

# Modelos de marcação e recaptura: populações fechadas

Leonardo Wedekin e Paulo Inácio Prado (<http://ecologia.ib.usp.br/bie5703>)

## 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.

Com o RMark instalado, abra o R e carregue-o:

```
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"` <sup>1</sup>:

---

<sup>1</sup>A lista de modelos implementados no RMark está no diretório onde o R instalou o pacote. Você pode consultá-lo lá ou no repositório de desenvolvimento do RMark: (<https://github.com/jlaake/RMark/blob/master/RMark/inst/MarkModels.pdf>).

```
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` ( $p$ , probabilidade da primeira captura) e `c` ( $c$ , probabilidade de recaptura)<sup>2</sup>. 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<sup>3</sup>:

```
## Fórmulas estatísticas para cada parâmetro do modelo sem heterogeneidade
## p e c constantes mas diferentes
t.dot <- list(formula=~1)
## p=c constantes (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<sup>4</sup>:

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

<sup>2</sup>O parâmetro  $f_0$  (número de indivíduos não registrados em nenhuma ocasião) é constante por definição, já que a população é fechada.

<sup>3</sup>Para entender completamente isso estude o comando `make.design.data` e o objeto que ele cria, que é a uma lista de matrizes de delineamento do modelo. Se você entender este objeto saberá quais covariáveis estão disponíveis e como manipulá-las. Veja também o [apêndice sobre o RMark](#) no guia *online* do MARK.

<sup>4</sup>Use sempre os argumentos `delete=TRUE` para remover os arquivos temporários do Mark que o ajuste cria e `adjust=TRUE` para que o número de parâmetros seja verificado e ajustado, para o cálculo do AIC.

## 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  $\pi$  dos indivíduos tem uma probabilidade de captura/recaptura e o restante  $(1 - \pi)$  tenha outra.

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, delete=TRUE)
boto.Mbh <- mark(botoH, model.parameters=list(c=t.mix, p=t.mix),
                model.name="Mbh", adjust=TRUE, delete=TRUE)
boto.Mth <- mark(botoH, model.parameters=list(p=t.timemixshared),
                model.name="Mth", adjust=TRUE, delete=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, delete=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  $\pi$  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	148.1513
## 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)		-1.9020480	0.4820135	-2.846795	-0.9573015
## p:(Intercept)		20.0222940	124.0183500	-223.053670	263.0982600
## p:time2		-19.6167860	124.0212700	-262.698490	223.4649100
## p:time3		-19.6167480	124.0212700	-262.698440	223.4649500
## p:time4		-18.6359790	124.0218100	-261.718730	224.4467800
## p:time5		14.1162190	484.5081200	-935.519720	963.7521600
## p:time6		-20.4277150	124.0212400	-263.509350	222.6539200
## p:time7		-19.6168420	124.0207300	-262.697490	223.4638000
## p:time8		-20.4277110	124.0216200	-263.510090	222.6546700
## p:time9		15.5656250	9098.0206000	-17816.555000	17847.6860000
## p:time10		-20.4277330	124.0220400	-263.510940	222.6554700
## p:time11		-20.4277240	124.0220200	-263.510880	222.6554300
## p:mixture2		-21.3534420	124.0184400	-264.429590	221.7227000
## p:time2:mixture2		19.9465390	124.0219400	-223.136460	263.0295400
## p:time3:mixture2		19.9465260	124.0219300	-223.136470	263.0295200
## p:time4:mixture2		19.6358850	124.0223100	-223.447850	262.7196200
## p:time5:mixture2		-13.3681070	484.5081400	-963.004070	936.2678600
## p:time6:mixture2		20.7574850	124.0219000	-222.325450	263.8404200
## p:time7:mixture2		20.3649520	124.0212700	-222.716740	263.4466500
## p:time8:mixture2		-27.7047590	0.0389281	-27.781058	-27.6284600
## p:time9:mixture2		-47.9045350	0.6486600	-49.175909	-46.6331620
## p:time10:mixture2		19.4398640	124.0237000	-223.646590	262.5263200
## p:time11:mixture2		19.7608730	124.0233200	-223.324830	262.8465800
## f0:(Intercept)		0.4040348	1.1145261	-1.780436	2.5885060

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

##	[[1]]		1	2	3	4	5	6
## mixture:1	1.0000000	0.6000102	0.6000193	0.8000033	1.0000000	0.4000105		
## mixture:2	0.2089695	0.2686671	0.2686721	0.4179384	0.3582342	0.2686705		
##		7	8	9	10	11		
## mixture:1	0.5999969	4.000115e-01	1.000000e+00	0.4000062	0.4000085			
## mixture:2	0.3582339	3.297795e-22	2.383852e-15	0.0895602	0.1194132			

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

```
## [[1]]  
##  
## mixture:1 0.1298769
```

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

```
## [[1]]  
##      1  
## 1.497856
```

### 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 confiança<sup>5</sup>:

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

```
## [1] 0.2573858 8.7167708
```

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] 37.25739 45.71677
```

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](#)

---

<sup>5</sup>Para uma lista com todos os coeficientes e seus intervalos use a função `summary` com a opção `se=TRUE`.