Estimating Population Size: Capture-Recapture

Erin Law
Kate Ng
Seunghyun Seo
Thomas Uriot

Year 4 Project
School of Mathematics
University of Edinburgh
June 6, 2021

Contents

We would like to give our sincere appreciation to our project supervisor Prof. Ruth King, who has been always supportive and caring for our project.

People who provided data from their studies for our use as examples include Chao, A. Efford M. and Laake J. We appreciate their help in illustrating various methods using real data.

Abstract

Estimating population size is a fundamental objective of many animal monitoring programmes. This report makes use of the capture-recapture method in order to estimate animal abundance in a closed population. The approach consists of repeated sampling of uniquely marked individuals, forming capture histories that are used in a general likelihood function. Logistic models are used to fit the probabilities in the general likelihood function, representing eight different ecological assumptions. Simulation and analysis of several data sets are performed for the simplest models, using the software R. Estimates of the population size and their associated confidence interval are computed. Model selection is made using the Akaike's criterion. It is shown that models perform poorly under extreme values of capture probabilities. Furthermore, for the capture-recapture method to give a good estimate of an unknown population size, it is desirable for data to be adequately large. Real data set analysis agrees with past research journals.

Introduction

Background

The first use of the capture-recapture concept dates back to the early 16th century, when the English statistician and demographer John Graunt attempted to estimate the effect of the plague on the population size of England. A couple of centuries later, in 1802, the French mathematician Pierre Simon Laplace used similar methods to estimate the size of the population in France [22]. Laplace estimated the number of newly born individuals in France, in 1802, to be approximately $1,000,000$. In the context of capture-recapture, these births can be thought as the marked individuals, in other words, the population is known to be of at least a million. Then, in order to obtain an estimate for the ratio between the total number of births in the population to the total population size, Laplace obtained census and live births from several communities, that were recorded each year. He found that in those communities, the number of births was $71,866$ and the total population of the communities (births included) was $2,037,615$. Therefore, the ratio of newly born to the population in these communities was 0.0353. Laplace then proceeded to use this ratio as an estimate for the whole population of France and concluded that the total population of France was

$$\frac{1,000,000}{0.0353} = 28,328,612.$$

A long time has passed since the first use of the capture-recapture principle and by the 1960s, the use of probability models with maximum-likelihood-estimation of the unknown parameters has become the standard approach. Nowadays, this method is principally used in epidemiological studies and estimating animal abundance.

Capture-Recapture

The capture-recapture method, sometimes called mark and recapture, consists of capturing individuals from the population of interest, uniquely marking them, releasing them in their environment and then repeating the process. The term individuals can range from objects such as taxis or golf tees to living animals. It is assumed that once an individual is marked, it cannot lose its mark. Different instances of marking could be attaching a numbered ring to a bird’s leg, or writing a number on a butterfly’s wings, as shown in Figure 1. Similarly, one could use cameras instead of properly capturing and marking the individuals, assuming that they can be uniquely differentiated. In this paper, it is assumed that the populations of interest are closed. In other words, there are no births, deaths or migrations, such that the population size does not vary over the period of study. The assumption that there are no deaths induces that marking an animal does not affect their survival. It also follows that marking has no effect on the individuals mobility, thus not affecting its catchability rate [27]. The assumption of closed population is usually valid when the data collection is carried out in a relatively short time and during a non-breeding season since there are more chance of births, deaths and migrations for longer capture periods, making the population 'open'. The geographical constraints need to be sensible for the population of study. The capture-recapture method needs a time frame and geographical limits, for example, it is particularly hard to estimate the population of blue whales since they undertake extensive migrations. Essentially, the overlapping information from all the captures - when an individual is present in more than one sample - is used in order to infer on the population size. Intuitively, in a two samples experiment, if a large part of the first sample is also found in the second sample with little new individuals in the second sample, it is likely that most of the population has been seen. However, if the second sample presents only a few individuals from the first sample, then it is likely that only a small portion of the population has been seen. The data collected will then be used in a general likelihood function in order to estimate the parameters of interest using the maximum-likelihood-estimates.

In summary, the key assumptions are

1. Once marked, individuals cannot lose their marks. If this assumption is violated, it can lead to positive bias in the estimation of the population size as less animals are thought to have been captured, which results in a potential increase in individuals seen but less recapture occasions.

2. All marks are correctly recorded. Violating this assumption has a similar result to the abundance estimation as 1, such that it leads to positive bias.

3. The population is closed, i.e. remain constant over the study period. If this were not the case, then open population estimation methods should be used.

4. All individuals in the population are available for capture.

5. Captures of individuals are independent. Capture probabilities should not depend on the practicalities of the experiment, such that the capture probability for one individual should not change just because the trap is full for example [6].

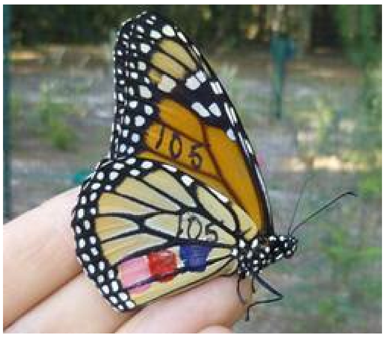

Figure 1: An example of marking [21].

Applications

The capture-recapture method is not only limited to estimate animal abundance, but can also be used in epidemiology to estimate the population size of drug addicts or modern slavery, for instance. Few of the main applications are listed below.

Commercial and ecology: Estimation of fishery stocks is very important for both the equilibrium of the ecosystem and for the fishing industry. An accurate estimate of the remaining stocks of fish is needed in order to place quotas on the legal amount of fish that fishermen can take [15].

Endangered species: Knowing the estimated population size of an endangered species provides insightful information for wildlife conservation organisations. For example, both species Microcebus tavaratra and Lepilemur milanoii in the Daraina region, northern Madagascar are threatened due to deforestation. The capture-recapture method was performed to determine which species requires more urgent conservation. The method also enables the endangered species' population to be monitored, which is useful for developing conservation strategies. [23].

Epidemiology: Capture-recapture was first applied to the epidemiological problem of estimating the size of a human population in the 1940s [30]. Note that Laplace used a method similar to capture-recapture as explained earlier, but he did not make use of any recaptures. Since then, it has been used to estimate the size of various populations such as drug users, homeless people or people with mental illness. This is done by evaluating the degree of overlap among incomplete lists from existing data sources (e.g, police and hospital records)[12].

Differences of Use Between Animal and Human Population

Note that there exists differences between the use of the capture-recapture method on wildlife and human populations. In wildlife studies, there is often a large number of capture instances, whereas most epidemiological studies are limited to only a few number of lists. While there exists a natural time ordering in animal experiments, this is not the case in human studies. Identical trapping methods are used in all capture occasions, hence behavioural responses exhibited by animals could be considered in model analysis. However, the behavioural response due to 'capture' is much harder to account for in human studies as there is no time ordering. This paper focuses on animal abundance and the models considered will require a time ordering as well as a behavioural factor. In addition to the capture-recapture approach, there are other methods which can be applied to find estimations of population size, such as *distance sampling*, *removal methods* and *catch-effort*, which are briefly discussed in the following section.

Various Estimation Methods

This section will cover three main methods, previously discussed by Borchers *et al.* (2002) [6].

Distance Sampling

The distance sampling method uses the notion of measuring the distances between an observer and observed individuals in order to estimate abundance. The two primary methods used in distance sampling are line transect sampling and point transect sampling. For line transect sampling, strips are placed on sections of the survey region - note that the area covered is smaller than the whole survey region. This area is called the covered region and is the area in which the individuals are searched for. In the middle of these strips are transect lines which are the paths that the observer will move along when trying to detect individuals. A key assumption is that all individuals on the transect line will be detected and the further away from the transect line the individual is, the smaller the probability of detection. The distance measured is the perpendicular distance between the transect line and the observed individual and this is sometimes easier calculated by finding the radial distance and angle between the observer and the individual, then using the sine rule to convert this [6]. After obtaining these distances, a detection function is fitted and this is used to estimate the proportion of individuals not seen in the population during the survey [31]. Point transect is similar to transect sampling but instead of strips, circular plots each containing a point are considered on the survey region. Observers detect individuals from these points and records the distance between them, where it is assumed that if the individual is close to or on the point, they will be detected and the further away from the point, the smaller the detection probability.

Removal Method (Point Estimation)

Removal methods estimate population size by removing a known number of individuals on, or after, the first capture occasion. When the population is surveyed for the second time, the reduction in captures between first and second capture occasion should then reflect critical information about the true population. For instance, if the number of captured individuals is similar for two occasions, despite removing a large number of individuals in-between, this indicates the removed proportion of individuals was negligible, suggesting a large population size N. Note that the capture probability is assumed to be constant over time, while in practice this is not always the case, given that there may be bias in reporting catch. Suppose observer A has removed n_1 animals that she has seen from the first capture and observer B has removed n_2 animals from the remaining $N - n_1$ animals. Assuming that observers A and B saw exactly the same proportion of the population present at the time of their captures, then one obtains the estimator $\hat{N} = {n_1}^2/(n_1 - n_2)$. A curve of predicted cumulative removals, specified by the constant probability p and abundance N, is fitted using the estimator to find the best fit in terms of likelihood. The point at which the curve flattens off is the point at which the model predicts that the whole population will have been removed. The estimated abundance is thus the height of the curve at this point.

Catch-Effort

The catch-effort method for closed populations is an extension of the removal method. Recognising that more effort may be put into some of the capture occasions, the capture probability in each capture occasion is assumed to depend on a measure of effort used in the study. Typically, an example of a measure of effort is the number of observers, a larger number of observers is expected to be better at catching individuals than a smaller group of observers. This assumption, however, violates the assumption made in the removal method, which assumes same capture proportion for every occasion. The capture probability is modelled as a functional form of effort and is given as follows

$$p(l_{co}) = 1 - \exp\{-\theta l_{co}\},$$

where l_{co} is the number of observers on occasion co. As l_{co} becomes larger, $p(l_{co})$ goes closer to 1. To account for the fact that one does not know how effort affects the capture probability, an unknown parameter, θ, is added to the function. These unknown parameters are estimated by fitting the function to the data (for instance through maximising the likelihood). The data and method described so far correspond to a 'capture effort' method because no removals occurred, whereas the more common method involves removing all captured individuals.

Comparison of Methods

While all of the methods mentioned above can estimate the abundance of closed populations, some methods perform better than others, depending on the context of application. The removal method and catch-effort method generally have poor precision due to potential for large bias. Provided that it is practical to apply capture-recapture or distance sampling method to a given population, the capture-recapture method has several advantages over line transect sampling. Detailed information may be recorded on each captured individual; survival rates may be estimated and recaptures provide information regarding movement. However, the cost of achieving precision on abundance estimates can be significantly higher for capture-recapture than line transact sampling. Typically, the capture-recapture method is more sensitive to failures of assumptions than line transect estimates and may not always estimate populations size reliably.

Literature Review

This paper aims to give an in-depth account of the use of the likelihood function in order to estimate animal abundance in particular. The starting point is given by Bishop et al. (1975), who supplies comprehension and general reference in log-linear and related models for the analysis of discrete multivariate data, in addition to illustrating possible capture-recapture applications, namely volunteer activity in Massachusetts. Schwarz and Seber (1999) review the developments in the subject from 1991 to 1999. They give an overview of all the methodologies available in order to estimate animal abundance such as removal methods, distance methods, change in ratio and relative population density. They also allocate a section to capture-recapture methods, which is the method applied in this paper. However, the authors do not give a detailed mathematical account of the capture-recapture method, but rather a wordy description. Otis et al. (1978) and Pollock (1981) provide a classical estimation of a closed population size, although their work are based on the assumption that all individuals in the population have the same capture probabilities. For models based on the assumption that individuals have unequal capture probabilities, the paper by Chao (1987) provides a point estimate of the population size with its associated confidence interval as well as numerical estimations using the Monte-Carlo method. A detailed account of capture-recapture and the associated ecological models for closed populations, is given in a more recent paper by Chao (2000). This paper is a good starting point and is the basis of the models presented here. Pollock et al. (1989) also provides methods to estimate animal abundance using a variety of methods, one of which is based on ecological models. For methods to estimate population size in open populations, Huggins and Yang (2003) propose models based on local polynomials for capture-recapture experiments. Pledger et al. (2003) propose a likelihood-based approach to estimate the size of open populations and make the use of finite-mixture models to deal with the heterogeneity of the capture probabilities. Huggins (1991), Burnham and Anderson (1992) and Lebreton et al. (1992) has recommended Akaikie's Information Criterion (AIC) as the basis of model selection in the analysis of capture-recapture data, which extended likelihood theory and put the MLE and model selection in one framework.

Aims and Summary

This paper aims to give a detailed review on how to use the general likelihood equation in order to estimate the size of closed population. It also intends to explore the accuracy of the models and compare them when applied to simulated data with known parameters. The paper is split into 3 parts:

Part I: Here, the notation and structure of the data used throughout the report is introduced and the general likelihood equation is described. The ecological assumptions are explained and linked to how they affect the capture probabilities through the use of logistic models.

Part II: This part discusses the theory of using different methods for model testing such as using AIC, BIC and LR-tests when analysing deviance. Confidence interval estimation is also considered, introducing various methods including the use of the asymptotic result of using the Fisher Information and corresponding Hessian matrix to construct the variance-covariance matrix. Non-parametric bootstrap and the delta method are also explained.

Part III: This part demonstrates the application of the capture-recapture method discussed in Part I, where R is used, firstly to simulate data in order to test the models formulated, and secondly to analyse the simulated data sets and other real-world data. The results are discussed.

Part I

Capture-Recapture:
General Likelihood Function &
Logistic Models

1 Data Collection and Structure

Notation is introduced here, as well as, the way the data is structured and stored for later use in the likelihood function. In the capture-recapture setting, there are 2 known quantities: the number of observed individuals, denoted by D and the number of captures or samples, denoted by T. The parameter of interest is the total number of individuals in the population and is denoted by N. In other words, one is interested in finding $N - D$, that is the number of unseen individuals. The individuals are assumed to behave independently. The data can be conveniently expressed as a $D \times T$ matrix where the entry $x_{ij} = 1$ if the ith individual is seen at time j or 0 otherwise, for $i = 1, \ldots, D$ and $j = 1, \ldots, T$. For instance, when the ith individual is captured only the first time out of 4 captures, the capture history can be written as $x_{i(1,0,0,0)}$. The following table illustrates the structure of the data.

	Capture Occasion			
Individuals	1	2	$\ldots$	T
1	x_{11}	x_{12}	x_{1j}	x_{1T}
2	x_{21}	x_{22}	x_{2j}	x_{2T}
$\vdots$	x_{i1}	x_{i2}	x_{ij}	x_{iT}
D	x_{D1}	x_{D2}	x_{Dj}	x_{DT}

Table 1: Structure of the data collection.

Thus, each individual is a Bernoulli trial with probability of success π_{ij} of being captured. The next section introduces the general maximum-likelihood-equation, different ecological models and the logistic models associated.

2 General Likelihood Function

Maximum-likelihood-estimation (MLE) has been the basis for most modern methods of estimating parameters from individual marking experiments, due to its general applicability and the desirable properties of the maximum-likelihood-estimators. In order to find the maximum-likelihood-estimates of the parameters of interest (π_{ij} and N), one needs to define the general likelihood function. It is worth noting that the capture probabilities π_{ij} are nuisance parameters, in a sense that they are only needed for the estimation of N. As mentioned in section 1, it is assumed that a closed population of unknown size N is sampled T times, with independence between individuals. An individual is permanently marked at its first capture to make it recognisable on any subsequent captures. The capture of the ith individual at the jth sample is taken to be a Bernoulli trial with probability π_{ij} of capture. Recall that D denote number of distinct individauls caught, the data form a $D \times T$ capture matrix with element $x_{ij} = 1$ if individual i is captured on occasion j; otherwise, $x_{ij} = 0$. Assuming that there are N individuals in the population, one may extend the capture matrix with $N - D$ rows of zeros to include the uncaught individuals.

The data structure follows a multinomial distribution and the capture probabilities π_{ij} of ecological models will be modelled as either fixed parameters or realisations of random variables. The general likelihood function can be written as

$$L(N, \pi_{ij}|X) \propto \frac{N!}{(N-D)!} \cdot \prod_{i=1}^{N} \prod_{j=1}^{T} \left\{ {\pi_{ij}}^{x_{ij}} \cdot (1-\pi_{ij})^{1-x_{ij}} \right\}, \tag{1}$$

where the fraction $\{N!/(N-D)!\}$ is part of the multinomial coefficient, which is an extension of the binomial coefficient with more than two possible mutually exclusive outcomes. The remaining part of the factorial term is omitted as it does depend on any of the parameters, which is why proportionality is used. The term $\prod_{i=1}^{N} \prod_{j=1}^{T} \{ {\pi_{ij}}^{x_{ij}} \cdot (1-\pi_{ij})^{1-x_{ij}} \}$ selects the corresponding capture probability for each x_{ij} from the data matrix. When an individual i is seen in capture j, the contribution to the likelihood function is π_{ij}, otherwise the individual is unseen and the contribution becomes $1 - \pi_{ij}$. This is taken for all individuals, across all capture occasions. The general likelihood function can be partitioned into seen and unseen individuals as

$$L(N, \pi_{ij}|X) \propto \frac{N!}{(N-D)!} \cdot \prod_{i=1}^{D} \prod_{j=1}^{T} \left\{ {\pi_{ij}}^{x_{ij}} \cdot (1-\pi_{ij})^{1-x_{ij}} \right\} \cdot \prod_{i=D+1}^{N} \prod_{j=1}^{T} (1-\pi_{ij}). \tag{2}$$

This form is particularly useful in order to maximise the likelihood function in R as it removes the dependence between the unknown N and the data x_{ij}.

3 Ecological Models

Heterogeneity is a concept that makes estimation using capture-recapture complex. There are many types of heterogeneity including time, behavioural dependence and dependence on each individuals characteristics. As an example, imagine there are different balls of varied size placed on a field which is covered in snow. Observer A walks around the field to mark the balls she observes (this is done by noting on a map where they are seen - since the balls are stationary they would not move and thus can be marked in this way). This is repeated by observer B. Intuitively, white balls would be harder to see than black balls, similarly small balls would be more difficult to observe than large balls. As a result of this, all the different classifications of balls should have a different capture probability. These illustrate some examples of heterogeneity. If analysis was made without taking into account different types of heterogeneity, this might lead to the analysis being skewed which would give unreliable estimations of the abundance. Models have been designed to classify different sources of heterogeneity. The simple notation used to classify all the models was developed by Pollock (1974) and Otis *et al.* (1978) where the subscript $*$ of M_* specifies the source of heterogeneity. The eight basic models M_0, M_t, M_b, M_h, M_{tb}, M_{bh}, M_{th}, and M_{tbh} are next described in section 4 where subscripts t, b and h indicate heterogeneity due to time dependence, behavioural dependence and individual characteristics respectively. Figure 2 below shows the hierarchy of the models, with M_{tbh} as the most general model and M_0 as the simplest model. Model M_0 is a special case of all models. For example, for the behavioural model M_b, if the capture probability up to first capture is set equal to the recapture probability, this model is equivalent to the M_0 model [33].

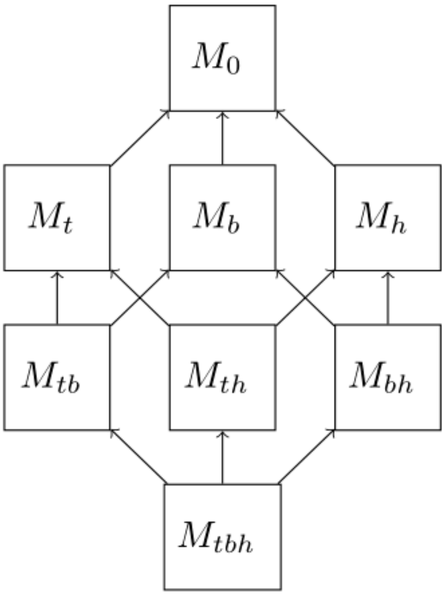

Figure 2: Relations among the models. Each arrow points from one model to an immediate special case of that model.

These are the regular additive models with no interaction effect. At the end of this section, interaction between factors will be briefly discussed. Note that, the following logistic models use corner-point constraints: all the parameters indexed by 1 are equal to 0. A desirable property of logistic models is that parameters representing probabilities are constrained between 0 and 1 due to the construction of $\text{logit}(x)$, where $\text{logit}(x) = \log[x/(1-x)]$. Furthermore, logistic models are more appropriate than log-linear models in order to deal with ecological assumptions such as behavioural effect. In the logistic models, the parameters α, β_j, δ_i and γ represent the constant effect, time effects, individual heterogeneity and behavioural response respectively.

4 Model Classes

Model M_0

This model does not take into account any heterogeneity. All individuals are assumed to have the same capture probability and are equally likely to be caught. The capture probabilities do not differ for each of the capture occasions. Thus, the capture probability π_{ij} is a constant in this case and $\pi_{ij} = \pi$. So, the likelihood function is

$$L(N, \pi_{ij}|X) \propto \frac{N!}{(N-D)!} \cdot \prod_{i=1}^{N}\prod_{j=1}^{T} \left\{ {\pi_{ij}}^{x_{ij}} (1-\pi_{ij})^{1-x_{ij}} \right\}, \tag{3}$$

where

$$\log\left(\frac{\pi_{ij}}{1-\pi_{ij}}\right) = \alpha, \quad i = 1, \ldots, N \quad j = 1, \ldots, T. \tag{4}$$

This can be rewritten as

$$L(N, \pi|X) \propto \frac{N!}{(N-D)!} \pi^{\sum_{i=1}^{N}\sum_{j=1}^{T} x_{ij}} (1-\pi)^{NT-\sum_{i=1}^{N}\sum_{j=1}^{T} x_{ij}}. \tag{5}$$

For this model, the maximum-likelihood-estimates $\hat{N}$ and $\hat{\pi}$ can be easily worked out.

Taking the log of (5) gives

$$\begin{aligned}\log[L(N, \pi|X)] = l(N, \pi|X) \propto \log(N!) - \log[(N-D)!] + \sum_{i=1}^{N}\sum_{j=1}^{T} x_{ij} \log(\pi) + \\ \left(NT - \sum_{i=1}^{N}\sum_{j=1}^{T} x_{ij}\right) \log(1-\pi).\end{aligned} \tag{6}$$

Now take the partial derivative with respect to π

$$\frac{\partial l}{\partial \pi} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{T} x_{ij}}{\pi} + \frac{\sum_{i=1}^{N}\sum_{j=1}^{T} x_{ij} - NT}{1-\pi}, \tag{7}$$

equating to 0 and solving for $\hat{\pi}$, which gives

$$\hat{\pi} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{T} x_{ij}}{\hat{N} T}. \tag{8}$$

Similarly, using the result that

$$\frac{d \log(x!)}{dN} \approx \log(x)$$

for sufficiently large x, the population size estimate is given by

$$\hat{N} = \frac{D}{1 - (1 - \hat{\pi})^T}. \tag{9}$$

In words, the population size estimate is the ratio of the number of individuals seen to the complement of the estimated probability of non capture over T subsequent captures. There may be several solutions to the polynomial equation, which arises by substituting (8) in (9), but there is only one sensible solution to the equation due to $\hat{N} \geq D$ (the total population must be larger than or equal to the number of individuals observed).

Model Class M_t

For this model, the capture probabilities depend on the capture occasion only. All individuals have equal capture probability for each capture occasion, but every capture occasion has a different capture probability. For example, individuals may be more likely to get captured in the morning when it is light rather than in the evening when it is dark. Thus, $\pi_{i1} \neq \ldots \neq \pi_{iT}$, for all individuals $i = 1, \ldots, N$ and $\pi_{1j} = \ldots = \pi_{Nj}$, for all capture occasions $j = 1, \ldots, T$. Similar to model M_0, the likelihood function is given by (1), with the logistic model

$$\log\left(\frac{\pi_{ij}}{1 - \pi_{ij}}\right) = \beta_j, \quad i = 1, \ldots, N \quad j = 1, \ldots, T. \tag{10}$$

Alternatively, given the fact that all individuals have the same capture probability for each capture occasion, the capture probability can be rewritten. In this case, let $\pi_{ij} = \pi_j$ where π_j denote the probability that an individual is captured at occasion j. Moreover, let n_j denote the number of individuals captured in the jth sample. The explicit likelihood is then given as

$$L(N, \pi_j | X) \propto \frac{N!}{(N-D)!} \cdot \prod_{j=1}^{T} \left\{ {\pi_j}^{n_j} \cdot (1 - \pi_j)^{N - n_j} \right\}. \tag{11}$$

Similar to the way the likelihood was maximized in M_0, one can proceed and find the maximum-likelihood-estimates for π_j and N.

$$\hat{\pi}_j = \frac{n_j}{\hat{N}}, \quad j = 1, \dots, T,$$

$$\hat{N} = \frac{D}{1 - (1 - \pi_j)^T}. \tag{12}$$

A Simple M_t Model: the Lincoln-Peterson Estimate

Consider a two-sample M_t model with $T = 2$. Let π_1 and π_2 denote the probability of an individual being captured at the first and second capture, respectively. Furthermore, let n_1 and n_2 be the number of individuals marked during the first and second capture, respectively. Suppose that of the n_2 individuals captured at the second occasion, m_2 were already marked.

Assuming that samples are independent, the resulting likelihood function is

$$\begin{aligned} L(N, \pi_1, \pi_2 | n_1, n_2, m_2) \propto \frac{N!}{(N-D)!} (\pi_1 \pi_2)^{m_2} \{\pi_1(1-\pi_2)\}^{n_1 - m_2} \times \\ \{(1-\pi_1)\pi_2)\}^{n_2 - m_2} \{(1-\pi_1)(1-\pi_2)\}^{N-D}, \end{aligned} \tag{13}$$

where $D = n_1 + n_2 - m_2$. Taking the log of (13), and the partial derivative with respect to N, one obtains

$$\frac{\partial \log(L)}{\partial N} = \log N - \log(N - D) + \log((1-\pi_1)(1-\pi_2)).$$

Equating to 0 and solving for N, after a few steps, gives

$$\hat{N} = \frac{D}{\hat{\pi}_2 + \hat{\pi}_1 - \hat{\pi}_1 \hat{\pi}_2}. \tag{14}$$

Now, taking the partial derivative with respect to π_1 yields

$$\frac{\partial \log(L)}{\partial \pi_1} = \frac{m_2}{\pi_1} + \frac{n_1 - m_2}{\pi_1} + \frac{m_2 - n_2}{1 - \pi_1} + \frac{D - N}{1 - \pi_1}.$$

Again, equating to 0 and solving for $\hat{\pi}_1$ gives

$$\hat{\pi}_1 = \frac{n_1}{\hat{N}}. \tag{15}$$

Similarly, for $\hat{\pi}_2$

$$\hat{\pi}_2 = \frac{n_2}{\hat{N}}. \tag{16}$$

Substituting (15) and (16) in (14) as well as replacing D with $n_1 + n_2 - m_2$, one obtain an expression for the estimated population size $\hat{N}$ in terms of n_1, n_2 and m_2 as

$$\hat{N} = \frac{n_1 n_2}{m_2}. \tag{17}$$

Model Class M_b

The capture probabilities for each individual and capture occasion are the same for this model. The model M_b was introduced to account for the behavioural response shown by animals in ecological studies. For instance, animals may exhibit trap-happiness, which increases the probability of being caught again after first capture. On the contrary, animals may become trap-shy, which decreases the probability of being recaptured. These behavioural responses cause difficulties for using capture-recapture methods as trap-happiness leads to underestimation and trap-shyness results in overestimation of animal abundance. The capture probabilities before and after the first capture are different, the likelihood function may be written as (3) with the capture probabilities given by

$$\log\left(\frac{\pi_{ij}}{1-\pi_{ij}}\right) = \alpha + \gamma I_{ij}, \quad i = 1, \dots, N \quad j = 1, \dots, T \tag{18}$$

where

$$I_{i,j} = \begin{cases} 1 & \text{, if individual } i \text{ has been captured before capture instance } j, \\ 0 & \text{, otherwise.} \end{cases}$$

Denote

$$\pi_{ij} = \begin{cases} \pi_{ijr} & \text{, if individual } i \text{ has been captured before capture instance } j, \\ \pi_{ijc} & \text{, otherwise.} \end{cases}$$

Thus for every individual i and capture j, $\pi_{ijc} \neq \pi_{ijr}$ where c and r denote first capture and recapture respectively. This is simplified to $\pi_{ijc} = \pi_c$ and $\pi_{ijr} = \pi_r$.

Model Class M_h

Only individual heterogeneity is considered in this model. For this case, every individual is assumed to have a different capture probability, but those capture probabilities are constant over all of the capture occasions and there is no behavioural effect. Therefore, $\pi_{i1} = \ldots = \pi_{iT}$, for all individuals $i = 1, \ldots, N$ and $\pi_{1j} \neq \ldots \neq \pi_{Nj}$, for all capture occasions $j = 1, \ldots, T$. Consequently, $\pi_{ij} = \pi_i$. The likelihood function for this model may be written as (3) where the capture probabilities are given by

$$\log\left(\frac{\pi_{ij}}{1-\pi_{ij}}\right) = \delta_i, \quad i = 1, \ldots, N \quad j = 1, \ldots, T. \tag{19}$$

However, using the above parametrization gives rise to $N + 1$ parameters (i.e. the unknown population N and the set of capture probabilities $\{\pi_i\}$ where $i = 1, \ldots, N$). This means that the number of parameters is generally greater than the number of data points which makes it impossible to compute the estimated parameters using maximum-likelihood methods. The number of data points is the number of unique capture histories. For example, there can only be two data points for a model with two capture instances (i.e $x_{(0,0)}$ and $x_{(1,1)}$). In practise, one can overcome the problem by grouping individuals sharing the same characteristics such as sex, size, age. Below, a particular case of the logistic model for M_h is given by

$$\log\left(\frac{\pi_{ij}}{1-\pi_{ij}}\right) = \alpha + \beta_{sex} + \beta_{age} + \beta_{weight} \cdot \text{weight}(i), \quad i = 1, \ldots, N \quad j = 1, \ldots, T. \tag{20}$$

where weight(i) is the weight of the ith individual and β_{weight} is the effect of one unit of weight [16]. Note that this is a particular case since one can take different characteristics into account.

Ways to approach estimation of heterogeneity models include using non-parametric MLE and mixture models. Non-parametric MLE involves using kernel to smooth the distribution with no assumption made on the model. Meanwhile, mixtures models are a generalization of (19) and (20), which can be either finite or infinite.

Finite Mixture

For a finite number of captures in discrete capture occasions, discrete distributions may be applied to this latent class model, in which individuals are grouped according to their characteristics. For instance, some individuals have higher capture probabilities than others, then the individuals may be sorted under two categories, a high capture probability group and a low capture probability group. Sorting individuals into categories, where each category group has its on parameter values, causes a reduction in the number of individual capture probability parameters and helps to estimate the abundance based on each group characteristic. This is done by attributing a weight ω_s and a probability π_s to group s [7]. Capture probability density function for two groups can be constructed as,

$$f_{S,2}(x; \pi_1, \pi_2) = \omega_1 f(x; \pi_1) + \omega_2 f(x; \pi_2), \quad \omega_1 + \omega_2 = 1[32].$$

For example, one may be interested in grouping males and females. Say there are 50% of females and let α_1 represent the proportion of females. Then $\alpha_1 = 0.5$.

Furthermore, if one is interested in the size of the individual, say big or small, and that there are 50% of big individuals for both sexes. Then there would be 4 coefficients represent the different combinations of sex and size, and $\alpha_k = 0.25$ for $k = 1, 2, 3, 4$. To each of the group a distinct capture probability π_s, for $s = 1, \ldots, S$. The likelihood function for finite mixture is given below,

$$L(N, \omega, \pi|X) \propto \frac{N!}{(N-D)!} \cdot \prod_{j=0}^{T} \left[\{\sum_{s=1}^{S} \omega_s \pi_s^j (1-\pi_s)^{T-j}\}^{f_j} \right], \tag{21}$$

where $\sum_{s=1}^{S} \omega_s = 1$ and f_j is the number of individuals with j captures [26]. Equation 21 is taken from Pledger (2000) page 436, equation (4). However, the equation from Pledger is thought to be wrong as one would expect the product to go from 0 to T, the number of captures, and not D, the number of individuals seen. The 'corrected' version in 21 is used in the R code to model M_h.

Infinite Mixture

An infinite mixture model can also be used to take into account the heterogeneity of the individuals in the population. In contrast to the finite mixture model where we have a countable set of events, the infinite mixture is made of an infinite uncountable number of events. While one can assign discrete density mass functions to a countable number of events, only a probability density function can be assigned to an infinite uncountable number of events. In particular, for infinite mixture models, the beta distribution is often used since it is defined for $x \in [0, 1]$, which is convenient in probabilistic cases. The likelihood function using the law of total probability for continuous functions is written as

$$L(\Theta|x_i) = P(x_i|\Theta) = \prod_{i=1}^{N} \int_{-\infty}^{\infty} (P(x_i|\Theta, b) \cdot f(b))\, \mathrm{d}\pi_i, \tag{22}$$

where

$$f(b) = Beta(\alpha, \beta) = \frac{1}{B(\alpha, \beta)} \pi_i^{\alpha-1} (1-\pi_i)^{\beta-1}$$

and

$$P(x_i|\Theta, b) = \pi_i^{x_i} (1-\pi_i)^{1-x_i},$$

as $x_i \sim Bin(N, \pi_i)$, where $\pi_i \sim Beta(\alpha, \beta)$.

In (22), x_i, is the number of captures of the ith individual, hence there is only one product sign in front of the integral since it already accounts for time capture in its data structure and $\Theta = [\alpha, \beta, N]$ is a vector of parameters. Since the beta density function is only defined on [0,1], (22) can be rewritten as

$$L(\Theta|x_i) = P(x_i|\Theta) = \prod_{i=1}^{N} \int_0^1 \left(\pi_i^{x_i} (1-\pi_i)^{T-x_i} \frac{1}{B(\alpha,\beta)} \pi_i^{\alpha-1} (1-\pi_i)^{\beta-1} \right) \mathrm{d}\pi_i. \quad (23)$$

$$= P(x_i|\Theta) = \prod_{i=1}^{N} \int_0^1 \left(\frac{1}{B(\alpha,\beta)} \pi_i^{x_i+\alpha-1} (1-\pi_i)^{\beta-1-x_i+T} \right) \mathrm{d}\pi_i.$$

The integrand is recognised to be the cumulative distribution function of the Beta Binomial distribution, $Beta(x_i+\alpha, \beta+T-x_i)$, this is then integrated, giving

$$P(x_i|\Theta) = \prod_{i=1}^{N} \frac{B(x_i+\alpha, \beta+T-x_i)}{B(\alpha,\beta)}, \quad (24)$$

where $B(\alpha,\beta)$ is the Beta function.

Model Class M_{th}

The capture probabilities for this model are dependent on the time of the capture and individual heterogeneity. Therefore, the capture probabilities are different for all individuals and these vary according to the capture occasion in which they are caught. Thus, $\pi_{1j} \neq \ldots \neq \pi_{Nj}$ and $\pi_{i1} \neq \ldots \neq \pi_{iT}$ for all $i = 1, \ldots N$ and $j = 1, \ldots T$. Again the likelihood function can be written as (3) with capture probabilities given by

$$\log\left(\frac{\pi_{ij}}{1-\pi_{ij}}\right) = \delta_i + \beta_j, \quad i = 1, \ldots, N \quad j = 1, \ldots, T. \quad (25)$$

Model Class M_{tb}

For this model classification, capture probabilities are dependent on capture occasion, as well as, behavioural responses (trap-happy/shy) but are assumed to be the same for each individual. Therefore, for all capture occasions $j = 1, \ldots T$, $\pi_{1jr} = \ldots = \pi_{Njr}$. However, $\pi_{i1r} \neq \ldots \neq \pi_{iTr}$ for all $i = 1, \ldots N$ and $\pi_{ijr} \neq \pi_{ijc}$ for all $i = 1, \ldots N$ and $j = 1, \ldots T$. Simplifying gives, $\pi_{ijr} = \pi_{jr}$ and $\pi_{ijc} = \pi_{jc}$. The likelihood function for this model may be written as (3) where the capture probabilities are given by

$$\log\left(\frac{\pi_{ij}}{1-\pi_{ij}}\right) = \beta_j + \gamma I_{ij}, \quad i = 1, \ldots, N \quad j = 1, \ldots, T. \quad (26)$$

where

$$I_{i,j} = \begin{cases} 1 & \text{, if individual } i \text{ has been captured before capture instance } j, \\ 0 & \text{, otherwise.} \end{cases}$$

Similarly to M_b, denote

$$\pi_{ij} = \begin{cases} \pi_{ijr} & \text{, if individual } i \text{ has been captured before capture instance } j, \\ \pi_{ijc} & \text{, otherwise.} \end{cases}$$

As for model M_h when each individual is attributed a unique capture probability, model M_{tb} can also have more parameters than observations. Since the model involves $(2T) - 1 + 1$ parameters ($2T - 1$ parameters for the capture probabilities and recapture probabilities and one for N), one must have $2T \leq z$, where z is the number of unique capture histories. The $2T - 1$ comes from the fact that the recapture probabilities can only happen at $j > 1$.

Model Class M_{bh}

Behavioural response and individual heterogeneity are considered in this model. The first capture probability is different to the recapture probability and these probabilities vary for each individual, however, they are the same for all capture occasions for $j \geq 2$. Thus, $\pi_{1jr} \neq \ldots \neq \pi_{Njr}$ for all $j = 1, \ldots, T$ and $\pi_{ijr} \neq \pi_{ijc}$ for all $j = 1, \ldots, T$ and for all $i = 1, \ldots, N$. Simplifying the notation gives, $\pi_{ijr} = \pi_{ir}$ and $\pi_{ijc} = \pi_{ic}$. The likelihood function can be written as (3) where the capture probabilities are given by

$$\log\left(\frac{\pi_{ij}}{1 - \pi_{ij}}\right) = \delta_i + \gamma I_{ij},\ i = 1, \ldots, N \quad j = 1, \ldots, T. \tag{27}$$

where

$$I_{i,j} = \begin{cases} 1 & \text{, if individual } i \text{ has been captured before capture instance } j, \\ 0 & \text{, otherwise.} \end{cases}$$

Again, denote

$$\pi_{ij} = \begin{cases} \pi_{ijr} & \text{, if individual } i \text{ has been captured before capture instance } j, \\ \pi_{ijc} & \text{, otherwise.} \end{cases}$$

Similarly to model M_h, this model can be easily over-parametrised. In (27), all individuals have a different capture probability, the number of parameters is then $2N + 1$ which is often much larger than the number of data points. Therefore, as for model M_h, partitioning individuals into groups with different capture probabilities is needed. Instead of using (27), one may use

$$\log\left(\frac{\pi_{ij}}{1-\pi_{ij}}\right) = \alpha+\beta_{sex}+\beta_{age}+\beta_{weight}\cdot weight(i)+\gamma I_{ij}, \quad i=1,\dots,N \;\; j=1,\dots,T. \tag{28}$$

where the notation is the same as in (20).

Model Class M_{tbh}

This model includes all sources of heterogeneity: behavioural, capture occasion and individual heterogeneity. It assumes that the first capture probability is different from the recapture probability for each individual and these are also dependent on which capture occasion individuals are captured in. So, $\pi_{1jr} \neq \ldots \neq \pi_{Njr}$ for all $j = 1,\dots,T$, $\pi_{i1r} \neq \ldots \neq \pi_{iTr}$ for all $i = 1,\dots,N$ and $\pi_{ijr} \neq \pi_{ijc}$ for all $i = 1,\dots,N$ and for all $j = 1,\dots,T$. This is the most complex model considered in this report. The logistic model is

$$\log\left(\frac{\pi_{ij}}{1-\pi_{ij}}\right) = \delta_i+\beta_j+\gamma I_{ij}, \quad i=1,\dots,N \;\; j=1,\dots,T. \tag{29}$$

where

$$I_{i,j} = \begin{cases} 1 & \text{, if individual } i \text{ has been captured before capture instance } j, \\ 0 & \text{, otherwise.} \end{cases}$$

and again the likelihood function is expressed as (3).

Models with Interaction

Up to this point, all the logistic models considered were additive models with no interaction. However, one could add interaction terms to account for dependence between effects. Since there are 3 main effects, the saturated model would involve 3 two-way interaction terms and 1 three-way interaction term. The saturated model is

$$\log\left(\frac{\pi_{ij}}{1-\pi_{ij}}\right) = \delta_i+\beta_j+\gamma I_{ij}+(\delta\beta)_{ij}+(\delta\gamma I_{ij})_i+(\beta\gamma I_{ij})_j++(\delta\beta\gamma I_{ij})_{ij}, \tag{30}$$

for $i = 1,\dots,N$ and $j = 1,\dots,T$.

There are two reasons that these interactive logistic models are not considered. Firstly, using interaction often leads to over-parametrisation since including any interaction term means that one must also include all the additive effects. This is because logistic models are hierarchical, in other words, higher-order interactions are unlikely in the absence of their lower-order relatives. Furthermore, while adding interaction terms to the model may improve the fit to the data, it makes the model less parsimonious as explained in section 6.2.

5 Conditional Likelihood

While this paper uses the full likelihood approach in order to find the parameter estimates, one could alternatively use the conditional likelihood approach. Instead of using the total population size N, the number of observed individuals D is used and the likelihood is conditioned on the probability that any individual is encountered at least once. So for example, for model M_t and T captures, the probability π_* of seeing all individuals at least once is

$$\pi_* = 1 - \prod_{j=1}^{T}(1-\pi_j).$$

Then, the conditional likelihood is

$$L(\pi|\mathbf{x}, D) = P(\mathbf{x}, D|\pi) \propto D! \cdot \prod_{i=1}^{F} \left(\frac{f_i}{\pi_*}\right)^{x_i}, \tag{31}$$

where f_i represents all the combination of observed and unobserved individuals across all capture instances, F is the total number of such combination and x_i is the number of individuals with the f_i capture history. For example, if there are 2 observed individuals and 2 capture instances, where both individuals are caught in the first capture but only one is caught in the second, the likelihood is

$$L(D, p_1, p_2|x_1, x_2, x_3, x_4) = P(x_1, x_2, x_3, x_4|n, p_1, p_2) \propto n! \cdot \left(\frac{f_1}{\pi_*}\right)^{x_1} \left(\frac{f_2}{\pi_*}\right)^{x_2} \left(\frac{f_3}{\pi_*}\right)^{x_3} \left(\frac{f_4}{\pi_*}\right)^{x_4}$$

$$= D! \cdot \left(\frac{\pi_1\pi_2}{\pi_*}\right)^{1} \left(\frac{\pi_1(1-\pi_2)}{\pi_*}\right)^{1} \left(\frac{\pi_2(1-\pi_1)}{\pi_*}\right)^{0} \left(\frac{(1-\pi_1)(1-\pi_2)}{\pi_*}\right)^{0}. \tag{32}$$

The main advantage of writing the likelihood in this form is that one can include the individual covariates to model the encounter process. In the full likelihood setting, this is not possible because it includes the term involving the probability of no individual being observed, and no covariate value is available for these individuals.

Part II

Model Testing & Inference

6 Model Testing

6.1 AIC and BIC

Criterion are essential in order to assess which model is the best fit to the data. Models described in the previous section can be assessed through the residual deviance, AIC (Akaike's Information Criterion) and BIC (Shwarz-Bayesian Information Criterion). One advantage of the AIC is that it does not need the models to be nested in order to compare them. On the other hand, the residual deviance can only be used to compare nested models. A nested model is one which is a special case (simplification) of a more general model. For example, M_t, M_b and M_h are nested models of M_{tbh}. However, M_{tb}, M_{th}, and M_{bh} are all non-nested. Both AIC and BIC discourage overfitting by including a penalty term for the number of parameters in the model. The formulas for AIC and BIC are

$$\text{AIC} = -2\log(L) + 2p \tag{33}$$

and

$$\text{BIC} = -2\log(L) + p\log(m), \tag{34}$$

where L is the likelihood function evaluated at the maximum-likelihood estimators, the total number of parameters estimated in the model is denoted by p and m is the total number of observations. The total number of observations is $m = NT$. The AIC is composed of a goodness-of-fit term $-2\log(L)$ and a penalty for the number of parameters estimated $(+2p)$.

Regarding model selection, the preferred model is the one with the minimum AIC and BIC values among a finite set of models. The main difference between AIC and BIC is that BIC is more sensitive towards large number of samples, because of the positive term involving $\log(m)$. In general, using BIC and AIC leads the the same model selection.

6.2 Deviance

As mentioned previously, the deviance can be used to compare the fit of nested models. Despite being limited to nested models, the deviance has an advantageous feature. When using the deviance, one can obtain a significance probability to compare the models. For instance, M_{tbh} may have a smaller AIC than M_{th} and so one would conclude that it offers a better fit. However, when using the deviance, one may find that while the AIC is smaller, model M_{tbh} is not a significantly better fit than the simpler nested model, in which case M_{th} would be deemed as the better model. In general, though, the concept of parsimony of a model prevails. A parsimonious model is a model that offers a good fit to the data with the least number of parameters.

This is because it is more convenient to use this model and it is more suitable to real life application. For instance, if the model of choice includes many factors, one will need to find an individual for whom all these factors are known or measurable which may be difficult in some cases. The deviance D of a model A based on the data $\mathbf{y}$ is calculated as follows

$$D(\mathbf{y}) = -2\log(LR), \tag{35}$$

where LR is the likelihood ratio given by

$$LR = \frac{L(\Theta_A|\mathbf{y})}{L(\Theta_S|\mathbf{y})}. \tag{36}$$

This is the ratio of the likelihood function for the fitted model A over the likelihood function of the saturated model. Now, if model B is a nested model of A, one can compare the fit of the model using the following result

$$D_B(\mathbf{y}) - D_A(\mathbf{y}) \sim \chi^2_{d_f}, \tag{37}$$

where d_f is the difference between the number of degrees of freedom of model A and B. If the significance probability is smaller than the chosen tolerance level α, then model B is chosen over model A with a $(1-\alpha)100\%$ confidence level. If the significance probability is larger, the simpler model is accepted as the better fit.

7 Inference

7.1 Asymptotic Distribution of the Maximum-Likelihood-Estimators

One can use the following theorem in order to compute the variance and confidence interval of the estimates of the parameters.

Theorem 1. *Let $Y_1, \cdots, Y_n \sim f(y|\theta)$ independent and identically distributed random variables, $\theta \in \Theta \subset \mathbb{R}^p$. Under the regularity conditions of Cramer-Rao inequality, MLE asymptotically satisfies*

$$\hat{\theta} \sim N_p(\theta, I_\theta^{-1}),$$

where

$$I_\theta = \mathbb{E}\left[\left(\frac{\partial l}{\partial \theta}\right)^2\right] = -\mathbb{E}\left[\left(\frac{\partial^2 l}{\partial \theta^2}\right)\right],$$

and l denotes the log-likelihood equation [5].

The matrix I_θ is called the Fisher information, whose inverse is the variance-covariance matrix of the maximum-likelihood estimates. The diagonal elements of I_θ^{-1} show the variance of the estimated parameters. The Fisher information can be computed using the Hessian matrix H where

$$H_{ij} = \frac{\partial^2 l}{\partial\theta_i \partial\theta_j},$$

and $I_\theta = -\mathbb{E}[H_\theta]$.

For the estimated Fisher information one has that

$$\hat{I}_\theta = -\mathbb{E}[\hat{H}_\theta] = -\hat{H}_\theta.$$

The estimated Hessian matrix will be used to compute the standard error for $\hat{N}$ in order to write a 95% confidence interval, using Theorem 1. The standard error of the first estimated parameter $\hat{N}$ is given by$\sqrt{-\hat{H}_{11}^{-1}}$. For instance, a 95% confidence interval is then given by

$$\hat{N} \pm 1.96 \cdot \sqrt{\hat{I}_{11}^{-1}}$$

or equivalently,

$$\hat{N} \pm 1.96 \cdot \sqrt{-\hat{H}_{11}^{-1}}.$$

However, this method can lead to inconsistency in computing the confidence intervals. Indeed, it is possible that the lower bound of the interval happens to be smaller than the number of seen individuals D. In which case, the lower bound is set to be D. Using a non-parametric bootstrap to compute confidence intervals is one way of dealing with this issue.

7.2 Bootstrap

An alternative approach to obtaining a standard error or confidence interval is to consider a non-parametric bootstrapping approach. No assumption on the distribution of the data is made and so this method can be used in any case. The results obtained using this approach will then be compared to the results using method in section 7.3. One of the advantages of using a non-parametric bootstrap compared to using the Hessian matrix, is that one never obtains a confidence interval where the lower bound of the interval is smaller than the number of seen individuals D. However, this is a computationally demanding method and one has to trade off efficiency for accuracy. The method consists of randomly re-sampling the rows of the original data matrix. Each row has the same probability of being picked and is replaced into the sample after being picked, so that the same row can be chosen several times. As a result, a new matrix of same dimensions as the original matrix is created. This process is repeated a large number of times, generally 999 times and adding the original matrix to get 1000 samples. For each sampled matrix, one would have to calculate the corresponding maximum-likelihood-estimators of the parameters. The sample standard error of the corresponding estimators of the parameters can be used to estimate the standard error. Similarly, a 95% (percentile) bootstrap confidence interval can be obtained for the parameters of interest, by considering the corresponding quantiles of the bootstrapped (sampled) estimators.

7.3 Delta Method

The delta method is another way of computing estimated standard errors. It is used when one already knows the variance of some parameters and wants to estimate the variance of a function of the parameters. Let f_p be a function of P variables (parameters) $p_1, \ldots, p_P$ and V be the variance-covariance matrix of the P parameters, then the estimated variance $\mathrm{Var}(\hat{f_p})$ is given by

$$Var(\hat{f_p}) = \nabla f_p^T V \nabla f_p,$$

where

$$\nabla f_p^T = \left(\frac{\partial f_p}{\partial p_1}, \ldots, \frac{\partial f_p}{\partial p_P} \right),$$

is evaluated at the maximum-likelihood-estimators of the parameters. This method will be used to estimate confidence intervals for the estimated probabilities since one obtains the estimated standard error of the parameter which the capture probability is a function of.

Part III

Analysis of Data Sets

8 Applying Theory

In this part, the likelihood equations for models M_0, M_t, M_b and M_h are used to estimate population size on both simulated and real world data. A single simulation will be run for each model to serve as an example before simulating a large number of capture histories in order to get consistent results. Since calculating the maximum-likelihood-estimators is computationally expensive for complex models such as M_t and M_b, the software R is used to optimize the likelihood functions. The MLE of the population size N and the capture probabilities π_{ij}, are then calculated. In this paper, the choice of R was natural as it explicitly shows how the likelihood function is maximized in order to obtain the maximum-likelihood-estimators. However, note that estimations of animal abundance can be made in a specialised program called MARK. This program can deal with many ecological models for both open and closed populations, but is not considered in this report.

In order to maximise the log-likelihood equation using R, an in-built optimisation function called `optim` was used. Note that by default this function performs minimisation, thus, it is necessary to use the negative of the log-likelihood equation in this function in order to find the MLE. Since the negative of the log-likelihood equation is minimized, this equation evaluated at the optimized parameter values found is equivalent to the negative of the maximum log-likelihood estimated at these parameter values. To use `optim` it is necessary to specify the initial search values for the parameters and the function to be minimized. Note that the values of the maximum-likelihood-estimators for models M_t and M_b are very sensitive to those initial search values. One should thus run this function several times using different initial search parameters, as the function may have only found the local minimum and not the global minimum using that initial search parameter. However, when the same initial search values are used for many sampled matrices (e.g, in the bootstrap code), one can obtain wrong estimates of the parameters as the `optim` function would not converge in cases where the sampled matrix would contain only extremes values. The function `optim` returns the following:

- The optimized estimates of the parameter values, $\hat{N}$ and $\hat{\pi}_{ij}$.
- The value of the function at the optimized parameter values, where in this case would be of the negative log-likelihood function, therefore, this value must be multiplied by -1 for the value of the log-likelihood at the MLE.
- A value for convergence which indicates the status of completion of the optimisation (0 indicates successful completion, 1 indicates that the maximum number of iterations have been reached and the estimates found have not fully converged yet - unsuccessful completion). When the error code is 1, the optimized parameter estimates should replace the current initial search parameters in `optim`, and the optimisation process should be repeated. This is done until full convergence (0) or until there are negligible changes to the estimates, which suggests that the estimates have converged, but perhaps not to the function's tolerance level.

- There is an option to output the Hessian matrix. Since the input function (negative log-likelihood) is used to calculate this matrix, it is required to multiply it by -1 since it is the Hessian of the log-likelihood equation which is needed. This is used to calculate the 'Hessian' confidence intervals in this section as mentioned in section 7.1. Remember though, that this is an asymptotic result and that it may not be accurate for a small number of data points (usually less than 30), where the number of data points is the number of distinct capture histories.

9 Simulated Data Sets

The R function `F.cjs.simulate` in package `mra` [18] was used to simulate data sets for several of the model classes. The function requires specified capture probabilities as well as a total number of individuals, N, and the total number of captures, T. This function can deal with either open and closed populations. Once the required arguments are imputed, the function generates a $D \times T$ matrix where the cell $x_{ij} = 1$ if the individual i is captured on the jth capture occasion and 0 otherwise for $i = 1, \ldots, D$ and $j = 1, \ldots, T$, alike the notation introduced before. Note that, in some cases, it may happen that all the individuals in the population are seen and one gets $D = N$. The motivation for simulating data before using real world data is that the probabilities π_{ij} and the population size N are known beforehand. This allows for investigation into the accuracy and limitations of the models by comparing the estimates obtained to the real values of the parameters, only then can real data sets be used. Data (capture histories) will be simulated under M_0, M_t, M_b, and M_h assumptions, in turn. For each set of simulated data, N will be estimated using all four models. The maximum-likelihood-estimators and the AIC will be computed and compared across the models, as well as the confidence intervals using both the Fisher information and the bootstrapping method described earlier. Note that since only the basic models are analysed, it is not appropriate to use deviance as a criteria for assessing model fitting as the models considered are not nested.

9.1 Single Data Set Examples

Before using repeated simulated data sets in section 9.3, to obtain more constructive results, the analysis is performed on small single data sets in order to introduce the key values of the analysis. The confidence intervals using the bootstrap methods make use of 1000 samples. Note that '$-$' in the tables below indicates that results were not produced for the corresponding values for the two following reasons. Firstly, the time taken to compute bootstrap confidence intervals for models M_b and M_t is too long, and thus are not calculated. Secondly, models may fail to estimate the maximum-likelihood-estimators.

Model M_0

	co_j						co_j				
Individual	1	2	3	4	5	Individual	1	2	3	4	5
1	0	0	0	1	0	10	1	0	0	0	1
2	0	0	1	0	0	11	1	1	0	0	0
3	0	1	0	0	1	12	0	0	0	0	1
4	0	1	1	0	0	13	0	1	0	0	1
5	1	0	0	1	1	14	1	0	0	0	0
6	0	0	1	0	0	15	0	0	1	0	0
7	0	1	0	0	0	16	1	0	1	1	0
8	0	0	0	1	0	17	0	1	0	1	0
9	0	0	0	1	0						

Table 2: Representative data for model M_0

The data in Table 2 above was simulated with 5 capture occasions (co), a population size set to $N = 20$ and a capture probability $\pi = 0.3$. Note that a data set of $N = 20$ is considered as very small. Generally, the number of individuals in the population is considered to be small for $N \leq 200$. In this example, 17 individuals were seen and 27 captures were recorded in the study out of 85 possible captures. This means there are 58 capture occasions where individuals were not being captured.

	M_0	M_t	M_b	M_h
$\hat{N}$	21.99	21.95	19.23	21.99
Convergence	0	1	0	0
Log-likelihood at MLE	-17.62	-17.47	-17.40	-17.61
AIC	39.24	46.94	40.80	43.24

Table 3: Value of $\hat{N}$, log-likelihood at MLE and AIC under all four models using data from Table 2.

Shown in Table 3 are the results for the population size estimates N under model assumptions M_0, M_t, M_b and M_h respectively, where the value of the log-likelihood at the MLE is used to calculate the AIC as shown in 33. Note that for M_h individuals are assumed to exhibit different characteristics and hence are split into 2 groups. For the M_0 simulated dataset, all models except M_t converged. M_t may not have converged as a result of requiring large number of capture probability parameters. While all models seem to give accurate point estimates for N, the model selection is based on the AIC criterion. Model M_0 gives the lowest AIC value, thus it is the most appropriate model to use for this simulated set of data as one would expect. However, often one is also interested in an interval estimate as well as a point estimate. Table 4 below summarises the values of the estimated standard error and confidence intervals.

	M_0	M_t	M_b	M_h
$\widehat{ese}(\hat{N})$	2.70	2.69	1.79	2.70
95% Hessian CI	[17.00 27.28]	[17.00 27.22]	[17.00 22.74]	[17.00 27.28]
95% bootstrap CI	[18.63 37.30]	–	–	[18.63 37.78]

Table 4: Value of estimated standard errors, 95% confidence intervals using the Hessian matrix and 95% bootstrap confidence intervals for N, under all four models using data from Table 2.

Note that the lower bounds of the confidence intervals using the Hessian matrix are truncated and set to $D = 17$. This happens when $\hat{N}$ is close to the number of captured individuals D, which may result in the lower bound being less than D once $1.96 \cdot \widehat{ese}(\hat{N})$ is subtracted. Setting the lower bound to D is equivalent to having no information on the lower bound. In this case, $N = 20$ belongs in all the 95% confidence intervals. The main differences between the confidence intervals using the Hessian matrix and the bootstrap is that the bootstrap gives information on the lower bound but seems to give a much larger upper bound, thus putting more uncertainty on the value of the real parameter N. Note that, as explained in section 7.3, the confidence intervals using bootstrap for M_t and M_b are not recorded due to very long computation time. Figure 3 below shows the frequency of the estimated N for the bootstrap under models M_h and M_0. Most of the estimations falls around 20, while there are few estimations larger than 40.

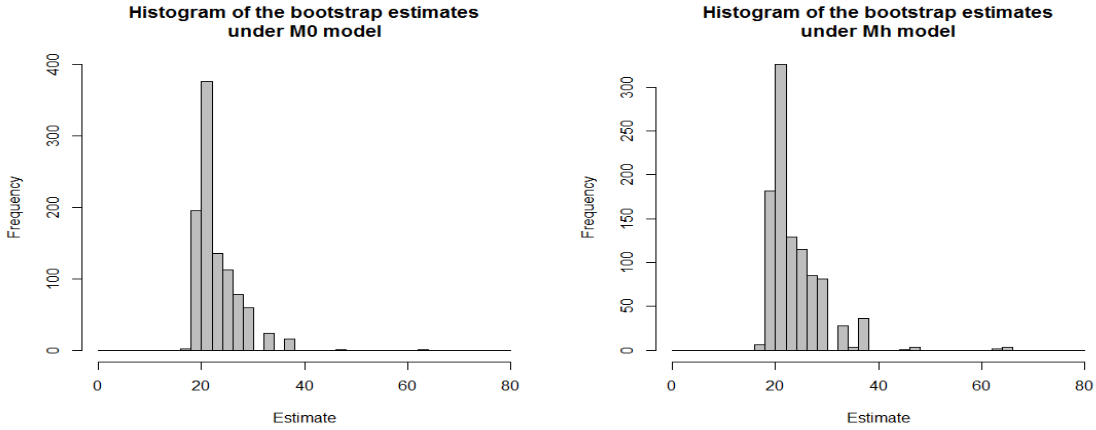

Figure 3: Histograms of the bootstrap estimates $\hat{N}$ for M_0 and M_h models. The data was simulated under M_0 and 1000 samples were used.

In addition, the confidence interval for the probability estimate, $\hat{\pi}$, of M_0 can be constructed by using the delta method introduced in section 7.3. The 95% confidence interval for π is [0.17 0.33] and the true probability, $\pi = 0.3$, falls in the interval.

Model M_t

Data below were simulated under model M_t with $N = 20$, $co = 5$, and capture probabilities $\pi_1 = 0.68$, $\pi_2 = 0.40$, $\pi_3 = 0.51$, $\pi_4 = 0.18$, $\pi_5 = 0.24$.

	co_j						co_j				
Individual	1	2	3	4	5	Individual	1	2	3	4	5
1	1	0	1	0	0	9	1	0	1	0	1
2	1	1	0	0	0	10	0	1	0	1	0
3	1	1	0	0	0	11	1	0	1	0	0
4	1	0	1	1	0	12	0	0	1	0	0
5	1	1	0	0	0	13	1	0	1	0	1
6	0	0	1	0	0	14	1	0	0	0	0
7	1	1	0	0	0	15	1	0	1	0	0
8	1	1	0	0	0	16	0	0	1	0	1

Table 5: Representative data for model M_t

	M_0	M_t	M_b	M_h
$\hat{N}$	17.21	16.01	–	17.25
Convergence	0	1	1	0
Log-likelihood at MLE	−22.78	−13.56	–	−22.73
AIC	49.57	39.13	–	53.45

Table 6: Values of $\hat{N}$ and AIC under all four models using data from Table 5.

	M_0	M_t	M_b	M_h
$\widehat{ese}(\hat{N})$	1.39	0.80	–	1.42
95% Hessian CI	[16.00 19.94]	[16.00 18.37]	–	[16.00 20.02]
95% bootstrap CI	[16.14 21.04]	–	–	[16.56 235.61]

Table 7: Value of estimated standard errors, 95% confidence intervals using the Hessian matrix and 95% bootstrap confidence intervals for N, under all four models using data from Table 5.

For the data set simulated under M_t, none of the population estimates $\hat{N}$ are accurate. Model M_b failed to compute maximum-likelihood-estimates for this data, which is why there there is no information in Table 7. While model M_t has the lowest AIC, model M_0 provides a better estimate for N. Even though M_t did not give an accurate $\hat{N}$ it estimated the probabilities accurately, $\hat{\pi}_1 = 0.79$, $\hat{\pi}_2 = 0.67$, $\hat{\pi}_3 = 0.61$, $\hat{\pi}_4 = 0.12$, and $\hat{\pi}_5 = 0.22$. Therefore, the relatively large log-likelihood value for model M_t is not due to accurately predicting N but rather due to the accuracy of the estimated probabilities. Model M_h is the only model that provides a confidence interval using the Hessian which contains the true value N. Confidence interval using the bootstrap method performs well for M_0 but gives a very large upper bound for M_h. This may be the result from extreme captures histories (for instance, many capture histories with very few captures) sampled into the sampled matrices, leading to overestimation of the upper bound. It is therefore useful to compute confidence intervals using both methods. Table 8 below provides 95% confidence intervals using the delta method for the probability estimates of model M_t. All true probabilities except π_2 fall in the intervals.

	$\hat{\pi}_1$	$\hat{\pi}_2$	$\hat{\pi}_3$	$\hat{\pi}_4$	$\hat{\pi}_5$
CI	[0.59 0.99]	[0.43 0.90]	[0.37 0.85]	[0.00 0.28]	[0.02 0.42]
	π_1	π_2	π_3	π_4	π_5
Probability	0.68	0.40	0.51	0.18	0.24

Table 8: 95% confidence intervals for model M_t probability estimates using the delta method.

Model M_b

For simulation under assumption M_b, the data has been generated under two different probabilities (capture and recapture) to reflect behavioural effects. The capture and recapture probabilities are respectively $\pi_c = 0.3$ and $\pi_r = 0.4$.

	co_j						co_j				
Individual	1	2	3	4	5	Individual	1	2	3	4	5
1	1	1	0	0	0	9	0	0	1	0	1
2	0	0	1	0	1	10	0	0	0	1	0
3	1	1	0	0	0	11	0	1	1	1	0
4	0	0	0	1	0	12	0	1	1	1	0
5	0	1	1	0	0	13	0	0	0	1	0
6	1	0	1	1	0	14	0	1	0	1	1
7	0	1	0	1	1	15	0	0	0	0	1
8	0	0	0	0	1	16	1	1	0	0	1

Table 9: Representative data for model M_b

	M_0	M_t	M_b	M_h
$\hat{N}$	16.96	16.87	20.87	16.96
Convergence	0	1	0	0
Log-likelihood at MLE	-23.27	-21.83	-22.37	-23.27
AIC	50.54	55.66	50.73	54.54

Table 10: Values of $\hat{N}$ and AIC under all four models using data from Table 9.

	M_0	M_t	M_b	M_h
$\widehat{ese}(\hat{N})$	1.28	1.23	2.68	1.28
95% Hessian CI	[16.00 19.48]	[16.00 19.30]	[16.00 26.13]	[16.00 19.48]
95% bootstrap CI	[16.22 19.36]	–	–	[16.08 22.23]

Table 11: Value of estimated standard errors, 95% confidence intervals using the Hessian matrix and 95% bootstrap confidence intervals for N, under all four models using data from Table 9.

As seen in Table 10, M_0 and M_b have the lower AIC values among the four models. Here, using AIC to choose the best fitting model can be misleading due to the small sample size of the data. This is because AIC is composed of a goodness-of-fit term $-2\log(L)$ and a penalty for the number of parameters estimated $(+2p)$, as mentioned in section 6.1. Model M_b has the best fit since its log-likelihood value is larger than the log-likelihood value of M_0 but estimates 3 parameters. Thus the AIC judges that the improvement in fit does not justify the larger number of parameters. Furthermore, M_b is the only model that includes N in the confidence interval using the Hessian. Therefore, based on the analysis, M_b is the most appropriate model for population estimation of this simulated data set under model M_b. Table 12 below shows the 95% confidence intervals for the capture and recapture probabilities. Capture probability π_c falls within the interval while recapture probability π_r does not. These results could be improved by increasing the number of captures or setting a larger population size.

	$\hat{\pi}_c$	$\hat{\pi}_r$
CI	[0.16 0.32]	[0.60 0.85]
	π_c	π_r
Probability	0.30	0.40

Table 12: 95% confidence intervals for model M_b probability estimates using the delta method.

Model M_h

Here, the data is simulated under model M_h for two groups. Group 1 has capture probability $\pi_1 = 0.31$ and group 2 has capture probability $\pi_2 = 0.12$.

Model M_h gives the minimum AIC for this data set while M_0 gives the $\hat{N}$ closest to N. Model M_h outputs probability estimates, $\hat{\pi}_1 = 0.18$ and $\hat{\pi}_2 = 0.18$. Although the estimated capture probabilities for the two groups are the same, due to a small sample size, these probability estimates are relatively close to the real capture probabilities. For confidence intervals using the Hessian, models M_0, M_t, and M_h include N inside the intervals. For model M_0, N falls in the bootstrap confidence interval. The bootstrap method used for model M_h fails to give an accurate upper bound to the confidence interval, similar to the previous case when the data was simulated under model M_t. The delta method is used to calculate the confidence intervals for

	co_j						co_j				
Individual	1	2	3	4	5	Individual	1	2	3	4	5
1	1	0	1	0	0	9	0	0	0	0	1
2	0	1	0	0	0	10	1	1	1	0	1
3	0	0	0	1	0	11	1	0	1	0	0
4	0	0	0	0	1	12	1	0	1	0	0
5	0	1	1	0	0	13	0	1	0	0	0
6	1	0	0	1	0	14	1	0	0	0	0
7	0	0	1	0	0	15	0	0	1	0	0
8	0	1	0	1	0						

Table 13: Representative Data: Model M_h

	M_0	M_t	M_b	M_h
$\hat{N}$	19.11	18.73	15.78	23.57
Convergence	0	1	0	0
Log-likelihood at MLE	-17.53	-15.72	-16.89	-11.15
AIC	39.06	43.45	39.77	30.30

Table 14: Values of $\hat{N}$ and AIC under all four models using data from Table 13.

	M_0	M_t	M_b	M_h
$\widehat{ese}(\hat{N})$	2.46	2.33	1.21	3.81
95% Hessian CI	[15.00 23.93]	[15.00 23.31]	[15.00 18.16]	[16.09 31.05]
95% bootstrap CI	[15.44 36.68]	–	–	[18.30 20871.15]

Table 15: Value of estimated standard error, 95% Hessian confidence interval and 95% bootstrap confidence interval of $\hat{N}$ respectively, under all four models using data from Table 13.

the capture probabilities. From Table 16, one can see that π_2 is contained in the corresponding confidence interval, but π_1 is not.

	$\hat{\pi}_1$	$\hat{\pi}_2$
CI	[0.09 0.27]	[0.00 0.42]
	π_1	π_2
Probability	0.31	0.12

Table 16: 95% confidence intervals for model M_h probability estimates using the delta method.

Summary of Models

The data were investigated with two methods: point estimation and interval estimation. Data were simulated with $N = 20$ and $co = 5$. Even though there were not enough individuals (N=20) to give a firm conclusion on this data analysis, there were some interesting outcomes. The best model was chosen to be the model with the lowest AIC value. However, it has been shown that AIC does not necessarily lead to the selection of the best model, in the case of small sample sizes. For all data, the best model to estimate the parameters was the one used to simulate the data. Additionally, model M_t was deemed to give the worst fit to the simulated data for models M_0, M_b and M_h due to the fact that AIC penalises for using a large number of parameters. This takes into account the parsimony of the model as explained in section 6.2.

Confidence intervals can provide useful information on which model gives a good estimation of the parameters (N in this case). Confidence intervals using the bootstrap method and the Hessian matrix were used here. Lower bounds for the confidence intervals using the Hessian were truncated whenever the lower bound was smaller than the number of captured animals. For data simulated using each model, N is contained in the confidence interval using the Hessian for the corresponding model, except for model M_t.

Section 9.2 highlights how the models estimate the parameters when different parameter values are used to simulate data.

9.2 Changes in Parameters

To test model sensitivity and accuracy, parameter values are modified to create new simulated data.

Model M_0 with $\pi = 0.13$ and $co = 5$

Data are simulated with $N = 20$, $co = 5$, and modified capture probability $\pi = 0.13$.

	co_j						co_j				
Individual	1	2	3	4	5	Individual	1	2	3	4	5
1	0	0	1	0	0	7	0	0	0	0	1
2	0	0	0	0	1	8	0	0	0	1	0
3	0	0	0	0	1	9	0	0	1	0	0
4	0	0	0	0	1	10	0	0	1	0	0
5	1	0	0	0	0	11	1	0	0	0	0
6	0	0	0	1	0						

Table 17: Representative Data: Model M_0 with $\pi = 0.13$ and $co = 5$.

	M_0	M_t	M_b	M_h
$\hat{N}$	43035.50	68.57	270.19	12893.64
Convergence	1	1	0	1
Log-likelihood at MLE	−2.32	−0.17	−2.35	−2.33
AIC	8.64	12.34	10.70	12.66

Table 18: Values of $\hat{N}$ and AIC under all four models using data from Table 17.

Capture probability was chosen such that only half of the individuals in the population should be seen. Therefore,

$$(1 - \pi)^T = 0.5$$

and so in this case, $\pi = 0.13$ for $T = 5$. Thus, there were only 11 captured individuals out of 20 total population. With such a small number of seen individuals, the models gave poor estimations and largely overestimated the estimated population size. Model M_t has the closest $\hat{N}$ and the largest log-likelihood value. Additionally, models M_0, M_t and M_h did not converge. Note that model M_t did not converge in any of the previous analysis but this does not necessarily mean that the results are poor, as the non-convergence is due to the large number of estimated parameters. However, models M_h and M_0 usually converges, which indicates that the results from these models are not meaningful here. Next, the capture probability is kept the same but the number of captures is increased from 5 to 8.

Model M_0 with $\pi = 0.13$ and $co = 8$

To test whether the models can have a better estimation with more data, 8 captures are performed. Other parameters N and π are the same as in the previous case, $N = 20$ and $\pi = 0.13$.

	co_j									co_j							
ID	1	2	3	4	5	6	7	8	ID	1	2	3	4	5	6	7	8
1	0	1	0	0	1	0	0	0	9	0	1	0	0	1	1	0	0
2	0	0	0	0	0	0	0	1	10	0	0	1	0	1	0	0	0
3	0	0	0	0	0	1	0	1	11	0	0	1	0	1	0	0	0
4	0	0	1	0	0	0	0	0	12	0	0	1	0	0	0	0	0
5	0	0	0	0	0	1	0	0	13	0	0	0	0	0	0	1	0
6	0	0	0	0	1	0	0	1	14	0	0	0	0	0	0	1	0
7	1	0	0	0	0	0	0	0	15	0	0	0	0	1	0	0	0
8	1	0	0	0	0	0	0	0	16	0	0	0	0	0	0	1	0

Table 19: Representative Data: Model M_0 with $p = 0.13$, $co = 8$

	M_0	M_t	M_b	M_h
$\hat{N}$	25.42	16.00	32.12	25.41
Convergence	0	1	0	0
Log-likelihood at MLE	−26.16	−68.44	−26.10	−26.16
AIC	56.32	154.88	58.20	60.33

Table 20: Values of $\hat{N}$ and AIC under all four models using data from Table 19.

Even though data were simulated under the same capture probability, $\pi = 0.13$, there were 16 captured individuals in this data set. With more capture occasions, it was possible to capture more than the previous case. Clearly, models have done better in estimating the population size. Here, models M_0, M_b, and M_h have converged and outputted $\hat{N} = 25.42$, $\hat{N} = 16.00$, $\hat{N} = 32.12$, and $\hat{N} = 20.53$, respectively. These estimates are relatively close to the true population, $N = 20$, compared to the previous simulated data under model M_0 ($\pi = 0.13$, $co = 5$). Here, model M_0 estimated the capture probability to be $\hat{\pi} = 0.11$. It has the lowest AIC and is thus taken to be the best fitting model. One can see that increasing the number of captures occasions from 5 to 8 has a positive effect on the models' abilities to estimate the population size for very low capture probabilities.

Model M_0 with $\pi = 0.13$ **,** $co = 10$**,** $N = 5000$

Here, large number of data is simulated with $N = 5000$, $co = 10$ and $\pi = 0.13$. Due to a large number of data, the table of the capture histories is not presented. In 10 capture occasions, there were 3729 captured individuals out of 5000.

	M_0	M_t	M_b	M_h
$\hat{N}$	4975.30	3730.73	4911.32	–
Convergence	0	1	0	–
Log-likelihood at MLE	10585.45	7340.892	10585.71	–
AIC	-21166.90	-14659.78	-21165.42	–

Table 21: Values of $\hat{N}$ and AIC using data simulated with $N = 5000$, $\pi = 0.13$ and $co = 10$.

Models M_0 and M_b were able to estimate N accurately. Model M_0 outputted $\hat{\pi} = 0.13$, when $\pi = 0.13$ was used to create the data. However, model M_b has outputted $\pi_c = 0.13$ and $\pi_r = 0.49$. It may have detected some pattern in the data such that it might be appropriate to consider a capture and a recapture probability. Since the AIC values for models M_0 and M_b are almost equal, M_0 is the more accurate model to use due to its better abundance estimate $\hat{N}$. Model M_h failed to estimate the abundance due to `optim` not being able to compute the MLE.

9.3 Multiple Simulations

This section investigates how rigorous the models are in terms of how many times the specified population size falls in the 95% confidence interval computed (using the Hessian), as well as, how the models perform in correctly identifying the best fitting model for the simulated data set by analysing the AIC values. In theory the best fitting model should be the model from which the simulation comes from.

Firstly, data were simulated and the maximum-likelihood estimate of N, AIC as well as the confidence interval using the Hessian were computed using each of the 4 models: M_0, M_t, M_b, and M_h. This process was repeated 50 times, simulating new data each time. For each iteration, the model with the lowest AIC was recorded and these were collated for each model formulation to give a summary of how many times each model had the lowest AIC. In addition, it was recorded whether N was contained in the confidence intervals found. Simulation under two models was considered in this section: M_0 and M_h. These were chosen since M_0 is the simplest model and M_h is the most complex model out of the 4 considered in this paper. Several values of the parameters were analysed in order to investigate the sensitivity of the models. For all investigations, the specified number of individuals in the population was set to $N = 500$ - since this is a compromise between $N = 20$ and $N = 5000$, which were the values used previously. Also, this was a reasonable number to allow for 50 iterations of the models to run in a reasonable amount of time. In addition, the number of capture occasions was set to 5. These parameter values were used throughout this section unless stated otherwise.

Representative Data Under M_0 Model

Considering data simulated from model M_0, it was investigated whether a large capture probability affects the robustness of the model. Setting $\pi = 0.6$, referring to Table 23, notice that model M_0 was indeed the best fitting model most often in terms of AIC, with the lowest AIC in 78% of the iterations. In terms of the number of times N was contained in the confidence intervals, all models apart from M_h performed well. This may be due to the capture probability being too high, causing too many individuals being seen (approximately 490 out of 500 in this case) which led to a slight overestimation. To see if this is the case, π was changed to 0.3 (Table 23) which resulted in a smaller number of individuals seen each time (approximately). Clearly, all models performed better, where the overestimation seen before has been removed and M_0 is the best model most often. Further reducing the capture probability to $\pi = 0.13$ (Table 22) such that only half the population is seen on average, it was shown that all models were still performing well. Notice that previously for $N = 20$, with the same number of capture occasions and capture probabilities (section 9.2), the models failed to estimate the population size correctly for a small sample, whereas for this larger sample size the models give reasonable estimates of N. However, changing the population size from $N = 500$ to $N = 100$ does not lead to any significant changes to the results found (Table 22), perhaps because N is still large enough in this case. Interestingly, model M_b has the lowest number of iterations where N is contained in the confidence interval suggesting that it is consistently the poorest model in terms of this criteria.

In conclusion, model M_0 is robust and is consistently the best model to estimate data simulated under M_0 assumptions. For extreme probabilities, model M_h tends to overestimate the population size for large capture probabilities and underestimate it for small capture probabilities.

	$N = 500$				$N = 100$			
	M_0	M_t	M_b	M_h	M_0	M_t	M_b	M_h
Interval	39	45	15	38	27	31	13	0
AIC	40	3	6	1	44	3	3	0

Table 22: The number of times N was included in the 95% confidence interval (using the Hessian) for each model and the number of times each model was the best model (lowest AIC) for $N = 500$ (left) and $N = 100$ (right), with $\pi = 0.13$ and $co = 5$.

	$\pi = 0.6$				$\pi = 0.3$			
	M_0	M_t	M_b	M_h	M_0	M_t	M_b	M_h
Interval	44	45	41	0	46	45	30	42
AIC	39	1	10	0	41	3	6	0

Table 23: The number of times N was included in the 95% confidence interval (using the Hessian) for each model and the number of times each model was the best model (lowest AIC) for $N = 500$, $\pi = 0.6$ (left) and $\pi = 0.3$ (right) with $co = 5$.

Representative Data Under M_h Model

As the M_h model is complex, the effects of extreme capture probabilities were investigated. For the real data set, an obvious criteria is 'sex' when it comes to grouping individuals into categories, thus 2 equal sized groups were used in this section. Firstly, π_1 and π_2 were set to 0.1 and 0.9. From Table 25, the analysis shows that for all the 50 iterations, model M_h best fits the data according to AIC, as expected since the capture probabilities are so different. Although this may seem like a good result, notice that N is almost contained in no confidence intervals, (which is not surprising for models M_0, M_t and M_b) but for M_h this shows poor performance. This may be due to the extreme differences between the capture probabilities. The number of individuals seen fluctuates around 350 (which is small compared to N) for each iteration and this causes major underestimation for N and the confidence intervals. Next, the capture probabilities are made to differ by a less extreme amount, setting $\pi_1 = 0.3$ and $\pi_2 = 0.7$. From Table 24 notice that again, the best model is M_h. The confidence intervals are still poor for the other models but have significantly improved for model M_h, with N being contained in the interval for 80% of the iterations, for $co = 5$. This shows that the difference in the capture probabilities were too severe previously as M_h does perform well for $\pi_1 = 0.3$ and $\pi_2 = 0.7$. It is checked for completeness whether increasing the number of capture occasions to 7 gives different results, but similar results were found. It was then considered having capture probabilities which are similar, such as $\pi_1 = 0.5$ and $\pi_2 = 0.6$. Referring to Table 25 it can be seen that model M_h is no longer considered as the best model, since the AIC is lower more frequently for models M_b and in particular M_0. This has happened because the capture probabilities are so similar and model M_0 has less parameters, causing the AIC to be lower. Looking at the confidence intervals, all models perform well. This is expected since the models M_t, M_b and M_h are all particular cases of the M_0 model, but they are equivalent when the probabilities are the same.

	$co = 5$				$co = 7$			
	M_0	M_t	M_b	M_h	M_0	M_t	M_b	M_h
Interval	0	0	0	40	0	0	2	45
AIC	0	0	0	50	0	0	0	50

Table 24: The number of times N was included in the 95% confidence interval (using the Hessian) for each model and the number of times each model was the best model (lowest AIC) for $N = 500$, $\pi_1 = 0.3$ and $\pi_2 = 0.7$ with $co = 5$ (left) and $co = 7$ (right).

	$\pi_1 = 0.1$ $\pi_2 = 0.9$				$\pi_1 = 0.5$ $\pi_2 = 0.6$			
	M_0	M_t	M_b	M_h	M_0	M_t	M_b	M_h
Interval	0	0	0	1	47	46	45	46
AIC	0	0	0	50	31	5	10	4

Table 25: The number of times N was included in the 95% confidence interval (using the Hessian) for each model and the number of times each model was the best model (lowest AIC) for $N = 500$, $\pi_1 = 0.1$ and $\pi_2 = 0.9$ (left), $\pi_1 = 0.5$ and $\pi_2 = 0.6$ (right) with $co = 5$.

10 Real Data Sets

In addition to simulated data sets, an analysis on real data sets was also performed using the previous tested model classes. The selected data sets are taken from closed populations, taking the key assumptions into account.

10.1 Reid's Deer Mice

Figure 4: An image of a deer mouse [19].

The dataset of deer mice (Peromyscus maniculatus), a small rodent native to North America which are about 8 to 10 cm long, not counting the length of the tail, is analysed and compared to the results in Otis. *et al.* (1978). This dataset has been widely used to discuss methods for estimating closed population size, for instance Huggins (1991) and Amstrup *et al.* (2005). This dataset is obtained using the R function `data(deermouse)` in the `secr` package [13]. V. H. Reid collected the data in Rio Blanco County, Colorado, in August 1975. Deer mice were livetrapped in a 'drainage bottom of sagebrush, gambel oak, and serviceberry with pinyon pine and juniper on the uplands' [1] for $T = 6$ consecutive nights. Traps were arranged in a 9×11 grid and spaced 50 feet (15.2 m) apart. A small mammal trap was placed at each grid point, thus 99 traps were used in total. Traps were checked on mornings and afternoons, but in this paper only the morning captures are considered. Listed below in Table 26 is a summary of the capture-recapture data, and full data are available in Appendix B. For the 38 deer mice captured, the age group, sex and weight of each of them are also given. The 38 marked individuals were captured 120 times in total during the study. This allows for the fitting of model M_h, with the logistic probabilities given by (20) mentioned in section 4. Sex and age are included as factors (with semi-adults being recorded as adults) and weight is included as a continuous covariate.

Similar to section 9, population size N of deer mice has been estimated using models M_0, M_t, M_b and M_h. Shown in Table 27 are the results of the maximum-likelihood-estimation. Models M_0 and M_t failed to give an estimate of the population size as they did not converge. Therefore, the only information that one has when using these models is that the population size is at least 38. Models M_b and M_h give an

j	No. Captures (C_j)	No. Recaptures (R_j)	Total tagged individuals (M_j)
1	15	0	0
2	20	12	23
3	16	10	29
4	19	16	32
5	25	22	35
6	25	22	38
	—	—	—
Total	120	82	38

Table 26: Capture-recapture data for deer mice (*Peromyscus maniculatus*)

estimated population size of 40.83 and 40.52, respectively. Note that the results for the 4 models agree with the results from the `secr` package in R. Based on the AIC, it is concluded that model M_h offers a best fit to the data. In Huggin (1991), it is also found that M_h also has the smallest AIC among the four simplest models [16]. On the other hand, Otis *et al.* (1978) analysis suggested that M_b is the best model based on their model selection. This is consistent to the findings present in this paper, given that M_b has a similar AIC value to M_h. Confidence intervals using the Hessian for M_h and M_b are very similar, so one may expect that the true value of the population size indeed lies within at least one of the two intervals. Since M_0 failed to provide an estimate for N, the bootstrap confidence interval was not computed. Bootstrap was only used for M_h model and it gives a much higher lower bound than for the two confidence intervals using the Hessian. Therefore, it is safe to assume that the true population size is smaller than the upper bound of the bootstrap confidence interval.

	M_0	M_t	M_b	M_h
$\hat{N}$	–	–	40.83	40.52
Convergence	1	1	0	0
Log-likelihood at MLE	–	–	−48.99	−46.38
AIC	–	–	103.98	100.77

Table 27: Value of $\hat{N}$ and AIC under all four models using deer mice data

	M_0	M_t	M_b	M_h
$\widehat{ese}(\hat{N})$	–	–	3.09	2.93
95% Hessian CI	–	–	[34.78 46.88]	[36.96 44.07]
95% bootstrap CI	–	–	–	[38.00 61.81]

Table 28: Value of estimated standard error, 95% Hessian confidence interval and 95% bootstrap confidence interval of $\hat{N}$ respectively, under all four models using deer mice data

10.2 Cottontail Rabbits

Figure 5: An image of a cottontail rabbit [21].

Another classic example is the estimation of cottontail rabbit (*Sylvilagus floridanus*) abundance conducted by Edwards and Eberhardt in 1967. The data, shown in Table 29, was obtained by livetrapping a confined population of known size on the Olentangy Wildlife Experiment Station, Delaware County, Ohio in 1961. In the study, N=135 wild cottontail rabbits were captured and placed in a 40-acre rabbit-proof enclosure. After allowing 4 days for the rabbits to adjust to their new surroundings, livetrapping was conducted for T=18 consecutive nights. As seen in

j	No. Captures (C_j)	No. Recaptures (R_j)	Total tagged individuals (M_j)
1	9	0	0
2	8	2	9
3	9	6	15
4	14	3	18
5	8	4	29
6	5	4	33
7	16	8	34
8	7	4	42
9	9	3	45
10	3	2	51
11	8	7	52
12	14	5	53
13	2	1	62
14	5	0	63
15	11	5	68
16	0	0	74
17	5	5	74
18	9	7	76
	—	—	—
Total	142	66	76

Table 29: Capture-recapture data for cottontail rabbits in a 40-acre pen, Olentangy Wildlife Experiment Stations, Delaware County, Ohio; October 24 – November 10, 1961.

Table 29, 76 out of 135 rabbits, i.e. 56% of the population, have been captured. Of these captured rabbits, 43 were captured once, 16 twice, 8 three times, 6 four times, 0 five times, 2 six times and 1 seven times respectively, resulting in a total number of 142 captures. The number of uncaptured rabbits is $135 - 76 = 59$, which is considerably high. Note that no mortality was detected during the study. The provision of the true population size N in this real data set is especially insightful for analysis and comparison. According to Otis (1978) the estimates were reliable due to high heterogeneity and low capture rates.

Bearing this in mind, the population size is estimated under model classes M_0, M_t, M_b and M_h and the results are displayed in Table 30. Note that for model M_h, information regarding sex and weight of the rabbits are not given in the data, hence assumptions are made and the population is split into two groups of equal numbers. The population size is underestimated for models M_0, M_t and M_b, whereas overestimated for model M_h, where the estimates range from 92 to 168. These results are consistent with the findings by Young and Young (1998) [34], who found that models M_t and M_h are preferred over model M_0. The model selection based on AIC indicates that model M_h is the preferred model, which agrees with the findings by Gosky and Ghosh (2009) who used the software MARK for their analysis. The true population size $N = 135$ does not lie within the 95% confidence interval (using the Hessian) for any of the four models. However, the true population size lies within [131.05, 204.95] which is the 99% confidence interval for model M_h. The confidence interval using the bootstrap method for model M_h contains $N = 135$ despite a very high upper bound of 1016.93. This supports the best model selection chosen using the lowest AIC criteria.

	M_0	M_t	M_b	M_h
$\hat{N}$	96.79	97.52	92.55	168.00
Convergence	0	0	0	0
Log-likelihood at MLE	−185.37	−165.33	−185.27	−130.70
AIC	374.74	368.67	376.54	269.41

Table 30: Value of $\hat{N}$ and AIC under all four models using cottontail rabbit data.

	M_0	M_t	M_b	M_h
$\widehat{ese}(\hat{N})$	7.02	7.57	10.00	14.32
95% Hessian CI	[83.03 110.55]	[82.68 112.36]	[72.95 112.15]	[139.93 196.07]
95% bootstrap CI	[85.36 120.47]	–	–	[100.73 1016.93]

Table 31: Value of estimated standard error, 95% Hessian confidence interval and 95% bootstrap confidence interval of $\hat{N}$ respectively, under all four models using cottontail data

Conclusions

This paper provided a detailed discussion of the capture-recapture method used to estimate unknown closed population sizes. Firstly, the capture-recapture method was introduced alongside the key assumptions of this method. A brief description of the other main methods used to estimate abundance were then discussed, as well as, an insight into the areas in which they can be applied.

One of the main aims of this paper was to establish a clear explanation of the maximum-likelihood functions used for the various models considered in the capture-recapture method. Thus we showed how the logistic models can be used in the general likelihood equation in order to account for the various ecological assumptions. We considered a selection of criterion used to assess the suitability of models such as AIC and deviance.

To test the efficiency of the models, simulations of data were generated using R. First, data were simulated under assumptions of the four basic models M_0, M_t, M_b, and M_h used as representative data. Analysis on the representative data was conducted using the likelihood models and the results were compared with the specified parameters used to simulate the data. AIC and confidence intervals using the Hessian and bootstrap were used as the criterion to choose the best model. However, for small data sets, AIC was not necessarily a good indication of best fitting model. This is because for small data sets, the log-likelihood values are generally small, which leads to the number of parameters having big impacts on the AIC values. Furthermore, we found that the model used to create the data, was chosen to be the best model every time.

To have more comprehensive understandings on model behaviour, parameters were altered and 50 simulations were performed to analyse the sensitivity of the models to changes in parameter values. Results from the multiple simulations showed that model M_0 is robust against changes, when used to estimate data simulated under the corresponding model. However, model M_h did not perform as well, in particular when capture probabilities were extreme.

From simulations, we have found that the likelihood models were capable of providing sensible estimates of abundance and capture probabilities. Thus, the models were applied to real data where the results were compared with previous research papers. The practicability of the capture-recapture method was demonstrated by the analysis of real datasets. Results were generally consistent with previous findings.

However, limitations were found regarding the accuracy in the estimates for cases when the probabilities were too extreme or when the sample size was too small. Moreover, the optimization function, `optim`, used in R was very sensitive to the starting values especially for model M_t. It is therefore advised to try several initial starting values for the parameters.

In further research on this subject, it would be of interest to implement more complex models in R such as M_{bt}, M_{bh} and M_{th} to examine whether they provide a better fit to real data sets, since these models are more realistic. Other optimization functions could be used in R, such as, `nlm`. It would also be beneficial to perform analysis on the same data using specialised software in estimating abundance such as MARK in order to have a comparison point for the results obtained. As an alternative to the general likelihood function, one can use the conditional likelihood function described in section 5. In addition to classical methods, Bayesian inference should be considered when there exists prior knowledge about the population of interest.

Bibliography

[1] Adke, S., White, G., Anderson, D., Burnham, K. and Otis, D. (1984). Capture-Recapture and Removal Methods for Sampling Closed Populations. *Journal of the American Statistical Association*, 79(388), p.952.

[2] Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6), pp.716-723.

[3] Anderson, D., Burnham, K. and White, G. (1994). *AIC Model Selection in Overdispersed Capture-Recapture Data.* Ecology, 75(6), p.1780.

[4] Bishop, Y., Fienberg, S. and Holland, P. (1975). *Discrete multivariate analysis.* Cambridge, Mass.: MIT Press.

[5] Bochkina, N. (2015). *Theory of Statistical Inference / Statistical theory.* [online] University of Edinburgh, School of Mathematics. Available at: `https://www.learn.ed.ac.uk/bbcswebdav/pid-1572228-dt-content-rid-2854323_1/courses/MATH100282015-6SV1SEM1/Notes_TSI_StT_2015.pdf` [Accessed 17 Mar. 2016].

[6] Borchers, D., Buckland, S. and Zucchini, W. (2002). *Estimating animal abundance.* London: Springer.

[7] Bohning, D., Dietz, E., Kuhnert, R. and Schon, D. (2016). *Mixture models for capture-recapture count data* [online] Available at: `http://www.personal.soton.ac.uk/dab1f10/sma.pdf` [Accessed 15 Mar. 2016].

[8] Buckland, S.T., Anderson, D.R., Burnham, K.P. and Laake, J.L. (1993). *Distance Sampling: Estimating Abundance of Biological Populations.* Chapman and Hall, London. 446pp.

[9] Chandler, R. (2015). *Modeling variation in abundance using capture-recapture data.* [online] Available at: `https://cran.r-project.org/web/packages/unmarked/vignettes/cap-recap.pdf` [Accessed 7 Mar. 2016].

[10] Chao, A. (1987). Estimating the Population Size for Capture-Recapture Data with Unequal Catchability. *Biometrics*, 43(4), p.783.

[11] Chao, A. (2001). An overview of closed capture-recapture models. *Journal of Agricultural, Biological, and Environmental Statistics*, 6(2), pp.158-175.

[12] Chao, A. (2015). Capture-Recapture for Human Populations. *Wiley StatsRef: Statistics Reference Online*. 1–16.

[13] Efford, M. (2016). Spatially Explicit Capture-Recapture. [online] *Package 'secr'*. Available at: `https://cran.r-project.org/web/packages/secr/secr.pdf` [Accessed 7 Mar. 2016].

[14] Gosky, R. and Ghosh, S. (2009). A Comparative Study of Bayesian Model Selection Criteria for Capture-Recapture Models for Closed Populations. [online] *Journal of Modern Applied Statistical Methods*. Available at: `http://digitalcommons.wayne.edu/cgi/viewcontent.cgi?article=1482&context=jmasm` [Accessed 16 Mar. 2016].

[15] Herzog, T. (n.d.). *Applications of Capture-Recapture Methods*. [online] Available at: `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.581.292&rep=rep1&type=pdf` [Accessed 17 Mar. 2016].

[16] Huggins, R. (1991). Some Practical Aspects of a Conditional Likelihood Approach to Capture Experiments. *Biometrics*, 47(2), p.725.

[17] King, R., Morgan, B., Gimenez, O. and Brooks, S. (2010). *Bayesian analysis for population ecology*. Boca Raton: Chapman & Hall/CRC.

[18] Laake, J. (2015). *R Code for Mark Analysis*. [online] R Code for Mark Analysis. Available at: `https://cran.r-project.org/web/packages/mra/mra.pdf` [Accessed 7 Mar. 2016].

[19] Lasley, G. (2007). *White-footed Deermouse (Peromyscus leucopus)*. [online] Greglasley.com. Available at: `http://greglasley.com/content/Mammals/White-footedDeermouse.php` [Accessed 16 Mar. 2016].

[20] Lebreton, J., Burnham, K., Clobert, J. and Anderson, D. (1992). *Modeling survival and testing biological hypotheses using marked animals*.

[21] Majewska, A., Altizer, S. and Davis, A. (2013). *Wormsloe Butterfly Gardens*. [online] Available at: `http://crew.uga.edu/butterflies.html` [Accessed 12 Mar. 2016].

[22] McCrea, R. and Morgan, B. (2014). *Analysis of capture-recapture Data*.

[23] Meyler, S., Salmona, J., Ibouroi, M., Besolo, A., Rasolondraibe, E., Radespiel, U., Rabarivola, C. and Chikhi, L. (2012). *Density estimates of two endangered nocturnal lemur species from northern Madagascar*.

[24] Otis, D., Burnham, K. P., White, G. C. and Anderson, D. R. (1978). *Statistical inference from capture data on closed animal populations*. [Washington]: Wildlife Society.

[25] Pcwallart.com. (n.d.). *Rabbit - image #1*. [online] Available at: `http://pcwallart.com/image-post/116-rabbit-wallpaper-4.jpg.html` [Accessed 16 Mar. 2016].

[26] Pledger, S. (2000). Unified maximum likelihood estimates for Closed Capture-Recapture Models Using Mixtures. *Biometrics*, 56(2), pp.434-442.

[27] Radford University. (n.d.). *A Method of Population Estimation: Mark & Recapture*. [online] Available at: `http://www.radford.edu/~jkell/mark_rec103.pdf` [Accessed 12 Mar. 2016].

[28] Rexstad, E. and Burnham, K. (1992). *Users Guide for Interactive Program CAPTURE*. [online] Available at: `http://www.mbr-pwrc.usgs.gov/software/doc/capturemanual.pdf` [Accessed 14 Mar. 2016].

[29] Seber, G. and Schwarz, C. (1999). Estimating Animal Abundance: Review III. *Statistical Science*, 14(4), pp.427-456.

[30] Sekar, C. and Deming, W. (1971). *On a method of estimating birth and death rates and the extent of registration*. Chapel Hill, N.C.: International Program of Laboratories for Population Statistics, Dept. of Biostatistics, School of Public Health, University of North Carolina.

[31] Thomas, L., Buckland, S., Burnham, K., Anderson, D., Laake, J., Borchers, D. and Strindberg, S. (2002). Distance sampling. [online] *Encyclopedia of Environmetrics*. Available at: `http://distancesampling.org/downloads/dist_encyc_env.pdf` [Accessed 15 Mar. 2016].

[32] Wallendorf, M. (2016). *Finite mixture models of heterogeneous capture*. [online] Lib.dr.iastate.edu. Available at: `http://lib.dr.iastate.edu/cgi/viewcontent.cgi?article=1458&context=rtd` [Accessed 15 Mar. 2016].

[33] White, G. and Leffler, B. (1982). *Capture-recapture and removal methods for sampling closed populations*. Los Alamos, N.M.: Los Alamos National Laboratory, pp.75-76

[34] Young, L. and Young, J. (1998). *Statistical ecology*. Boston: Kluwer Academic Publishers.

Appendix A

R Code

The codes for simulation and estimation were written using R 3.0.1 (R Core Team, 2013).

Simulation

The following R function simulates capture-recapture data under model assumptions of M_0, M_t, M_b and M_h respectively.

```

##Simulations
#Simulation of data sets for specified parameter values is done using the MRA package which utilises the function F.cjs.simulate to carry out the simulation.
#The function F.cjs.simulate has the following input parameters:

#F.cjs.simulate(super.p, super.s, N1, R):
#super.p: array of capture probabilities where the number of columns indicates the number of capture occasions.
#super.s: array of death probabilities with [(the number of capture occasions) - 1] columns.  It is 1 less than the number of capture occasions since the survival probabilities apply between each capture occasion. (note that since this paper assumes closed populations, all elements will be 1.)
#N1: true population size.
#R: the number of replications of the simulation, where R independent capture histories are produced.


#Initialisation
library(mra)    #MRA package used to create simulated data
co <- 5       #Number of capture occasions
N <- 30       #Total number of individuals in population
-------------------------------------------------------------------------
#M0
prob.m0 <- 0.3    #Capture probability
```

```
M0 <- F.cjs.simulate(super.p =rep(prob.m0, co), super.s = rep(1,co-1), N1= N, R = 1)
        #Simulate Data and Assign to M0
M0          #Print out M0 data on Console

------------------------------------------------------------------------

#Mt
prob.mt <- runif(co)  #Creates a random capture probability for each capture occasion
Mt <- F.cjs.simulate(super.p = prob.mt, super.s = rep(1,co-1), N1= N, R = 1)
        #Simulate Data and Assign to Mt
Mt        #Print out Mt data on Console

------------------------------------------------------------------------

#Mb
  #Mb
  #Generate probability matrix of behaviour before capture
  prob.Mbc <- 0.3     # capture probability
Mb.c <- F.cjs.simulate(super.p =rep(prob.Mbc, co), super.s = rep(1,co-1), N1= N, R = 1)
#Simulate Data and Assign to Mb.c
Mb.c        #Print out Mb.c data on Console
Mb.c <- read.Table('C:/Users/s1305557/Downloads/mbc.txt', header=TRUE)
#Import Data
#Detect the First captures and replace rest of matrix with 2
i.dim <- dim(Mb.c)[1]   #Number of Seen individuals
j.dim <- dim(Mb.c)[2] #Number of Capture Occasions
for(i in 1:i.dim){
  for (j in 2:j.dim) if (Mb.c[i,j-1]>0.1  ) for(k in 0:j.dim) Mb.c[i,j+k]=2
}       #After the first captures, replace elements with 2
Mb.c <- Mb.c[,1:j.dim]  #Choose data according to original number of capture occasions
Mb.c        #Display modified Mb.c Matrix
prob.Mbr <- 0.4     # recapture probability
Mb.r <- F.cjs.simulate(super.p =rep(prob.Mbr, co), super.s = rep(1,co-1), N1= N, R = 1)
#Simulate Data and Assign to Mb.r
Mb.r <- Mb.r[[1]]$hists       #Print out Mb.r data on Console
idim.Mbr <- dim(Mb.r)[1] #number of individuals recaptured
zero.mat <- matrix(0, nrow=N-idim.Mbr, ncol=co) #create a matrix of zeros the size of (the number of individuals not recaptured + not seen)
Mb.r <- rbind(Mb.r, zero.mat) #makes the individuals not seen/not recaptured have a capture history with zero for each capture occasion
for(i in 1:i.dim){ for (j in 2:j.dim) if (Mb.c[i,j]==2) Mb.c[i,j]=Mb.r[i,j]}
#Replace behavior after the first capture
Mb <- Mb.c      #Assign to Mb matrix
Mb        #Print out Mb on Console

------------------------------------------------------------------------
```

```
#Mh
no.of.groups <- 2      #Set number of groups
prob.mh <- runif(no.of.groups) #Creates a random capture probability for each group
capture <- NULL      #Initialise a probability matrix of the groups
           #Creates a matrix of the probabilities for each group
for(i in 1:length(prob.mh)-1){
  capture1 <- rbind(capture, rep(prob.mh[i+1],co))
  capture <- capture1}     #Assign values to the empty matrix
capture        #Present probabilities
Mh <- F.cjs.simulate(super.p = capture ,super.s = rep(1,co-1), N1= N, R = 1)
           #Simulate Data and Assign to Mh
Mh           #Print out Mh data on Console
```

MLE

The following R function uses the function `optim` to optimise likelihood functions and estimate the population size $\hat{N}$.

```

##MLE

##Analyse the simulated data

#The function optim is used to maximise the likelihood estimators.
#By default, optim minimises the function which is to be optimized.
#Let x be the vector of parameters which are to be optimized over. x[1] is N_hat in all the models.

--------------------------------------------------------------------------

  #M0
  seen <- sum(Capture1.df)    #Total number of captures; sum(x_ij)
  unseen <- i.dim*j.dim - seen  #Total number of times that seen animals are not captured

  f <- function(x) {      #Starts the function to be optimized over
    p <- 1/(1+exp(-x[2]))   #Use the logistic model to estimate capture probability p, where alpha = x[2]
    if (x[1] <= i.dim+0.000000001) { loglik <- -10000000000} #If N_hat is smaller than the number of individuals seen, then give a nonsense loglikelihood value.
    else {            #Otherwise,
      loglik <-         #let loglik be the MLE equation
        lgamma(x[1]+1) - lgamma(x[1]-i.dim+1) + seen*log(p) + unseen*log(1-p) + (x[1]-i.dim)*j.dim*log(1-p)
    }
    loglik <- -loglik       #Multiply by -1 to maximize loglik instead of minimize
```

```r
}

optim.m0 <- optim(c(N,0), f, hessian = T) #optimise the parameter vector x over the loglikelihood equation, with the specified initial search values where the Hessian matrix is to be returned.
----------------------------------------------------------------------------------------------------
#Mt
f <- function(x) {      #Starts the function to be optimized over
  p <- NULL
  for (j in 1:j.dim){     #For each capture occasion j,
    p[j] <- 1/(1+exp(-x[1+j]))  #Use the logistic model to estimate probability p for capture occasion j, where x[1+j] is beta_j.
  }
  if (x[1] <= i.dim+0.000000001) { loglik1 <- -10000000000} #If N_hat is smaller than the number of individuals seen, then give a nonsense loglikelihood value.
  else {                            #Otherwise,
  loglik1 <- lgamma(x[1]+1) - lgamma(x[1]-i.dim+1)      #Factorial terms
  #Sum over the number of individuals i and the number of capture occasions j.
  for (j in 1:j.dim){
    for (i in 1:i.dim){
      loglik1 <- loglik1 + Capture1.df[i,j]*log(p[j]) - Capture1.df[i,j]*log(1-p[j])
    }
    loglik1 <- loglik1 + x[1]*log(1-p[j])
  }
  }
  loglik1 <- -loglik1             #Multiply by -1 to maximize loglik instead of minimize
}
optim.mt <- optim(c(N,0,0,0,0,0), f, hessian = T) #Optimise the parameter vector x over the loglikelihood equation, with the specified initial search values where the Hessian matrix is to be returned.
----------------------------------------------------------------------------------------------------
#Mb
indicator <- matrix(data = NA, nrow = i.dim, ncol = j.dim)      #Initialize an indicator matrix which will has 1's for capture occasions after first capture and 0 o.w for all individuals
position <- NULL                        #Initialize a vector which tells you what capture occasion individual i is first seen
for(i in 1:i.dim){
  position[i]<-   match(1,Capture1.df[i,])            #Finds what capture occasion individual i is first seen
}
for(i in 1:i.dim){
  for(j in 1:j.dim){
    if(j <= position[i]) indicator[i,j] = 0 else indicator[i,j] = 1 #Updates the indicator matrix
  }
}
f <- function(x) {                    #Starts the function to be
```

```
 optimized over
  p <- matrix(data = NA, nrow = i.dim, ncol = j.dim)

  for(i in 1:i.dim){
    for (j in 1:j.dim){
      p[i,j] <- 1/(1+exp(-x[2]-x[3]*indicator[i,j]))        #Use the
 logistic model to estimate probability p for individual i & capture
  occasion j, where x[2] is alpha and x[3] is gamma.
    }
  }
  if (x[1] <= i.dim+0.000000001) { loglik1 <- -10000000000}       #If
  N_hat is smaller than the number of individuals seen, then give a
 nonsense loglikelihood value.
  else {                        #otherwise,
    loglik1 <- lgamma(x[1]+1) - lgamma(x[1]-i.dim+1) + ((x[1]-i.dim)*
 j.dim*log(1-(1/(1+exp(-x[2])))))
    for (i in 1:i.dim){
      for (j in 1:j.dim){
        loglik1 <-loglik1 + Capture1.df[i,j]*log(p[i,j]) + (1-
 Capture1.df[i,j])*log(1-p[i,j])
      }
    }
  }
  loglik1 <- -loglik1             #Multiply by -1 to maximize loglik
 instead of minimize
}
optim.mb <- optim(c(N, 0,0), f, hessian = T)  #Optimise the parameter
  vector x over the loglikelihood equation, with the specified
 initial search values where the Hessian matrix is to be returned.
------------------------------------------------------------------------

#Mh
f <- function(x) { #Starts the function to be optimized over
  p <- NULL         #Initialize p
  for (a in 1:no.groups){
    p[a] <- 1/(1+exp(-x[1+a]))  #Use the logistic model to estimate
 probability p for each group
  }
  w.old <- NULL       #Initialize non standardized weights
  w.old[1] = 1        #Set weight of first group initially to 1 since
  it is not a free parameter
  for (i in 2:no.groups)
  {
    w.old[i] <- 1/(1+exp(-x[no.groups+i])) #For the remaining groups
 use logistic model to estimate weight for each group
  }
#Add up all the non standardized weights
  w.sum <- w.old[1]
  for (i in 2:no.groups){
    w.sum = w.sum + w.old[i]
  }
#Standardize the weights
  w <- NULL
  for (i in 1:no.groups){
    w[i] = w.old[i]/w.sum
  }
```

```
#Find how many individuals were captured j times for j = 1,...,T
  A <- NULL
  for(i in 1:i.dim){
    A[i] <- sum(Capture1.df[i,]) #Adds up number of captures for each individual
  }
d1 <- 1:j.dim
d2 <- c(rep(0,times = j.dim))
C <- cbind(d1,d2)        #Creates matrix to be updated
  for(i in 1:j.dim){
C[i,2] <- sum(A == i)      #Updates the matrix and shows how many individuals were caught j times
  }
#Initialisations
  g<-0
  k<-0
  l<-0
  loglik1 <- 0

  if (x[1] <= i.dim+0.0000000000001) { loglik1 <- -10000000000} #If N_hat is smaller than the number of individuals seen, then give a nonsense loglikelihood value.
  else {                    #Otherwise,
#Log-likelihood for captured
  for(j in 1:j.dim){
    for(a in 1:no.groups){
      k <- k + w[a]*(p[a]^j)*((1-p[a])^(j.dim-j))
    }
    k <- k^C[j,2]
    g <- g+log(k)
  }
#Log-likelihood for not captured
    for(a in 1:no.groups){
      l <- l + w[a]*(1-p[a])^j.dim
    }
    l <- (x[1]-i.dim)*log(l)
#Sum of coefficients and log-likelihoods
    loglik1 <- lgamma(x[1]+1) - lgamma(x[1]-i.dim+1) + g + l
  }
  loglik1 <- -loglik1              #Multiply by -1 to maximize loglik instead of minimize
}
optim.mh <- optim(c(i.dim,0,0,0), f, hessian = T) #Optimise the parameter vector x over the loglikelihood equation, with the specified initial search values where the Hessian matrix is to be returned.
---------------------------------------------------------------------
```

Multiple Simulations

The following R function simulates data under model M_h 50 times. The MLE, AIC and confidence interval of each dataset were computed using models M_0, M_t, M_b and M_h. The code also records the number of times each model had the lowest AIC, as well as the number of times N was contained in the confidence intervals found. Note that this was also performed on simulated data under model M_0, the only difference is the simulation, refer to Simulation.

```
##Simulations
#Initialisation
library(mra)    #MRA package used to create simulated data
co <- 5       #Number of capture occasions
N <- 500      #Total number of individuals in population
it <- 50      #Number of iterations to carry out

  results <- matrix(0, nrow=it, ncol=17) #Initialize results matrix

  for(y in 1:it){

  no.groups <- 2        #Set number of groups
  #prob.mh <- runif(no.groups)  #Creates a random capture probability for each group
  prob.mh <- c(0.25,0.35)   #Or specify probabilities for each group
  capture <- NULL       #Initialise a probability matrix of the groups
  #Creates a matrix of the probabilities for each group
  for(i in 1:length(prob.mh)-1){
    capture1 <- rbind(capture, rep(prob.mh[i+1],co))
    capture <- capture1}    #Assign values to the empty matrix
  capture         #Present probabilities
  Mh <- F.cjs.simulate(super.p = capture ,super.s = rep(1,co-1), N1= N, R = 1)
  #Simulate Data and Assign to Mh
  Mh            #Print out Mh data on Console
  Capture1.df <- Mh[[1]]$hists  #Assign the capture history of simulated data to Capture1.df
    i.dim <- dim(Capture1.df)[1]#Number of individuals seen; D
  j.dim <- dim(Capture1.df)[2]  #Number of capture occasions; T
  results[y,17] <- i.dim        #Shows the number of individuals seen for each iteration

  #M0
  seen <- sum(Capture1.df)    #Total number of captures; sum(x_ij)
  unseen <- i.dim*j.dim - seen  #Total number of times that seen animals are not captured

  f <- function(x) {      #Starts the function to be optimized over
    p <- 1/(1+exp(-x[2]))   #Use the logistic model to estimate capture probability p, where alpha = x[2]
    if (x[1] <= i.dim+0.000000001) { loglik <- -10000000000} #If N_hat is smaller than the number of individuals seen, then give a nonsense loglikelihood value.
    else {            #otherwise,
      loglik <-         #let loglik be the MLE equation
      lgamma(x[1]+1) - lgamma(x[1]-i.dim+1) + seen*log(p) + unseen*log(1-p) + (x[1]-i.dim)*j.dim*log(1-p)
  }
  loglik <- -loglik       #Multiply by -1 to maximize loglik instead of minimize
}

optim.m0 <- optim(c(N,0), f, hessian = T) #optimise the parameter vector x over the loglikelihood equation, with the specified initial search values where the Hessian matrix is to be returned.

#Updates the results matrix
results[y,1] <- optim.m0$par[1]                          #N
results[y,5] <- -2*(-optim.m0$value) + 2*2                   #AIC
results[y,9] <- optim.m0$par[1] - 1.96*sqrt(solve(optim.m0$hessian[1,1]))   #Lower bound of CI
results[y,10] <- optim.m0$par[1] + 1.96*sqrt(solve(optim.m0$hessian[1,1]))  #Ipper bound of CI

#Mt
f <- function(x) {      #Starts the function to be optimized over
  p <- NULL
  for (j in 1:j.dim){     #For each capture occasion j,
    p[j] <- 1/(1+exp(-x[1+j]))  #Use the logistic model to estimate probability p for capture occasion j, where x[1+j] is beta_j.
  }
  if (x[1] <= i.dim+0.000000001) { loglik1 <- -10000000000} #If N_hat is smaller than the number of individuals seen, then give a nonsense loglikelihood value.
  else {                    #otherwise,
  loglik1 <- lgamma(x[1]+1) - lgamma(x[1]-i.dim+1)      #Factorial terms
  #Sum over the number of individuals i and the number of capture occasions j.
  for (j in 1:j.dim){
    for (i in 1:i.dim){
      loglik1 <- loglik1 + Capture1.df[i,j]*log(p[j]) - Capture1.df[i,j]*log(1-p[j])
    }
    loglik1 <- loglik1 + x[1]*log(1-p[j])
  }
  }
  loglik1 <- -loglik1             #Multiply by -1 to maximize loglik instead of minimize
}
optim.mt <- optim(c(N,0,0,0,0,0), f, hessian = T) #Optimise the parameter vector x over the loglikelihood equation, with the specified initial search values where the Hessian matrix is to be returned.

#Updates the results matrix
results[y,2] <- optim.mt$par[1]                          #N
results[y,6] <- -2*(-optim.mt$value) + 2*(j.dim+1)              #AIC
results[y,11] <- optim.mt$par[1] - 1.96*sqrt(solve(optim.mt$hessian[1,1])) #Lower bound of CI
results[y,12] <- optim.mt$par[1] + 1.96*sqrt(solve(optim.mt$hessian
```

```r
[1,1])) #Upper bound of CI

#Mb
indicator <- matrix(data = NA, nrow = i.dim, ncol = j.dim)  #Initialize an indicator matrix which will has 1's for capture occasions after first capture and 0 o.w for all individuals
position <- NULL                                       #Initialize a vector which tells you what capture occasion individual i is first seen
for(i in 1:i.dim){
  position[i]<-   match(1,Capture1.df[i,])            #Finds what capture occasion individual i is first seen
}
for(i in 1:i.dim){
  for(j in 1:j.dim){
    if(j <= position[i]) indicator[i,j] = 0 else indicator[i,j] = 1 #Updates the indicator matrix
  }
}
f <- function(x) {                                     #Starts the function to be optimized over
  p <- matrix(data = NA, nrow = i.dim, ncol = j.dim)

  for(i in 1:i.dim){
    for (j in 1:j.dim){
      p[i,j] <- 1/(1+exp(-x[2]-x[3]*indicator[i,j]))          #Use the logistic model to estimate probability p for individual i & capture occasion j, where x[2] is alpha and x[3] is gamma.
    }
  }
  if (x[1] <= i.dim+0.000000001) { loglik1 <- -10000000000}         #If N_hat is smaller than the number of individuals seen, then give a nonsense loglikelihood value.
  else {                       #otherwise,
    loglik1 <- lgamma(x[1]+1) - lgamma(x[1]-i.dim+1) + ((x[1]-i.dim)*j.dim*log(1-(1/(1+exp(-x[2])))))
    for (i in 1:i.dim){
      for (j in 1:j.dim){
        loglik1 <-loglik1 + Capture1.df[i,j]*log(p[i,j]) + (1-Capture1.df[i,j])*log(1-p[i,j])
      }
    }
  }
  loglik1 <- -loglik1 #Multiply by -1 to maximize loglik instead of minimize
}
optim.mb <- optim(c(N, 0,0), f, hessian = T)  #Optimise the parameter vector x over the loglikelihood equation, with the specified initial search values where the Hessian matrix is to be returned.

#Updates the results matrix
results[y,3] <- optim.mb$par[1]       #N
results[y,7] <- -2*(-optim.mb$value) + 2*3  #AIC
results[y,13] <- optim.mb$par[1] - 1.96*sqrt(solve(optim.mb$hessian[1,1])) #Lower bound of CI
results[y,14] <- optim.mb$par[1] + 1.96*sqrt(solve(optim.mb$hessian[1,1])) #Upper bound of CI
```

```


#Mh
f <- function(x) { #Starts the function to be optimized over
  p <- NULL         #Initialize p
  for (a in 1:no.groups){
    p[a] <- 1/(1+exp(-x[1+a]))  #Use the logistic model to estimate probability p for each group
  }
  w.old <- NULL       #Initialize non standardized weights
  w.old[1] = 1        #Set weight of first group initially to 1 since it is not a free parameter
  for (i in 2:no.groups)
  {
    w.old[i] <- 1/(1+exp(-x[no.groups+i])) #For the remaining groups use logistic model to estimate weight for each group
  }
#Add up all the non standardized weights
  w.sum <- w.old[1]
  for (i in 2:no.groups){
    w.sum = w.sum + w.old[i]
  }
#Standardize the weights
  w <- NULL
  for (i in 1:no.groups){
    w[i] = w.old[i]/w.sum
  }
#Find how many individuals were captured j times for j = 1,...,T
  A <- NULL
  for(i in 1:i.dim){
    A[i] <- sum(Capture1.df[i,]) #Adds up number of captures for each individual
  }
d1 <- 1:j.dim
d2 <- c(rep(0,times = j.dim))
C <- cbind(d1,d2)        #Creates matrix to be updated
  for(i in 1:j.dim){
C[i,2] <- sum(A == i)      #Updates the matrix and shows how many individuals were caught j times
  }
#Initialisations
  g<-0
  k<-0
  l<-0
  loglik1 <- 0

  if (x[1] <= i.dim+0.0000000000001) { loglik1 <- -10000000000} #If N_hat is smaller than the number of individuals seen, then give a nonsense loglikelihood value.
  else {                      #Otherwise,
#Log-likelihood for captured
  for(j in 1:j.dim){
    for(a in 1:no.groups){
      k <- k + w[a]*(p[a]^j)*((1-p[a])^(j.dim-j))
    }
    k <- k^C[j,2]
    g <- g+log(k)
  }
#Log-likelihood for not captured
    for(a in 1:no.groups){
      l <- l + w[a]*(1-p[a])^j.dim
    }
    l <- (x[1]-i.dim)*log(l)
#Sum of coefficients and log-likelihoods
    loglik1 <- lgamma(x[1]+1) - lgamma(x[1]-i.dim+1) + g + l
  }
  loglik1 <- -loglik1
}
#Calculate MLEs
optim.mh <- optim(c(i.dim,0,0,0), f, hessian = T)
#Updates the results matrix
results[y,4] <- optim.mh$par[1]                       #N
results[y,8] <- -2*(-optim.mh$value) + 2*(2*no.groups)          #AIC
results[y,15] <- optim.mh$par[1] - 1.96*sqrt(solve(optim.mh$hessian[1,1]))  #Lower bound of CI
results[y,16] <- optim.mh$par[1] + 1.96*sqrt(solve(optim.mh$hessian[1,1]))  #Upper bound of CI
}                                         #Ends the iterations loop
###################################################
#Truncates confidence interval if the lower bound is less than the number of individuals seen
for(i in 1:it){
if(results[i,9] < results[i,17]){
  results[i,9] = results[i,17]}
if(results[i,11] < results[i,17]){
  results[i,11] = results[i,17]}
if(results[i,13] < results[i,17]){
  results[i,13] = results[i,17]}
if(results[i,15] < results[i,17]){
  results[i,15] = results[i,17]}
}
###################################################
#Finds the smallest AIC value for each iteration
minimum.aic <- NULL
for(i in 1:it){
  minimum.aic[i] <- min(results[i,5:8])
}
#Finds which model the smallest AIC value comes from (best model) (1=M0, 2=Mt, 3=Mb, 4=Mh)
model <- NULL
for(i in 1:it){
  for(j in 5:8){
    if(minimum.aic[i] == results[i,j]) model[i] = j - 4
  }
}
no.each.model <- Table(model) #shows how many times each model gives the best fit
no.each.model
###################################################
#Finds how many times N is contained in the confidence interval for each model
```

```
  CI <- matrix(data = 0, nrow = it, ncol = 4)
  for(i in 1:it){
  if (N > results[i,9] & N < results[i,10]) {CI[i,1] = 1}
  if (N > results[i,11] & N < results[i,12]) {CI[i,2] = 1}
  if (N > results[i,13] & N < results[i,14]) {CI[i,3] = 1}
  if (N > results[i,15] & N < results[i,16]) {CI[i,4] = 1}
  }
  no.in.CI.m0 <- sum(CI[,1])
  no.in.CI.mt <- sum(CI[,2])
  no.in.CI.mb <- sum(CI[,3])
  no.in.CI.mh <- sum(CI[,4])
  no.in.CI.m0
  no.in.CI.mt
  no.in.CI.mb
  no.in.CI.mh
  #####################################################
```

Bootstrap

The R code below performs non-parametric bootstrap to construct bootstraps confidence intervals for models M_0, M_t, M_b and M_h respectively.

```
#Bootstrap

-------------------------------------------------------------------------

#Boostrap for model M0

#Attach the data matrix
Capture1.df <- read.Table('C:/Users/user/Downloads/m0.txt', header=TRUE)
attach(Capture1.df)
Capture1.df

#Compute the total number of captures (add the 1s across the matrix) and the number of 0s
n <- Capture1.df
n
seen <- sum(n)
i.dim <- dim(Capture1.df)[1] # number of individuals seen; D
j.dim <- dim(Capture1.df)[2] # number of capture occasions; T
unseen <- i.dim*j.dim - seen

#Sum each row of the matrix and store it in a vector to be used in sample() function

kk <- NULL
for (i in 1:i.dim)
{
kk[i] = sum(n[i,])
}
```

```
kk

# Usual optim function
f <- function(x) {
  p <- 1/(1+exp(-x[2]))
  if (x[1] <= i.dim+0.000000001) { loglik <- -10000000000}
  else {
    loglik <- lgamma(x[1]+1) - lgamma(x[1]-i.dim+1) + seen*log(p) + unseen*log(1-p)
              + (x[1]-i.dim)*j.dim*log(1-p)

    loglik <- -loglik
  }
}
res <- optim(c(19,0.4), f, hessian = T)


# bootstrap function

#Initializes with arguments number of samples and data matrix
bootpo <- function(bootsamples,nn=data) {

  # Set up vector to store the bootstrapped MLE values:

  bootpo <- array(0,bootsamples+1)

  for (i in 1:bootsamples) {

#Gives equal probability for any cell of vector kk to be chosen
prob <- rep(c(1/(i.dim)), times=i.dim)

#Takes 18 random sample from kk with equal probability for each cell
nn <- sample(kk, i.dim, replace = TRUE, prob) # takes 18 random sample from kk with equal probability for each cell

#Collects the number of ones and zeros to be used in the function to optimize
newseen <- sum(nn)
newunseen <- i.dim*j.dim-sum(nn)
newseen
newunseen

f <- function(x) {
  p <- 1/(1+exp(-x[2]))
  if (x[1] <= i.dim+0.000000001) { loglik <- -10000000000}
  else {
    loglik <- lgamma(x[1]+1) - lgamma(x[1]-i.dim+1) + newseen*log(p)
              + newunseen*log(1-p) + (x[1]-i.dim)*j.dim*log(1-p)
    }
    loglik <- -loglik
  }

res1 <- optim(c(19,0.4), f, hessian = T)

#Store values of N_hat in bootpo vector
   bootpo[i] <- res1$par[1]
}

  # Add in the MLE for N from the observed data
  bootpo[bootsamples+1] <- res$par[1]

  # Output the 95% CI as the lower and upper 2.5% quantiles of
  # MLE estimates from bootstrapped sample

  quantile(bootpo,c(0.025,0.975))

}
#Run function for 999 samples and n as data matrix
bootpo(999,n)

------------------------------------------------------------------------

# Bootstrap for model Mt

# This is essentially the same as for model M0, except that one cannot use the sum
# of all 1s in the matrix as there is a dependence on j (capture). Therefore, in this code,
# a new matrix is sampled from the original data, each time.

#Attach the data matrix
Capture1.df <- read.Table('C:/Users/user/Downloads/m0.txt', header=TRUE)
attach(Capture1.df)
Capture1.df

# Get the dimensions of the matrix
n <- Capture1.df
n
i.dim <- dim(Capture1.df)[1] # number of individuals seen; D
j.dim <- dim(Capture1.df)[2] # number of capture occasions; T


# Usual optim function
f <- function(x) {
  p <- NULL
  for (j in 1:j.dim){
    p[j] <- 1/(1+exp(-x[1+j]))
  }
  loglik1 <- lgamma(x[1]+1) - lgamma(x[1]-i.dim+1)

  for (j in 1:j.dim){
    for (i in 1:i.dim){
      loglik1 <- loglik1 + n[i,j]*log(p[j]) - n[i,j]*log(1-p[j])
    }

    loglik1 <- loglik1 + x[1]*log(1-p[j])
  }

  loglik1 <- -loglik1
}
res <- optim(c(25,0,0,0,0,0), f)
```

```


# bootpo function

bootpo <- function(bootsamples,nn=data) {

  # Set up vector to store the bootstrapped MLE values:

i.dim <- dim(nn)[1] # number of individuals seen; D
j.dim <- dim(nn)[2] # number of capture occasions; T

bootpo <- NULL   # Initialize bootpo vector
Sampledmatrix <- array(0, dim=c(i.dim,j.dim)) # Initializes sampled matrix
Sampledmatrix <- as.data.frame(Sampledmatrix) # Change matrix type to data frame type

  for (i in 1:bootsamples) {

# Create the sampled matrix. It consists of the same number of randomly picked row from the
# original matrix nn.

x <- 1:i.dim
xx <- sample(x, replace = TRUE)
for(j in 1:i.dim) Sampledmatrix[j,] <- nn[xx[j],]

# MLE function per usual

f <- function(x) {
  p <- NULL
  for (j in 1:j.dim){
    p[j] <- 1/(1+exp(-x[1+j]))
  }
  loglik1 <- lgamma(x[1]+1) - lgamma(x[1]-i.dim+1)  # factorial terms

  for (j in 1:j.dim){
    for (i in 1:i.dim){
      loglik1 <- loglik1 + Sampledmatrix[i,j]*log(p[j]) - Sampledmatrix[i,j]*log(1-p[j])
    }

    loglik1 <- loglik1 + x[1]*log(1-p[j])
  }

  loglik1 <- -loglik1
}
res1 <- optim(c(25,0,0,0,0,0), f)

# Add the estimated N in the bootpo vector
 bootpo[i] <- res1$par[1]
}

  # Add in the MLE for N from the observed data
  # (these are included in bootstrap replicates)
bootpo[bootsamples+1] <- res$par[1]
```

```

  # Output the 95% CI as the lower and upper 2.5% quantiles of
  # MLE estimates from bootstrapped sample

  quantile(bootpo,c(0.025,0.975))

}
bootpo(999,n)

-----------------------------------------------------------------------

# Bootstrap for model Mb

Capture1.df <- read.Table('C:/Users/user/Downloads/m0.txt', header=TRUE)
attach(Capture1.df)
Capture1.df

n <- Capture1.df
n
i.dim <- dim(Capture1.df)[1] # number of individuals seen; D
j.dim <- dim(Capture1.df)[2] # number of capture occasions; T


# Usual optim function
i.dim <- dim(Capture1.df)[1] # number of individuals seen; D
j.dim <- dim(Capture1.df)[2] # number of capture occasions; T
indicator <- matrix(data = NA, nrow = i.dim, ncol = j.dim)
position <- NULL
for(i in 1:i.dim){
  position[i]<-   match(1,Capture1.df[i,])
}
for(i in 1:i.dim){
  for(j in 1:j.dim){
    if(j <= position[i]) indicator[i,j] = 0 else indicator[i,j] = 1
  }
}
indicator

f <- function(x) {
  p <- matrix(data = NA, nrow = i.dim, ncol = j.dim)

  for(i in 1:i.dim){
    for (j in 1:j.dim){
      p[i,j] <- 1/(1+exp(-x[2]-x[3]*indicator[i,j]))
    }
  }
  if (x[1] <= i.dim+0.000000001) { loglik1 <- -10000000000}
  else {
    loglik1 <- lgamma(x[1]+1) - lgamma(x[1]-i.dim+1) + ((x[1]-i.dim)*j.dim*log(1-(1/(1+exp(-x[2])))))
    for (i in 1:i.dim){
      for (j in 1:j.dim){
        loglik1 <-loglik1 + Capture1.df[i,j]*log(p[i,j]) + (1-Capture1.df[i,j])*log(1-p[i,j])
      }
    }
  }
  loglik1 <- -loglik1
}
res <- optim(c(20,0.5,0), f)

# bootpo function

bootpo <- function(bootsamples,nn=data) {

  bootpo <- NULL

i.dim <- dim(nn)[1]
j.dim <- dim(nn)[2]
Sampledmatrix <- array(0, dim=c(i.dim,j.dim))
Sampledmatrix <- as.data.frame(Sampledmatrix)

  for (i in 1:bootsamples) {

x <- 1:i.dim
xx <- sample(x, replace = TRUE)
for(j in 1:i.dim) Sampledmatrix[j,] <- nn[xx[j],]

indicator <- matrix(data = NA, nrow = i.dim, ncol = j.dim)
position <- NULL
for(l in 1:i.dim){
  position[l]<-   match(1,Sampledmatrix[l,])
}
for(l in 1:i.dim){
  for(j in 1:j.dim){
    if(j <= position[l]) indicator[l,j] = 0 else indicator[l,j] = 1
  }
}
indicator

f <- function(x) {
  p <- matrix(data = NA, nrow = i.dim, ncol = j.dim)

  for(i in 1:i.dim){
    for (j in 1:j.dim){
      p[i,j] <- 1/(1+exp(-x[2]-x[3]*indicator[i,j]))
    }
  }
  if (x[1] <= i.dim+0.000000001) { loglik1 <- -10000000000}
  else {
    loglik1 <- lgamma(x[1]+1) - lgamma(x[1]-i.dim+1) + ((x[1]-i.dim)*j.dim*log(1-(1/(1+exp(-x[2])))))
    for (i in 1:i.dim){
      for (j in 1:j.dim){
        loglik1 <-loglik1 + Sampledmatrix[i,j]*log(p[i,j]) + (1-Sampledmatrix[i,j])*log(1-p[i,j])
      }
    }
  }
  loglik1 <- -loglik1
}
```

```
res1 <- optim(c(25,0.5,0), f)

   bootpo[i] <- res1$par[1]
}
  # Add in the MLE for N from the observed data
  # (these are included in bootstrap replicates)

  bootpo[bootsamples+1] <- res$par[1]

  # Output the 95% CI as the lower and upper 2.5% quantiles of
  # MLE estimates from bootstrapped sample

  quantile(bootpo,c(0.025,0.975))

}
bootpo(999,n)

-----------------------------------------------------------------------


# Bootstrap for model Mh

Capture1.df <- read.Table('C:/Users/user/Downloads/m0.txt', header=TRUE)
attach(Capture1.df)
Capture1.df

# Get the dimensions of the data matrix and set the number of groups
n <- Capture1.df

# Usual optim function
no.groups <-2

  i.dim <- dim(Capture1.df)[1]  #number of individuals seen; D
  j.dim <- dim(Capture1.df)[2]  #number of capture occasions; T
  seen <- sum(Capture1.df) # total number of seen
  unseen <- i.dim*j.dim - seen # total number of unseen
  f <- function(x) {

    p <- NULL
    for (a in 1:no.groups){
      p[a] <- 1/(1+exp(-x[1+a]))
    }

    w.old <- NULL
    w.old[1] = 1
    for (i in 2:no.groups)
    {
      w.old[i] <- 1/(1+exp(-x[no.groups+i]))
    }
    w.sum <- w.old[1]
    for (i in 2:no.groups){
      w.sum = w.sum + w.old[i]
    }
    w <- NULL
    for (i in 1:no.groups){
```

```r
      w[i] = w.old[i]/w.sum
    }

    A <- NULL
    for(i in 1:i.dim){
      A[i] <- sum(Capture1.df[i,])
    }
    A
d1 <- 1:j.dim
d2 <- c(rep(0,times = j.dim))
   C <- cbind(d1,d2)
    for(i in 1:j.dim){
C[i,2] <- sum(A == i)
   }


    g<- 0
    k<- 0
    l<-0
    loglik1 <- 0
    for(j in 1:j.dim){
      for(a in 1:no.groups){
        k <- k + w[a]*(p[a]^j)*((1-p[a])^(j.dim-j))
      }
      k <- k^C[j,2]
      g <- g+log(k)
    }

    if (x[1] <= i.dim+0.0000000000001) { loglik1 <- -10000000000} # if N_hat is smaller than the number of individuals seen, then give a nonsense loglikelihood value.
    else {        #otherwise,
      for(a in 1:no.groups){
        l <- l + w[a]*(1-p[a])^j.dim
      }
      l <- (x[1]-i.dim)*log(l)

      loglik1 <- lgamma(x[1]+1) - lgamma(x[1]-i.dim+1) + g + l
    }
    loglik1 <- -loglik1
  }
  res <- optim(c(20,0.2,0.3,0), f, hessian = T)
############################################ bootpo

bootpo <- function(bootsamples,nn=data) {

  # Set up vector to store the bootstrapped MLE values
  bootpo <- NULL

 i.dim <- dim(nn)[1]
 j.dim <- dim(nn)[2]
 Sampledmatrix <- array(0, dim=c(i.dim,j.dim))  # Set up a matrix filled of zero with same dimension as nn
 Sampledmatrix <- as.data.frame(Sampledmatrix)  # Change matrix type to data frame type

```

```
  for (i in 1:bootsamples) {

x <- 1:i.dim
xx <- sample(x, replace = TRUE)
for(j in 1:i.dim) Sampledmatrix[j,] <- nn[xx[j],]

no.groups <-2

  f <- function(x) {

    p <- NULL
    for (a in 1:no.groups){
      p[a] <- 1/(1+exp(-x[1+a]))
    }

    w.old <- NULL
    w.old[1] = 1
    for (i in 2:no.groups)
    {
      w.old[i] <- 1/(1+exp(-x[no.groups+i]))
    }
    w.sum <- w.old[1]
    for (i in 2:no.groups){
      w.sum = w.sum + w.old[i]
    }
    w <- NULL
    for (i in 1:no.groups){
      w[i] = w.old[i]/w.sum
    }




    A <- NULL
    for(i in 1:i.dim){
      A[i] <- sum(Sampledmatrix[i,])
    }

d1 <- 1:j.dim
d2 <- c(rep(0,times = j.dim))
   C <- cbind(d1,d2)
    for(i in 1:j.dim){
C[i,2] <- sum(A == i)
   }


    g<- 0
    k<- 0
    l<-0
    loglik1 <- 0
    for(j in 1:j.dim){
      for(a in 1:no.groups){
        k <- k + w[a]*(p[a]^j)*((1-p[a])^(j.dim-j))
      }
      k <- k^C[j,2]
      g <- g+log(k)
    }

    if (x[1] <= i.dim+0.0000000000001) { loglik1 <- -10000000000} # if N_hat is smaller than the number of individuals seen, then give a nonsense loglikelihood value.
    else {        #otherwise,
      for(a in 1:no.groups){
        l <- l + w[a]*(1-p[a])^j.dim
      }
      l <- (x[1]-i.dim)*log(l)

      loglik1 <- lgamma(x[1]+1) - lgamma(x[1]-i.dim+1) + g + l
    }
    loglik1 <- -loglik1
  }
  res1 <- optim(c(20,0.2,0.3,0), f, hessian = T)

# Add the estimated N to the bootpo vector

  bootpo[i] <- res1$par[1]
}
  # Add in the MLE for N from the observed data
  # (these are included in bootstrap replicates)

  bootpo[bootsamples+1] <- res$par[1]
  # Output the 95% CI as the lower and upper 2.5% quantiles of
  # MLE estimates from bootstrapped sample


  return(quantile(bootpo,c(0.025,0.975)))

}
# Call the function with chosen arguments
bootpo(999,n)
```

Appendix B

Reid’s Deer Mice Data

			Capture occasion (co_j)									Capture occasion (co_j)					
Sex	Age	Weight (g)	1	2	3	4	5	6	Sex	Age	Weight (g)	1	2	3	4	5	6
m	y	12	1	1	1	1	1	1	m	y	13	0	1	1	0	1	0
f	y	15	1	0	0	1	1	1	f	y	5	0	1	0	1	0	1
m	y	15	1	1	0	0	1	1	f	a	20	0	1	0	0	0	1
m	y	15	1	1	0	1	1	1	m	y	12	0	1	0	0	1	1
m	y	13	1	1	1	1	1	1	f	y	6	0	0	1	0	0	0
m	a	21	1	1	0	1	1	1	f	a	22	0	0	1	1	1	1
m	y	11	1	1	1	1	1	0	f	y	10	0	0	1	0	1	1
m	sa	15	1	1	1	0	0	1	f	y	14	0	0	1	1	1	1
m	y	14	1	1	1	1	1	1	f	a	19	0	0	1	0	0	0
m	y	13	1	1	0	1	1	1	f	a	20	0	0	0	1	0	0
f	a	22	1	1	1	0	1	1	m	sa	16	0	0	0	1	1	1
m	y	14	1	1	1	1	1	1	f	y	11	0	0	0	1	1	0
m	y	11	1	0	1	1	1	0	m	y	14	0	0	0	0	1	0
f	y	10	1	0	0	1	0	0	f	y	11	0	0	0	0	1	0
f	a	23	0	1	0	0	1	0	m	a	24	0	0	0	0	1	0
f	y	7	0	1	1	0	0	1	m	y	9	0	0	0	0	0	1
m	y	8	0	1	0	0	0	1	m	sa	16	0	0	0	0	0	1
m	a	19	0	1	0	1	0	1	f	a	19	0	0	0	0	0	1

Table B.1: Captures of *Peromyscus maniculatus* collected by *V. Reid at East Stuart Gulch, Colorado.* The columns represent the *sex* (m or f), the *ages* (y: young, sa: semi-adult, a: adult). the *weights in grams*, and the capture histories of 38 individuals over 6 trapping occasions (1: captured, 0: not captured).