

Predação

```
predacao<-function(N, Vi, Vi2, Pi, r, s, a, a2, b, b2, q, t, graphic=TRUE)
{
  ##Verificando os parâmetros globais - aparecem em ambos modelos

  if (N!="1" && N!="2") ##Verifica se N foi inserido corretamente
  {
    stop ("N deve ser '1' para uma presa ou '2' para duas presas") ##Se N
diferente de "1" ou "2", função para e exibe mensagem para usuário corrigir
entrada
  }
  if (Vi<1) ##Verifica se valor de Vi é menor que 1
  {
    stop ("Insira um valor de Vi maior ou igual a 1") ##Se valor de Vi é <1,
função para e exibe mensagem para usuário corrigir entrada\\
  }
  if (Pi<1) ##Verifica se valor de Pi é menor que 1
  {
    stop ("Insira um valor de Pi maior ou igual a 1") ##Se valor de Pi é <1,
função para e exibe mensagem para usuário corrigir entrada
  }
  if (r<0 | r>1) ##Verifica se valor de r está entre 0 e 1
  {
    stop ("Insira um valor de r entre 0 e 1") ##Se valor de r é <0 ou >1,
função para e exibe mensagem para usuário corrigir entrada
  }
  if (a<0 | a>1) ##Verifica se valor de a está entre 0 e 1
  {
    stop ("Insira um valor de a entre 0 e 1") ##Se valor de a é <0 ou >1,
função para e exibe mensagem para usuário corrigir entrada
  }
  if (b<0 | b>1) ##Verifica se valor de b está entre 0 e 1
  {
    stop ("Insira um valor de b entre 0 e 1") ##Se valor de b é <0 ou >1,
função para e exibe mensagem para usuário corrigir entrada
  }
  if (q<0 | q>1) ##Verifica se valor de q está entre 0 e 1
  {
    stop ("Insira um valor de q entre 0 e 1") ##Se valor de q é <0 ou >1,
função para e exibe mensagem para usuário corrigir entrada
  }
  if (t<1) ##Verifica se valor de t é menor que 1
  {
    stop("Insira um valor de t maior ou igual a 1") ##Se valor de t é <1,
função para e exibe mensagem para usuário corrigir entrada
  }
#####
#####
```

```
#####Simulando a interação predador-presa para uma única espécie de presa
e uma única espécie de predador#####
#####
#####
if (N == 1) ##Se argumento "N" for igual a "1", roda modelo com apenas uma
espécie de presa e uma de predador
{
  ##Criando os objetos
  Presa<-rep(NA,t) ##Cria o vetor "Presa" com t NAs para guardar o tamanho
populacional da presa ao longo do tempo t
  Presa[1]<-Vi ##Insere o valor inicial da população da presa na posição 1
do vetor "Presa"
  Predador<-rep(NA,t) ##Cria o vetor Predadores com t NAs para guardar o
tamanho populacional do predador ao longo do tempo t
  Predador[1]<-Pi ##Insere o valor inicial da população do predador na
posição 1 do vetor "Predador"
  Tempo<- rep(1:t) ##Cria o vetor "Tempo" de tamanho 1 até t
  ##Calculando os tamanhos populacionais da presa e do predador ao longo
do tempo t
  for(i in 2:t) ##Cria loop for com contador i de 2 até t
  {
    crescimento.V1=r*Vi ##Calcula crescimento da população da presas
    captura.V1 = a*Vi*Pi ##Calcula captura de presas
    crescimento.P = b*Vi*Pi ##Calcula crescimento da população de
predadores
    mortalidade.P = q*Pi ##Calcula mortalidade da população de predadores
    dV1=crescimento.V1-captura.V1 ##Calcula o crescimento final da
população de presas
    dP=crescimento.P-mortalidade.P ##Calcula o crescimento final da
população de predadores
    V1=Vi+dV1 ##Calcula o tamanho da população de presas no próximo tempo
    P1=Pi+dP ##Calcula o tamanho da população de predadores no próximo
tempo
    Vi=V1 ##Atualiza o valor da população de presas
    Pi=P1 ##Atualiza o valor da população de predadores
    Presa[i]<-V1 ##Insere os valores calculados do tamanho populacional da
presa em cada tempo t no vetor "Presa"
    Predador[i]<-P1 ##Insere os valores calculados do tamanho populacional
do predador em cada tempo t no vetor "Predador"
  } ##Fim do loop
  ##Criando o data.frame
  predador.presa<-data.frame(Tempo, Presa, Predador) ##Cria data.frame com
todos os valores das populações de presas e predadores ao longo do tempo t,
unindo os vetores "Tempo", "Presa" e "Predador"
  ##Criando os gráficos
  par(mfrow=c(1,2), mar=c(3.5, 3.5, 3, 1.2), las=1, cex.axis=0.8,
cex.lab=1.1, font.lab=2, mgp=c(2.2,0.6,0))##Cria espaço para 2 gráficos e
ajusta parâmetros gráficos em comum
  matplot(predador.presa[,-1], type = "l", col=c("#CC0066","black"),
lwd=c(3,3), lty=c(1,1), xlab = "Tempo (t)", ylab = "Tamanho da população
```

```

(N)", main="Dinâmica predador-presa") ##Gera o gráfico do tamanho
populacional da presa e do predador x tempo
  legend("topleft", c("Presa","Predador"),
col=c("#CC0066","black"),lwd=c(3,3), lty=c(1,1), bty="n") ##Adiciona legenda
ao gráfico
  plot(predador.presa$Predador~predador.presa$Presa,xlab="N presa",ylab="N
predador", main="Plano de fase", col="#CC0066", type="l", lwd=3, lty=1)
##Gera o gráfico de plano de fase predador x presa
  ##Retorno da função
  return(predador.presa) ##Retorna data.frame da interação predador-presa
para o usuário
}
#####
#####
#####Simulando a interação predador-presa para duas espécies de presas e
uma única espécie de predador#####
#####
#####
  if (N == 2) ##Se argumento "N" for igual a "2", roda modelo com duas
espécies de presa e uma de predador
  {
    #Verificando os parâmetros específicos
    if (Vi2<1) ##Verifica se valor de Vi2 é menor que 1
    {
      stop ("Insira um valor de Vi2 maior ou igual a 1") ##Se valor de Vi2 é
<1, função para e exibe mensagem para usuário corrigir entrada
    }
    if (s<0 | s>1) ##Verifica se valor de s está entre 0 e 1
    {
      stop ("Insira um valor de s entre 0 e 1") ##Se valor de s é <0 ou >1,
função para e exibe mensagem para usuário corrigir entrada
    }
    if (a2<0 | a2>1) ##Verifica se valor de a2 está entre 0 e 1
    {
      stop ("Insira um valor de a2 entre 0 e 1") ##Se valor de a2 é <0 ou
>1, função para e exibe mensagem para usuário corrigir entrada
    }
    if (b2<0 | b2>1) ##Verifica se valor de b2 está entre 0 e 1
    {
      stop ("Insira um valor de b2 entre 0 e 1") ##Se valor de b2 é <0 ou
>1, função para e exibe mensagem para usuário corrigir entrada
    }
    ##Criando os objetos
    Presa.1<-rep(NA,t) ##Cria o vetor "Presa.1" com t NAs para guardar o
tamanho populacional da presa nº1 ao longo do tempo t
    Presa.1[1]<-Vi ##Insere o valor inicial da população da presa nº1 na
posição 1 do vetor "Presa.1"
    Presa.2<-rep(NA,t) ##Cria o vetor "Presa.2" com t NAs para guardar o
tamanho populacional da presa nº2 ao longo do tempo t
    Presa.2[1]<-Vi2 ##Insere o valor inicial da população da presa nº2 na
posição 1 do vetor "Presa.2"

```

```
Predador<-rep(NA,t) ##Cria o vetor "Predador" com t NAs para guardar o
tamanho populacional do predador ao longo do tempo t
Predador[1]<-Pi ##Insere o valor inicial da população do predadores na
posição 1 do vetor "Predador"
Tempo<- rep(1:t) ##Cria o vetor "Tempo" de tamanho 1 até t
##Calculando os tamanhos populacionais das duas espécies de presa e do
predador ao longo do tempo t
for(i in 2:t) ##Cria loop for com contador i de 2 até t
{
  crescimento.V1=r*Vi ##Calcula crescimento da população de presas nº1
  crescimento.V2=s*Vi2 ##Calcula crescimento da população de presas nº2
  captura.V1 = a*Vi*Pi ##Calcula captura de presas nº1
  captura.V2 = a2*Vi2*Pi ##Calcula captura de presas nº2
  crescimento.P = b*Vi*Pi+b2*Vi2*Pi ##Calcula crescimento da população
de predadores
  mortalidade.P = q*Pi ##Calcula mortalidade da população de predadores
  dV1=crescimento.V1-captura.V1 ##Calcula o crescimento final da
população de presas nº1
  dV2=crescimento.V2-captura.V2 ##Calcula o crescimento final da
população de presas nº2
  dP=crescimento.P-mortalidade.P ##Calcula o crescimento final da
população de predadores
  V1=Vi+dV1 ##Calcula tamanho da população de presas nº1 no próximo
tempo
  V2=Vi2+dV2 ##Calcula tamanho da população de presas nº2 no próximo
tempo
  P1=Pi+dP ##Calcula tamanho da população de predadores no próximo tempo
  Vi=V1 ##Atualiza o valor da população de presas nº1
  Vi2=V2 ##Atualiza o valor da população de presas nº2
  Pi=P1 ##Atualiza o valor da população de predadores
  Presa.1[i]<-V1 ##Insere os valores calculados do tamanho populacional
da presa nº1 em cada tempo t no vetor "Presa.1"
  Presa.2[i]<-V2 ##Insere os valores calculados do tamanho populacional
da presa nº2 em cada tempo t no vetor "Presa.2"
  Predador[i]<-P1 ##Insere os valores calculados do tamanho populacional
do predador em cada tempo t no vetor "Predador"
} ##Fim do loop
##Criando o data.frame
predador.presa<-data.frame(Tempo, Presa.1, Presa.2, Predador) ##Cria
data.frame com todos os valores das populações de presas e predadores ao
longo do tempo t, unindo os vetores "Tempo", "Vítimas" e "Predadores"
##Criando os gráficos
layout(matrix(c(1,1,2,3), 2, 2, byrow = TRUE)) ##Cria espaço para 3
gráficos
par(mar=c(3.5, 3.5, 1.5, 1.5), las=1, cex.axis=0.8, cex.lab=1.1,
font.lab=2, mgp=c(2.2,0.6,0))##Ajusta parâmetros gráficos em comum para os 3
matplot(predador.presa[,-1], type = "l", col=c("#CC0066", "#3399CC",
"black"), lwd=c(3,3,3), lty=c(1,1,1), xlab = "Tempo (t)", ylab = "Tamanho da
população (N)", main="Dinâmica predador-presa") ##Gera o gráfico do tamanho
populacional de presas e predador x tempo
```

```
  legend("topleft", c("Presas 1", "Presas 2", "Predador"), col=c("#CC0066",
"#3399CC", "black"),lwd=c(3,3,3), lty=c(1,1,1), bty="n") ##Adiciona legenda
ao gráfico
  plot(predador.presas$Predador~predador.presas$Presas.1,xlab="N presas
1",ylab="N predador", main="Plano de fase", col="#CC0066", type="l", lwd=3,
lty=1) ##Gera o gráfico de plano de fase predador x presas 1
  plot(predador.presas$Predador~predador.presas$Presas.2,xlab="N presas
2",ylab="N predador", main="Plano de fase", col="#3399CC", type="l", lwd=3,
lty=1) ##Gera o gráfico de plano de fase predador x presas 2
  ##Retorno da função
  return(predador.presas) ##Retorna data.frame da interação predador-presas
para o usuário
}
} ##Fim da função
```

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2019:alunos:trabalho_final:cassiasqr:predacao



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