

Micael Eiji Nagai



Mestrando em Ecologia, Instituto de Biologia, Unicamp. O título da minha tese é “Biologia populacional e reprodutiva de *Exogone (Exogone) brevia antennata* Hartmann-Schröder, 1959 (Polychaeta: Exogoninae)”, orientado pela Profa. A. Cecília Z. Amaral.

Meus Exercícios

[exec](#)

Proposta do Trabalho Final

Proposta A

Gerar uma função que calcula diferentes modelos de regressões (linear, quadrática, cúbica e segmentada) para um mesmo conjunto de variáveis. Esta função deve calcular também o critério de Akaike modificado por fatores de correção (AICc), a diferença entre o AIC entre os modelos e o peso de Akaike. Gerando os gráficos de cada modelo.

Comentários da Proposta Principal

Função interessante e de aplicabilidade geral. Boa idéia. Tenha claro como os dados entrarão na função. O output está bem definido.

Gabriel

Proposta B

Gerar uma função que realiza uma regressão segmentada (piecewise linear regression).

Comentários

Creio que essa está bem simples, pode seguir com a principal.

Gabriel

Trabalho Final

Função da proposta A

```
##### Trabalho Final - Construção de Função
#####
### Micael Eiji Nagai ###

# Função que calcula as regressões linear, quadrática, cúbica e segmentada
# (borken-stick) para um mesmo conjunto de dados
# Os objetos de entrada são dois vetores de dados numéricos
# Como saída retorna um objeto da classe data.frame com os valores de
# coeficientes, soma de quadrados dos resíduos,
# o valor de AIC de cada modelo, valor da diferença de AIC (di) e o peso
# de AIC (wi).
# A função gera gráficos com as linhas de tendências para cada modelo.

mod.regr=function(lx, ly, nomex="varx", nomey="vary", na.val="NA", AICc=FALSE,
log10=FALSE, save.img=FALSE)

{
  #####
  ##### valores de x e y #####
  varx=lx
  vary=ly

  ### Designando valores de NA diferentes de NA
  if(na.val!="NA")
  {
    varx[varx==na.val]=NA
    vary[vary==na.val]=NA
  }
  ### Retirando os NA's
  nax=!is.na(varx)
  nay=!is.na(vary)
  naxy=nax&nay
  n.nax=length(varx)-sum(nax)
  n.nay=length(vary)-sum(nay)
  n.naxy=length(varx)-sum(naxy)
  varx=varx[naxy]
  vary=vary[naxy]
  cat("\n\n Foram encontrados",n.nax,"NA's da variável x (", nomex,")
e",n.nay,"da variável y (",nomey,"),\n sendo excluidos",n.naxy,"conjuntos de
dados.\n\n\n")

  if( log10==TRUE)  {
```

```

    varx=log(varx,base=10)
    vary=log(vary,base=10)
  }
#####
##### modelos da regressão #####
mod.l=lm(vary~varx)
mod.q=lm(vary~varx+I(varx^2))
mod.c=lm(vary~varx+I(varx^2)+I(varx^3))
  ### Achando o break point
val.x=sort(unique(varx))
rss=rep(NA,each=length(val.x))

for(i in 1:length(val.x)){
  m=lm(vary~(varx*(varx<val.x[i])+varx*(varx>=val.x[i])))
  s=summary(m)
  rss[i]=sum(s$residuals^2)
}
rss.minimo=min(rss)
bp.pos=match(rss.minimo,rss) #localizando a posição do menor erro
residual
bp=val.x[bp.pos] #o ponto de x que possui uma descontinuidade (break-
point)
  ### Modelo do broken - stick
  ### coef1+coef2(bp-x)(para x<bp)+coef3(x-bp)(para x>=bp)
  ### a1= coef(mod.bs)[1]+coef(mod.bs)[2]*bp
  ### b1=-coef(mod.bs)[2]
  ### a2= coef(mod.bs)[1]-coef(mod.bs)[3]*bp
  ###b2=coef[3]
lhs = function(x) ifelse(x < bp,bp-x,0)
rhs = function(x) ifelse(x < bp,0,x-bp)
mod.bs = lm(vary ~ lhs(varx) + rhs(varx))
  ### Switching regression
mod.ds=lm(vary~varx*(varx<bp)+varx*(varx>=bp))
  ### Separando os coeficientes e os coeficientes de regressão
#a1, a2, b1, b2, b3, bp, r2 ajustado
coef.l=round(  as.numeric(
c(coef(mod.l)[1],NA,coef(mod.l)[2],NA,NA,NA,summary(mod.l)[9])
), 3)
coef.q=round(  as.numeric(
c(coef(mod.q)[1],NA, coef(mod.q)[2:3],NA,NA,summary(mod.q)[9])
), 3)
coef.c=round(  as.numeric(
c(coef(mod.c)[1],NA,coef(mod.c)[2:4],NA,summary(mod.c)[9])
), 3)
coef.bs=round(  as.numeric(
c( coef(mod.bs)[1]+coef(mod.bs)[2]*bp,NA, -
coef(mod.bs)[2],coef(mod.bs)[3],NA,bp,summary(mod.bs)[9])
), 3)
coef.ds=round(  as.numeric(
c(coef(mod.ds)[1]+coef(mod.ds)[3],coef(mod.ds)[1],coef(mod.ds)[2]+coef(mod.d
s)[5],coef(mod.ds)[2],NA,bp,summary(mod.ds)[9])

```

```
) ,3)

### Valores de akaike, diferenca de akaike e peso de akaike
if(AICc==TRUE){
  library(AICcmodavg)
  akaike=round(c(
AICc(mod.l),AICc(mod.q),AICc(mod.c),AICc(mod.bs),AICc(mod.ds)
), 3)
  cat("Foi utilizado o critério de akaike modificado para pequenas
amostras - AICc\n\n\n")
}
else{
  akaike=as.vector(round(AIC(mod.l,mod.q,mod.c,mod.bs,mod.ds)[[2]],3))
  cat("Foi utilizado o critério de akaike - AIC\n\n\n")
}
min.aic=min(akaike)
di=akaike-min.aic
wi=round((exp(-0.5*(di))/sum(exp(-0.5*(di)))) ,3)
### tabela de resultados
tab.res=data.frame(
  linear=c(coef.l,akaike[1],di[1],wi[1]),
  quadratico=c(coef.q,akaike[2],di[2],wi[2]),
  cubico=c(coef.c,akaike[3],di[3],wi[3]),
  broken=c(coef.bs,akaike[4],di[4],wi[4]),
  doisSeg=c(coef.ds,akaike[5],di[5],wi[5])
)

rownames(tab.res)=c("a1","a2","b1","b2","b3","bp","r2","AIC","Dif.
AIC","wi")
#####
##### Gráficos #####
### Salvando
if(save.img==TRUE){
  nomearq=paste(deparse(substitute(lx)),deparse(substitute(ly)))
png(file=paste(nomearq,"%02d.jpg"),width=1024,height=768,unit="px",res=150,r
estoreConsole=TRUE)
par(mfrow=c(2,3),tcl=0.2,bty="l",cex.main=1.2,cex.lab=1.3,cex.axis=1.2)
##### gráfico do modelo linear
plot(vary~varx,xlab=nomex,ylab=nomey,main="Linear")
abline(mod.l,col="red",lwd=2)
##### gráfico do modelo quadrático
plot(vary~varx,xlab=nomex,ylab=nomey,main="Quadrático")
curve(mod.q$coef[1]+mod.q$coef[2]*x+mod.q$coef[3]*x^2,col="red",lwd=2,add=TR
UE)

##### gráfico do modelo cúbico
plot(vary~varx,xlab=nomex,ylab=nomey,main="Cúbico")
curve(mod.c$coef[1]+mod.c$coef[2]*x+mod.c$coef[3]*x^2+mod.c$coef[4]*x^3,add=
TRUE,col="red",lwd=2)
##### gráfico do modelo broken stick
```

```

        py =
mod.bs$coef[1]+mod.bs$coef[2]*lhs(val.x)+mod.bs$coef[3]*rhs(val.x)
        plot(vary~varx,xlab=nomex,ylab=nomey,main="Broken-
stick",font.main=3)
        lines(val.x,py,col="red",lwd=2)
        abline(v=bp,lty=5)
        ##### Dois segmentos
        plot(vary~varx,xlab=nomex,ylab=nomey,main="Dois segmentos")
        abline(v=bp,lty=5)
segments(min(val.x),(mod.ds$coef[1]+mod.ds$coef[3])+(mod.ds$coef[2]+mod.ds$co
coef[5])*min(val.x),
bp,(mod.ds$coef[1]+mod.ds$coef[3])+(mod.ds$coef[2]+mod.ds$coef[5])*bp,col="r
ed", lwd=2)
        segments(max(val.x),mod.ds$coef[1]+mod.ds$coef[2]*max(val.x),
bp,mod.ds$coef[1]+mod.ds$coef[2]*bp,col="red",lwd=2)

        dev.off()
    }
    ##### Plotando os gráficos #####
    x11()
    par(mfrow=c(2,3),tcl=0.2,bty="l",cex.main=1.2,cex.lab=1.3,cex.axis=1.2)
    ##### gráfico do modelo linear
    plot(vary~varx,xlab=nomex,ylab=nomey,main="Linear")
    abline(mod.l,col="red",lwd=2)
    ##### gráfico do modelo quadrático
    plot(vary~varx,xlab=nomex,ylab=nomey,main="Quadrático")
    curve(mod.q$coef[1]+mod.q$coef[2]*x+mod.q$coef[3]*x^2,col="red",lwd=2,add=TR
UE)
    ##### gráfico do modelo cúbico
    plot(vary~varx,xlab=nomex,ylab=nomey,main="Cúbico")
    curve(mod.c$coef[1]+mod.c$coef[2]*x+mod.c$coef[3]*x^2+mod.c$coef[4]*x^3,add=
TRUE,col="red",lwd=2)
    ##### gráfico do modelo broken stick
    py = mod.bs$coef[1]+mod.bs$coef[2]*lhs(val.x)+mod.bs$coef[3]*rhs(val.x)
    plot(vary~varx,xlab=nomex,ylab=nomey,main="Broken-stick",font.main=3)
    lines(val.x,py,col="red",lwd=2)
    abline(v=bp,lty=5)
    ##### Dois segmentos
    plot(vary~varx,xlab=nomex,ylab=nomey,main="Dois segmentos")

    abline(v=bp,lty=5)
segments(min(val.x),(mod.ds$coef[1]+mod.ds$coef[3])+(mod.ds$coef[2]+mod.ds$c
coef[5])*min(val.x),
bp,(mod.ds$coef[1]+mod.ds$coef[3])+(mod.ds$coef[2]+mod.ds$coef[5])*bp,col="r
ed", lwd=2) #segmento da esquerda
        segments(max(val.x),mod.ds$coef[1]+mod.ds$coef[2]*max(val.x),
bp,mod.ds$coef[1]+mod.ds$coef[2]*bp,col="red",lwd=2) # segmento da
direita

    return(tab.res)

```

```
}
```

Arquivo da função

[Mod.regr](#)

Página de help da função

mod.regr package:nenhum R Documentation

Modelos de regressões linear, quadrático, cúbico e segmentado.

Description:

Utilizando dois vetores de dados a função calcula os modelos linear, quadrático, cúbico e segmentado. Plotando gráficos de cada modelo com a respectiva linha de tendência

Usage:

```
mod.regr=function(lx, ly, nomex="varx", nomey="vary", na.val="NA",  
AICc=FALSE, log10=FALSE, save.img=FALSE)
```

Arguments:

lx: vetor com os dados da variável independente (preditora)

ly: vetor com os dados da variável dependente (resposta)

nomex: nome da variável x, será usada para a plotagem do gráfico

nomey: nome da variável y

na.val: valor que será considerado NA, fora o próprio NA

AICc: se TRUE calcula o valor do AICc, ao invés do AIC. Caso seja FALSE calcula o AIC

log10: transforma os valores de x e y em log na base 10

save.img: salva o gráfico gerado

Details:

0 AICc é o valor de Akaike com correções para amostras pequenas.

Value:

Retorna um data frame contendo os valores, para cada modelo, do:

intercepto, inclinação e o break-point (para a regressão segmentada);
coeficiente de regressão ajustado;
valores de Akaike (AIC);
diferença de Akaike (delta i) nesta função com a abreviação Dif. AIC;
e o peso de Akaike (wi).
Gera gráficos para cada modelo com a sua respectiva linha de tendência.

Warning:

Caso não tenha o pacote AICcmodavg instalado, a função irá emitir um erro

```
"Erro em library(AICcmodavg) : não há nenhum pacote chamado  
'AICcmodavg' "
```

Note:

Para o cálculo do AICc é necessário estar instalado o pacote AICcmodavg

Author(s):

Micael Eiji Nagai
michaelnagai@gmail.com

References:

Burnham, K.P. & Anderson, D.R. (2002). Model selection and multimodel inference: a practical information-theoretic approach. 2nd ed. Springer
Crawley, M.J. (2007). The R book. Wiley.
Faraway, J.J. (2002). Practical regression and anova using R. (cran.r-project.org/doc/contrib/Faraway-PRA.pdf)
Katsanevakis, S.; Legaki, M.T.; Karlou-Riga, C.; Lefkaditou, E.; Dimitriou, E. & Verriopoulos, G. (2007). Information-theory approach to allometric growth of marine organisms. Mar Bio, 151:949-959.

See Also:

```
lm(), segmented()
```

Examples:

```
#Exemplo 1, verificando a relação entre a largura e o comprimento da  
petala, dados do data frame iris.  
largura=iris$Petal.Width  
comprimento=iris$Petal.Length  
mod.regr(comprimento,largura,nomex="Comprimento",nomey="Largura")
```

#Exemplo2, relação entre o assassinato e assalto nos estados dos EUA (data frame USArrests), foram incluídos valores de NA e valor 0 que representa NA

```
assalto=USArrests$Assault  
assassinato=USArrests$Murder  
assalto[15]=NA  
assassinato[4]=0  
mod.regr(assalto,assassinato,na.val="0")
```

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2011:alunos:trabalho_final:micael:start 

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