

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.

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"`¹:

¹A 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)². O objeto criado na seção acima tem uma covariável de tempo chamada `time`, que então pode ser usado nas fórmulas para expressar diferenças entre ocasiões³:

```
## 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`.

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

³Para 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 π 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 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	148.1513
## 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)		-1.9020422	4.820128e-01	-2.846787	-0.9572972
## p:(Intercept)		22.1255500	1.130957e+02	-199.541930	243.7930300
## p:time2		-21.7200840	1.130989e+02	-243.393840	199.9536800
## p:time3		-21.7200790	1.130989e+02	-243.393840	199.9536900
## p:time4		-20.7392530	1.130994e+02	-242.414160	200.9356500
## p:time5		15.4402550	4.435730e+02	-853.962900	884.8434100
## p:time6		-22.5310120	1.130988e+02	-244.204710	199.1426900
## p:time7		-21.7200860	1.130983e+02	-243.392690	199.9525100
## p:time8		-22.5310100	1.130992e+02	-244.205530	199.1435100
## p:time9		17.2270850	6.445849e-08	17.227085	17.2270850
## p:time10		-22.5310100	1.130997e+02	-244.206440	199.1444200
## p:time11		-22.5310120	1.130997e+02	-244.206400	199.1443700
## p:mixture2		-23.4567010	1.130958e+02	-245.124390	198.2109900
## p:time2:mixture2		22.0498720	1.130996e+02	-199.625320	243.7250700
## p:time3:mixture2		22.0498740	1.130996e+02	-199.625330	243.7250700
## p:time4:mixture2		21.7391600	1.131000e+02	-199.936830	243.4151500
## p:time5:mixture2		-14.6921470	4.435730e+02	-884.095320	854.7110300
## p:time6:mixture2		22.8608050	1.130995e+02	-198.814330	244.5359300
## p:time7:mixture2		22.4681940	1.130989e+02	-199.205570	244.1419500
## p:time8:mixture2		-30.3973760	0.000000e+00	-30.397376	-30.3973760
## p:time9:mixture2		-52.5557010	0.000000e+00	-52.555701	-52.5557010
## p:time10:mixture2		21.5431220	1.131015e+02	-200.135880	243.2221200
## p:time11:mixture2		21.8641480	1.131011e+02	-199.814040	243.5423400
## f0:(Intercept)		0.4040034	1.114558e+00	-1.780531	2.5885379

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

##	[[1]]	1	2	3	4	5	6
## mixture:1	1.0000000	0.6000002	0.6000015	0.8000004	1.0000000	0.4000007	
## mixture:2	0.208969	0.2686736	0.2686749	0.4179379	0.3582326	0.2686744	
##		7	8	9	10	11	
## mixture:1	0.5999996	4.000013e-01	1.000000e+00	0.4000013	0.4000007		
## mixture:2	0.3582326	2.725102e-24	1.199128e-16	0.0895584	0.1194114		

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

```
## [[1]]  
##  
## mixture:1 0.1298775
```

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

```
## [[1]]  
##      1  
## 1.497809
```

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 confiança⁴:

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

```
## [1] 0.2573687 8.7168022
```

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] 37.25737 45.71680
```

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](#)

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