

# 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:

```
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`.

---

<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 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.

## 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)
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  $\pi$  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	148.1514
## 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)		-1.9020360	4.820119e-01	-2.846779	-0.9572926
## p:(Intercept)		17.5327250	1.406182e+02	-258.079000	293.1444500
## p:time2		-17.1272710	1.406208e+02	-292.744060	258.4895200
## p:time3		-17.1272840	1.406208e+02	-292.744070	258.4895100
## p:time4		-16.1464650	1.406213e+02	-291.764250	259.4713200
## p:time5		12.5487350	5.452421e+02	-1056.125800	1081.2233000
## p:time6		-17.9382090	1.406208e+02	-293.554950	257.6785300
## p:time7		-17.1272710	1.406203e+02	-292.743130	258.4885900
## p:time8		-17.9382210	1.406211e+02	-293.555610	257.6791700
## p:time9		13.5982440	2.262272e-01	13.154839	14.0416500
## p:time10		-17.9382150	1.406215e+02	-293.556340	257.6799100
## p:time11		-17.9382100	1.406215e+02	-293.556290	257.6798700
## p:mixture2		-18.8638800	1.406183e+02	-294.475770	256.7480100
## p:time2:mixture2		17.4570730	1.406214e+02	-258.160860	293.0750100
## p:time3:mixture2		17.4570700	1.406214e+02	-258.160860	293.0750000
## p:time4:mixture2		17.1463770	1.406218e+02	-258.472270	292.7650300
## p:time5:mixture2		-11.8006250	5.452421e+02	-1080.475200	1056.8740000
## p:time6:mixture2		18.2680040	1.406214e+02	-257.349870	293.8858800
## p:time7:mixture2		17.8753820	1.406208e+02	-257.741400	293.4921700
## p:time8:mixture2		-24.5164870	2.738599e-10	-24.516487	-24.5164870
## p:time9:mixture2		-42.3979690	3.943849e-01	-43.170963	-41.6249740
## p:time10:mixture2		16.9503190	1.406230e+02	-258.670670	292.5713100
## p:time11:mixture2		17.2713330	1.406226e+02	-258.348990	292.8916600
## f0:(Intercept)		0.4039794	1.114593e+00	-1.780623	2.5885821

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

##	[[1]]		1	2	3	4	5	6
## mixture:1	1.0000000	0.5999973	0.5999941	0.7999944	1.0000000	0.3999955		
## mixture:2	0.2089683	0.2686754	0.2686721	0.4179379	0.3582322	0.2686741		
##		7	8	9	10	11		
## mixture:1	0.5999973	3.999926e-01	1.000000e+00	0.3999939	0.3999952			
## mixture:2	0.3582324	9.639199e-20	8.209658e-14	0.0895574	0.1194096			

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

```
## [[1]]  
##  
## mixture:1 0.1298782
```

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

```
## [[1]]  
##      1  
## 1.497773
```

### 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>4</sup>:

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

```
## [1] 0.2573528 8.7169221
```

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 45.71692
```

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>4</sup>Para uma lista com todos os coeficientes e seus intervalos use a função `summary` com a opção `se=TRUE`.