

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 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**.

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	148.1513
## 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) -1.9020423 4.820127e-01 -2.846787 -0.9572974
## p:(Intercept)  19.7698030 1.260079e+02 -227.205730 266.7453300
## p:time2        -19.3643380 1.260108e+02 -266.345520 227.6168500
## p:time3        -19.3643260 1.260108e+02 -266.345510 227.6168600
## p:time4        -18.3834990 1.260113e+02 -265.365720 228.5987300
## p:time5         13.9572760 4.926224e+02 -951.582650 979.4972000
## p:time6        -20.1752620 1.260108e+02 -267.156380 226.8058600
## p:time7        -19.3643420 1.260103e+02 -266.344480 227.6158000
## p:time8        -20.1752650 1.260112e+02 -267.157120 226.8065900
## p:time9         15.3662460 0.000000e+00  15.366246  15.3662460
## p:time10       -20.1752620 1.260116e+02 -267.157940 226.8074100
## p:time11       -20.1752640 1.260115e+02 -267.157890 226.8073600
## p:mixture2     -21.1009520 1.260080e+02 -268.076670 225.8747600
## p:time2:mixture2  19.6941210 1.260115e+02 -227.288340 266.6765900
## p:time3:mixture2  19.6941220 1.260115e+02 -227.288340 266.6765800
## p:time4:mixture2  19.3834040 1.260118e+02 -227.599790 266.3666000
## p:time5:mixture2 -13.2091710 4.926224e+02 -978.749110 952.3307700
## p:time6:mixture2  20.5050500 1.260114e+02 -226.477340 267.4874400
## p:time7:mixture2  20.1124500 1.260108e+02 -226.868730 267.0936300
## p:time8:mixture2 -27.3814360 4.085290e-02 -27.461507 -27.3013640
## p:time9:mixture2 -47.3461890 8.831559e-16 -47.346189 -47.3461890
## p:time10:mixture2 19.1873740 1.260132e+02 -227.798500 266.1732500
## p:time11:mixture2 19.5084010 1.260128e+02 -227.476740 266.4935400
## f0:(Intercept)  0.4040094 1.114546e+00 -1.780501  2.5885196
```

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

```
## [[1]]
##              1          2          3          4          5          6
## mixture:1 1.0000000 0.6000001 0.6000028 0.8000016 1.0000000 0.4000015
## mixture:2 0.2089694 0.2686730 0.2686754 0.4179382 0.3582323 0.2686740
##              7          8          9         10         11
## mixture:1 0.5999992 4.000007e-01 1.000000e+00 0.4000015 0.4000012
## mixture:2 0.3582333 5.865114e-22 3.413317e-15 0.0895586 0.1194119
```

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

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

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

```
## [[1]]  
##      1  
## 1.497818
```

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] 0.2573737 8.7167371
```

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 45.71674
```

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