

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)	-1.9020354	0.4820129	-2.846781	-0.957290
##	p:(Intercept)	17.5878900	140.1413300	-257.089120	292.264900
##	p:time2	-17.1826560	140.1439200	-291.864750	257.499430
##	p:time3	-17.1827760	140.1439200	-291.864860	257.499310
##	p:time4	-16.2018710	140.1444200	-290.884940	258.481200
##	p:time5	12.5842330	542.6611200	-1051.031600	1076.200000
##	p:time6	-17.9935740	140.1438900	-292.675600	256.688450
##	p:time7	-17.1824080	140.1434500	-291.863580	257.498760
##	p:time8	-17.9933490	140.1442200	-292.676040	256.689340
##	p:time9	13.6429390	3.6336488	6.520988	20.764891
##	p:time10	-17.9933820	140.1446000	-292.676800	256.690040
##	p:time11	-17.9934350	140.1445900	-292.676830	256.689960
##	p:mixture2	-18.9190350	140.1414100	-293.596210	255.758140
##	p:time2:mixture2	17.5126430	140.1445100	-257.170590	292.195880
##	p:time3:mixture2	17.5126210	140.1445000	-257.170610	292.195850
##	p:time4:mixture2	17.2017420	140.1448600	-257.482190	291.885680
##	p:time5:mixture2	-11.8361700	542.6611200	-1075.452000	1051.779600
##	p:time6:mixture2	18.3234490	140.1444700	-256.359720	293.006620
##	p:time7:mixture2	17.9304280	140.1439200	-256.751670	292.612520
##	p:time8:mixture2	-24.5880840	0.0652480	-24.715970	-24.460198
##	p:time9:mixture2	-42.5210950	0.8347261	-44.157158	-40.885032
##	p:time10:mixture2	17.0053670	140.1460700	-257.680930	291.691660
##	p:time11:mixture2	17.3263730	140.1457400	-257.359280	292.012020
##	f0:(Intercept)	0.4038651	1.1147273	-1.781000	2.588731

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

##	[[1]]		1	2	3	4	5	6
##	mixture:1	1.00000	0.5999445	0.5999157	0.7999560	1.0000000	0.3999475	
##	mixture:2	0.20897	0.2687138	0.2686859	0.4179307	0.3582235	0.2686919	
##			7	8	9	10	11	
##	mixture:1	0.6000039	4.000014e-01	1.000000e+00	0.3999934	0.3999808		
##	mixture:2	0.3582137	8.491987e-20	7.590446e-14	0.0895485	0.1193913		

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

```
## [[1]]  
##  
## mixture:1 0.1298783
```

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

```
## [[1]]  
##      1  
## 1.497602
```

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] 0.257286 8.717192
```

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] 37.25729 45.71719
```

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

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