

Propostas de Trabalho Final

Proposta 1 Desenvolver uma função para o cálculo do modelo de competição presa-predador de Lotka-Volterra baseado nas equações para as duas espécies: $dx/dt = x(\alpha - \beta y)$ e $dy/dt = -y(\gamma - \delta x)$. Na entrada da função o usuário deverá fornecer como argumento os valores de Alfa, Beta, Gamma e Delta. Aonde: Alfa é a taxa linear de crescimento da presa e beta a taxa que a presa é comida pelo predador. E o predador come a presa em uma taxa (delta), e tem uma taxa de mortalidade (gamma). Outro argumento da função será o tempo aonde essas duas espécies coexistiram. E ao final da função ela irá retornar os valores do modelo em um data frame e o gráfico das populações das duas espécies em função do tempo (t).

Proposta 2 Função para calcular diferentes índices de similaridades de fauna. Os argumento "índice" que estará na função indicará qual o índice que será usado. Esta função calculará os índices de Jaccard, Sorensen e a distancia Euclidiana. Para os índices: Sorensen e Jaccard a entrada na função deve ser $fx(a, b, c, index=J, porc=T, dis=F)$ ou $fx(a, b, c, index=S, porc=T, dis=F)$. Aonde: a = numero de espécies em comum a ambas áreas estudadas b = numero de espécies únicas a primeira área c = numero de espécies únicas a segunda área Se o argumento porc=T ele me retornará o valor em porcentagem, e se dis=T ele retornará além do valor de similaridade o índice de dissimilaridade (1-índice). E quando o objetivo for a distância Euclidiana a função se comportará como $fx(a, b, index=E)$ Aonde o usuário deverá prover vetores de tamanhos iguais para a e b.

Comentários:

A proposta A é interessante, e abre [muitas possibilidades](#) de implementação. Sugiro que a função tenha como argumentos: alfa, beta, gama e delta; tempo final; intervalo de tempo para a integração numérica; e alguns "switches" de qual método numérico utilizar e quais gráficos apresentar (pode ser o gráfico de população ao longo do tempo, o [espaço de fase](#), nenhum ou ambos).

A proposta B está muito rasa. Se você quiser usar o plano B, detalhe mais a proposta. —[André Chalom](#)

TRABALHO FINAL

COMO TRABALHO FINAL FOI UTILIZADO A PROPOSTA 2

```
sim.index = function(a, b, c, index, pct=T, dis=F) #Cálculo de índice de
similaridade
{
  if(index=="J"|index=="j") #Escolha do índice para Jaccard
  {
    iJ = a/(a+b+c)#Formula do índice de Jaccard
    if(pct==T) #Caso o resultado do índice volte em porcentagem
      {pJ = iJ/100 #Objeto com o valor em porcentagem
        if(dis==F) #Não voltar o indice de dissimilaridade (1-D) junto ao
de Jaccard.
          {return(pJ)} #INdice de Jaccard
```

```
        if(dis==T) #Voltar o índice de dissimilaridade junto ao de
Jaccard
        {dJ = (1 - iJ) #Índice de dissimilaridade
          dJ = dJ/100 #Volta o índice de dissimilaridade em
porcentagem tb
          return(c(pJ, dJ))} #Jaccard e (1-J)
    if(pct==F) #Caso o valor não volte em porcentagem
      { if(dis==F) ##Voltar so o índice de Jaccard
        {return(iJ)} #Índice de Jaccard
        if(dis==T) #Voltar junto ao índice de Jaccard, o índice de
dissimilaridade (1-J)
          {dJ = (1 - iJ) #(1-J)
            return(c(iJ, dJ))} #Jaccard e (1-J)
      }
}

if(index=="S"|index=="s") #Escolha do índice para Sorensen
{
  iS = 2*a*((2*a)+b+c) #Índice de Sorensen
  if(pct==T) #Caso o resultado do índice volte em porcentagem
  {pS = iS/100 #Objeto com o valor em porcentagem
    if(dis==F) #Não voltar o índice de dissimilaridade (1-S) junto ao de
Sorensen.
    {return(pS)} #Índice de Sorensen
    if(dis==T) #Voltar o índice de dissimilaridade junto ao de Sorensen
    {dS = (100 - iS) #Índice de dissimilaridade
      dS = dS/100 #Volta o Índice de dissimilaridade em porcentagem tb
      return(c(pS, dS))} #Sorensen e (1-S)
    if(pct==F) #Caso o valor não volte em porcentagem
    { if(dis==F) ##Voltar so o índice de Sorensen
      {return(iS)} #Índice de Sorensen
      if(dis==T) #Voltar junto ao índice de Sorensen, o índice de
dissimilaridade (1-J)
        {dS = (100 - iS) #(1-S)
          return(c(iS, dS))} #Jaccard e (1-S)
        }
    }

if(index=="E"|index=="e") #Distancia Euclidiana, neste caso não existe
argumento "c" e, "a" e "b" devem ser vetores do mesmo tamanho
{
  ED = sqrt(sum((a-b)^2)) #Distancia Euclidiana, no caso "a" e "b" são
vetores com as abundâncias das n espécies nos dois ambientes de comparação
  if(length(a)!=length(b)) #Caso tamanho e A e B forem diferentes
  {stop
    return("Erro: Vetores de tamanhos diferentes")} #Para a função

  if(length(a)==length(b)) #Caso A e B sejam vetores do mesmo tamanho
  {
    return(ED) #Retorna a distancia Euclidiana
  }
}
```

```
}  
  
}
```

Em anexo o script da função [trabalho_final_-_final.r](#)

HELP DA FUNÇÃO

sim.index package:unknown R Documentation

Function to calculate similarity indexes

Description:

A simple function to calculate similarity index measures. Built in this function you can easily calculate Jaccard and Sorensen similarity indexes, and also the Euclidean distance. This function has arguments to return the dissimilarity indexes for Jaccard (1 - Jaccard) and Sorensen (1-Sorensen) as well. You can also have the similarity index in percentage through one of the arguments.

Usage:

```
sim.index((a, b, c, index, pct=T, dis=F)
```

Arguments:

a number of species common to (shared by) quadrats, in case the chosen index be "E", this argument should provide a vector containing abundance of Nth species in the quadrat.
b number of species unique to the first quadrat, in case the chosen index be "E", this argument should provide a vector containing abundance of Nth species in the quadrat.
c number of species unique to the second quadrat.
index Type of index chosen to analyses. It could be "j" or "J" for Jaccard, "s" or "S" for Sorensen and "e" or "E" for Euclidean distance.
pct If True it will return the index in percentages.
dis If True it will return the dissimilarity measure as well.

Details:

#NOTE#

The arguments pct and dis are not valid when using Euclidean Distance as index.

Warning:

When using the Euclidean Distance as index both of the vectors should be the same length.

Author(s):

Silva, A. R.

References:

University of Arizona

<http://ag.arizona.edu/classes/rnr555/lecnotes/10.html>

Examples:

```
#For Jaccard index without the dissimilarity measure
sim.index(a, b, c, index="J", pct=T, dis=F)
#For Euclidean Distance
sim.index(a, b, index="E")
#For Sorensen index with the dissimilarity measure
sim.index(a, b, c, index="S", pct=T, dis=T)
```

From:

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

Permanent link:

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

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