

Modelos de marcação e recaptura: populações fechadas

Contents

Preparação	1
Ajuste dos modelos	1
Seleção de modelos	3
Valores das estimativas	4

-
- [Arquivo em pdf](#)
 - [Arquivo em markdown](#) (para executar os comandos no R studio)
-

Preparação

Vamos usar o pacote *RMark*, que é um pacote do R para usar o programa [MARK](#). Siga as instruções do [site do RMark](#) para instalar o pacote.

Abra o R e carregue o pacote

```
library(RMark)
```

Usaremos dados de registros fotográficos de indivíduos do boto cinza (*Sotalia guianensis*) em 11 ocasiões. [Aqui](#) há mais informações sobre este caso de estudo.

Os dados estão no formato nativo do MARK (*.inp*). Use os comandos abaixo para importá-lo para o R:

```
## Link dos dados na página da disciplina
url <- "http://ecologia.ib.usp.br/bie5703/lib/exe/fetch.php?media=roteiros:botos_2002.inp"
## Importa arquivo inp
boto2002 <- convert.inp(url)
```

Ajuste dos modelos

Processamento dos dados

O primeiro passo é usar a função `process.data` para criar um objeto com as informações que o Mark usa para ajustar o modelo. Uma delas é o tipo de modelo, que é indicado no argumento `model`.

Para o modelo de populações fechada sem heterogeneidade e de verossimilhança condicionada este argumento é `model="Closed"`:

```
boto <- process.data(data=boto2002, model="Closed")
```

E para o modelo com heterogeneidade o argumento é `model="FullHet"`

```
botoH <- process.data(data=boto2002, model="FullHet")
```

Ajuste dos modelos sem heterogeneidade

Para ajustar os modelos, crie listas que especificam a fórmula de cada termo. No modelo **Closed** os nomes parâmetros que podem variar são p (probabilidade da primeira captura), c (probabilidade de recaptura) ¹. O objeto criado na seção acima tem uma covariável de tempo chamada **time**, que então pode ser usado nas fórmulas para expressar diferenças entre ocasiões:

```
## Fórmulas estatísticas para cada parâmetro do modelo sem heterogeneidade
## p e c constantes mas diferentes
t.dot <- list(formula=~1)
## p=c contantes (use o argumento share=TRUE)
t.dotshared=list(formula=~1,share=TRUE)
## Parametros dependem do tempo
t.time <- list(formula=~time)
## Parametro p=c dependem do tempo
t.timeshared <- list(formula=~time, share=TRUE)
```

E usamos a função **mark** para fazer os ajuste:

```
boto.M0 <- mark(boto, model.parameters=list(p=t.dotshared),
               model.name="M0", adjust=TRUE)
boto.Mb <- mark(boto, model.parameters=list(c=t.dot, p=t.dot),
               model.name="Mb", adjust=TRUE)
```

```
##
## Note: only 2 parameters counted of 3 specified parameters
##
## AICc and parameter count have been adjusted upward
```

```
boto.Mt <- mark(boto, model.parameters=list(p=t.timeshared), model.name="Mt",
               adjust=TRUE)
boto.Mtb <- mark(boto, model.parameters=list(c=t.time, p=t.time),
               model.name="Mtb", adjust=TRUE)
```

```
##
## Note: only 18 parameters counted of 22 specified parameters
##
## AICc and parameter count have been adjusted upward
```

Se omitimos a função de um parâmetro ela será constante. Portanto para todos os modelos acima a expressão para o parâmetro f_0 é **formula=~1**.

Ajuste dos modelos com heterogeneidade

Para os modelos com heterogeneidade acrescente o termo **mixture** nas fórmulas do parâmetro p . O *default* é uma mistura de duas subpopulações, o que representa que uma proporção π dos indivíduos tem uma probabilidade de captura/recaptura e o restante $(1 - \pi)$ tenha outra.

¹O parâmetro f_0 (número de indivíduos não registrados em nenhuma ocasião) é constante por definição, já que a população é fechada.

No modelo com efeito de ocasião, use uma fórmula com interação. Com isso as probabilidades de captura e recaptura de cada subpopulação poderão ser diferentes a cada ocasião.

```
## Fórmulas estatísticas para cada parâmetro do modelo com heterogeneidade
## p com heterogeneidade
t.mix <- list(formula=~mixture)
## p=c com heterogeneidade (use o argumento share=TRUE)
t.mixshared=list(formula=~mixture,share=TRUE)
## Parametros diferem entre ocasiões
t.timemixshared <- list(formula=~time*mixture, share=TRUE)
t.timemix <- list(formula=~time*mixture)
```

E ajuste os modelos

```
boto.Mh <- mark(botoH, model.parameters=list(p=t.mixshared),
               model.name="Mh", adjust=TRUE)
boto.Mbh <- mark(botoH, model.parameters=list(c=t.mix, p=t.mix),
                model.name="Mbh", adjust=TRUE)
boto.Mth <- mark(botoH, model.parameters=list(p=t.timemixshared),
                model.name="Mth", adjust=TRUE)
```

```
##
## Note: only 19 parameters counted of 24 specified parameters
##
## AICc and parameter count have been adjusted upward
```

```
boto.Mtbh <- mark(botoH, model.parameters=list(c=t.timemix, p=t.timemix),
                 model.name="Mtbh", adjust=TRUE)
```

```
##
## Note: only 25 parameters counted of 44 specified parameters
##
## AICc and parameter count have been adjusted upward
```

Note que em todos os modelos acima os parâmetros f_0 e π são constantes, pois omitimos suas fórmulas.

Seleção de modelos

A função abaixo retorna a tabela de seleção de modelos:

```
collect.models(lx=c("boto.M0", "boto.Mb", "boto.Mt", "boto.Mtb",
                  "boto.Mh", "boto.Mbh", "boto.Mth", "boto.Mtbh"))
```

```
## Warning in model.table(x, type, pf = 2, adjust = adjust): Model list contains models of differing ty
```

```
##  model npar    AICc DeltaAICc    weight Deviance
## 7   Mth   24 252.9300  0.00000 9.997964e-01 148.1514
## 3    Mt   12 270.1989 17.26893 1.778327e-04 191.7698
## 6   Mbh    6 275.2837 22.35377 1.399112e-05 209.4365
## 5    Mh    4 275.6888 22.75878 1.142628e-05 213.9520
## 4   Mtb   22 282.6256 29.69561 3.561171e-07 182.3529
## 1    M0    2 289.7349 36.80490 1.018190e-08 232.0679
## 2    Mb    3 290.1108 37.18087 8.436994e-09 230.4140
## 8  Mtbh   44 291.2933 38.36331 4.671170e-09 138.7168
```

Valores das estimativas

A função `coef` retorna os coeficientes na escala de ligação (logito). Para as estimativas na escala de probabilidades use a função `get.real`:

```
coef(boto.Mth, data=boto2002)
```

##		estimate	se	lcl	ucl
##	pi:(Intercept)	-1.9020487	0.4820141	-2.846796	-0.9573011
##	p:(Intercept)	17.9475570	137.2541800	-251.070640	286.9657600
##	p:time2	-17.5420770	137.2568300	-286.565460	251.4813100
##	p:time3	-17.5420780	137.2568300	-286.565460	251.4813000
##	p:time4	-16.5612100	137.2573400	-285.585600	252.4631800
##	p:time5	12.8103450	528.3486000	-1022.752900	1048.3736000
##	p:time6	-18.3529980	137.2568000	-287.376320	250.6703300
##	p:time7	-17.5420790	137.2563400	-286.564520	251.4803600
##	p:time8	-18.3530100	137.2571400	-287.377020	250.6710000
##	p:time9	13.9267110	1.1539141	11.665039	16.1883830
##	p:time10	-18.3529990	137.2575200	-287.377740	250.6717500
##	p:time11	-18.3530170	137.2575000	-287.377720	250.6716800
##	p:mixture2	-19.2787000	137.2542700	-288.297070	249.7396700
##	p:time2:mixture2	17.8718580	137.2574200	-251.152700	286.8964100
##	p:time3:mixture2	17.8718660	137.2574200	-251.152690	286.8964200
##	p:time4:mixture2	17.5611060	137.2577900	-251.464170	286.5863800
##	p:time5:mixture2	-12.0622390	528.3486100	-1047.625500	1023.5011000
##	p:time6:mixture2	18.6827940	137.2573900	-250.341700	287.7072800
##	p:time7:mixture2	18.2901790	137.2568300	-250.733210	287.3135700
##	p:time8:mixture2	-25.0485270	0.0000000	-25.048527	-25.0485270
##	p:time9:mixture2	-43.3163840	0.7086233	-44.705286	-41.9274820
##	p:time10:mixture2	17.3650980	137.2590200	-251.662580	286.3927800
##	p:time11:mixture2	17.6861500	137.2586700	-251.340850	286.7131500
##	f0:(Intercept)	0.4039764	1.1145841	-1.780609	2.5885613

```
## Na escala de probabilidades  
get.real(boto.Mth, parameter="p")
```

```
## [[1]]  
##           1           2           3           4           5           6  
## mixture:1 1.0000000 0.6000037 0.6000034 0.8000085 1.0000000 0.4000059  
## mixture:2 0.2089705 0.2686740 0.2686753 0.4179374 0.3582341 0.2686768  
##           7           8           9          10          11  
## mixture:1 0.6000033 4.000031e-01 1.000000e+00 0.4000057 0.4000012  
## mixture:2 0.3582329 3.739765e-20 4.551132e-14 0.0895581 0.1194121
```

```
get.real(boto.Mth, parameter="pi")
```

```
## [[1]]  
##  
## mixture:1 0.1298768
```

```
get.real(boto.Mth, parameter="f0") ##N de indivíduos não registrados
```

```
## [[1]]  
##      1  
## 1.497769
```

Estimativa do tamanho populacional

A estimativa de interesse é o tamanho da população, que é obtido somando-se a f_0 ao total de indivíduos registrados. Usamos a função `get.real` com argumento `se=TRUE` para obter os intervalos de confiança²:

```
(boto.f0.ic <- as.numeric(get.real(boto.Mth, parameter="f0", se=TRUE)[,5:6]))
```

```
## [1] 0.2573545 8.7168098
```

O número de indivíduos registrados é a soma das frequências no objeto processado

```
(boto.Nobs <- sum(boto$freq))
```

```
## [1] 37
```

E finalmente temos o intervalo de confiança do tamanho populacional

```
(boto.Nobs + boto.f0.ic)
```

```
## [1] 37.25735 45.71681
```

A estimativa do tamanho populacional parece bastante precisa, mas com o pacote *Rcapture* os intervalos são mais conservadores. Confira isto executando o [roteiro do Rcapture](#)

²Para uma lista com todos os coeficientes e seus intervalos use a função `summary` com a opção `se=TRUE`.