Função final

Plano A: Tabela de Vida

Página de Ajuda

```
 tabvida                          package:unknown                    R
Documentation

CONTRUÇÃO DE TABELAS DE VIDA

Description:

   Esta função serve para construir tabelas de vida de espécies e deve ser
utilizada
   para descrever os estágios de desenvolvimento de quaisquer espécies em
que a idade
   de vida esteja representada numericamente (dias, anos, tamanhos, etc.). A
função
   retorna uma tabela de vida, que opcionalmente pode ser salva
(data.frame), e um
   gráfico da curva de sobrevivência da espécie analisada.

Usage:

   tabvida(x,x1,idade,Nxi=FALSE,save=TRUE)

Arguments:

   x: Vetor numérico contendo a primeira idade de cada classe de idade, por
exemplo:
   o primeiro dia de uma classe, o primeiro ano de uma classe, o primeiro
tamanho de
   uma classe, etc.. No exemplo de *Phlox drummondii* (tabela de vida no
trabalho de:
   Leverich & Levin, 1979) a primeira classe de idade é representada pelo
“0” e a
   classe seguinte pelo "63", isso significa que a 1º classe vai de "0 até
62 dias"
   e que a 2º classe começa no dia "63" (o primeiro dia “x” da 2º classe).

   x1: Vetor numérico contendo a primeira idade seguinte a idade “x”. No
exemplo
   de *Phlox drummondii*, na 1º classe, a primeira idade seguinte a “x” é
   representada pelo “63”.

   idade: Data.frame contendo o dia, ano, tamanho ou quaisquer outros
```

```
valores que
   possam representar a idade de morte dos indivíduos. Cada linha do
data.frame
   deve representar um indivíduo e conter a idade de morte do indivíduo.

   Nxi: Número total de indivíduos que o estágio de desenvolvimento é
acompanhado
   ao longo do tempo. Deve ser um número inteiro que corresponda ao total de
   indivíduos a partir do qual a tabela de vida é construída.
   save: Opção para salvar a tabela de vida que é calculada. O default da
função
   é =TRUE indicando que a tabela deve ser salva, caso não exista interesse
em
   salvar a tabela deve ser informado save=FALSE.
Details:
   Esta função pode ser utilizada para construir tabelas de vida de coortes
e não
   coortes. Será uma tabela de coorte quando todos os indivíduos que nascem
dentro
   de um mesmo intervalo são acompanhados do nascimento até a morte do
último
   sobrevivente. Caso seja uma tabela de coorte "Nxi" não precisa ser
informado
   (o default da função é para uma tabela de coorte: Nxi=FALSE), porém, caso
não
   seja uma coorte "Nxi" deve ser informado, pois é necessário saber o total
de
   indivíduos inicial a partir do qual a tabela é construída.
   Os parâmetros calculados na tabela de vida foram estabelecidos e
definidos
   conforme as descrições que constam no trabalho de Leverich & Levin (1979)
e
   no livro de Begon et al. (2007).
   O argumento “x” deve ser um vetor que contém zero na primeira posição,
pois
   o primeiro dia da primeira classe sempre será zero (é o dia inicial); “x”
   deve estar em ordem crescente, pois a idade aumenta com o passar do
   desenvolvimento; e as idades de “x” devem ser menores do que as idades de
   “x1”, pois “x1” delimita o final das classes de idade (intervalo fechado
em “x1”).
   O argumento “x1” não deve conter zero em nenhuma posição do vetor e deve
estar
   em ordem crescente, pois a idade aumenta com o passar do desenvolvimento.
   O argumento "idade" deverá ser da classe data.frame, pois a forma mais
usual e
   fácil de acompanhar o desenvolvimento dos indivíduos é anotando a idade
de morte
   em uma planilha de trabalho.
Value:
```

```
   Dx: Duração dos intervalos ou classes de idades (x-x1), que descrevem os
estágios
   de desenvolvimento das espécies, por exemplo, para *Phlox drummondii* o
primeiro
   intervalo de idade possui 63 dias (de 0 até 62).
   Nx : Número total de indivíduos vivos no dia "x". O dia "x" representa o
primeiro
   dia de uma classe de idade, por exemplo, para *Phlox drummondii* o
primeiro
   dia da 1º classe de "x" é igual a zero e, nesse caso, o total de
indivíduos é
   igual a 996, que representa o total de indivíduos a partir do qual o
   desenvolvimento foi acompanhado. A primeira 1º linha da coluna “Nx” será
igual a
   “Nxi” e a 2º linha será igual a “Nxi – dxi”... E assim sucessivamente até
calcular
   todas as linhas.
   lx: Proporção de sobreviventes no dia "x" (1*Nx/Nxi).
   dx: Número total de indivíduos mortos por cada intervalo de idade de
vida. A idade
   de morte de cada indivíduo deve ser informada no data.frame "idade" e, a
partir
   disso, conta-se o número de indivíduos que morreram em cada intervalo.
   p.dx: Proporção de mortos durante cada idade "dx", que corresponde a
diferença entre
   os sucessivos valores de lx (lxi – lxii...). Este parâmetro é calculado
no livro de
   Begon et al. (2007).
   qx: Taxa de mortalidade específica por idade ((dx/Nx)/Dx).
   Gráfico da curva de sobrevivência: plota no eixo y “lx”, que corresponde
à proporção
   de sobreviventes no dia “x”, e plota no eixo x a idade no dia “x”.
Warning:

   A função retornará erro:
   Se “x” não for um vetor da classe numérica com zero na primeira posição
do vetor, se
   os componentes de “x” não estiverem em ordem crescente e se as idades de
“x” forem
   maiores do que as idades de “x1”;

   Se “x1” não for um vetor da classe numérica, se “x1” possuir zero em
alguma de suas
   posições e se “x1” não estiver em ordem crescente;

   Se “idade” não for da classe data.frame;

   Se “x” e “X1” não possuírem o mesmo comprimento.
Author(s):
   Débora Samira Gongora Negrão
   e-mail: deborasamira@gmail.com
```

```
References:

Begon, M.; Townsend, C.R.; Harper, J.L.. Porto Alegre: Artmed, 2007.752p.

Leverich, W. J. & Levin, D. A. 1979. Age-Specific Survivorship and
Reproduction in Phlox drummondii. The American Naturalist, 113(6), 881–903.

Examples:

   Dados para exemplo:
   x<-
c(0,63,124,184,215,231,247,264,271,278,285,292,299,306,313,320,327,334,341,3
48,355)
x1<c(63,124,184,215,231,247,264,271,278,285,292,299,306,313,320,327,334,341,
348,355,362)
   idade (idade.txt)
   Ler data.frame: idade<-read.table("idade.txt", header=T, sep="\t",
as.is=T)
   Obtenha a tabela de vida:
   Coorte: tabvida(x, x1, idade, Nxi=FALSE, save=TRUE)
   Não coorte: tabvida(x, x1, idade, Nxi=996, save=TRUE)
   Os dados fornecidos no help foram construídos para calcular a tabela de
vida de
   *Phlox drummondii*, uma espécie de ciclo anual. A construção do
data.frame contendo
   os dias de morte dos indivíduos (idade) retrata apenas o intervalo das
classes de
   idades, ou seja, o número de indivíduos que morreram em um intervalo,
mas, não
   condiz com o dia de morte real dos indivíduos de *Phlox drummondii*, veja
o código
   utilizado para construir o data.frame idade (tabvida_entrada.r).
```

idade:idade.txt tabvida_codigo_entrada.r

Código tabvida

```
##############################
####FUNÇÃO TABELA DE VIDA####
##############################

tabvida<-function(x,x1,idade,Nxi=FALSE,save=TRUE){ #inicia a função
#verifica os argumentos de entrada
if(class(x)!="numeric")#verifica se "x" é numérico
{
stop("x precisa ser numérico!")#PARAR se "x" não for numérico
}
if(x[1]!=0)#verifica se o 1º elemento de "x" é !=0
{
```

```
stop("o 1º elemento de x precisa ser ==0")#PARAR se o 1º elemento de "x" não
for =0
}
for(a in 1:(length(x)-1))#abre fluxo for com índice a indo de 1 até o número
de linhas de "x"-1
{
if(x[a]>x[a+1])#verifica as condições: se o 1º elemento de "x" for maior do
que o 2º elemento de "x" (e assim sucessivamente até verificar todos os
elementos):
{
stop("x precisa estar em ordem crescente")#PARAR quando "x" não estiver em
ordem crescente
}
}
if(class(x1)!="numeric")#verifica se "x1" é numérico
{
stop("x1 precisa ser numérico!")#PARAR se "x1" não for numérico
}
for(b in 1:length(x1))#abre fluxo for com o índice b indo de 1 até o número
de linhas de "x1"
{
if(x1[b]==0)#se "x1" possuir algum zero, então:
{
stop("x1 não deve conter zeros")#PARAR "x1" não deve conter zero em nenhuma
de suas posições
}
}
for(c in 1:(length(x1)-1))#abre fluxo for com o índice c indo de 1 até o
número de linhas de "x1"-1
{
if(x1[c]>x1[c+1])#verifica as condições: se o 1º elemento de "x1" for maior
do que o 2º elemento de "x1" (e assim sucessivamente até verificar todos os
elementos):
{
stop("x1 precisa estar em ordem crescente")#PARAR quando "x1" não estiver em
ordem crescente
}
if(length(x)!=length(x1))#verifica se "x" tem o mesmo comprimento de "x1",
caso não:
{
stop("x e x1 precisam ter o mesmo tamanho")#PARAR se o comprimento de
"x!=x1"
}
for(i in 1:length(x))#abre fluxo for com o índice i indo de 1 até o número
de linhas de "x"
{
if(x[i]>x1[i])#se os elementos de "x" forem maiores do que "x1":
{
stop("os elementos de x precisam ser menores do que os de x1")#PARAR os
elementos de "x" devem ser menores do que os elementos de "x1"
}
```

```
}
if(class(idade)!="data.frame")#verifica se "idade" é um data.frame
{
stop("idade precisa ser da classe data.frame!")#PARAR se "idade" não for um
data.frame
}
if(Nxi!=FALSE)#se "Nxi" é !=FALSE, então, não é uma tabela de coorte, logo:
{
if(class(Nxi)!="numeric")#"Nxi" deve ser informado pelo usuário e deve ser
numérico, caso não:
{
stop("Nxi precisa numérico")#PARAR se "Nxi" não for numérico
}
}
if(Nxi!=FALSE)#se "Nxi" é !=FALSE, então, não é uma tabela de coorte, logo:
{
if(Nxi==0)#verifica se "Nxi" é =0
{
stop("Nxi precisa ser != de zero")#PARAR se "Nxi" for =0
}
}
if(Nxi!=FALSE)#se "Nxi" é !=FALSE, então, não é uma tabela de coorte, logo:
{
if(Nxi!=round(Nxi))#verifica se "Nxi" é um número inteiro
{
stop("Nxi precisa ser um número inteiro")#PARAR se "Nxi" for != de inteiros
}
}
}
#calcula os parâmetros da Tabela de Vida
Dx<-abs(x-x1)#calcula o número de dias entre os intervalos (em módulo)
{
if(Nxi==FALSE)#então é uma coorte: qual o total de indivíduos?
{
Nxi<-nrow(idade)#cada linha do data.frame "idade" é um indivíduo (conta o
número de linhas, usa nrow por ser data.frame)
}
}
#cálculo "dx"
n<-1 #cria contador "n" para percorrer os vetores "x", "x1" e "dx"
dx<-rep(0, length(x))#cria vetor "dx" com zeros e possuindo o tamanho do
comprimento ="x" (length por ser vetor class numeric)
for(d in 1:nrow(idade))#abre o fluxo for com o índice d indo de 1 até o
número de linhas de "idade"
{
if(x[n]<=idade[d,1] && idade[d,1]<x1[n])#verifica as condições: se "idade" é
maior ou igual a "x" e menor do que "x"
{
dx[n]<-dx[n]+1 #guarda no vetor "dx" o total de "idades" que satisfazem "x"
e "x1"
```

```
}else{ #então
n<-n+1 #contador n = n+1 para olhar para a próxima posição (ou linha) dos
vetores "x" e "x1"
dx[n]<-dx[n]+1 #guarda no vetor "dx" o total de "idades" que satisfazem "x"
e "x1"
}
}
#cálculo "Nx"
Nx<-rep(0, length(x))#cria vetor "Nx" com zeros e possuindo o tamanho do
comprimento ="x"
{
if(Nxi==FALSE)#se for uma tabela de coorte:
{
Nx[1]<-nrow(idade)#"Nxi" é = ao total de linhas do data.frame "idade"
}
if(Nxi!=FALSE)#se não for uma coorte:
{
Nx[1]<-Nxi#"Nxi" é informado pelo usuário e guardado na primeira posição de
"Nx"
}
for(e in 2:length(x))#abre o fluxo for com o índice e indo de 2 até o número
de linhas de "x"; a primeira posição já está ocupada por "Nxi"
{
Nx[e]<-Nx[e-1]-dx[e-1]#calcula Nx (1º elemento de "Nx" - o 1º elemento de
"dx"... e assim sucessivamente)
}
}
#cálculo "lx" e "qx"
lx<-(1*Nx/Nx[1])#calcula "lx"
lx<-round(lx,4)#arredonda para 4 casas decimais
qx<-((dx/Nx)/(Dx))#calcula "qx"
qx<-round(qx,4)#arredonda para 4 casas decimais
qx[is.na(qx)]<-0 #última linha da coluna "qx" pode retornar NaN (0/0=NaN),
para não confundir como um dado faltante substitui por zero
#cálculo "p.dx"
p.dx<-rep(0, length(lx))#cria "p.dx" com zeros e possuindo o tamanho do
comprimento ="x"
for(f in 1:(length(lx)-1))#abre o fluxo for com o índice f indo de 1 até o
número de linhas de "lx"-1 (o número de repetições do fluxo será "lx"-1)
{
p.dx[f]<-abs(lx[f]-lx[f+1])#calcula "p.dx" (1º elemento de "lx" - o 2º
elemento de "lx"... e assim sucessivamente)
}
p.dx<-round(p.dx,4)#arredonda para 4 casas decimais
#gráfico da curva de sobrevivência
par(cex=1.2)#aumenta todas as fontes e pontos
plot(x, lx,cex=1.2,type="o",xlab="Idade (em x)",ylab="Sobrevivência
(lx)",col="grey") #plota o gráfico
#salva data.frame
tabvida<-data.frame(x,x1,Dx,Nx,lx,dx,p.dx,qx)#salva todas as colunas da
tabela de vida em um data.frame
```

```
if(save==TRUE)#se save=TRUE
{
write.table(tabvida,"tabvida.txt",sep="\t", col.names = NA, row.names =
TRUE)#exporta tabvida
}
return(tabvida)#retorna o data.frame
}
```

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

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2018:alunos:trabalho_final:deboranegrao:funcao_final&rev=1597223093

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