

Myldred Ometto Spinelli



Mestranda em Ecologia - Instituto de Biociências da USP. Orientada do Prof. Dr. Luciano Martins Verdade com o projeto "Uso do espaço e dieta de felídeos de pequeno porte em paisagem silvicultural do Estado de São Paulo, Brasil".

Meus exercícios

Linque para a página com meus exercícios resolvidos [exec](#)

Plano A

Criar uma função chamada dieta que calcule a frequência de ocorrência (FO%) e a porcentagem de ocorrência (PO%) de cada item consumido por cada espécie estudada, assim como a amplitude de nicho (Índice de Levin) para cada espécie e a sobreposição de nicho (Índice de Pianka) entre as espécies.

PO, FO, AMPLITUDE, SOBREPOSIÇÃO → argumentos da função.

ENTRADA: matriz onde as linhas seriam as espécies estudadas e a colunas seriam os itens alimentares (i1,i2,i3,etc) e o número de fezes em que o determinado item foi encontrado (n1,n2,n3,etc).

A função encurtaria as diversas contas que precisariam ser feitas neste tipo de análise, 'entrando' uma matriz com os dados coletados e 'devolvendo' como resultado os cálculos ditos acima, sendo 'devolvidos' também um gráfico para FO e PO (espécies x frequência de ocorrência de cada item) (espécies x porcentagem de ocorrência de cada item). A sobreposição de nicho poderia, se possível, ser representada também pelo Diagrama de Venn.

SAÍDA:

- FO e GRÁFICO (espécies x frequência de ocorrência de cada item)
- PO e GRÁFICO (espécies x porcentagem de ocorrência de cada item)
- AMPLITUDE: B e BA (amplitude e amplitude padronizada)
- SOBREPOSIÇÃO: O (sobreposição) e diagrama de venn (se possível)

Plano B

CALCULADORA DE BARMAN

Criar uma função que devolva uma lista do que o barman irá precisar para uma noite de trabalho de acordo com o número de convidados (x), a duração da festa (horas) e os drinks escolhidos.

Eu iria criar um banco de dados com as receitas (quantidades) de diversos drinks clássicos. Primeiramente, seria necessário digitar o nome da função para que ela retornasse o banco de dado com as receitas que foi criado para que o barman escolhesse quais drinks gostaria de servir em sua festa.

Na função, através dos argumentos, seria possível escolher quais drinks seriam utilizados e em que proporção. O padrão seria que todos os drinks seriam calculados em iguais proporções. Lembrando que o resultado seriam de drinks clássicos, mas por se tratar de uma função do R, seria possível modificá-la incluindo ou excluindo itens de sua preferência.

Base seria: pessoas costumam beber y drinks em z horas.

No argumento: a ordem do drinks dirá sua preferência. Desta forma, se a divisão for um número quebrado, a função irá ciclar os drinks começando do primeiro para o último.

Dúvida: como criar a base de dados com as receitas.

A proposta A é interessante e factível, e será um bom desafio para implementar como função. Sugiro seguir com ela. Você pode começar implementando a função para o cálculo dos índices para uma espécie somente. Se conseguir criar esta função facilmente você pode generalizar para múltiplas espécies e, finalmente, calcular os índices de sobreposição de nicho. A proposta B parece interessante, mas eu deixaria os drinks para alguma outra ocasião... 😊 —

[Leonardo](#)

Help da função

indice.dieta package: nenhum R Documentation

Índice de dieta.

Description:

indice.dieta calcula índices relacionados a dieta, como frequência de

ocorrência (F0%), porcentagem de ocorrência (P0%), amplitude de nicho (índice de Levin) e amplitude de nicho padronizada (índice de Hulbert). O índice lê tabelas padrões utilizadas nesse tipo de estudo, onde uma das colunas deve indicar a quantidade de cada item encontrado nas fezes (i1,i2,i3,etc) e a outra coluna deve indicar o numero de vezes que determinado item foi encontrado nas fezes (n1,n2,n3,etc) - é preciso que exista o título "i" e "n" nas colunas.

As linhas da matriz contem os diferentes itens alimentares encontrados nas fezes da espécie estudada - se estas linhas estiverem nomeadas, os gráficos de saída serão mais informativos, entretanto, se não estiverem nomeadas, os gráficos irão numerá-las apenas.

Usage:

```
indice.dieta(x, indice)
```

Arguments:

x tabela de entrada - deve conter uma coluna nomeada com "i" e outra nomeada com "n"

indice especificar o índice que deseja calcular - opções: "frequencia", "porcentagem", "amplitude" e "amplitude.padronizada" (os argumentos precisam ser escritos entre aspas " ")

Details:

É necessário que as colunas da tabela de entrada sejam nomeadas com "n" e "i" independente da ordem e isto deve ser especificado quando a tabela for enviada ao R. Veja o exemplo: `objeto<-read.table(file="tabela.csv", sep=";", header=TRUE)`.

É necessário que o argumento seja escrito entre aspas " ".

É recomendado (porém não é exigido) que as linhas da matriz sejam nomeadas, pois os gráficos de saída serão mais informativos acerca da dieta da espécie estudada. Veja o exemplo: `objeto<-read.table(file="tabela.csv", sep=";", header=TRUE, row.names=1)`.

Value:

indice.dieta retorna valores e gráficos com os seguintes componentes:

porcentagem porcentagem de ocorrência dos itens consumidos pela espécie estudada - valores em porcentagem - e gráfico do tipo barplot

frequencia frequencia de ocorrência dos itens consumidos pela espécie estudada - valores em porcentagem - e gráfico do tipo barplot

amplitude amplitude de nicho calculada pelo Índice de Levin

amplitude.padronizada amplitude de nicho padronizada calculada pelo

Índice de Hulbert - valores de 0 a 1

Author:

Myldred Ometto Spinelli

myldred.spinelli@gmail.com

myldred.spinelli@usp.br

References:

HURLBERT, S.H. The measurement of niche overlap and some relatives. Ecology, Tempe, v.59, n.1, p.67-77, jan. 1978.

KREBS, C.J. Ecological methodology. New York: Harper Collins Publishers, 1989. 581 p.

Examples:

Baixe o arquivo tabela.csv para rodar os exemplos abaixo:

```
tabela<- read.table(file="tabela.csv", sep=";", header=TRUE, row.names=1) #  
é preciso especificar que a primeira linha é o título das colunas  
(header=TRUE) e é recomendável que a primeira coluna contenha os nomes das  
linhas (itens alimentares) (row.names=1).
```

```
indice.dieta(tabela,"porcentagem") # retorna os valores das porcentagens de  
ocorrência e o gráfico para os diferentes itens alimentares
```

```
indice.dieta(tabela, "frequencia") # retorna os valores das frequências de  
ocorrência e o gráfico para os diferentes itens alimentares
```

```
indice.dieta(tabela, "amplitude") # retorna o valor da amplitude de nicho
```

```
indice.dieta(tabela, "amplitude.padronizada") # retorna o valor da amplitude  
de nicho padronizada
```

Código da função

```
#A função calcula índices relacionados a dieta da espécie estudada, como  
frequência de ocorrência (F0%),  
# porcentagem de ocorrência (P0%), amplitude de nicho (índice de Levin) e  
amplitude de nicho padronizada
```

```
 #(índice de Hulbert).
```

```
 #A entrada é uma matriz padrão para esse tipo de estudo, onde uma das  
 colunas deve indicar a quantidade
```

```
 #de cada item encontrado nas fezes (i1,i2,i3,etc) e a outra coluna deve  
 indicar o numero de vezes que
```

```
 #determinado item foi encontrado nas fezes (n1,n2,n3,etc) - é preciso que  
 exista o título "i" e "n" nas
```

```
 #colunas.
```

```
 #As linhas da matriz contem os diferentes itens alimentares encontrados nas  
 fezes da espécie estudada -
```

```
 #se estas linhas estiverem nomeadas, os gráficos de saída serão mais  
 informativos, entretanto, se não
```

```
 #estiverem nomeadas, os gráficos irão numerá-las apenas.
```

```
 indice.dieta<-function(dados,indice)
```

```
{  
  if (indice=="frequencia")  
  {  
    dados$n[is.na(dados$n)] <- 0 #substitui os NA por 0, pois para este tipo  
 de estudo NAs não são valores faltantes, mas sim observações iguais a 0  
    n<-dados$n #seleciona a coluna onde estão os valores de "n"  
    N<-sum(dados$n) #valor de "N" é a somatória dos valores de "n"  
    F=(n/N)*100 #fórmula que calcula a frequencia de ocorrencia  
    x11() #abre uma janela de grafico  
    midpts <- barplot(1:length(rownames(dados))) #insere o nome das linhas  
 no eixo x de forma central  
    barplot((F),ylab="Frequência de ocorrência", ylim=c(0,100)) #grafico de  
 barras onde é mostrado as frequencias de ocorrencia  
    axis(side=1,labels=rownames(dados), at=midpts, tick=FALSE,las=1) #coloca  
 o nome das linhas (itens alimentares) no eixo x do gráfico. O argumento  
 las=1 diz que o eixo deve ser escrito na horizontal, porém dependendo do  
 número de linhas, é melhor mudar o argumento para las=3 (escrita na  
 vertical)  
    return(F)  
  }  
  if (indice=="porcentagem")  
  {  
    dados$i[is.na(dados$i)] <- 0 ##substituo os NA por 0, pois NAs não são  
 valores faltantes, mas sim observações iguais a 0  
    i<-dados$i #seleciona a coluna onde estão os valores de "i"  
    I=sum(dados$i) #número de itens alimentares é igual a soma de "i"  
    P=(i/I)*100 #fórmula da porcentagem de ocorrência  
    x11() #abro uma janela de gráfico  
    midpts <- barplot(1:length(rownames(dados))) #insere o nome das linhas  
 no eixo x de forma central  
    barplot((P), xlab="Itens alimentares", ylab="Porcentagem de ocorrência",  
 ylim=c(0,100)) #gráfico de barras onde é mostrado as porcentagens de  
 ocorrencia  
    axis(side=1,labels=rownames(dados), at=midpts, tick=FALSE, las=1)  
 #coloca o nome das linhas (itens alimentares) no eixo x do gráfico. O
```

```
argumento las=1 diz que o eixo deve ser escrito na horizontal, porém
dependendo do número de linhas, é melhor mudar o argumento para las=3
(escrita na vertical)
    return(P)
}
if (indice=="amplitude")
{
    dados$i[is.na(dados$i)] <- 0 ##substituo os NA por 0, pois NAs não são
valores faltantes, mas sim observações iguais a 0
    i<-dados$i #seleciona a coluna onde estão os valores de "i"
    I=sum(dados$i) #número de itens alimentares é igual a soma de "i"
    B<-1/(sum((i/I)^2)) # B é a somatória de todos os valores de i/I ao
quadrado.
    return(B)
}
if (indice=="amplitude.padronizada")
{
    dados$i[is.na(dados$i)] <- 0 ##substituo os NA por 0, pois NAs não são
valores faltantes, mas sim observações iguais a 0
    i<-dados$i #seleciona a coluna onde estão os valores de "i"
    I=sum(dados$i) #número de itens alimentares é igual a soma de "i"
    B<-1/(sum((i/I)^2)) #calcula a amplitude de nicho
    BA<-(B-1)/(I-1) #calcula a amplitude de nicho padronizada
    cat("Amplitude de nicho padronizado (valores 0 a 1) \n Valores máximos
-> nicho amplo/mínima especialização \n Valores mínimos -> nicho
estreito/máxima especialização \n")
    return(BA)
}
}
```

Arquivo para download

tabela

From:

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

Permanent link:

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

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