

Lab 12 – Estimating abundance with closed-population capture-mark-recapture data

Due before Monday

The purpose of this lab is to learn how to estimate abundance using mark-recapture data from surveys of closed populations. A closed population does not experience recruitment, mortality, immigration or emigration. The closure assumption is usually only valid over very short time periods. We will learn how to work with data from open populations later.

Work through the problems below, put your code and your answers in a Word file, and then upload it to ELC. Name the file something like “Chandler-lab12.docx”.

Assignment

Exercise I: Lincoln-Peterson estimation

1. Suppose you capture, mark, and release 100 largemouth bass (*Micropterus salmoides*) at Lake Herrick. The next day, you return and capture 50 individuals, 25 of which were marked on the first occasion. What is the Lincoln-Peterson estimate of abundance (N)? Do the calculations in R and show your work.

Exercise II: Closed-population models in R

1. Use the R package ‘mra’ to fit the four mark-recapture models described in Table 1 to the stinkpot (*Sternotherus odoratus*) data found in the file **CH-S0-Andy07.inp**. These data were collected using baited ‘hoop traps’ at Andy’s Pond in 2007. Software instructions are in the next section.
2. Summarize your results by creating a table in which each row is a model, and include the following columns: model description, the estimates of N (abundance), the standard errors of N (SE), and the AICc values.
3. Use model M_t to make a graph of capture probability (p_t) at each time point. You can do this in either Excel or R.
4. The model with the lowest AICc is considered the best in the set of models. Which model has the lowest AICc? Do your results suggest that there was a behavioral (i.e., a trap-happy or trap-shy) response? Explain.
5. Why do you think model M_0 is the worst model in the set? Look at the capture histories when answering this question.

Table 1: The four models to be fitted to the stinkpot data.

Model	Description
M_0	The most basic model in which p and c are constant
M_t	p differs among sampling occasions and $p_t = c_t$.
M_b	Behavioral response model in which p and c differ. Can describe trap happiness or trap shyness.
M_{tb}	A combination of models M_t and M_b .

Parameter definitions

- p – capture probability. The probability of capturing an individual on a single occasion
- p_t – capture probability on occasion t
- c – recapture probability. The probability of capturing an individual that has been captured previously.
- n – the number of individuals captured
- N – abundance. The number of individuals in the population.



Figure 1: A stinkpot

Example analysis in R

Open R (or RStudio) and install the “mra” package using the following command:

```
install.packages("mra")
```

Now load the package like so:

```
library(mra)
```

The example bear data are formatted for program MARK, but we can import them using `read.table`. The only trick is to tell R that the capture histories should be treated as a character string, rather than as a numeric variable. The `colClasses` arguments let’s us do that.

```
capture.histories <- read.table("CH-example.inp", sep=" ",
                                colClasses=c("character", "character"),
                                col.names=c("ch", "freq"))
```

It’s a small dataset, so we can display it in full:

```
capture.histories
##      ch freq
## 1 101001   1;
## 2 100001   1;
## 3 100001   1;
## 4 010000   1;
## 5 101000   1;
## 6 010001   1;
## 7 001000   1;
## 8 001011   1;
## 9 000100   1;
## 10 001001   1;
## 11 001000   1;
## 12 000010   1;
## 13 000010   1;
## 14 000101   1;
```

We need to convert the capture histories from a character vector to a matrix. The following code does the trick.

```
ch.mat <- t(sapply(capture.histories$ch,
                   function(x) as.integer(strsplit(x, "")[[1]])))
dimnames(ch.mat) <- list(paste0("Bear", 1:nrow(capture.histories)),
                        paste0("Time", 1:6))
```

Now the capture histories look like this:

```
ch.mat
##      Time1 Time2 Time3 Time4 Time5 Time6
## Bear1     1     0     1     0     0     1
## Bear2     1     0     0     0     0     1
## Bear3     1     0     0     0     0     1
## Bear4     0     1     0     0     0     0
## Bear5     1     0     1     0     0     0
## Bear6     0     1     0     0     0     1
## Bear7     0     0     1     0     0     0
## Bear8     0     0     1     0     1     1
## Bear9     0     0     0     1     0     0
## Bear10    0     0     1     0     0     1
## Bear11    0     0     1     0     0     0
## Bear12    0     0     0     0     1     0
## Bear13    0     0     0     0     1     0
## Bear14    0     0     0     1     0     1
```

We will use the `F.huggins.estim` function to fit the closed population models. Model M_0 can be fit like this:

```
M0 <- F.huggins.estim(capture=~1, recapture=NULL, histories=ch.mat)
```

Model M_b like this:

```
Mb <- F.huggins.estim(capture=~1, recapture=~1, histories=ch.mat)
```

To fit model M_t , we have to create a time variable.

```
time <- tvar(factor(1:6), nan=nrow(ch.mat)) ## 6 time periods. 14 animals.
Mt <- F.huggins.estim(capture=~time, recapture=NULL, histories=ch.mat)
```

You should be able to figure out how to fit model M_{tb} by extending the code above.

Estimates of abundance along with AICc values can be found by typing the name of the fitted model object. For example:

```
M0

## Call:
## F.huggins.estim(capture = ~1, recapture = NULL, histories = ch.mat)
##
## Capture and Recapture model:
## Variable      Est      SE
## (Intercept) -1.25004  0.32308
##
## Population Size Estimate (se): 17.962 (3.1462)
## 95% confidence interval for population size: 15.01 to 29.6
## Individuals observed: 14
## Effective sample size: 84
##
## Message = SUCCESS: Convergence criterion met
## Number of estimable coefficients (estimated) = 1
## Log likelihood = -47.6734852287387
## Deviance = 95.3469704574774
## AIC = 97.3469704574774
## AICc = 97.3957509452822
```

You can extract the capture probability estimates for each individual on each sampling occasion:

```
round(M0$p.hat, digits=3)

##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,] 0.223 0.223 0.223 0.223 0.223 0.223
## [2,] 0.223 0.223 0.223 0.223 0.223 0.223
## [3,] 0.223 0.223 0.223 0.223 0.223 0.223
## [4,] 0.223 0.223 0.223 0.223 0.223 0.223
## [5,] 0.223 0.223 0.223 0.223 0.223 0.223
## [6,] 0.223 0.223 0.223 0.223 0.223 0.223
## [7,] 0.223 0.223 0.223 0.223 0.223 0.223
## [8,] 0.223 0.223 0.223 0.223 0.223 0.223
## [9,] 0.223 0.223 0.223 0.223 0.223 0.223
## [10,] 0.223 0.223 0.223 0.223 0.223 0.223
## [11,] 0.223 0.223 0.223 0.223 0.223 0.223
## [12,] 0.223 0.223 0.223 0.223 0.223 0.223
## [13,] 0.223 0.223 0.223 0.223 0.223 0.223
## [14,] 0.223 0.223 0.223 0.223 0.223 0.223
```

For model M_0 , capture probability is the same for all individuals during all time periods. This won't be the case for the other models.