Methods for JS Lewis analysis

Brandon

2025-02-20

# Methods

This is an individual based model describing the estimate of carcass abundance in the Lewis River. This model is different than that of Schwarz. The goal is to estimate the total carcass abundance (i.e., )

To understand the likelihood, it is best to start with a model that has no spatial component.

Breaking down the likelihood into it’s components, the probability of the observed total number of carcasses is where, is total population size, and p is the capture probability of a carcass. The probability of the arrival process is, where, is a latent effect, and vector of arrival probabilities for each time-step. The probability of the initial detection for any carcass, where, , is the latent estimate of arrival time, is the survival probability, and is the detection probability. The probability of carcass being tagged is, where, is a carcass was tagged, is the tagging rate. Finally, there is probability of recapture,

where, for a carcass recaptured on day .

## Create the data

We begin by reading the data and transforming some of the output to make it easier to work in RTMB.

#Data list  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tidyr)  
  
d <- read.csv("data/simpleData2.csv") %>%  
 mutate(t\_wk = lubridate::week(lubridate::mdy(TagDate)),  
 r\_wk = lubridate::week(lubridate::mdy(RecapDate)),  
 t\_yr = lubridate::year(lubridate::mdy(TagDate)),  
 r\_yr = lubridate::year(lubridate::mdy(RecapDate))) %>%  
 filter(t\_yr == 2024) %>%   
 filter(t\_wk > 10) %>%  
 mutate(t\_k = t\_wk - min(t\_wk) + 1,  
 r\_k = r\_wk - min(t\_wk) + 1) %>%  
 mutate(t\_l = TagState,  
 r\_l = RecapState) %>%  
 filter(is.na(r\_wk) | r\_k>0) %>%   
 mutate(tag = ifelse(Tag1=="",FALSE,TRUE)) %>%  
 group\_by(t\_k,r\_k,t\_l,r\_l,tag) %>%  
 summarise(n = n())

## `summarise()` has grouped output by 't\_k', 'r\_k', 't\_l', 'r\_l'. You can  
## override using the `.groups` argument.

## Just model the CJS part of the likelihood

It is possible to breaks down the analysis in to separate components. We can start by just looking at the Cormack-Jolly-Seber part of the model. For those tagged individuals that are tagged, we can estimate survival and detection probability. For now, we can simply assume all fish are equal and the detection and survival is constant across time and location.

I am going to use a matrix-algebra approach to analyzing the data. And start with the simplest data set possible.

data <- list(t\_l = d$t\_l, #tagging location  
 r\_l = d$r\_l, #recapture location  
 t\_k = d$t\_k, #tagging week  
 r\_k = d$r\_k,  
 tag = d$tag,  
 n = d$n) #recapture week, last week if not recapture  
  
# Initial parameter values  
parameters <- list(  
 phi\_par = 0,  
 p\_par = 0  
 )

Now lets create the likelihoods function given the data.

f <- function(parms){  
  
 RTMB::getAll(data,  
 parms)  
   
 #Negative  
 nll <- rep(0,length(d$t\_l))  
   
 #Survival  
 phi <- matrix(0,2,2)  
 phi[1,1] <- -exp(phi\_par)  
 phi[1,2] <- -phi[1,1]  
 phi <- Matrix::expm(phi)  
   
 #Detection probability  
 p <- matrix(0,2,2)  
 p[1,1] <- -exp(p\_par)  
 p[1,2] <- -p[1,1]  
 p <- Matrix::expm(p)  
   
 #This is the amount of time to first detection  
 for(i in 1:length(t\_l)){  
   
 #Accumulator  
 m <- matrix(0,2,2)  
 diag(m) <- 1  
   
 #Initial state  
 delta <- rep(0,2)  
 delta[1] <- 1  
 if(tag[i] == TRUE ){  
 if(is.na(r\_k[i])){  
 Ui <- 16  
 }else{  
 Ui <- r\_k[i]  
 }  
 for(j in (t\_k[i]+1):Ui){  
 if(j<Ui){ #not detected  
 m <- m %\*% phi %\*% diag(p[,2])  
 }else{#last observation  
 if(is.na(r\_k[i])){#not detected  
 m <- m %\*% phi %\*% diag(p[,2])  
 }else{#Detected  
 m <- m %\*% phi %\*% diag(p[,1])  
 }  
 }  
 }  
 }  
 nll[i] <- log(t(delta) %\*% m %\*% rep(1,2))  
 }  
  
 RTMB::REPORT(nll)   
 RTMB::REPORT(phi)   
 RTMB::REPORT(p)   
 return(-sum(nll\*n))  
 # return(0)  
}

Next we can optimize.

## outer mgc: 964.4787   
## outer mgc: 477.1059   
## outer mgc: 397.2693   
## outer mgc: 83.36825   
## outer mgc: 115.5598   
## outer mgc: 23.24397   
## outer mgc: 4.240043   
## outer mgc: 6.129297   
## outer mgc: 6.249639   
## outer mgc: 1.467554   
## outer mgc: 2.381205   
## outer mgc: 0.07488745   
## outer mgc: 0.0007851369   
## outer mgc: 1.521512e-05

Finally, we can look at the results of and matrixes,

#Survival/persistence  
print(round(rep$phi,2))

## 2 x 2 Matrix of class "dgeMatrix"  
## [,1] [,2]  
## [1,] 0.51 0.49  
## [2,] 0.00 1.00

#Survival/persistence  
print(round(rep$p,2))

## 2 x 2 Matrix of class "dgeMatrix"  
## [,1] [,2]  
## [1,] 0.61 0.39  
## [2,] 0.00 1.00

In this simple model, for the tagged individuals, the “persistence/survival” between time steps is 0.51 and the detection probability is 0.61.

## Just model the CJS part of the likelihood

# Tables