

Gabriel

Bacharel e licenciado em Ciências Biológicas, estou me preparando para o mestrado em Ecologia Vegetal.

Meus Exercícios

Exercício 1
Exercício 2
Exercício 3
Exercício 9 parcial

Proposta de Trabalho Final

Plano A

Criar uma função que demonstra gráficamente o comportamento de populações (número a ser escolhido pelo usuário) em competição a partir dos parâmetros do modelo de competição de Lotka-Volterra. A função irá perguntar ao usuário todos os valores dos parâmetros. Alternativamente poder-se-ia entrar com uma tabela padronizada com esses parâmetros. A partir do modelo inicial em que a capacidade de suporte de cada população é constante, pode-se implementar capacidade de suporte variável com o tempo. Implementar também efeito de perturbação em um dado tempo.

Comentários PI

Muito boa. O R é uma ótima ferramenta para estudar modelos teóricos por simulação. Apenas atenção à distinção entre modelos de crescimento discreto, que vc pode simular com *loops*, e modelo contínuos como sistemas de equação diferenciais do tipo Lotka-Volterra. Neste segundo caso, o usuário definiria os parâmetros das equações diferenciais, que então devem ser integradas no tempo para vc ter os valores dos tamanhos populacionais. O R tem rotinas para isto. Se vc ir por esta via veja o pacote *odesolve*, geral, e o pacote *simecol*, específico para simulações ecológicas. Mas nada contra a solução discreta, apenas colocando os pingos nos i's.

Resposta

Acabei trabalhando com *loops* em um modelo de crescimento discreto. Ainda quero implementar vários detalhes, e possivelmente refazer a função com modelo contínuo, como sugerido.

Plano B

Faltou o plano B.

Resposta

Como eu já havia começado a função, e sou cabeça dura, me empenhei no plano A. Funcionou, a função roda legal! :D

Página de Ajuda

compet package:unknown R Documentation

Simulação do modelo de competição de Lotka-volterra

Description:

Uma função totalmente interativa para simular a competição entre populações através do modelo de competição de Lotka-Volterra. Gera gráficos do comportamento das populações no tempo, com e sem competição. Todos os parâmetros são determinados pelo usuário.

Usage:

```
compet()
```

Details:

Os parâmetros definidos pelo usuário são:

- número de populações a simular;
- tamanho das populações (N);
- capacidade de suporte das populações (K);
- taxa de crescimento intrínseca das populações (r);
- tempo de observação (t);
- coeficientes de competição entre populações.

Warning:

Não simule muitas populações, pois quanto mais populações, mais parâmetros terá que dar à função.

Ainda assim, é possível simular quantas populações quiser.

Note:

A função simula tanto a competição intra como interespecífica.

Além da competição, é possível incluir efeitos da Facilitação. Basta usar coeficientes de competição negativos.

Ainda por implementar:

- possibilidade do usuário entrar com uma tabela com os valores dos

```
parâmetros;  
-entrada dos parâmetros como argumentos da função;  
-capacidade de suporte variável com o tempo;  
-competição intraespecífica para o gráfico sem competição  
interespecífica;  
-impacto de perturbação em algum tempo;  
-retardos de efeito sobre populações.
```

Author(s):

Gabriel Ponzoni Frey

References:

GOTELLI, Nicolas J. Ecologia. Terceira edição. Londrina. Ed. Planta, 2007

Código da Função

```
compet= function()  
  
{  
cat("\n")  
cat("\n")  
cat("\n")  
cat("\n")  
cat("\n")  
cat("\n")  
cat("\n")  
cat("\n")  
cat("\n")  
  
cat("    BEM VINDO A FUNCAO INTERATIVA DO MODELO DE COMPETICAO DE LOTKA-  
VOLTERRA    ")  
cat("\n")  
cat("\n")  
cat("\n")  
cat("A função vai calcular o comportamento das populacoes discretamente no  
tempo,")  
cat("\n")  
cat("e retornara os resultados em forma de graficos sem e com competicao.")  
cat("\n")  
cat("\n")  
cat("Além da competicao, voce podera ver os efeitos da facilitação no  
modelo. Basta seguir as instrucoes!")  
cat("\n")  
cat("\n")  
cat("OBS: As formulas foram obtidas no cap. 5 do livro Ecology de Nicolas J.  
Gotelli.")  
cat("\n")
```

[illegible]

npop

```
#####  
#Qual a capacidade de suporte de cada população?#  
#####  
  
capacidade=c(rep(NA,pop))  
capacidade=matrix(capacidade,1,pop)  
rownames(capacidade)="K"  
  
for(i in 1:pop)  
{  
  capacidade[i]=as.numeric((readline(paste("Capacidade de suporte da  
populacao", i, "? ")  )))  
} # pergunta a capacidade de suporte de cada população  
  
for(i in 1:pop)  
{  
  colnames(capacidade)=paste("K", 1:pop,sep="")  
} #define os nomes das colunas  
  
#capacidade  
  
#####IMPLEMENTAR:  Capacidade de suporte oscila com o tempo?#####  
  
#####  
#Qual a taxa de crescimento intrínseca de cada população (r)?#  
#####  
  
taxa.crescimento=c(rep(NA,pop))  
taxa.crescimento=matrix(taxa.crescimento,1,pop)  
rownames(taxa.crescimento)="r"  
  
for(i in 1:pop)  
{  
  taxa.crescimento[i]=as.numeric((readline(paste("Taxa de crescimento  
intrinseca da população", i, "? ")  )))  
} #define a taxa de crescimento intrínseca de cada população  
  
for(i in 1:pop)  
{  
  colnames(taxa.crescimento)=paste("r", 1:pop,sep="")  
} #define os nomes das colunas
```

```
#taxa.crescimento
```

```
#####  
#Quanto tempo de observação?#  
#####
```

```
tempo=as.numeric(readline("Quanto tempo de observacao? "))  
#tempo
```

```
#####  
#Gerando então a matriz de tamanho populacional (sem competição ainda).#  
#####
```

```
npop.tempo=matrix(npop, nrow=tempo+1, ncol=pop) #cria a matriz de tamanho  
populacional para t tempos  
npop.tempo
```

```
for (i in 2:(tempo+1))  
{  
  npop.tempo[i,]=npop.tempo[i-1,]+(npop.tempo[i-1,]*taxa.crescimento*((capacid  
ade-npop.tempo[i-1,])/capacidade))  
}  
colnames(npop.tempo)=colnames(npop)  
rownames(npop.tempo)=seq(from=0, to=tempo)  
round(npop.tempo)
```

```
#####  
##Definindo os coeficientes de competição#  
#####
```

```
# 1.Quantos coeficientes há?  
#R: numero de coeficientes = numero de populacoes*(numero de populacoes  
-1)  
# 2. Construir uma matriz de tamanho pop x pop)
```

```
coef.comp=matrix(nrow=pop, ncol=pop)
```

```
# 3. Perguntar quais os coeficientes:  
cat("\n")  
cat("\n")  
cat("\n")  
cat("Para facilitacao, coloque coeficientes negativos.")
```

```

cat("\n")
cat("\n")
cat("\n")
for (i in 1:pop)
{ for (j in 1:pop)
  { #aqui entra o efeito da competicao intra-especifica
    #if(readline("Há competição intraespecífica (S ou N)")=="N")
    #{
      #{ if(i==j) ###se não houver competição intraespecifica, ligar
este if
        #{
          #coef.comp[i,j]= 0
        }
      #else
      #{
        if(i==j)
        {
          cat("    Comepeticao INTRAespecífica    ")
        }
        coef.comp[i,j]= as.numeric((readline(paste("Coeficiente de
competição entre as populações ", i, "e", j, "? ")    )))
      }
    }
  }
}

coef.comp ##### FUNCIONAAAAA :D #####

#####
#Populações em competição#
#####

competindo=matrix(npop, nrow=tempo+1, ncol=pop) #cria a matriz de tamanho
populacional para t tempos
competindo
colnames(competindo)=colnames(npop)
rownames(competindo)=seq(from=0, to=tempo)

#matriz de perdas: calcula quanto cada populacao perde por unidade de tempo
perdas=matrix(npop,ncol=pop, nrow=tempo+1)
perdas

for (g in 1:pop)
{ for(i in 1:(tempo+1))
  {
    perdas[i,g]=sum(coef.comp[,g]*competindo[i,])
  }
}

```

```
}  
round(perdas)  
  
#incluindo as perdas  
for(i in 2:(tempo+1))  
{  
#competindo[i,]=competindo[i-1,]+(competindo[i-1,]*taxa.crescimento*((capaci  
dade-competindo[i-1,])/capacidade)) # formula sem as perdas  
competindo[i,]=competindo[i-1,]+(competindo[i-1,]*taxa.crescimento*((capacid  
ade-competindo[i-1,]-perdas[i-1,])/capacidade))  
}  
  
colnames(competindo)=colnames(npop)  
competindo  
  
#npop.tempo  
  
#####  
#          Perturbação# #IMPLEMENTAR          #  
#Perturbação em que tempo? Qual a magnitude?#  
#####  
  
#Gráficos estáticos  
#Implementar com graficos animados  
  
x11(width=60, height=60) #faz uma janela bem grande para melhor visualização  
dos gráficos.  
par(mfrow = c(2, 1))  
for(g in 1:pop)  
{  
    plot(x=as.numeric(rownames(npop.tempo)), y=npop.tempo[,g],type="l",  
bty="l", xlim=c(0,tempo+1), ylim=c(0,max(npop.tempo+5, competindo+5)),  
xlab="tempo", ylab="Nº de indivíduos")  
    lines(x=as.numeric(rownames(npop.tempo)), y=npop.tempo[,g], col=g)  
    title(main="Comportamento sem competição")  
    par(new=TRUE)  
}  
    par(new=FALSE)  
par(xpd=NA) #permite escrever fora das áreas do plot  
legend(x=0, y=-max(npop.tempo,competindo)/3,  
legend=colnames(npop),lty=1,col=1:pop, bty="n", ncol=3) #coloca a legenda  
entre os dois gráficos
```



```
for(g in 1:pop)
{
  plot(x=as.numeric(rownames(competindo)), y=competindo[,g], type="l",
    bty="l", xlim=c(0,tempo+1), ylim=c(0,max(npop,tempo+5,competindo+5)),
    xlab="tempo", ylab="Nº de indivíduos")
  lines(x=as.numeric(rownames(competindo)), y=competindo[,g], col=g)
  title(main="Comportamento com competição")
  par(new=TRUE)
}
par(new=FALSE)
par(mfrow=c(1,1))
}
```

Arquivo da Função

Competicao

From:

<http://labtrop.ib.usp.br/> - Laboratório de Ecologia de Florestas Tropicais

Permanent link:

http://labtrop.ib.usp.br/doku.php?id=cursos:ecor:05_curso_antigo:r2010:alunos:trabalho_final:gapfrey:start



Last update: 2020/07/27 18:46