

Modelos de marcação e recaptura: populações abertas: Cormak-Jolly-Seber

Contents

Preparação	1
Ajuste dos modelos	1
Seleção de modelos	3
Valores das estimativas	3
Para saber mais	5

-
- [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 marcação e recaptura de andorinhões (*Apus apus*) em 8 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. Como o arquivo tem uma covariável (tipo de colônia) com dois estados, você deve informar isso com o argumento `group.df` ¹:

```
## Link dos dados na página da disciplina
url <- "http://ecologia.ib.usp.br/bie5703/lib/exe/fetch.php?media=roteiros:aa.inp"
## Importa arquivo inp
and.raw <- convert.inp(url, group.df=data.frame(colony=c("exposed","protected")))
```

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`.

¹Veja a ajuda da função `convert.inp` para detalhes.

Vamos usar o modelo Cormack-Jolly-Seber, cuja a sigla no RMark é “CJS” ²:

```
andor <- process.data(data=and.raw, model="CJS", groups="colony")
```

Ajuste dos modelos

Para ajustar os modelos, crie listas que especificam a fórmula de cada termo. No modelo CJS os nomes parâmetros são Φ (ϕ , probabilidade de sobrevivência entre capturas), p (p , probabilidade de recaptura).

Além da covariável de tipo de colônia que indicamos neste caso, o RMark já cria algumas outras para cada tipo de modelo. No modelo CJS há uma covariável chamada `time`, com um nível para cada ocasião de captura. Há também uma covariável de identidade da coorte (`cohort`), entre outras ³.

Podemos então criar objetos com as fórmulas para diferentes combinações dessas covariáveis. Algumas possibilidades:

```
## Fórmulas estatísticas para cada parâmetro do modelo
## formula para expressar um parâmetro constante
f.dot <- list(formula=~1)
## formula para expressar um parametro que varia em funcao do tempo
f.time <- list(formula=~time)
## parametro depende do tipo de colonia
f.col <- list(formula=~colony)
## parametro depende do tipo de colonia e tempo
f.coltime <- list(formula= ~colony + time)
## Interacao tempo x colonia
## o efeito de colonia varia com o tempo de maneira diferente entre colonias
f.coltime2 <- list(formula= ~colony * time)
## Sobrevivencia depende da coorte e da colonia
f.cohcol <- list(formula=~cohort+colony)
##
```

E usamos a função `mark` para fazer os ajustes, usando os objetos com as fórmulas:

```
a1 <- mark(andor, model.parameters=list(p=f.dot, Phi=f.dot))
a2 <- mark(andor, model.parameters=list(p=f.time, Phi=f.time))
```

```
##
## Note: only 13 parameters counted of 14 specified parameters
##
## AICc and parameter count have been adjusted upward
```

```
a3 <- mark(andor, model.parameters=list(p=f.time, Phi=f.col))
a4 <- mark(andor, model.parameters=list(p=f.time, Phi=f.coltime))
a5 <- mark(andor, model.parameters=list(p=f.time, Phi=f.coltime2))
```

```
##
## Note: only 20 parameters counted of 21 specified parameters
##
## AICc and parameter count have been adjusted upward
```

²A 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>).

³Para 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.

```
a6 <- mark(andor, model.parameters=list(p=f.time, Phi=f.cohcol))
```

Seleção de modelos

A função abaixo retorna a tabela de seleção de modelos:

```
collect.models(lx=c("a1", "a2", "a3", "a4", "a5", "a6"))
```

```
##              model npar      AICc DeltaAICc      weight
## 3      Phi(~colony)p(~time)    9 369.8080  0.000000 9.594874e-01
## 1              Phi(~1)p(~1)    2 376.9136  7.105612 2.748371e-02
## 6 Phi(~cohort + colony)p(~time) 15 379.1125  9.304565 9.153319e-03
## 4      Phi(~colony + time)p(~time) 15 381.0747 11.266695 3.431686e-03
## 2              Phi(~time)p(~time) 14 385.1905 15.382559 4.382874e-04
## 5      Phi(~colony * time)p(~time) 21 393.9027 24.094763 5.622480e-06
##      Deviance
## 3 111.6644
## 1 133.6472
## 6 107.3258
## 4 109.2879
## 2 115.7384
## 5 107.5633
```

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(a3, data=and.raw)
```

```
##              estimate      se      lcl      ucl
## Phi:(Intercept)  0.3107153 0.3161186 -0.3088772  0.9303078
## Phi:colonyprotected 0.8973282 0.3759843  0.1603989  1.6342576
## p:(Intercept)    2.3000686 1.0331540  0.2750866  4.3250505
## p:time3          -1.3106856 1.1495134 -3.5637320  0.9423608
## p:time4          -2.1525247 1.1281067 -4.3636139  0.0585645
## p:time5          -1.4604208 1.1436618 -3.7019980  0.7811564
## p:time6          -0.4981603 1.2596787 -2.9671306  1.9708100
## p:time7          -0.4818698 1.3516538 -3.1311113  2.1673717
## p:time8          -2.4470973 1.0999975 -4.6030926 -0.2911021
```

```
## Na escala de probabilidades
get.real(a3, parameter="Phi")
```

```
## $`Group:colonyexposed`
## $`Group:colonyexposed`$pim
##           1           2           3           4           5           6           7
## 1 0.5770598 0.5770598 0.5770598 0.5770598 0.5770598 0.5770598 0.5770598
## 2      NA 0.5770598 0.5770598 0.5770598 0.5770598 0.5770598 0.5770598
## 3      NA      NA 0.5770598 0.5770598 0.5770598 0.5770598 0.5770598
```

```
## 4      NA      NA      NA 0.5770598 0.5770598 0.5770598 0.5770598
## 5      NA      NA      NA      NA 0.5770598 0.5770598 0.5770598
## 6      NA      NA      NA      NA      NA 0.5770598 0.5770598
## 7      NA      NA      NA      NA      NA      NA 0.5770598
##
## $`Group:colonyexposed`$group
## [1] 1
##
##
## $`Group:colonyprotected`
## $`Group:colonyprotected`$pim
##      1      2      3      4      5      6      7
## 1 0.7699526 0.7699526 0.7699526 0.7699526 0.7699526 0.7699526 0.7699526
## 2      NA 0.7699526 0.7699526 0.7699526 0.7699526 0.7699526 0.7699526
## 3      NA      NA 0.7699526 0.7699526 0.7699526 0.7699526 0.7699526
## 4      NA      NA      NA 0.7699526 0.7699526 0.7699526 0.7699526
## 5      NA      NA      NA      NA 0.7699526 0.7699526 0.7699526
## 6      NA      NA      NA      NA      NA 0.7699526 0.7699526
## 7      NA      NA      NA      NA      NA      NA 0.7699526
##
## $`Group:colonyprotected`$group
## [1] 2
```

```
get.real(a3, parameter="p")
```

```
## $`Group:colonyexposed`
## $`Group:colonyexposed`$pim
##      2      3      4      5      6      7      8
## 1 0.9088827 0.728966 0.5368192 0.698391 0.8583811 0.8603499 0.4633089
## 2      NA 0.728966 0.5368192 0.698391 0.8583811 0.8603499 0.4633089
## 3      NA      NA 0.5368192 0.698391 0.8583811 0.8603499 0.4633089
## 4      NA      NA      NA 0.698391 0.8583811 0.8603499 0.4633089
## 5      NA      NA      NA      NA 0.8583811 0.8603499 0.4633089
## 6      NA      NA      NA      NA      NA 0.8603499 0.4633089
## 7      NA      NA      NA      NA      NA      NA 0.4633089
##
## $`Group:colonyexposed`$group
## [1] 1
##
##
## $`Group:colonyprotected`
## $`Group:colonyprotected`$pim
##      2      3      4      5      6      7      8
## 1 0.9088827 0.728966 0.5368192 0.698391 0.8583811 0.8603499 0.4633089
## 2      NA 0.728966 0.5368192 0.698391 0.8583811 0.8603499 0.4633089
## 3      NA      NA 0.5368192 0.698391 0.8583811 0.8603499 0.4633089
## 4      NA      NA      NA 0.698391 0.8583811 0.8603499 0.4633089
## 5      NA      NA      NA      NA 0.8583811 0.8603499 0.4633089
## 6      NA      NA      NA      NA      NA 0.8603499 0.4633089
## 7      NA      NA      NA      NA      NA      NA 0.4633089
##
## $`Group:colonyprotected`$group
## [1] 2
```

Para saber mais

- Para outro exemplo de ajuste do modelo CJS e outras funções úteis do RMark digite `?dipper`
- Veja também os pacotes Rcapture e marked