
c(rgb(1,0,0,0.2),rgb(0,0,0,0.2)), bty="n", text.font=0.5, lwd=2) # legenda
do gráfico
  savePlot("acumulacao.especies.jpg", type= "jpg") #salva o grafico em
jpg
  x11()
  ric.all<-specnumber(sp.all) #Cria objeto com a riqueza de especie por
dia;
  hist(ric.all, main="", xlab="Riqueza de Especies observada",
ylab="Frequencia", col="grey") # riqueza de espécies
  savePlot("riqueza.especies.jpg", type= "jpg") #salvar o grafico em jpg
  pool.all <- specpool(sp.all) #Estimativa da riqueza de especie para a
area de estudo por diversos estimadores.
  resultados<- list(total.Especies=n.spp,
Estimadores.riqueza=summary(pool.all))
  #Grafico da estimativa de riqueza de ID.spp para a area de estudo.
}

if(period=="annual") #para period="annual"
{
  resultados<- list() # cria o objeto resultado como lista
  riqueza<-rep(NA, length(years))
  for( j in 1:length(years)) #for para inserir os resultados em lista
  {
    resultados[[j]]<- list(NA) #cria uma lista vazia
    names(resultados[[j]])<- paste0("ano.",years[j]) #nomeia cada posição da
lista por ano de analise
  }
  for(i in 1:length(years)) #for para realizar as analises por ano
separadamente
  {
    dados.ano<- dados[dados$ano==years[i],]
    n.spp<-data.frame(sort(table(dados.ano[,1]), decreasing = T)) #Cria um
dataframe com a frequencia de registro por ID.spp
    colnames(n.spp)<- c("Especies", "Frequencia") #Nome das colunas
    #n.spp.10<-n.spp[1:10,] #extraindo as 10 especies mais frequentes
    sp.all <- with(dados.ano, table(dados.ano[,3],dados.ano[,1])) #Cria
objeto com contagem do numero acumulado de ID.spp observado por dia
    accum.sp<-specaccum(sp.all) # riqueza de especie e desvio padrao ao
longo do esforco amostral
    accum.sp2 <- specaccum(sp.all, "random") #riqueza de especie estimada e
desvio padrao ao longo do esforco amostral
    x11()
    plot(accum.sp, ci.type="poly", col="red", lwd=2, ci.lty=0,
ci.col=rgb(1,0,0,0.2), ylab="Riqueza de Especies", xlab="Esforco amostral
(dias)", main= paste0("Curva de Acumulacao de Especies.",years[i])) #grafico
da curva de acumulacao observada
    par(new=TRUE) #permite que os graficos sejam plotados uma sobre o outro
para comparacao
```



```

    plot(accum.sp2, ci.type="poly", col="black", lwd=2, ci.lty=0,
ci.col=rgb(0,0,0,0.2), xaxt="n", yaxt="n", xlab="", ylab="") # grafico da
curva de acumulacao estimada
    legend (x=40, y=15, legend= c("Observada", "Estimada"), col=
c(rgb(1,0,0,0.2),rgb(0,0,0,0.2)), bty="n", text.font=0.5, lwd=2) # legenda
do grafico
    savePlot(paste0("acumulacao.especies.",years[i],".jpg"), type= "jpg")
#salva o grafico em jpg
    ric.all<-specnumber(sp.all) #Cria objeto com a riqueza de especie por
dia;
    x11()
    hist(ric.all, main= paste0("Riqueza de Especie.",years[i]),
xlab="Riqueza de Especies observada", ylab="Frequencia", col="grey") #
riqueza de especies
    savePlot(paste0("riqueza.especies.",years[i],".jpg"), type= "jpg")
#salvar o grafico em jpg
    pool.all <- specpool(sp.all) #Estimativa da riqueza de especie para a
area de estudo por diversos estimadores.
    resultados[[i]][[1]]<-n.spp #frequencia de ID.spp por ano
    resultados[[i]][[2]]<-summary(pool.all) #resultado dos estimadores de
riqueza por ano
    names(resultados[[i]][[1]][,1])<- paste0("total.Especies.",years[i])
#renomeando os resultados
    names(resultados[[i]][[2]][,1])<-
paste0("Estimadores.riqueza.",years[i]) #renomeando os resultados
    riqueza[i] <- as.numeric(sub("Mean      :", "" ,
x=resultados[[i]][[2]][4,4]))
  }
  riqueza
  years
  y <- data.frame(cbind(years,riqueza)) # criando o objeto para o
grafico
  x11()
  plot(ggplot(data=y,aes(x=years,y=riqueza)) + #grafico de estimativa
de riqueza por ano
    geom_line(size=2, alpha=0.5) + #configuração do tamanho dos pontos
    geom_point(size=1.5, alpha=0.5) + #configuração do tamanho das linhas
    ylim(0,2*max(riqueza)) + #configuração do limite do eixo y para a
curva ficar localizada no meio da area do grafico
    scale_x_continuous(breaks=c(years)) + #configuração do limite do eixo
x como numeros discretos
    theme(panel.background = element_blank(), axis.line =
element_line(colour = "black")) + # configuração da area de plotagem
    theme(axis.text=element_text(size=10),axis.title=element_text(size=10)) +
    theme(axis.title.x = element_blank()) +
    ylab("Estimativa Média de Riqueza de Especies (Jack1)")
    savePlot("estimativa.riqueza.jpeg", type="jpeg")
  }
  return(resultados)
}

```

Help - Função monitora

monitora

package:nenhum

R Documentation

~~Estimativa de riqueza de espécie com dados de armadilhamento fotográfico
~~

Description:

~~ A funcao faz uma preliminar da comunidade de uma área de estudo baseada em levantamento de armadilhamento fotográfico, calculando o número e a riqueza de espécies em uma área, podendo ser feito separadamente para cada ano de monitoramento na área. Para essa função é importante a instalação de pacotes adicionais como "lubridate", "vegan" e "ggplot2". ~~

Usage:

~~ monitora(filedata=NULL, ID.spp=NULL, sampling.units=NULL, date=NULL, period="all") ~~

Arguments:

~~

filedata	dataframe com os dados de armadilhamento fotográfico, com as identificação das espécies, data dos registros e informações da unidade amostral.
ID.spp	vetor com a identificação das espécies por registros.
sampling.units	vetor com os dados da unidade amostral
date	vetor com as datas dos registros
period	definição do tipo de análise que será feita, "all" - análise conjunta de todos os anos; "annual" - análise por ano

~~

Details:

~~ A Curva de Acumulação de Espécies é gerada para todo área de trabalho ou para cada ano de monitoramento, levando em consideração o total das espécies registradas nas armadilhas fotográficas e é um dos métodos utilizados para saber se a amostragem foi suficiente para obter um número de espécies próximo ao que existe realmente na área estudada. A Riqueza de Espécie pode apresentar diferenças entre a observada e a estimada. Os estimadores de riqueza estimam a quantidade de espécies que se pode encontrar em uma área, sem levar em consideração a quantidade de indivíduos por espécie (abundância). A função estima a riqueza por 4 diferentes estimadores, Chao, Jack1, Jack2, Boot, retornando as informações das médias e quantiles de cada um dos estimadores. A função

considera a média de estimativa pelo Jackknife1 para o gráfico de estimativa de riqueza entre os anos.

~~

Value:

~ a funcao monitora retorna resultados diferentes dependendo das opções do argumento period.

Para o period = "all", retorna:

n.spp : um dataframe com a frequencia das espécies registradas em todos os anos de monitoramento

plot: um gráfico com curva de acumulação de espécie observada e estimada para a área de estudo nos anos de monitoramento

pool.all : um dataframe com as informações da riqueza das espécies (sumário) e estimativas de riqueza de 5 estimadores diferentes (sumário)

hist: um histograma com a frequencia de espécies registradas na área de estudo

Para o period = "annual", retorna:

n.spp : um dataframe com a frequencia das espécies registradas para cada ano de monitoramento

plot: um gráfico com curva de acumulação de espécie observada e estimada para cada ano de monitoramento

pool.all : um dataframe com as informações da riqueza das espécies (sumário) e estimativas de riqueza de 5 estimadores diferentes (sumário) para cada ano de monitoramento

hist: um histograma com a frequencia de espécies registradas para cada ano de monitoramento

...

Warning:

Essa funcao retorna alertas se os seus dados tiverem NA. Faz-se necessário a omissão de NAs para o correto andamento da função.

A função verifica se os pacotes necessários ("lubridate", "vegan" e "ggplot2") já estão instalados, caso não a função instala automaticamente.

....

Author(s):

~~ Livia de Almeida Rodrigues ~~

References:

~~Peroni, N & Hernández, M.I.M. (2011). Ecologia de populações e comunidades. CCB/EAD/UFSC - Florianopolis. ~~

~~ Dias, S.C.(2004). Planejando estudos de diversidade e riqueza: uma abordagem para estudantes de graduação. Acta Scientiarum Biological Sciences, 26(4): 373-379.~~

See Also:

~~~~

## Examples:

```
##Teste da funcao com os dados reais, análise para todos os anos
filedata <-
read.table("http://ecologia.ib.usp.br/bie5782/lib/exe/fetch.php?media=bie578
2:01_curso_atual:alunos:trabalho_final:livrodrigues:dados_monitora.csv",
sep=";", header =TRUE)
monitora(filedata, period="all")
##Teste da funcao com os dados reais, análise para cada ano separadamente
filedata <-
read.table("http://ecologia.ib.usp.br/bie5782/lib/exe/fetch.php?media=bie578
2:01_curso_atual:alunos:trabalho_final:livrodrigues:dados_monitora.csv",
sep=";", header =TRUE)
monitora(filedata, period="annual")

##Teste da funcao com os dados reais, análise para cada ano separadamente
inserindo os dados pelos argumentos em vetores
filedata <-
read.table("http://ecologia.ib.usp.br/bie5782/lib/exe/fetch.php?media=bie578
2:01_curso_atual:alunos:trabalho_final:livrodrigues:dados_monitora.csv",
sep=";", header =TRUE)
monitora(ID.spp=filedata$bin, sampling.units=filedata$Camera.Name.Trap,
date=filedata$Photo.Date, period="annual")
```

1)

não sei se faz sentido, precisa avaliar

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

Permanent link:  
[http://ecor.ib.usp.br/doku.php?id=05\\_curso\\_antigo:r2019:alunos:trabalho\\_final:livrodrigues:start](http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2019:alunos:trabalho_final:livrodrigues:start)

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

