

Braz Titon Junior



Doutorando em Fisiologia Geral, Instituto de Biociências - USP.

[exec](#)

Proposta de Trabalho Final

Plano A:

Criar uma função para gerar modelos lineares onde eu forneço a variável dependente e as variáveis independentes e a função me retorna os possíveis modelos desde os mais simples até os mais complexos com todas as interações para posteriormente rodar um teste de seleção de modelos.

Plano B:

Uma função para churrasco onde eu forneço número de homens, número de mulheres, e número de crianças menores de 12 anos e ele me retorna a quantidade e os tipos de carne recomendados e a quantidade e o tipo de bebida recomendado.

Comentários

Daniel (Musgo): Você pretende fazer todas as combinações possíveis de parâmetros nos seus modelos, correto? Lembre-se somente que o número de combinações cresce bastante em função do número de parâmetros e que o número de parâmetros depende do usuário da função. Ou seja, sua função deve atender tanto poucas como muitas variáveis independentes (determine um número máximo para entrada da função). Talvez você poderia incluir um argumento para fazer os testes dos modelos e como output gerar uma tabela de resultados, o que acha?

Concordo com o Musgo. Acho que uma tabela tipo a que é produzida pelo `AICTab` do pacote `bbmle` pode ser legal! Uma outra possibilidade é fazer o modelo cheio e ele ir tirando hierarquicamente as interações e variáveis até o modelo selecionado par a par! Como o procedimento descrito no *R Book* do Crowley! Não gosto muito da abordagem, mas como desafio de função é interessante. —

[Alexandre Adalardo de Oliveira](#) 2012/04/03 21:00

Página de Ajuda

`sel.mod`

`package:unknown`

R Documentation

Melhores modelos lineares possíveis à partir de uma variável dependente e no máximo cinco independentes.

Description:

Produz todos os modelos lineares possíveis à partir de uma variável dependente e no máximo cinco

independentes. Depois compara os modelos através do índice AIC selecionando o(s) modelo(s) que

apresentar(em) o(s) menor(es) valor(es).

Usage:

```
sel.mod(x)
```

Arguments:

x: Matriz com no máximo seis colunas contendo a variável dependente na primeira coluna e as variáveis

independentes nas demais colunas, sem nomes de linhas ou colunas.

Details:

A função cria todos os modelos lineares possíveis à partir de uma variável dependente e no máximo cinco

variáveis independentes de uma maneira genérica onde a variável dependente é chamada de "y" (primeira

coluna da matriz de dados) e as variáveis independentes chamadas de "a", "b", "c", "d" e "e"

(respectivamente segunda, terceira, quarta, quinta e sexta coluna da matriz de dados). Então é calculado

o índice AIC (Akaike Information Criteria) corrigido de acordo com o comprimento da amostra, sendo

selecionados o(s) modelo(s) com menor valor de AICc com uma diferença menor ou igual a 2 do modelo

anterior, sendo que a dAICc do modelo que apresenta o menor AICc de todos é

considerada 0.

Value:

A função retorna como resultado uma matriz com cinco colunas contendo respectivamente:

Fórmula: o(s) modelo(s) selecionado(s) em ordem crescente de AICc.

AICc: valor(es) do(s) AICc(s) do(s) modelo(s) selecionado(s)

df: graus de liberdade/número de parâmetros

dAICc: diferença entre o AICc do modelo com menor índice com o modelo anterior, sendo que

a dAICc do modelo que apresenta o menor AICc de todos é considerada 0.

Weights: $\exp(-dAICc/2) / \sum(\exp(-dAICc/2))$

Warning:

Variáveis com um número de observações baixo compromete o funcionamento da função de acordo com

o aumento do número de variáveis independente.

Author(s):

Braz Titon Junior

References:

Burnham and Anderson 2002

Chambers, J. M. (1992) Linear models. Chapter 4 of Statistical Models in S eds J. M. Chambers

and T. J. Hastie, Wadsworth & Brooks/Cole.

Wilkinson, G. N. and Rogers, C. E. (1973) Symbolic descriptions of factorial models for

analysis of variance. Applied Statistics, 22, 392–9.

See Also:

'AICctab' do pacote "bbmle": Compute table of information criteria and auxiliary info

'lm' do pacote "base": Fitting Linear Models

Examples:

```
## Criando um objeto para teste:
t1=rnorm(40,15,1)
t2=rnorm(40,21,2)
t3=rnorm(40,11,1)
t4=rnorm(40,23,2)
teste=matrix(c(t1,t2,t3,t4), ncol=4)
## Executando o comando:
sel.mod(teste)
```

Código da Função

```
sel.mod<-function(x)
{
  pegaFormula<-function(x)
  {
    return(formula(get(x)))
  }
  require(bbmle)
  dados<-x
  NV<-as.factor(dim(dados)[2])
  if(NV==2)
  {
    y<-dados[,1]
    a<-dados[,2]
    m00<-lm(y~1)
    m01<-lm(y~a)
    res<-AICctab(m00,m01, base=T, weights=T, nobs=dim(dados)[1])
    class(res)<-paste("data.frame")
    res1<-res[res$dAICc<=2,]
    res2<-matrix(paste(row.names(res1)))
    res3<-tapply(res2,INDEX=res2[,1],FUN=pegaFormula)
    res3<-matrix(paste(res3))
    res4<-matrix(res1$dAICc)
    res4<-round(res4,digits=4)
    res5<-matrix(res1$df)
    res6<-matrix(res1$dAICc)
    res6<-round(res6,digits=4)
    res7<-matrix(res1$weight)
```

```
res7<-round(res7,digits=4)
res8<-matrix(ncol=5,nrow=dim(res2)[1])
res8[,1]<-res3
res8[,2]<-res4
res8[,3]<-res5
res8[,4]<-res6
res8[,5]<-res7
rownames(res8)<- paste("Modelo", 1:dim(res2)[1])
colnames(res8)<- paste(c("Fórmula","AICc","df","dAICc","Weight"))
}
if(NV=="3")
{
y<-dados[,1]
a<-dados[,2]
b<-dados[,3]
m00<-lm(y~1)
m01<-lm(y~a)
m02<-lm(y~b)
m03<-lm(y~a+b)
m04<-lm(y~a*b)
res<-AICctab(m00,m01,m02,m03,m04, base=T, weights=T, nobs=dim(dados)[1])
class(res)<-paste("data.frame")
res1<-res[res$dAICc<=2,]
res2<-matrix(row.names(res1))
res3<-tapply(res2,INDEX=res2[,1],FUN=pegaFormula)
res3<-matrix(paste(res3))
res4<-matrix(res1$AICc)
res4<-round(res4,digits=4)
res5<-matrix(res1$df)
res6<-matrix(res1$dAICc)
res6<-round(res6,digits=4)
res7<-matrix(res1$weight)
res7<-round(res7,digits=4)
res8<-matrix(ncol=5,nrow=dim(res2)[1])
res8[,1]<-res3
res8[,2]<-res4
res8[,3]<-res5
res8[,4]<-res6
res8[,5]<-res7
rownames(res8)<- paste("Modelo", 1:dim(res2)[1])
colnames(res8)<- paste(c("Fórmula","AICc","df","dAICc","Weight"))
}
if(NV==4)
{
y<-dados[,1]
a<-dados[,2]
b<-dados[,3]
c<-dados[,4]
m00<-lm(y~1)
m01<-lm(y~a)
m02<-lm(y~b)
```

```
m03<-lm(y~c)
m04<-lm(y~a+b)
m05<-lm(y~a+c)
m06<-lm(y~b+c)
m07<-lm(y~a*b)
m08<-lm(y~a*c)
m09<-lm(y~b*c)
m10<-lm(y~a+b+c)
m11<-lm(y~a+b*c)
m12<-lm(y~a*b+c)
m13<-lm(y~a*b*c)
res<-AICctab(m00,m01,m02,m03,m04,m05,m06,m07,m08,m09,m10,m11,m12,m13,
base=T, weights=T, nobs=dim(dados)[1])
class(res)<-paste("data.frame")
res1<-res[res$dAICc<=2,]
res2<-matrix(row.names(res1))
res3<-tapply(res2,INDEX=res2[,1],FUN=pegaFormula)
res3<-matrix(paste(res3))
res4<-matrix(res1$dAICc)
res4<-round(res4,digits=4)
res5<-matrix(res1$df)
res6<-matrix(res1$dAICc)
res6<-round(res6,digits=4)
res7<-matrix(res1$weight)
res7<-round(res7,digits=4)
res8<-matrix(ncol=5,nrow=dim(res2)[1])
res8[,1]<-res2
res8[,2]<-res3
res8[,3]<-res4
res8[,4]<-res5
res8[,5]<-res6
rownames(res8)<- paste("Modelo", 1:dim(res2)[1])
colnames(res8)<- paste(c("Fórmula","AICc","df","dAICc","Weight"))
}
if(NV==5)
{
y<-dados[,1]
a<-dados[,2]
b<-dados[,3]
c<-dados[,4]
d<-dados[,5]
m00<-lm(y~1)
m01<-lm(y~a)
m02<-lm(y~b)
m03<-lm(y~c)
m04<-lm(y~d)
m05<-lm(y~a+b)
m06<-lm(y~a+c)
m07<-lm(y~a+d)
m08<-lm(y~b+c)
```

```
m09<-lm(y~b+d)
m10<-lm(y~c+d)
m11<-lm(y~a*b)
m12<-lm(y~a*c)
m13<-lm(y~a*d)
m14<-lm(y~b*c)
m15<-lm(y~b*d)
m16<-lm(y~c*d)
m17<-lm(y~a+b+c)
m18<-lm(y~a+b+d)
m19<-lm(y~a+c+d)
m20<-lm(y~b+c+d)
m21<-lm(y~a+b*c)
m22<-lm(y~a+b*d)
m23<-lm(y~a+c*d)
m24<-lm(y~b+c*d)
m25<-lm(y~a*b+c)
m26<-lm(y~a*b+d)
m27<-lm(y~a*c+d)
m28<-lm(y~b*c+d)
m29<-lm(y~a*b*c)
m30<-lm(y~a*b*d)
m31<-lm(y~a*c*d)
m32<-lm(y~b*c*d)
m33<-lm(y~a+b+c+d)
m34<-lm(y~a+b+c*d)
m35<-lm(y~a+b*c+d)
m36<-lm(y~a+b*c*d)
m37<-lm(y~a*b+c+d)
m38<-lm(y~a*b*c*d)
m39<-lm(y~a*b*c+d)
m40<-lm(y~a*b*c*d)
res<-
AICctab(m00,m01,m02,m03,m04,m05,m06,m07,m08,m09,m10,m11,m12,m13,m14,m15,m16,
m17,m18,m19,m20,m21,m22,m23,m24,m25,m26,m27,m28,m29,m30,m31,m32,m33,m34,m35,
m36,m37,m38,m39,m40, base=T, weights=T, nobs=dim(dados)[1])
class(res)<-paste("data.frame")
res1<-res[res$dAICc<=2,]
res2<-matrix(paste(row.names(res1)))
res3<-tapply(res2,INDEX=res2[,1],FUN=pegaFormula)
res3<-matrix(paste(res3))
res4<-matrix(res1$dAICc)
res4<-round(res4,digits=4)
res5<-matrix(res1$df)
res6<-matrix(res1$dAICc)
res6<-round(res6,digits=4)
res7<-matrix(res1$weight)
res7<-round(res7,digits=4)
res8<-matrix(ncol=5,nrow=dim(res2)[1])
res8[,1]<-res3
res8[,2]<-res4
```

```
res8[,3]<-res5
res8[,4]<-res6
res8[,5]<-res7
rownames(res8)<- paste("Modelo", 1:dim(res2)[1])
colnames(res8)<- paste(c("Fórmula","AICc","df","dAICc","Weight"))
}
if(NV==6)
{
y<-dados[,1]
a<-dados[,2]
b<-dados[,3]
c<-dados[,4]
d<-dados[,5]
e<-dados[,6]
m00<-lm(y~1)
m01<-lm(y~a)
m02<-lm(y~b)
m03<-lm(y~c)
m04<-lm(y~d)
m05<-lm(y~e)
m06<-lm(y~a+b)
m07<-lm(y~a+c)
m08<-lm(y~a+d)
m09<-lm(y~a+e)
m10<-lm(y~b+c)
m11<-lm(y~b+d)
m12<-lm(y~b+e)
m13<-lm(y~c+d)
m14<-lm(y~c+e)
m15<-lm(y~d+e)
m16<-lm(y~a*b)
m17<-lm(y~a*c)
m18<-lm(y~a*d)
m19<-lm(y~a*e)
m20<-lm(y~b*c)
m21<-lm(y~b*d)
m22<-lm(y~b*e)
m23<-lm(y~c*d)
m24<-lm(y~c*e)
m25<-lm(y~d*e)
m26<-lm(y~a+b+c)
m27<-lm(y~a+b+d)
m28<-lm(y~a+b+e)
m29<-lm(y~a+c+d)
m30<-lm(y~a+c+e)
m31<-lm(y~a+d+e)
m32<-lm(y~b+c+d)
m33<-lm(y~b+c+e)
m34<-lm(y~b+d+e)
m35<-lm(y~c+d+e)
```


m36<-lm(y~a+b*c)
m37<-lm(y~a+b*d)
m38<-lm(y~a+b*e)
m39<-lm(y~a+c*d)
m40<-lm(y~a+c*e)
m41<-lm(y~a+d*e)
m42<-lm(y~b+c*d)
m43<-lm(y~b+c*e)
m44<-lm(y~b+d*e)
m45<-lm(y~c+d*e)
m46<-lm(y~a*b+c)
m47<-lm(y~a*b+d)
m48<-lm(y~a*b+e)
m49<-lm(y~a*c+d)
m50<-lm(y~a*c+e)
m51<-lm(y~a*d+e)
m52<-lm(y~b*c+d)
m53<-lm(y~b*c+e)
m54<-lm(y~b*d+e)
m55<-lm(y~c*d+e)
m56<-lm(y~a*b*c)
m57<-lm(y~a*b*d)
m58<-lm(y~a*b*e)
m59<-lm(y~a*c*d)
m60<-lm(y~a*c*e)
m61<-lm(y~a*d*e)
m62<-lm(y~b*c*d)
m63<-lm(y~b*c*e)
m64<-lm(y~b*d*e)
m65<-lm(y~c*d*e)
m66<-lm(y~a+b+c+d)
m67<-lm(y~a+b+c+e)
m68<-lm(y~a+b+d+e)
m69<-lm(y~a+c+d+e)
m70<-lm(y~b+c+d+e)
m71<-lm(y~a+b+c*d)
m72<-lm(y~a+b+c*e)
m73<-lm(y~a+b+d*e)
m74<-lm(y~a+c+d*e)
m75<-lm(y~b+c+d*e)
m76<-lm(y~a+b*c+d)
m77<-lm(y~a+b*c+e)
m78<-lm(y~a+b*d+e)
m79<-lm(y~a+c*d+e)
m80<-lm(y~b+c*d+e)
m81<-lm(y~a+b*c*d)
m82<-lm(y~a+b*c*e)
m83<-lm(y~a+b*d*e)
m84<-lm(y~a+c*d*e)
m85<-lm(y~b+c*d*e)
m86<-lm(y~a*b+c+d)

```
m87<-lm(y~a*b+c+e)
m88<-lm(y~a*b+d+e)
m89<-lm(y~a*c+d+e)
m90<-lm(y~b*c+d+e)
m91<-lm(y~a*b+c*d)
m92<-lm(y~a*b+c*e)
m93<-lm(y~a*b+d*e)
m94<-lm(y~a*c+d*e)
m95<-lm(y~b*c+d*e)
m96<-lm(y~a*b*c+d)
m97<-lm(y~a*b*c+e)
m98<-lm(y~a*b*d+e)
m99<-lm(y~a*c*d+e)
m100<-lm(y~b*c*d+e)
m101<-lm(y~a*b*c*d)
m102<-lm(y~a*b*c*e)
m103<-lm(y~a*b*d*e)
m104<-lm(y~a*c*d*e)
m105<-lm(y~b*c*d*e)
m106<-lm(y~a+b+c+d+e)
m107<-lm(y~a+b+c+d*e)
m108<-lm(y~a+b+c*d+e)
m109<-lm(y~a+b+c*d*e)
m110<-lm(y~a+b*c+d+e)
m111<-lm(y~a+b*c*d+e)
m112<-lm(y~a+b*c*d*e)
m113<-lm(y~a+b*c*d*e)
m114<-lm(y~a*b+c+d+e)
m115<-lm(y~a*b+c+d*e)
m116<-lm(y~a*b+c*d+e)
m117<-lm(y~a*b+c*d*e)
m118<-lm(y~a*b*c+d+e)
m119<-lm(y~a*b*c*d+e)
m120<-lm(y~a*b*c*d+e)
m121<-lm(y~a*b*c*d*e)
res<-
AICctab(m00,m01,m02,m03,m04,m05,m06,m07,m08,m09,m10,m11,m12,m13,m14,m15,m16,
m17,m18,m19,m20,m21,m22,m23,m24,m25,m26,m27,m28,m29,m30,m31,m32,m33,m34,m35,
m36,m37,m38,m39,m40,m41,m42,m43,m44,m45,m46,m47,m48,m49,m50,m51,m52,m53,m54,
m55,m56,m57,m58,m59,m60,m61,m62,m63,m64,m65,m66,m67,m68,m69,m70,m71,m72,m73,
m74,m75,m76,m77,m78,m79,m80,m81,m82,m83,m84,m85,m86,m87,m88,m89,m90,m91,m92,
m93,m94,m95,m96,m97,m98,m99,m100,m101,m102,m103,m104,m105,m106,m107,m108,m10
9,m110,m111,m112,m113,m114,m115,m116,m117,m118,m119,m120,m121, base=T,
weights=T, nobs=dim(dados)[1])
class(res)<-paste("data.frame")
res1<-res[res$AICc<=2,]
res2<-matrix(row.names(res1))
res3<-tapply(res2,INDEX=res2[,1],FUN=pegaFormula)
res3<-matrix(paste(res3))
res4<-matrix(res1$AICc)
```

```
res4<-round(res4,digits=4)
res5<-matrix(res1$df)
res6<-matrix(res1$dAICc)
res6<-round(res6,digits=4)
res7<-matrix(res1$weight)
res7<-round(res7,digits=4)
res8<-matrix(ncol=5,nrow=dim(res2)[1])
res8[,1]<-res3
res8[,2]<-res4
res8[,3]<-res5
res8[,4]<-res6
res8[,5]<-res7
rownames(res8)<- paste("Modelo", 1:dim(res2)[1])
colnames(res8)<- paste(c("Fórmula","AICc","df","dAICc","Weight"))
}
return(res8)
}
```

Arquivo da Função

Fórmula para Seleção de Modelos

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:alunos2012:alunos:trabalho_final:titonjr.b:start

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