

Pedro Dias



Graduando de bacharelado em Ciências Biológicas pela Universidade Federal de São Carlos, campus Sorocaba. Tem experiência na área de Ecologia, com ênfase em Microbiologia Ambiental e Ictiologia e na área de Zoologia, com ênfase em Taxonomia de Grupos Recentes. Atualmente, desenvolve um projeto sobre a dinâmica da comunidade de peixes da zona de arrebentação na ilha do Prumirim em Ubatuba, SP.

pedrodias888@icloud.com

[exec](#)

Proposta final com as alterações sugeridas pela Dra. Sara Mortara

Função Cohort Life Table Calculations

`colife.calc(x,y,colifeout=c("all","null","mortality","survivor"))`

O objetivo da função é produzir uma tabela de vida de coorte (dataframe) para uma espécie anual. Dessa forma, o usuário colocaria no argumento **x** da função um vetor com o número de sobreviventes (ax) referentes as classes de idade (x-x') da espécie analisada em seu projeto e, no argumento **y**, outro vetor com o número total de descendentes produzidos durante cada período (Fx). A partir do número de sobreviventes, será calculado a) a proporção da coorte original sobrevivente no dia x (lx), b) a proporção da coorte original morta durante o intervalo (dx), c) a taxa de mortalidade por dia (qx), d) o log na base 10 de lx e e) a força diária de morte (kx). A partir do número de descendentes produzidos durante cada período (Fx), será calculado g) o número médio de sementes produzidas por cada indivíduo sobrevivente (mx) e h) l_xm_x. Esses vetores calculados serão organizados em um dataframe como o representado pela imagem a seguir e, através do comando `return ()`, será retornado o dataframe e a taxa reprodutiva líquida (Ro) através da equação

$$Ro = \sum l_x m_x.$$

Por fim, serão elaborados dois gráficos `plot()`: i) a taxa de mortalidade diária específica por idade (qx) e a intensidade de mortalidade (kx) serão representadas por duas retas segundo a classe de idade (eixo x) e ii) a curva de sobrevivência: variação log₁₀lx (eixo y) em relação a classe de idade (eixo x). O usuário da função, através do argumento **colifeout**, poderá escolher como output todos os gráficos ("all"), nenhum ("null"), apenas o gráfico **i** ("mortality") ou apenas o gráfico **ii** ("survivor"), sendo que ele poderá escrever o argumento por extenso ou apenas a primeira letra.

Proposta A

Função Cohort Life Table Calculations

colife.calc(x,y)

O objetivo da função é produzir uma tabela de vida de coorte (dataframe) para uma espécie anual. Dessa forma, o usuário colocaria no argumento **x** da função um vetor com o número de sobreviventes (ax) referentes as classes de idade (x-x') da espécie analisada em seu projeto e, no argumento **y**, outro vetor com o número total de descendentes produzidos durante cada período (Fx). A partir do número de sobreviventes, será calculado a) a proporção da coorte original sobrevivente no dia x (lx), b) a proporção da coorte original morta durante o intervalo (dx), c) a taxa de mortalidade por dia (qx), d) o log na base 10 de lx e e) a força diária de morte (kx). A partir do número de descendentes produzidos durante cada período (Fx), será calculado g) o número médio de sementes produzidas por cada indivíduo sobrevivente (mx) e h) l_xm_x. Esses vetores calculados serão organizados em um dataframe como o representado pela imagem a seguir e, através do comando return (), será retornado o dataframe e a taxa reprodutiva líquida (Ro) através da equação

$$Ro = \sum l_x m_x.$$

Por fim, serão elaborados dois gráficos plot(): i) a taxa de mortalidade diária específica por idade (qx) e a intensidade de mortalidade (kx) serão representadas por duas retas segundo a classe de idade (eixo x) e ii) a curva de sobrevivência: variação loglx (eixo y) em relação a classe de idade (eixo x).



Olá Pedro, está legal a proposta porém ainda é bastante simples. Apenas uma série de contas a serem feitas. Sugiro que você coloque um argumento para que o usuário decida se quer ou não plotar os gráficos, podendo ser plotar todos, nenhum, apenas o 1 ou apenas o 2. Isto já adiciona algum nível de dificuldade. Se conseguir pensar em algum outro incremento, mande bala!

— [Sara Mortara](#)

Proposta B

Função Rain Water Collector (RWC)

RWC(zone="Sul", roof.area=100, people=1)

A captação da água da chuva e seu armazenamento em cisternas pode ser útil não só em tempos de estiagem mas também pode contribuir na redução da conta de água tratada pelas companhias de saneamento.

A função conterà um banco de dados no formato dataframe com a precipitação mensal (mm) dos últimos 10 anos das zonas norte, sul, leste, oeste e central do estado de São Paulo. Os índices pluviométricos serão obtidos no site do DAEE e, para melhorar a acurácia e precisão dos dados, o índice pluviométrico mensal dos meses de janeiro a dezembro de cada zona será composto pelos

dados provenientes dos seguintes postos (estação pluviométrica):

Zona norte

- * Santana
- * Jaçanã
- * Vila Maria
- * Freguesia do Ó
- * Perus
- * Piritiba

Zona sul

- * Santo Amaro
- * Capela do Socorro
- * Campo Limpo
- * Jabaquara

Zona leste

- * Mooca
- * Vila Prudente
- * Aricanduva
- * Penha
- * São Miguel Paulista
- * Itaquera
- * São Mateus
- * Guaianazes

Zona oeste

- * Pinheiros
- * Lapa
- * Butantã

Centro

- * Sé
- * Vila Mariana
- * Ipiranga
- * Afonso Pena

Posteriormente, será obtida a média da precipitação mensal referente a zona determinada pelo usuário no argumento **zone** da função. Levando em consideração a variação da precipitação mensal dentro de cada zona, o primeiro produto será a) um gráfico plot () das médias mensais com os respectivos intervalos de confiança a 95% dispostos em linhas a partir do ponto da média mensal. Será determinado o potencial de captação nos meses de janeiro a dezembro através da equação

Potencial de captação=área do telhado (m²) x índice pluviométrico mensal (mm) x runoff

originando o argumento **roof.area**, no qual o usuário deverá determinar a área do telhado (m²) no

qual será feita a coleta de água. A determinação do volume da cisterna é de fundamental importância para que a captação seja máxima e também para que não seja desperdiçado dinheiro numa cisterna cujo volume máximo nunca é atingido. Dessa forma, através da determinação do argumento **people**, na qual o usuário informará a quantidade de pessoas residentes, será determinado o consumo mensal na residência, visto que, de acordo com os dados mundiais, o gasto médio de água tratada e encanada é em torno de 5,4 m³/pessoa/mês (Fundação PROCON SP). Com efeito, com o potencial de captação e o consumo mensais, poderá ser determinado, através da soma acumulativa da água excedente de cada mês, o volume da cisterna mais indicado para o caso. Por fim, será elaborado b) um gráfico barplot() com a captação e o consumo nos meses de janeiro a dezembro além de uma reta desta diferença acumulada num eixo secundário. Além disso, o comando return() irá retornar, o volume mais indicado da cisterna e quanto será economizado por ano em volume de água e monetariamente.

OBS: runoff é o coeficiente de escoamento superficial. Para telhados perfeitos sem fuga utiliza-se de 0,7 a 0,95. Para a função, será 0,8.

Olá novamente,

A idéia está bem explicada e você já fez um bom levantamento sobre o tema. Contudo, a função está muito simples. A implementação envolve bastante trabalho a priori, procurando os dados e as equações. Uma vez que isto esteja pronto, a função se resume apenas a uma série de contas. Gostaríamos de ver uma função mais complexa. Você pode incrementar sua idéia inicial ou propor algo inteiramente novo. Vale a pena dar uma olhada nas propostas dos anos interiores!

Fico no aguardo de um plano B.

—[Mauro Sugawara](#)

Tudo bom Mauro,

Concordo com você. A função está bem simples mesmo. Essa proposta B era a que eu estava menos disposto a fazer, deve ser por isso que eu não me empenhei muito nela. Assim facilitaria sua decisão em prol da minha escolha.rs No entanto, creio que você não viu minha proposta A referente a criação de uma tabela de vida de coorte, pois a mesma encontra-se sem crítica.

Obrigado

Olá Pedro, de novo a proposta é apenas uma conta. Brincar

com as saídas da função (tanto gráficas como sumário estatístico) pode ser um bom caminho para adicionar desafios. Outra coisa seria fazer o usuário dizer em que região está e a função buscar em algum repositório on line os dados necessários para o cálculo. Isto é sempre divertido! Sugiro que você siga com a proposta que mais toca o seu coração e trabalhar para incorporar algum desafio em termos de programação. — [Sara Mortara](#)

Trabalho Final

Help da função Cohort Life Table Calculations

OBS: A tabela, o R_0 e os gráficos provenientes do exemplo referente a espécie *Phlox drummondii* (ver o item **Examples** no help da função) podem ser conferidos na tabela 4.1 (página 98) e figura 4.7 (página 100) do livro Ecologia: de indivíduos a ecossistemas (Begon et al., 2007).

colife.calc

package:nenhum

R Documentation

Cálculo dos componentes (l_x , dx , q_x , $\log_{10}l_x$, k_x , m_x e l_xm_x) referentes a tabela de vida de coorte para espécies anuais, assim como a taxa reprodutiva líquida (R_0) da espécie.

Description:

A função `colife.calc` é uma função que calcula, através do número de sobreviventes de uma única coorte separados categoricamente pela diferença do intervalo de idade em dias ($x-x'$), a proporção da coorte original sobrevivente no período x (l_x), a proporção da coorte original morta durante o intervalo (dx), a taxa de mortalidade por dia (q_x), o $\log_{10}(l_x)$ e a força diária de morte (k_x). Além disso, mediante o número total de descendentes produzidos durante cada intervalo (F_x) é também calculado o número médio de descendentes produzidos por cada indivíduo sobrevivente (m_x) e l_xm_x . Por fim, é obtida a taxa reprodutiva líquida (R_0).

Usage:

```
colife.calc(i,x,y,colifeout = "all")
```

Arguments:

<code>i</code>	vetor numérico da diferença do intervalo de idade ($x-x'$) em dias
<code>x</code>	vetor numérico do número de sobreviventes no intervalo x (ax)
<code>y</code>	vetor numérico do número total de descendentes produzidos durante cada intervalo (F_x)

colifeout gráfico de saída, sendo "all", "null", "mortality" ou "survivor"

Details:

Existem dois tipos diferentes de gráficos de saída: i) a taxa de mortalidade diária específica por idade (q_x) e intensidade de mortalidade (k_x) e ii) a curva de sobrevivência (variação de $\log_{10} l_x$ em relação à idade); sendo que dentre as possibilidades de saída especificadas no argumento colifeout encontram-se I) os dois gráficos disponibilizados lado a lado ("all"), II) nenhum dos gráficos ("null"), III) apenas o gráfico i ("mortality") ou apenas o gráfico ii ("survivor").

Value:

Data frame da tabela de vida de coorte para a espécie analisada

Vetor da taxa reprodutiva líquida (R_0)

Warning:

No argumento i deve ser informado a diferença em dias do intervalo de idade. Ex: o primeiro intervalo da espécie *Phlox drummondii*, ou seja, aquele que representa o nascimento de um grupo de indivíduos no mesmo intervalo de 0 a 63 dias, deve ser informado no vetor numérico como 63. O próximo intervalo de 63 a 124 dias, 61 e assim sucessivamente.

No argumento y, referente ao número total de descendentes produzidos durante cada intervalo (F_x), os intervalos cujos indivíduos não produziram prole devem ser registrados obviamente com NA, pois o resultado zero indicaria erroneamente que os indivíduos reprodutivamente ativos não geraram descendentes.

Note:

Embora a função seja destinada para as espécies anuais, ela pode ser usada para espécies iteróparas desde que tomados os devidos cuidados. Uma coorte deve ser reconhecida na população e acompanhada (muitas vezes por vários anos), mesmo que os organismos estejam misturados e coexistindo com as demais coortes, mais velhas ou mais jovens; o que torna o delineamento experimental complexo e muitas vezes inviável.

Author(s):

Pedro Dias Barreto Neto

References:

BEGON, M.; TOWNSEND, C.R.; HARPER, J.L. Ecologia: de indivíduos a ecossistemas - 4.ed.- Porto Alegre: Artmed, 2007. 752 p.

LEVERICH, W.J. & LEVIN, D.A. (1979) Age-specific survivorship and reproduction in *Phlox drummondii*. *American Naturalist*, 113, 881-903p.

Examples:

```
#Dados referentes a espécie //Phlox drummondii//(Leverich e Levin, 1979)
i=c(63,61,60,31,49,14,14,14,14,14,14,0)#intervalo
x=c(996,668,295,190,176,172,167,159,154,147,105,22,0)#ax
y=c(rep(NA,7),53.0,485.0,802.7,972.7,94.8,NA)#Fx
colife.calc(i,x,y,colifeout="all")#retorna uma lista ([[1]]data frame da
tabela de vida de coorte e [[2]]vetor R0) e dois gráficos i e ii dispostos
lado a lado (ver o item details) em uma nova janela
```

Função Cohort Life Table Calculations

```
colife.calc=function(i,x,y,colifeout="all")
{
  tabela=matrix(NA,length(x),10)
  tabela[,1]=i#Coluna referente ao intervalo de dias (x-x'). O último valor
  deve ser zero pois não há um intervalo.
  tabela[,2]=x#Coluna referente ao número de sobreviventes no dia x (ax).
  tabela[,3]=1#Os valores da coluna da proporção da coorte original
  sobrevivente no dia x (lx) necessitam ser padronizados, dessa forma, o valor
  inicial é 1 e todos os demais são produzidos em relação a ele.
  for(ii in 2:length(x))#Calculando os valores da coluna lx.
  {
    tabela[ii,3]=round(((1*tabela[ii,2])/tabela[1,2]),3)
  }
  for(j in 1:(length(x)-1))#Calculando os valores da coluna dx referentes a
  proporção da coorte original morta durante o intervalo.
  {
    tabela[j,4]=round((tabela[j,3]-tabela[(j+1),3]),3)
  }
  tabela[,5]=round(((tabela[,4])/tabela[,3])/tabela[,1]),3)#Calculando os
  valores da coluna qx referentes a taxa de mortalidade por INTERVALO.
  tabela[(length(x)-1),5]=1#No penúltimo intervalo da coluna qx, a taxa de
  mortalidade sempre será 100%.
  tabela[,6]=round((log10(tabela[,3])),2)#Calculando a coluna log10lx
  tabela[length(x),6]=NA#Pois sempre o log10(0)=-Inf
  for(k in 1:(length(x)-1))#(1) Calculando os valores da coluna kx
  referentes a força diária de morte.
  {
    tabela[k,7]=round(log10(tabela[k,2])-log10(tabela[(k+1),2]),3)
  }
  tabela[,7]=round((tabela[,7])/tabela[,1]),3)#(2) Calculando os valores da
  coluna kx referentes a força diária de morte.
  tabela[tabela[,7]==Inf,7]=NA#Retirando os valores infinitos da coluna kx.
  tabela[,8]=y#Coluna referente ao número total de sementes produzidas
  durante cada período.
```

```
tabela[,9]=round((tabela[,8]/tabela[,2]),2)#Cálculo dos valores da coluna
mx referente ao número médio de sementes produzidas por cada indivíduo
sobrevivente.
tabela[,10]=round((tabela[,9]*tabela[,3]),2)#Cálculo da coluna lxx
tabela.final=as.data.frame(tabela)
colnames(tabela.final)=c("intervalo","ax","lx","dx","qx","Log10lx","kx","Fx",
,"mx","lxx")
R0=sum(tabela[,10],na.rm = TRUE)#Cálculo da taxa reprodutiva líquida
INDICES=c("all","null","mortality","survivor")
colifeout=match.arg(colifeout,INDICES)
if(colifeout=="all"){
  x11()
  par(mfrow=c(1,2))
  #A) Gráfico da Taxa de mortalidade diária específica por idade (qx) e
intensidade de mortalidade (kx).
  tab5.graph=tabela[,5]
  tab5.graph[(length(x)-1)]=NA#Nos pontos de dispersão referentes a coluna
qx, o valor 1, referente a taxa de mortalidade de 100%, não precisa estar no
gráfico.
  plot(c(rep(cumsum(tabela[,1]),2)),c(tab5.graph,tabela[,7]),type =
"n",ylab = "qx e kx",xlab="Dias",bty="l",tck=0.01)
  points(cumsum(tabela[,1]),tab5.graph,type="o",pch=16)
  points(cumsum(tabela[,1]),tabela[,7],type = "o",pch=8)
  legend(min(cumsum(tabela[,1]),na.rm =
TRUE),max(c(tab5.graph,tabela[,7]),na.rm = TRUE),c("qx","kx"),lty=c(1,1),pch
= c(16,8))
  #B) Gráfico da curva de sobrevivência
  plot(tabela[,6]~cumsum(tabela[,1]),type ="o",
pch=16,bty="l",tck=0.01,xlab="Dias",ylab="Log10 lx")
}
if (colifeout=="mortality"){
  #A) Gráfico da Taxa de mortalidade diária específica por idade (qx) e
intensidade de mortalidade (kx).
  x11()
  tab5.graph=tabela[,5]
  tab5.graph[(length(x)-1)]=NA#Nos pontos de dispersão referentes a coluna
qx, o valor 1, referente a taxa de mortalidade de 100%, não precisa estar no
gráfico.
  plot(c(rep(cumsum(tabela[,1]),2)),c(tab5.graph,tabela[,7]),type = "n",ylab
= "qx e kx",xlab="Dias",bty="l",tck=0.01)
  points(cumsum(tabela[,1]),tab5.graph,type="o",pch=16)
  points(cumsum(tabela[,1]),tabela[,7],type = "o",pch=8)
  legend(min(cumsum(tabela[,1]),na.rm =
TRUE),max(c(tab5.graph,tabela[,7]),na.rm = TRUE),c("qx","kx"),lty=c(1,1),pch
= c(16,8))
}
if(colifeout=="survivor"){
  #B) Gráfico da curva de sobrevivência
  x11()
  plot(tabela[,6]~cumsum(tabela[,1]),type ="o",
```



```
pch=16,bty="l",tck=0.01,xlab="Dias",ylab="Log10 lx")
}
else{
  (colifeout=="null")
}
return(list(tabela.final,R0))
}
```

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2016:alunos:trabalho_final:pedrodias888:start



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