

Modelos de marcação e recaptura: delineamento robusto

Contents

| | |
|--|---|
| Preparação | 1 |
| Ajuste dos modelos | 1 |
| Seleção de modelos | 4 |
| Valores das estimativas | 4 |
| Coda: refinando a seleção de modelos | 6 |
| Para saber mais | 9 |

-
- [Arquivo em pdf](#)
 - [Arquivo em markdown](#) (para executar os comandos no R studio)
-

Preparação

Vamos usar o *RMark*, que é um pacote do R para usar o programa [MARK](#). Siga as instruções do [site do RMark](#) para instalação.

Com o RMark instalado, abra o R e carregue-o:

```
library(RMark)
```

Usaremos dados de marcação e recaptura de albatrozes (*Phoebastria immutabilis*) no atol de Midway. Foi usado o [delineamento robusto de Pollock](#), com quatro ocasiões primárias com duas ocasiões secundárias cada. [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:albatroz.inp"
## Importa arquivo inp
## use.comments=TRUE usa os rótulos das linhas (id dos indivíduos)
alb.raw <- convert.inp(url, use.comments=TRUE)
```

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 precisa para ajustar o modelo. Uma delas é o tipo de modelo, que é indicado no argumento `model`.

Vamos usar o modelo robusto para populações fechadas, cuja a sigla no RMark é “Robust” ¹.

Para esta classe de modelos indique os intervalos de tempo com um vetor em que os zeros indicam os momentos em que a população é considerada fechada. Os elementos diferentes de zero indicam a duração de cada intervalo. Neste exercício os intervalos entre as ocasiões são todos de um ano, com duas instâncias secundárias cada, o que representamos com um vetor de valores 1 e 0. Note que omitimos o último valor, que é sempre não zero, por definição.

```
## Vetor de intervalos (zero = fechamento)
ti <- c(0, 1, 0, 1, 0, 1, 0)
## Processa os dados
alb <- process.data(data=alb.raw, model="Robust", time.intervals=ti)
```

O objeto resultante é uma lista, na qual podemos conferir o número de ocasiões primárias e secundárias, por exemplo:

```
## N de ocasiões primárias
alb$noocc
```

```
## [1] 4
```

```
## N de ocasiões secundárias em cada primária
alb$noocc.secondary
```

```
## [1] 2 2 2 2
```

Ajuste dos modelos

Para ajustar os modelos, crie listas que especificam a fórmula de cada termo. No modelo **Robust** os nomes parâmetros são:

- **S** : probabilidade de sobrevivência aparente entre capturas (S),
- **p** : probabilidade de recaptura (c),
- **GammaPrime** : probabilidade de permanecer não amostrável entre duas amostragens (γ')
- **GammaDoublePrime** : probabilidade de tornar-se não amostrável entre duas amostragens (emigração temporária, γ'')
- **f0** : número de indivíduos que não foram registrados.

Quando processamos os dados o RMark já cria algumas covariáveis para cada tipo de modelo. No modelo robusto há uma covariável chamada **session**, com um nível para cada ocasião primária de captura, e outra chama **time**, com um nível para cada intervalo entre seções primárias ².

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

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>).

²Para 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.

```
## 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 das ocasiões primarias
f.occ <- list(formula=~1+session)
## formula para expressar um parametro que varia a cada ano
f.time <- list(formula=~1+time)
## formula para expressar os parâmetro fixos em zero
f.zero <- list(formula=~1, fixed=0)
```

Para indicar que dois parâmetros devem ser iguais, acrescente à lista `share=TRUE` ³:

```
## Fórmulas estatísticas para parametros relacionados iguais
## (usa-se para fazer
## GammaPrime = GammaDoublePrime e c = p)
## formula para expressar dois parâmetros relacionados constantes
f.dot.s <- list(formula=~1, share=TRUE)
## formula para expressar que os parametros variam em funcao das ocasiões primarias
f.occ.s <- list(formula=~1+session, share=TRUE)
## formula para expressar parametros que variam a cada ano
f.time.s <- list(formula=~1+time, share=TRUE)
## formula para expressar parametros que variam entre ocasioes secundárias
## Para os parametros de captura/recaptura
f.time2.s <- list(formula=~1+session:time, share=TRUE)
```

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

```
alb1 <- mark(alb, model.parameters=
  list(S=f.time, GammaDoublePrime=f.time, GammaPrime=f.time,
    p=f.time2.s, f0=f.occ))
```

```
##
## Note: only 17 parameters counted of 20 specified parameters
##
## AICc and parameter count have been adjusted upward
```

```
alb2 <- mark(alb, model.parameters=
  list(S=f.time, GammaDoublePrime=f.time.s,
    p=f.time2.s, f0=f.occ))
```

```
##
## Note: only 16 parameters counted of 18 specified parameters
##
## AICc and parameter count have been adjusted upward
```

³O RMark permite reduzir pares de alguns parâmetros a uma só estimativa. No caso do modelo robusto, pode-se substituir os parâmetros γ' e γ'' por um só (*random emigration model*); e também substituir os parâmetros p e c por um único (modelo sem resposta comportamental à captura). Veja final da seção C.3 e exemplo na seção C.19 do apêndice sobre o RMark no guia *online* para mais detalhes.

```
alb3 <- mark(alb, model.parameters=
  list(S=f.time, GammaDoublePrime=f.dot, GammaPrime=f.dot,
    p=f.time2.s, f0=f.occ))
alb4 <- mark(alb, model.parameters=
  list(S=f.time, GammaDoublePrime=f.dot.s,
    p=f.time2.s, f0=f.occ))
```

```
##
## Note: only 14 parameters counted of 16 specified parameters
##
## AICc and parameter count have been adjusted upward
```

```
alb5 <- mark(alb, model.parameters=
  list(S=f.time, GammaDoublePrime=f.zero, GammaPrime=f.zero,
    p=f.time2.s, f0=f.occ))
```

Seleção de modelos

A função abaixo retorna a tabela de seleção de modelos:

```
collect.models(lx=c("alb1", "alb2", "alb3", "alb4", "alb5"))
```

```
##
## 3          S(~-1 + time)Gamma'(~1)Gamma'(~1)p(~-1 + session:time)c()f0(~-1 + session)      model
## 4          S(~-1 + time)Gamma'(~1)Gamma'(~1)p(~-1 + session:time)c()f0(~-1 + session)
## 2          S(~-1 + time)Gamma'(~-1 + time)Gamma'(~1)p(~-1 + session:time)c()f0(~-1 + session)
## 1 S(~-1 + time)Gamma'(~-1 + time)Gamma'(~-1 + time)p(~-1 + session:time)c()f0(~-1 + session)
## 5          S(~-1 + time)Gamma'(~1)Gamma'(~1)p(~-1 + session:time)c()f0(~-1 + session)
##   npar      AICc  DeltaAICc      weight  Deviance
## 3   17 -103394.1    0.00000 9.999996e-01 -10978.60
## 4   16 -103363.6   30.44502 2.448768e-07 -10946.16
## 2   18 -103363.0   31.03527 1.822955e-07 -10949.57
## 1   20 -103312.9   81.14670 0.000000e+00 -10903.47
## 5   15 -102135.0 1259.01033 0.000000e+00  -9715.58
```

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(alb3$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"
```

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

Os coeficientes na escala da função de ligação estão no dataframe `beta` desta lista. Verifique estes valores para o modelo selecionado:

```
alb3$results$beta
```

| ## | estimate | se | lcl | ucl |
|---------------------------------|------------|-----------|------------|------------|
| ## S:time1 | 2.7013443 | 0.1430024 | 2.4210595 | 2.9816291 |
| ## S:time2 | 3.0507041 | 0.3157838 | 2.4317679 | 3.6696404 |
| ## S:time3 | 5.1538522 | 4.3323440 | -3.3375422 | 13.6452470 |
| ## GammaDoublePrime:(Intercept) | -0.6238018 | 0.0384548 | -0.6991732 | -0.5484303 |
| ## GammaPrime:(Intercept) | -1.4274964 | 0.1773193 | -1.7750421 | -1.0799506 |
| ## p:session1:time1 | 1.5319563 | 0.0820832 | 1.3710733 | 1.6928394 |
| ## p:session2:time1 | 2.1758718 | 0.0929140 | 1.9937604 | 2.3579832 |
| ## p:session3:time1 | 1.6228615 | 0.0714008 | 1.4829159 | 1.7628070 |
| ## p:session4:time1 | 0.6567638 | 0.0635849 | 0.5321374 | 0.7813902 |
| ## p:session1:time2 | -0.2980583 | 0.0456685 | -0.3875686 | -0.2085480 |
| ## p:session2:time2 | -0.1161159 | 0.0409353 | -0.1963490 | -0.0358827 |
| ## p:session3:time2 | -0.1888300 | 0.0401241 | -0.2674733 | -0.1101867 |
| ## p:session4:time2 | -0.6902724 | 0.0455277 | -0.7795067 | -0.6010380 |
| ## f0:session1 | 5.4922234 | 0.1085583 | 5.2794491 | 5.7049977 |
| ## f0:session2 | 4.9626018 | 0.1284600 | 4.7108201 | 5.2143834 |
| ## f0:session3 | 5.5807828 | 0.0979300 | 5.3888399 | 5.7727257 |
| ## f0:session4 | 6.6183176 | 0.0779625 | 6.4655112 | 6.7711240 |

Os coeficientes na escala original de probabilidades estão no dataframe `real`

```
alb3$results$real
```

| ## | estimate | se | lcl | ucl |
|---------------------------|-------------|------------|-------------|-------------|
| ## S g1 c1 c1 a0 t1 | 0.9371059 | 0.0084283 | 0.9184192 | 0.9517373 |
| ## S g1 c1 c1 a1 t2 | 0.9548129 | 0.0136246 | 0.9192179 | 0.9751477 |
| ## S g1 c1 c1 a2 t3 | 0.9942561 | 0.0247417 | 0.0343055 | 0.9999988 |
| ## Gamma'' g1 c1 c1 a0 t1 | 0.3489173 | 0.0087359 | 0.3319956 | 0.3662287 |
| ## Gamma' g1 c1 c1 a1 t2 | 0.1934891 | 0.0276709 | 0.1449164 | 0.2535154 |
| ## p g1 s1 t1 | 0.8222924 | 0.0119946 | 0.7975535 | 0.8445972 |
| ## p g1 s1 t2 | 0.4260322 | 0.0111673 | 0.4043028 | 0.4480511 |
| ## p g1 s2 t1 | 0.8980618 | 0.0085060 | 0.8801404 | 0.9135667 |
| ## p g1 s2 t2 | 0.4710036 | 0.0101994 | 0.4510698 | 0.4910303 |
| ## p g1 s3 t1 | 0.8351894 | 0.0098282 | 0.8150126 | 0.8535609 |
| ## p g1 s3 t2 | 0.4529323 | 0.0099421 | 0.4335275 | 0.4724812 |
| ## p g1 s4 t1 | 0.6585330 | 0.0142982 | 0.6299815 | 0.6859796 |
| ## p g1 s4 t2 | 0.3339725 | 0.0101270 | 0.3144262 | 0.3541062 |
| ## f0 g1 a0 s1 t0 | 242.7964400 | 26.3575730 | 196.3840200 | 300.1777500 |
| ## f0 g1 a0 s2 t0 | 142.9652700 | 18.3653230 | 111.2577600 | 183.7091600 |
| ## f0 g1 a0 s3 t0 | 265.2792000 | 25.9788040 | 219.0495100 | 321.2655100 |
| ## f0 g1 a0 s4 t0 | 748.6844300 | 58.3692770 | 642.7414800 | 872.0899400 |
| ## | fixed | note | | |
| ## S g1 c1 c1 a0 t1 | | | | |
| ## S g1 c1 c1 a1 t2 | | | | |
| ## S g1 c1 c1 a2 t3 | | | | |
| ## Gamma'' g1 c1 c1 a0 t1 | | | | |
| ## Gamma' g1 c1 c1 a1 t2 | | | | |
| ## p g1 s1 t1 | | | | |

```
## p g1 s1 t2
## p g1 s2 t1
## p g1 s2 t2
## p g1 s3 t1
## p g1 s3 t2
## p g1 s4 t1
## p g1 s4 t2
## f0 g1 a0 s1 t0
## f0 g1 a0 s2 t0
## f0 g1 a0 s3 t0
## f0 g1 a0 s4 t0
```

E os tamanhos populacionais a cada ocasião (um parâmetro derivado) estão no dataframe `derived`:

```
alb3$results$derived
```

```
##      estimate      se      lcl      ucl
## 1 2384.796 26.35757 2338.384 2442.178
## 2 2659.965 18.36532 2628.258 2700.709
## 3 2947.279 25.97880 2901.050 3003.266
## 4 3293.684 58.36928 3187.741 3417.090
```

Coda: refinando a seleção de modelos

As probabilidades de sobrevivência por ocasião estimadas pelo modelo selecionado são muito parecidas. Isso sugere que este parâmetro não varie entre ocasiões. Vamos incluir modelos com S constante em nossa seleção para verificar:

```
alb6 <- mark(alb, model.parameters=
  list(S=f.dot, GammaDoublePrime=f.time, GammaPrime=f.time,
    p=f.time2.s, f0=f.occ))
alb7 <- mark(alb, model.parameters=
  list(S=f.dot, GammaDoublePrime=f.time.s,
    p=f.time2.s, f0=f.occ))
alb8 <- mark(alb, model.parameters=
  list(S=f.dot, GammaDoublePrime=f.dot, GammaPrime=f.dot,
    p=f.time2.s, f0=f.occ))
alb9 <- mark(alb, model.parameters=
  list(S=f.dot, GammaDoublePrime=f.dot.s,
    p=f.time2.s, f0=f.occ))
alb10 <- mark(alb, model.parameters=
  list(S=f.dot, GammaDoublePrime=f.zero, GammaPrime=f.zero,
    p=f.time2.s, f0=f.occ))
```

E repetimos a seleção de modelos:

```
collect.models(lx=c("alb1", "alb2", "alb3", "alb4", "alb5",
  "alb6", "alb7", "alb8", "alb9", "alb10"))
```

```
##                                                                 model
## 3      S(~-1 + time)Gamma'(~1)Gamma'(~1)p(~-1 + session:time)c()f0(~-1 + session)
```

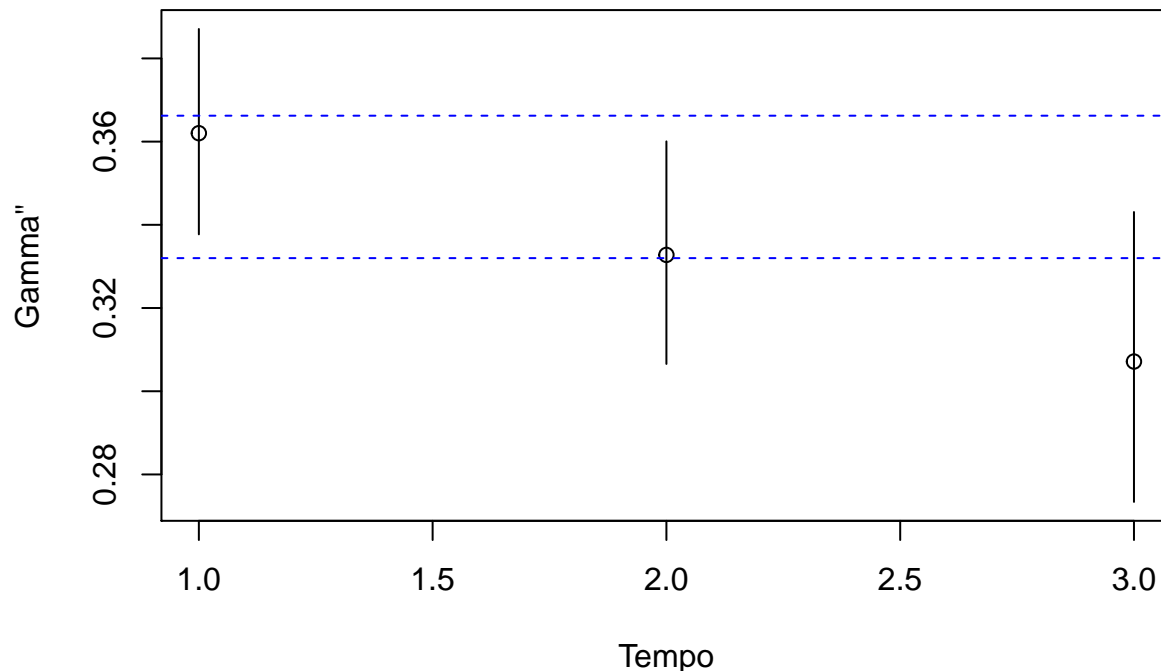
```
## 6      S(~1)Gamma'(~-1 + time)Gamma'(~-1 + time)p(~-1 + session:time)c()f0(~-1 + session)
## 8      S(~1)Gamma'(~1)Gamma'(~1)p(~-1 + session:time)c()f0(~-1 + session)
## 4      S(~-1 + time)Gamma'(~1)Gamma'(~1)p(~-1 + session:time)c()f0(~-1 + session)
## 2      S(~-1 + time)Gamma'(~-1 + time)Gamma'(~1)p(~-1 + session:time)c()f0(~-1 + session)
## 7      S(~1)Gamma'(~-1 + time)Gamma'(~1)p(~-1 + session:time)c()f0(~-1 + session)
## 9      S(~1)Gamma'(~1)Gamma'(~1)p(~-1 + session:time)c()f0(~-1 + session)
## 1 S(~-1 + time)Gamma'(~-1 + time)Gamma'(~-1 + time)p(~-1 + session:time)c()f0(~-1 + session)
## 5      S(~-1 + time)Gamma'(~1)Gamma'(~1)p(~-1 + session:time)c()f0(~-1 + session)
## 10     S(~1)Gamma'(~1)Gamma'(~1)p(~-1 + session:time)c()f0(~-1 + session)
##      npar      AICc    DeltaAICc      weight  Deviance
## 3      17 -103394.1      0.000000 4.690098e-01 -10978.60
## 6      18 -103393.7      0.315273 4.006095e-01 -10980.29
## 8      15 -103391.5      2.560334 1.303804e-01 -10972.03
## 4      16 -103363.6     30.445021 1.148497e-07 -10946.16
## 2      18 -103363.0     31.035273 8.549841e-08 -10949.57
## 7      16 -103362.1     31.945021 5.425114e-08 -10944.65
## 9      14 -103353.6     40.415942 7.851844e-10 -10932.17
## 1      20 -103312.9     81.146700 0.000000e+00 -10903.47
## 5      15 -102135.0    1259.010334 0.000000e+00 -9715.58
## 10     13 -102033.3    1360.751842 0.000000e+00 -9609.83
```

E agora temos o empate com um modelo com S constante mas parâmetros de migração variando com o tempo.

Será que os dois modelos plausíveis dão estimativas similares dos parâmetros de interesse? Vamos verificar.

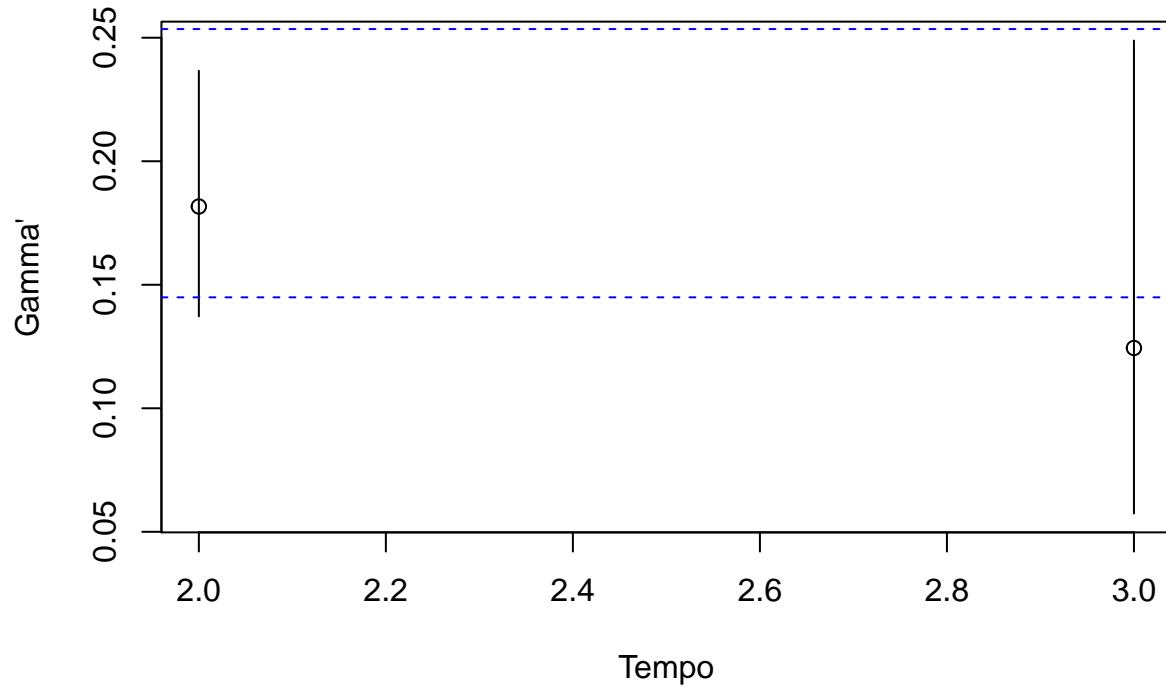
O gráfico a seguir mostra os intervalos de confiança dos valores de γ'' estimados para cada tempo pelo modelo 6 (barras pretas) e o intervalo de confiança da estimativa de γ'' constante pelo modelo 3 (linhas azuis):

```
plot(1:3, alb6$results$real[2:4,1], xlab="Tempo", ylab="Gamma'",
     ylim=range(alb6$results$real[2:4,3:4]))
segments(x0=1:3, y0=alb6$results$real[2:4,3], x1=1:3, y1=alb6$results$real[2:4,4])
abline(h=alb3$results$real[4,3:4], lty=2, col="blue")
```



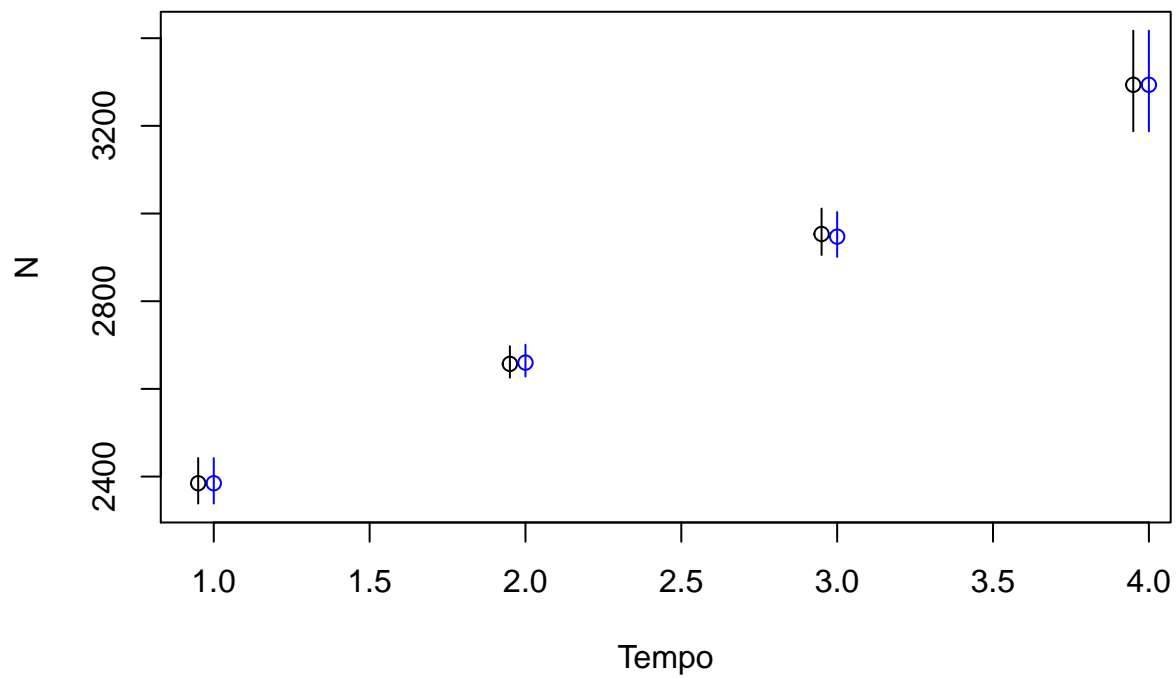
A mesma comparação para o parâmetro γ' :

```
plot(2:3, alb6$results$real[5:6,1], xlab="Tempo", ylab="Gamma' ",
     ylim=range(alb6$results$real[5:6,3:4]))
segments(x0=2:3, y0=alb6$results$real[5:6,3], x1=2:3, y1=alb6$results$real[5:6,4])
abline(h=alb3$results$real[5,3:4], lty=2, col="blue")
```



E por fim comparamos as estimativas dos tamanhos populacionais

```
tempo <- 1:4 -0.05
plot(tempo, alb6$results$derived[,1], xlab="Tempo", ylab="N",
     ylim=range(alb6$results$derived[,3:4]))
segments(x0=tempo, y0=alb6$results$derived[,3], x1=tempo, y1=alb6$results$derived[,4])
tempo <- tempo+0.05
points(tempo, alb3$results$derived[,1], col="blue")
segments(x0=tempo, y0=alb3$results$derived[,3], x1=tempo, y1=alb3$results$derived[,4], col="blue")
```

Para saber mais

- Para outro exemplo de ajuste do modelo robusto e outras funções úteis do RMark digite `?robust`
- Veja também o pacote Rcapture