

Juliano van Melis



Doutorando em Biologia Vegetal, Instituto de Biologia, Unicamp.

Interesses em ecologia de comunidades vegetais e de ecossistemas.

[Lattes](#)

Meus Exercícios

[exec](#)

Meus planos

Principal

Pensei numa função “curve finder” onde seria utilizada o AIC para que indicasse a curva que melhor se ajusta em uma relação bivariada (dois vetores). Os principais modelos de curvas a serem analisados nessas relações bivariadas seriam: no modelo linear (simples, quadrático e - talvez - polinomial), exponencial (e o modificado), o logarítmico, geométrico, (e talvez outros). A inspiração foi no [curve expert](#), que faz bem isso, mas usa estatística F para definir a curva com melhor ajuste e possui uns graficos bem feinhos. Aí o usuário do [R](#) receberia como resposta os graficos X por Y acrescentado as linhas para cada um dos modelos e a ordem dos modelos com os melhores ajustes e suas fórmulas.

Comentário PI

Boa proposta. Falta só definir se todos os modelos presumirão distribuição normal dos resíduos. Isto vai afetar a função que vc usará para ajuste. Se sim, será a função de regressão gaussiana não linear (nlm). Se não, terá que partir para glm's, mas aí é complicar demais. O importante é deixar claro que os modelos são Gaussianos.

Script da Função

```
fit.model=function(x,y, res.normal=TRUE)
{
  if(res.normal==TRUE)
  {
    lm(y~x) -> ls
    lm(y~x+I(x^2)) -> lq
```

```
lm(y~x+I(x^2)+I(x^3))->lp
nls(y~a+exp(b*x), start=c("a"=1, "b"=2)) ->expo
nls(y~a+b*log(x), start=c("a"=1e-5, "b"=1e-5))->loga
nls(y~a*x/(b+x), start=c("a"=1, "b"=1))->sat
nls(y~a*sin(b*x), start=c("a"=1, "b"=2))->sinus
nls(y~a*I(x^b), start=c("a"=1, "b"=1)) -> pm
AIC(ls, lq, lp, expo, loga, sat, sinus, pm)->AT
AT[,2]->AT
AT[AT<0]*(-1)->Akaike
layout(matrix(c(1:9), 3, 3, byrow=TRUE))
plot(x,y, main="Linha de tendencia"); panel.smooth(x, y)
plot(x, y, main="Modelo Linear Simples"); abline(x, y, col="red")
plot(x, y, main="Modelo Linear Quadratico"); lines(x,
lq$coef[1]+lq$coef[2]*x+lq$coef[3]*x^2, col="red")
plot(x, y, main="Modelo linear Polinomial"); lines(x,
lp$coef[1]+lp$coef[2]*x+lp$coef[3]*x^2+lp$coef[4]*x^4, col="red")
plot(x, y, main="Modelo Exponencial"); coef(expo)->coexp;
lines(coexp[1]+exp(coexp[2]*x), col="red")
plot(y~x, main="Modelo Logarítmico"); coef(loga)->cologa;
lines(cologa[1]+cologa[2]*log(x), col="red")
plot(y~x, main="Modelo de Saturação"); coef(sat)->cosat;
lines(cosat[1]*x/(cosat[2]+x), col="red")->l1
plot(x,y, main="Modelo Sinusoidal"); coef(sinus)->cosin;
lines(cosin[1]*sin(cosin[2]*x), col="red")
plot(x,y, main="Modelo de Potência"); coef(pm)->copm;
lines(copm[1]*x^copm[2], col="red")
x11()
AIC(ls, lq, lp, expo, loga, sat, sinus, pm)->AT
AT[,2]->AT
AT[AT<0]*(-1)->Akaike
layout(matrix(c(1:9), 3, 3, byrow=TRUE))
plot(y~x, main="Dispersão dos pontos")
qqnorm(residuals(ls), main="Modelo Linear simples", xlab="Quantis
Teóricos",
ylab="Quantis amostrados"); qqline(residuals(ls))
qqnorm(residuals(ls), main="Modelo Linear Quadrático", xlab="Quantis
Teóricos",
ylab="Quantis amostrados"); qqline(residuals(ls))
qqnorm(residuals(ls), main="Modelo Linear Polinomial", xlab="Quantis
Teóricos",
ylab="Quantis amostrados"); qqline(residuals(ls))
qqnorm(residuals(ls), main="Modelo Exponencial", xlab="Quantis
Teóricos",
ylab="Quantis amostrados"); qqline(residuals(ls))
qqnorm(residuals(ls), main="Normalidade dos Resíduos da Regressão \n no
modelo Logarítmico", xlab="Quantis Teóricos",
ylab="Quantis amostrados"); qqline(residuals(ls))
qqnorm(residuals(ls), main="Modelo de Saturação", xlab="Quantis
Teóricos",
ylab="Quantis amostrados"); qqline(residuals(ls))
```

```

qqnorm(residuals(ls), main="Modelo Sinusoidal", xlab="Quantis Teóricos",
ylab="Quantis amostrados"); qqline(residuals(ls))
qqnorm(residuals(ls), main="Modelo de Potência", xlab="Quantis
Teóricos",
ylab="Quantis amostrados"); qqline(residuals(ls))
c("a+b*x", "a+b*x+c*x^2", "a+b*x+c*x^2+d*x^3", "a+exp(b*x)",
"a+b*log(x)", "a*x/(b+x)", "a*sin(b*x)", "a*(x^b)")->formulas
Modelo=c("Linear Simples", "Linear Quadrático", "Linear Polinomial",
"Exponencial", "Logarítmico", "de Saturação", "Sinusoidal", "de Potência")
as.vector(ls$coef)-> lsc; as.vector(lq$coef)-> lqc;
as.vector(lq$coef)->lpc; as.vector(coef(expo))->coexp;
as.vector(coef(loga))->cologa; as.vector(coef(sat))->cosat;
as.vector(coef(sinus))->cosin; as.vector(coef(pm))->copm

a=c(lsc[1], lqc[1], lpc[1], coexp[1], cologa[1], cosat[1], cosin[1],
copm[1])
b=c(lsc[2], lqc[2], lpc[2], coexp[2], cologa[2], cosat[2], cosin[2],
copm[2])
c=c("NA", lqc[3], lpc[3], "NA", "NA", "NA", "NA", "NA")
d=c("NA", "NA", lpc[4], "NA", "NA", "NA", "NA", "NA")
AIC(ls, lq, lp, expo, loga, sat, sinus, pm)->AT
AkaikeIndexCriterion<-AT[,2]
Ordem=order(AkaikeIndexCriterion)
resultado=data.frame(Ordem, Modelo, formulas, a, b, c, d,
AkaikeIndexCriterion)
print("Alerta: Verifique se a relação entre os quantis teóricos e os quantis
amostrados dos resíduos dos modelos seguem a normalidade. \n
Caso os resíduos dos modelos não sigam a normalidade: Use 'res.normal=FALSE'
na função.")
return(resultado)
}
if(res.normal==FALSE)
{
formulas=c("y~a+b*x", "y~a+b*x+c*x^2", "y~a+b*x+c*x^2+d*x^3")
glm(y~x)->simp
glm(y~x+I(x^2))->quad
glm(y~x+I(x^2)+I(x^3))->poli
AIC(simp, quad, poli)->ATG
ATG[,2]->AkaikeIndexCriterion
Ordem=order(AkaikeIndexCriterion)
a=c(simp$coef[1], quad$coef[1], poli$coef[1])
b=c(simp$coef[2], quad$coef[2], poli$coef[2])
c=c("NA", quad$coef[3], poli$coef[3])
d=c("NA", "NA", poli$coef[4])
Gaussiano<-data.frame(Ordem, Modelo, formulas, a, b, c, d,
AkaikeIndexCriterion)
Y=y/max(y)
X=x/max(x)
glm(Y~X, family='binomial')->binos
glm(Y~X+I(X^2), family=binomial())->binosq
glm(Y~X+I(X^2)+I(X^3), family=binomial())->binop

```

```
AIC(binop, binoq, binop)->ATG
ATG[,2]->AkaikeIndexCriterion
Ordem<-order(AkaikeIndexCriterion)
a=c(binop$coef[1], binoq$coef[1], binop$coef[1])
b=c(binop$coef[2], binoq$coef[2], binop$coef[2])
c=c("NA", binoq$coef[3], binop$coef[3])
d=c("NA", "NA", binop$coef[4])
Binomial<-data.frame(Ordem, Modelo, formulas, a, b, c, d,
AkaikeIndexCriterion)
glm(y~x, family=gamma()) -> gamms
glm(y~x+I(x^2)+I(x^3), family=inverse.gaussian()) ->gammq
glm(y~x+I(x^2)+I(x^3), family=inverse.gaussian()) ->gamp
AIC(gamms, gammq, gamp)->ATG
ATG[,2]->AkaikeIndexCriterion
Ordem<-order(AkaikeIndexCriterion)
a=c(gamms$coef[1], gammq$coef[1], gamp$coef[1])
b=c(gamms$coef[2], gammq$coef[2], gamp$coef[2])
c=c("NA", gammq$coef[3], gamp$coef[3])
d=c("NA", "NA", gamp$coef[4])
Gamma<-data.frame(Ordem, Modelo, formulas, a, b, c, d,
AkaikeIndexCriterion)
glm(y~x, family=inverse.gaussian()) ->iguass
glm(y~x+I(x^2)+I(x^3), family=inverse.gaussian()) ->igausq
glm(y~x+I(x^2)+I(x^3), family=inverse.gaussian()) ->igausp
AIC(iguass, igausq, igausp)->ATG
ATG[,2]->AkaikeIndexCriterion
Ordem<-order(AkaikeIndexCriterion)
a=c(iguass$coef[1], igausq$coef[1], igausp$coef[1])
b=c(iguass$coef[2], igausq$coef[2], igausp$coef[2])
c=c("NA", igausq$coef[3], igausp$coef[3])
d=c("NA", "NA", igausp$coef[4])
GaussianoInverso<-data.frame(Ordem, Modelo, formulas, a, b, c, d,
AkaikeIndexCriterion)
glm(y~x, family=poisson())->poiss
glm(y~x+I(x^2), family=poisson()) -> poisq
glm(y~x+I(x^2)+I(x^3), family=poisson()) ->poisp
AIC(poiss, poisq, poisp)->ATG
ATG[,2]->AkaikeIndexCriterion
Ordem<-order(AkaikeIndexCriterion)
a=c(poiss$coef[1], poisq$coef[1], poisp$coef[1])
b=c(poiss$coef[2], poisq$coef[2], poisp$coef[2])
c=c("NA", poisq$coef[3], poisp$coef[3])
d=c("NA", "NA", poisp$coef[4])
Poisson<-data.frame(Ordem, Modelo, formulas, a, b, c, d,
AkaikeIndexCriterion)
glm(y~x, family=quasi())->quasis
glm(y~x+I(x^2), family=quasi()) -> quasiaq
glm(y~x+I(x^2)+I(x^3), family=quasi()) ->quasip
AIC(quasis, quasiaq, quasip)->ATG
ATG[,2]->AkaikeIndexCriterion
```

```

Ordem<-order(AkaikeIndexCriterion)
a=c(quasis$coef[1], quasiaq$coef[1], quasip$coef[1])
b=c(quasis$coef[2], quasiaq$coef[2], quasip$coef[2])
c=c("NA", quasiaq$coef[3], quasip$coef[3])
d=c("NA", "NA", quasip$coef[4])
QuasiGaussiano=data.frame(Ordem, Modelo, formulas, a, b, c, d,
AkaikeIndexCriterion)
Y=y/max(y)
X=x/max(x)
glm(Y~X, family=quasibinomial())->quabis
glm(Y~X+I(X^2), family=quasibinomial()) -> quabiq
glm(Y~X+I(X^2)+I(X^3), family=quasibinomial()) ->quabip
AIC(quabis, quabiq, quabip)->ATG
ATG[,2]->AkaikeIndexCriterion
Ordem=order(AkaikeIndexCriterion)
a=c(quabis$coef[1], quabiq$coef[1], quabip$coef[1])
b=c(quabis$coef[2], quabiq$coef[2], quabip$coef[2])
c=c("NA", quabiq$coef[3], quabip$coef[3])
d=c("NA", "NA", quabip$coef[4])
QuasiBinomial=data.frame(Ordem, Modelo, formulas, a, b, c, d,
AkaikeIndexCriterion)
glm(y~x, family=quasipoisson())->quapos
glm(y~x+I(x^2), family=quasipoisson()) -> quapoq
glm(y~x+I(x^2)+I(x^3), family=quasipoisson()) ->quapop
AIC(quapos, quapoq, quapop)->ATG
ATG[,2]->AkaikeIndexCriterion
Ordem=order(AkaikeIndexCriterion)
a=c(quapos$coef[1], quapoq$coef[1], quapop$coef[1])
b=c(quapos$coef[2], quapoq$coef[2], quapop$coef[2])
c=c("NA", quapoq$coef[3], quapop$coef[3])
d=c("NA", "NA", quapop$coef[4])
QuasiPoisson=data.frame(Ordem, Modelo, formulas, a, b, c, d,
AkaikeIndexCriterion)
resultglm=array(Gaussiano, Binomial, Gamma, GaussianoInverso, Poisson,
QuasiGaussiano, QuasiBinomial, QuasiPoisson)
return(resultglm)
print("\n\n Os valores dos vetores deve estar entre 0 e 1 para que
ocorra o modelo generalizado binomial \n\n
Cuidado: Para o modelo generalizado Gamma, ao menos 1 argumento deve
passar por 'gamma'.")
}
}

```

Página de Ajuda

fit.model

package:nenhum

R Documentation

Ajuste de modelos entre dois vetores numéricos, usando o Índice de Critério de Akaike.

Description

Ajusta modelos lineares e não-lineares na relação entre dois vetores, classificando o melhor ajuste e no caso de distribuição normal dos resíduos no modelo (res.norm=TRUE)

Usage:

```
fit.model(x, y, res.normal = TRUE)
```

Arguments:

x um objeto da classe "vector", sendo com valores numéricos. Variável preditora

y: um objeto da classe "vector", sendo com valores numéricos. Varipavel resposta

res.normal: para resíduos com distribuição não normal use res.normal=FALSE
para resíduos com distribuição gaussiana use o argumento
res.normal=TRUE(default)

Details:

A função promove o ajuste do melhor modelo entre dois vetores, usando o Índice de Critério de Akaike (Akaike Index Criterion - AIC).

Para a distribuição normal dos resíduos (res.normal=TRUE), também são dados os gráficos da

variável resposta vs a variável preditora com o as curvas dos modelos. Além disso, é fornecido

a distribuição teórica gaussiana vs observada dos quantis dos resíduos dos modelos. Dessa forma, o usuário pode ver se os resíduos do melhor modelo ajustado possui distribuição gaussiana.

Para modelos generalizados (res.normal=FALSE), são ajustados três tipos de modelos: simples, quadrático e polinomial.

Nesse caso, a resposta fornecida será um "array" com os dados de AIC e as fórmulas para cada

um dos modelos generalizados testados (Gaussiano, Binomial, Gamma, GaussianoInverso, Poisson, QuasiGaussiano, QuasiBinomial, QuasiPoisson). Esses modelos são chamados de "family" pela função "glm".

Warning:

Os vetores não podem possuir NA's ou NAN's.

Author(s):

Versão original por Juliano van Melis

jvmelis@yahoo.com.br

References:

Kelley,

C. T.(1999). "Iterative Methods for Optimization, SIAM Frontiers in Applied Mathematics",

no 18, ISBN: 0-89871-433-8.

Nelder, J. & Wedderburn, R. (1972). "Generalized Linear Models". Journal of the Royal

Statistical Society. Series A (General) 135 (3): 370–384.

See Also:

glm

nls

Examples:

Plano B

Envolveria estatística circular. A minha idéia seria comparar a riqueza ou abundância de determinada(s) espécie(s) em diferentes ecounidades florestais (Oldeman 1990: clareira-construção-bioestasia-degradação). Transformaria cada ecounidade em valores angulares e os valores a serem comparados nos eixos que saem do centro do gráfico. O teste V (Jammalamadaka & SenGupta 2001) seria utilizado para verificar a preferência da(s) espécies e/ou da sinúsia vegetal por determinado ângulo (ecounidade). Como o número de parcelas em cada ecounidade não é uniforme (p.ex: ecounidades maduras 28-37%, [Cassola 2008](#)), teria que ser utilizado valores proporcionais. Talvez fazer uma rarefação, reduzindo o número de parcelas para aquela que tiver menor número.

Novamente, a inspiração vem de outro software ([Oriana](#)), que eu já tive que usar e que tem saídas gráficas sofríveis...Além de não ser um freeware.

Bibliografia

Cassola, H. (2008)Aspectos da estrutura fitossociológica e silvigenética em fragmentos de floresta estacional semidecídua com diferentes histórias de perturbação em Botucatu, SP. Dissertação Mestrado - ESALQ/USP, Piracicaba

Jammalamadaka, S.R., SenGupta, A. (2001)Topics in Circular Statistics. Series on multivariate analysis, vol. 5. World Scientific Publishing Co.. Singapore

Oldeman, R.A.A. (1990) Forests: Elements of Silvology. Springer-Verlag. Berlin

Comentário PI

Também muito boa. A minha crítica não é à função em si, mas à subjetividade de converter cada ecounidade em um ângulo. Se vc tivesse tempo de regeneração a coisa ficaria mais precisa, concordas?

Mas isto não invalida o exercício de construir a função, que é adequado.

Comentário ao Comentário PI

Usei no relatório FAPESP do ano passado essa “subjetividade” pois o meu problema era exatamente como descrever a floresta amostrada. Eu tinha classificado essas ecounidades por Oldeman (1990), e como em campo eu não achava que cada parcela de 100m² era considerada como uma única ecounidade s.s., as vezes eu achava que uma parcela estava entre clareira e construção. Por isso não teríamos somente quatro angulos, mas 8 ângulos.

A “subjetividade” ficou na análise visual em campo, pois não tinha experiência nessa classificação (Oldeman 1990). Para refinar a caracterização das ecounidades terei que usar análise multivariada para ver se as ecounidades formam grupos usando algumas variáveis características (altura do dossel médio, n de árvores mortas, IIC médio, n de árvores do presente/passado/futuro (Oldeman 1990), etc...). Não vou colocar o n de lianas pq talvez viesse a relação que eu quero (ecounidades vs sinúsia de trepadeiras).

Concordo que se eu tivesse o tempo de regeneração (que talvez eu consiga usando os dados do Roque (2001 e 2007), o uso da estatística circular seria mais *robusta*.

Plano C

Se as outras duas propostas forem muito complicadas, também seria útil para um dos meus projetos a análise de covariância (ANCOVA), onde as retas de dois modelos lineares seriam comparadas, verificando se as retas são iguais (mesma inclinação) e paralelas (diferentes interceptos), ou se são distintas (valores de intercepto e inclinação significativamente diferentes).

Comentário PI

Este é tão fácil de fazer diretamente no R que não vejo muito sentido em criar uma função para isto. Mas as duas propostas anteriores são viáveis.

Plano D

Viro hippie.

Comentário PI

Não é original, mas é sempre uma opção.

Comentário ao Comentário PI

Não é original, mas seria desafiadora. O único artesanato que sou capaz de fazer são aviões de papel, no formato "[estrela da morte](#)" (by *Star Wars*).

From:

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

Permanent link:

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



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