

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	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 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

Os objetos dos modelos ajustados são uma lista com todo o output do Mark. Esta lista tem um elemento `results`, com os valores dos coeficientes e muito mais ⁴:

```
names(a3$results)
```

```
## [1] "lnl"          "deviance"      "deviance.df"
## [4] "npar"         "n"             "AICc"
## [7] "beta"         "real"          "beta.vcv"
## [10] "derived"      "derived.vcv"   "covariate.values"
## [13] "singular"     "real.vcv"
```

Os coeficientes na escala da função de ligação estão no dataframe `beta` desta lista

```
a3$results$beta
```

```
##              estimate      se      lcl      ucl
## Phi:(Intercept)  0.3107152 0.3161186 -0.3088773  0.9303078
## Phi:colonyprotected 0.8973283 0.3759844  0.1603989  1.6342576
## p:(Intercept)    2.3000673 1.0331573  0.2750790  4.3250555
## p:time3          -1.3106843 1.1495163 -3.5637363  0.9423678
## p:time4          -2.1525237 1.1281097 -4.3636187  0.0585713
## p:time5          -1.4604192 1.1436651 -3.7020028  0.7811644
## p:time6          -0.4981590 1.2596825 -2.9671368  1.9708187
## p:time7          -0.4818681 1.3516592 -3.1311202  2.1673839
## p:time8          -2.4470960 1.1000010 -4.6030980 -0.2910940
```

⁴Consulte o [apêndice sobre o RMark](#) no guia *online* do MARK.

E os coeficientes na escala original de probabilidades estão no dataframe `real`

```
a3$results$real
```

```
##               estimate      se      lcl      ucl fixed
## Phi gexposed c1 c1 a0 t1  0.5770598 0.0771525 0.4233888 0.7171377
## Phi gprotected c1 c1 a0 t1 0.7699526 0.0389282 0.6850936 0.8373725
## p gexposed c1 c1 a1 t2    0.9088826 0.0855609 0.5683394 0.9869400
## p gexposed c1 c1 a2 t3    0.7289660 0.1025871 0.4929224 0.8815383
## p gexposed c1 c1 a3 t4    0.5368191 0.1137048 0.3210910 0.7395922
## p gexposed c1 c1 a4 t5    0.6983911 0.1037709 0.4685583 0.8587843
## p gexposed c1 c1 a5 t6    0.8583811 0.0878042 0.5953706 0.9614914
## p gexposed c1 c1 a6 t7    0.8603499 0.1050437 0.5261259 0.9715791
## p gexposed c1 c1 a7 t8    0.4633089 0.0948795 0.2900973 0.6458517
##               note
## Phi gexposed c1 c1 a0 t1
## Phi gprotected c1 c1 a0 t1
## p gexposed c1 c1 a1 t2
## p gexposed c1 c1 a2 t3
## p gexposed c1 c1 a3 t4
## p gexposed c1 c1 a4 t5
## p gexposed c1 c1 a5 t6
## p gexposed c1 c1 a6 t7
## p gexposed c1 c1 a7 t8
```

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`