CHAPTER 15

Closed population capture-recapture models

Paul Lukacs, *University of Montana*

A fair argument could be made that the marking of individuals in a wild population was originally motivated by the desire to estimate a fundamental parameter: abundance (i.e., population size). By comparing the relative proportions of marked and unmarked animals in successive samples, various estimators of animal abundance could be derived. In this chapter, we consider the theory and mechanics of estimation of abundance from *closed* population capture-recapture data, using program **MARK**.* Here, the population of interest is assumed to be closed geographically – no movement on or off the study area – and demographically – no births or deaths.

15.1. The basic idea

How many individuals are there in the sampled population? If the population is (or assumed to be) closed, then the number of individuals in the population being sampled is a constant over time. Meaning, the population size does not change at each sampling event. With a little thought, you quickly realize that the canonical estimate of population size is a function of (i) how many unique individuals are encountered over all sampling events, and (ii) what the probability is of encountering an individual at least once. For a single sampling event, we can express this more formally as

$$\hat{N} = \frac{n}{\hat{p}},$$

where the numerator (n) is the number of unique individuals encountered, and the denominator (p) is the probability that any individual will be encountered.

This expression makes good intuitive sense. For example, suppose that you capture 50 individuals ($n = 50$), and the encounter probability is $p = 0.5$, then clearly, since there is a 50:50 chance that you will miss an individual instead of encountering it, then

$$\hat{N} = \frac{n}{\hat{p}} = \frac{50}{0.5} = 100.$$

* Prior to **MARK**, program **CAPTURE** was a widely used application for closed population abundance estimation. All of the likelihood-based models from **CAPTURE** can be built in **MARK**, plus numerous models that have been developed since then. Further, there are some important differences between **MARK** and **CAPTURE**: (i) for likelihood-based models, **CAPTURE** returns the estimate from the integer, and not the floating point value that maximizes the likelihood; (ii) all of the heterogeneity models in **CAPTURE** (except M_{bh}) are not likelihood based, so will give quite different estimates than those from **MARK**.

15.1.1. The Lincoln-Petersen estimator – a quick review

The most general approach to estimating abundance, and p, in closed populations is based on what is known as the Lincoln-Petersen estimator (hereafter, the 'LP' estimator). The LP estimator is appropriate when there are just two sampling occasions, and the population is closed between the two occasions.

Imagine you go out on the first occasion, capture a sample of individuals, mark and release them back into the population. On the second occasion, you re-sample from (what you hope is) the same population. In this second sample, there will be two types of individuals: those that are unmarked (not previously captured) and those with marks (individuals captured and marked on the first occasion). The basic sampling structure is shown in Fig. (15.1).

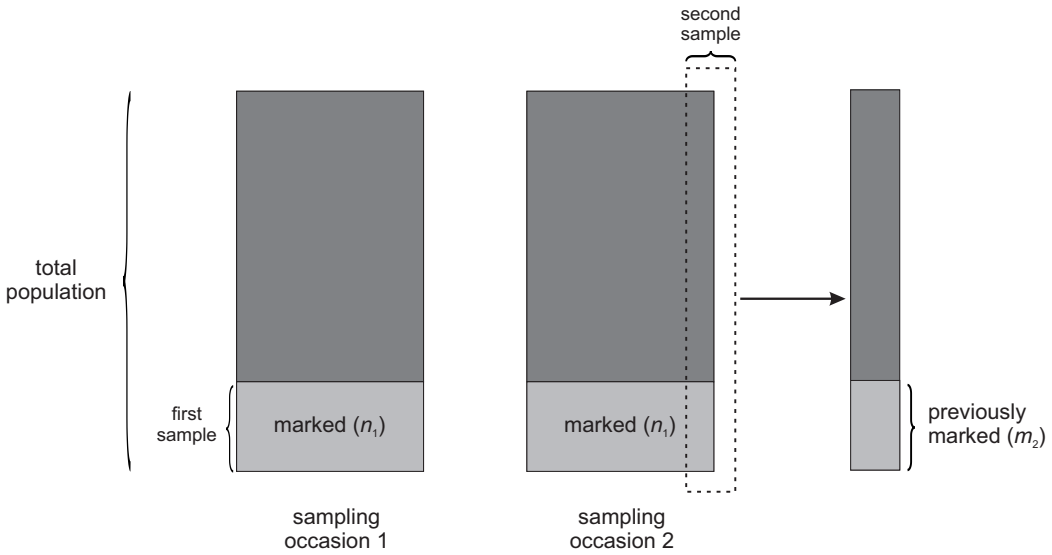


Figure 15.1: *Schematic representation of the LP sampling scheme. The entire left-most vertical bar (the sum of light- and dark-grey areas) represents the total population, N. The light-grey represents the proportion of the total population that is sampled on the first sampling occasion. The number encountered, and marked, during this first sample, is n_1. The middle bar is the same population at the time of the second sample, with the same total abundance, N, which we assume is constant between sampling occasions. During the second sample, indicated as the proportion of the total population bounded by the dashed-line box, some of the n_2 total sampled individuals are newly encountered – dark-grey – while some (m_2, the light-grey portion) were previously encountered. Adapted from Powell & Gale (2015).*

We develop the LP estimator by noting that the proportion of marked animals in the population after the first sample is simply n_1/N, where N is the size of the population (and which, of course, is what we're trying to estimate). Note that the numerator of this expression (n_1) is known, whereas the denominator (N) is not. In the second sample (Fig. 15.1), the ratio of the previously marked to the total number of individuals sampled is, simply, m_2/n_2.

Now, the key step, based on the following assumption – we assume that all individuals (marked or not) are equally catchable (meaning, we assume random mixing of marked and unmarked after the first sample). Under this assumption, then this proportion of previously marked individuals in the second sample should be equivalent to the proportion of newly marked individuals in the first sample:

$$\frac{m_2}{n_2} = \frac{n_1}{N}.$$

Next, a little algebraic rearrangement of this equation, and we come up with the familiar LP estimator for abundance, as

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

We might also use the canonical form noted earlier, where abundance is estimated as the count statistic divided by the encounter probability:

$$\hat{N} = \frac{n}{\hat{p}}.$$

If n_1 is the number of animals caught and marked at the first sampling occasion, and if m_2 is the number of the animals caught in both occasions, then assuming that (i) all n_1 individuals are alive and in the sample at occasion 2, and (ii) that marked and unmarked individuals have the same probability of detection, then the probability of encountering any of those n_1 marked individuals is

$$\hat{p} = \frac{m_2}{n_2}.$$

Thus, the ratio of the count statistic to the detection probability is the Lincoln-Petersen estimator:

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

15.2. Likelihood

While the 'algebraic' (LP) estimator for N developed in the preceding section is simple, reasonably intuitive and undoubtedly quite familiar, here we consider a more formal approach, based upon maximum likelihood estimation.

15.2.1. Full likelihood approach

We start by re-visiting the simple two sample study we used to motivate the LP estimator introduced in the previous section. For such a study, there are only 4 possible encounter histories: '`11`', '`10`', '`01`', and '`00`'. The number of individuals with encounter history '`00`' is not known directly, but must be estimated. So, the estimation of abundance proceeds by using the number of individuals observed who were encountered at least once.

We can express the probability distribution for n_1, n_2, and m_2, given the r (total) observed frequencies of the 3 observable encounter histories ('`11`', '`10`' and '`01`'), as

$$P\left(n_1, n_2, m_2 \mid N, p_1, p_2\right) = \frac{N!}{m_2!(n_1 - m_1)!(n_2 - m_2)!(N - r)!} \times \left(p_1 p_2\right)^{m_2} \left[p_1(1 - p_2)\right]^{(n_1 - m_2)} \left[(1 - p_1)p_2\right]^{(n_2 - m_2)} \left[(1 - p_1)(1 - p_2)\right]^{(N - r)}.$$

Two important things to note in this expression. First, N appears in the multinomial coefficient of the likelihood function. Second, the probability expression is written including a term for each encounter history, and with the exponent representing the number of individuals with a given encounter history

(expressed in the standard notation introduced earlier). For example, the probability of encounter history '11' is $p_1 p_2$, the probability of encounter history '10' is $p_1(1 - p_2)$, and so on.

Note also that the encounter history representing individuals that were never caught (i.e., '00' for a two occasion case) also appears (as the final term) in the likelihood (but not in the encounter histories file – since (obviously) there are no data for individuals that were never captured!).

More generally, we can write the likelihood as

$$\mathcal{L}(N, \mathbf{p} \mid \text{data}) \propto \frac{N!}{(N - M_{t+1})!} \prod_h P[h]^{n_h} \cdot P[\text{not encountered}]^{N - M_{t+1}},$$

where $\mathbf{p}$ is the vector of encounter probability parameters, M_{t+1} is the number of unique animals encountered (i.e., r in the expression on the previous page), and n_h is the number (frequency) of individuals with encounter history h.

Now, it is possible to rewrite the likelihood in terms of the number of individuals never caught, f_0, such that $f_0 = N - M_{t+1}$ (the notation 'f_0' originates from the frequency (count) of animals observed 0 times). The likelihood now becomes

$$\mathcal{L}(f_0, \mathbf{p} \mid \text{data}) \propto \frac{(f_0 + M_{t+1})!}{f_0!} \prod_h P[h]^{n_h} \cdot P[\text{not encountered}]^{f_0}.$$

The f_0 parametrization is useful computationally because f_0 is bounded on the interval $[0, \infty]$, thus forcing the logical constraint that $\hat{N} \geq M_{t+1}$. In fact, **MARK** uses the f_0 parametrization for ease of computation by using the log link function to constrain $\hat{f}_0 \geq 0$, but presents the results in terms of $\hat{N}$ as a *derived* parameter (i.e., $\hat{N} = \hat{f}_0 + M_{t+1}$ and $\widehat{\text{var}}[\hat{N}] = \widehat{\text{var}}[\hat{f}_0]$).

The fact that **MARK** uses f_0, the number of individuals never caught, in the likelihood has important implications you must keep in mind. Consider a study with two different sites (say, sampling plots) – you may be interested as to whether or not there is a difference between sites in abundance. How would you build a model where (say) you set $\hat{N}_1 = \hat{N}_2$? Answer – you can't. You can only apply constraints to parameters that are included in the likelihood. Abundance N isn't in the likelihood, so you can't build models that constrain N.

But, $\hat{N} = \hat{f}_0 + M_{t+1}$, and since M_{t+1} is a constant, then $\hat{N} \propto \hat{f}_0$. So wouldn't constraining $\hat{f}_0$ be equivalent to constraining $\hat{N}$? If you think about it for a moment, you should realize the answer is 'no, this is generally not reasonable'. Why? Consider setting $\hat{f}_{0,1} = \hat{f}_{0,2}$. This is easy enough to do in **MARK**, but, does it really make sense to say that 'the number never caught is the same in the 2 locations...'? Probably not. So, in short, you cannot constrain N in any meaningful way. [But, that doesn't mean you can't address the question of whether or not abundance differs between groups – we address one approach in Addendum 1.]

15.2.2. Conditional likelihood

It is sometimes convenient to use a conditional likelihood approach to estimating abundance, where N (or, equivalently, f_0) is not a parameter in the likelihood. This is possible if you 'condition' the analysis only on those individuals which are encountered (i.e., r).

Recall that the probability that any individual in the population is encountered at least once during a two-sample study is

$$p^* = 1.0 - (1 - p_1)(1 - p_2).$$

Thus, we can re-write the conditional probability expression for the capture histories as

$$P\left(\{x_{ij}\} \mid r, p_1, p_2\right) = \frac{r!}{x_{11}!x_{10}!x_{01}!} \times \left(\frac{p_1 p_2}{p^*}\right)^{x_{11}} \left(\frac{p_1(1-p_2)}{p^*}\right)^{x_{10}} \left(\frac{(1-p_1)p_2}{p^*}\right)^{x_{01}}.$$

The ML estimates for this model are again fairly easy to derive (see Williams, Nichols & Conroy 2002 for the details).

The primary advantage of using this conditional likelihood approach is that individual covariates can be used to model the encounter process. Individual covariates cannot be used with the full likelihood approach introduced in the preceding section, because the term $(1-p_1)(1-p_2)\ldots(1-p_t)$ is included in the likelihood, and no covariate value is available for animals that were never captured.

In contrast, the conditional likelihood approach conditions this multinomial term out of the likelihood, and so an individual covariate can be measured for each of the animals included in the likelihood. When individual covariates are used, a Horvitz-Thompson estimator is used to estimate N:

$$\hat{N} = \sum_{i=1}^{M_{t+1}} \frac{1}{1 - \left[1-\hat{p}_1(x_i)\right]\left[1-\hat{p}_2(x_i)\right]\ldots\left[1-\hat{p}_t(x_i)\right]}.$$

An example is perhaps the best way to illustrate the difference between the full and conditional likelihood approaches. Consider the 4 possible encounter histories for 2 sampling occasions:

encounter history	*probability*
`11`	$p_1 p_2$
`10`	$p_1(1-p_2)$
`01`	$(1-p_1)p_2$
`00`	$(1-p_1)(1-p_2)$

For each of the encounter histories except the last, the number of animals with the specific encounter history is known. For the last encounter history, the number of animals is $f_0 = (N - M_{t+1})$, i.e., the population size (N) minus the number of animals known to have been in the population (M_{t+1}).

The approach (first described by Huggins 1989, 1991) was to condition this last encounter history out of the likelihood by dividing the quantity '1 minus this last history' into each of the others. The result is a new multinomial distribution that still sums to one. The derived parameter N is then estimated as

$$\hat{N} = \frac{M_{t+1}}{\left[1 - (1-\hat{p})(1-\hat{p})(1-\hat{p})\right]},$$

for data with no individual covariates. A more complex estimator is required for models that include individual covariates to model the p parameters.

Here's a simple example of how this works, given 2 occasions. Let $p_1 = 0.4$, $p_2 = 0.3$. At the top of the next page, we tabulate both the *unconditional* probability of a given encounter history (i.e., where N is in the likelihood), and the *conditional* probability of the encounter history, where the individuals not seen are not included (i.e., are 'conditioned out'). Note that if $p_1 = 0.4$ and $p_2 = 0.3$, then the probability of not being captured at all is $(1-p_1)(1-p_2) = 0.42$, such that the probability of being captured at least once is $p^* = (1 - 0.42) = 0.58$.

history	*unconditional Pr(history)*		*Pr(history | captured)*	
11	$p_1 p_2$	$(0.4 \times 0.3) = 0.12$	$(p_1 p_2)/p^*$	$0.12/0.58 = 0.207$
10	$p_1(1-p_2)$	$0.4\,(1-0.3) = 0.28$	$\left[p_1(1-p_2)\right]/p^*$	$0.28/0.58 = 0.483$
01	$(1-p_1)p_2$	$(1-0.4)\,0.3 = 0.18$	$\left[(1-p_1)p_2\right]/p^*$	$0.18/0.58 = 0.310$
00	$(1-p_1)(1-p_2)$	$(1-0.4)(1-0.3) = 0.42$	(not included because not captured)	

In either case, the probabilities for all 4 histories sum to 1.0 (i.e., $(0.12 + 0.28 + 0.18 + 0.42) = 1.0$, and $(0.207 + 0.48 + 0.310) = 1.0$). Each forms a multinomial likelihood that can be solved for p_1 and p_2, by maximizing the likelihood expression.

As noted earlier, the derived parameter N is then estimated as

$$\hat{N} = \frac{M_{t+1}}{\left[1 - (1-\hat{p})(1-\hat{p})(1-\hat{p})\right]},$$

for data with no individual covariates.

Regardless of whether or not you include individuals not encountered in the likelihood, the key to understanding the fitting of closed capture models is in realizing that the event histories are governed entirely by the encounter probability.

In fact, the process of estimating abundance for closed models is in effect the process of estimating detection probabilities – the probability that an animal will be caught for the first time (if at all), and the probability that if caught at least once, that it will be caught again. The different closed population models differ conceptually on how variation in the encounter probability (e.g., over time, among individuals) is handled. The mechanics of fitting these models in **MARK** is the subject of the rest of this chapter.

begin sidebar

What does 'closure' really mean?

The 'closed captures' data types all assume the population of interest is closed during the sampling period (White *et al*. 1982). Meaning, the models assume that no births or deaths occur and no immigration or emigration occurs. Typically, we refer to a closed population as one that is free of unknown changes in abundance, as we can usually account for known changes.

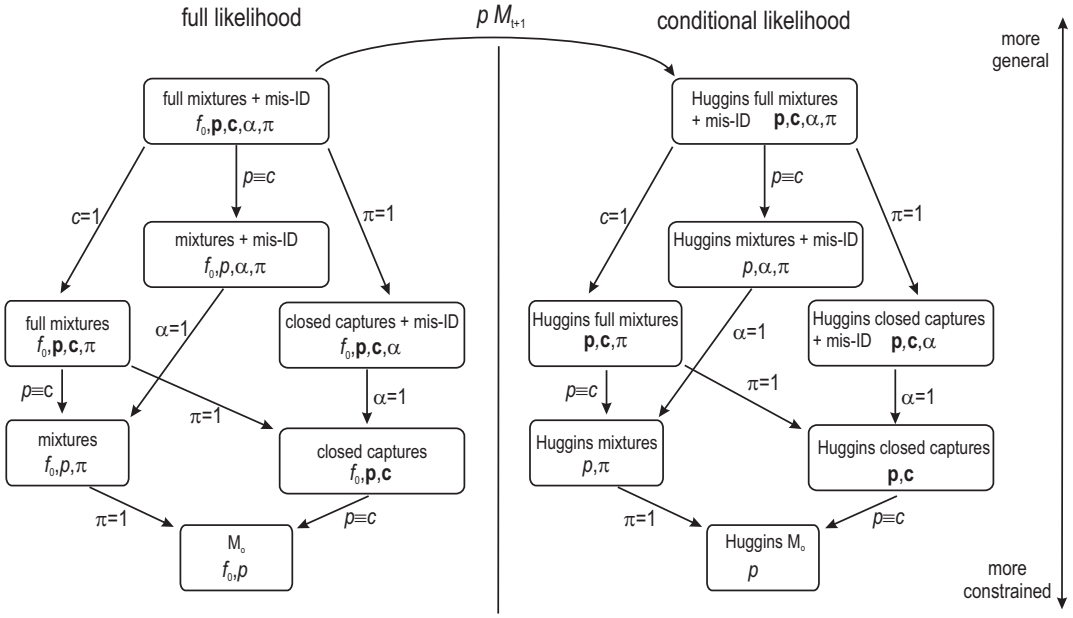
A few methods have been developed to test for closure violations. Program **CloseTest** (Stanley & Burnham 1999) can test the assumption of closure in some cases, although it is no longer in widespread use. The Pradel model with '**survival and recruitment**' parameterization has also been used to explore closure violations (Boulanger *et al*. 2002; see chapter 13 for details of the Pradel model), and offers some flexibility. By analyzing closed population capture-recapture data with the Pradel '**survival and recruitment**' parameterization, one could test for closure and violations of closure. For example, a model with φ fixed to 1 (no losses), and f fixed to 0 (no entries) would represent a model with 'full closure', and could be compared to a model where both φ and f are unconstrained. To test violations of closure due to emigration, you could construct a model with φ fixed to 1, with f unconstrained. Alternatively, to test for violation of closure due to immigration, you could construct a model with f fixed to 0, with φ unconstrained.

Heterogeneity in capture probability can cloud our ability to detect closure violations. In situations where the population is truly closed, heterogeneity in capture probability can cause both the tests of immigration and emigration to reject the null hypothesis of closure.

end sidebar

15.3. Model types

MARK currently supports 12 different closed population capture-recapture data types. These different data types can be classified within a hierarchy of dichotomous divisions – as shown in the diagram, below:



The first and most important split is between the models with abundance (or, rather, f_0) *in* the likelihood (Otis *et al*. 1978) and those with abundance *conditioned out* of the likelihood (Huggins 1989). We refer to the former as 'full likelihood' models, and the latter as either 'conditional likelihood' or 'Huggins' models. This is a major division that results in the two types of models not being comparable with standard AIC-based model selection techniques.

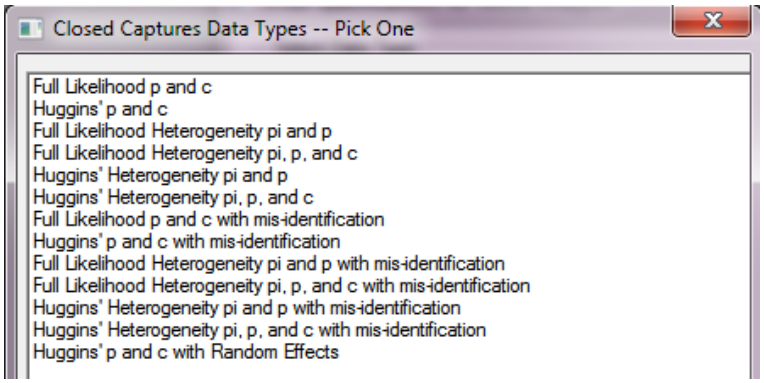
The remainder of the splits reflect one or more constraints on different parameters, and which parameters are included in the likelihoods. As noted earlier, the encounter histories in **MARK** are determined by the underlying encounter probabilities only. Minimally, most models in **MARK** are parameterized in terms of two different encounter parameters:

- ***p*** – the probability of first capture (i.e., the probability that an animal in the population will be captured – and marked - *for the very first time*)
- ***c*** – the probability of recapture (conditional on having been captured at least once before). The *c* parameter is generally used to model for behavioral effects following initial capture.

Both *p* and *c* can be time specific, although some specific constraints are required to ensure identifiability (discussed later). As a matter of convention in this chapter, we will use bold **p**'s and **c**'s to indicate a set (vector) of parameters that are (potentially) time varying, italic, un-subscripted *p*'s and *c*'s to indicate constant parameters, and italic, subscripted *p*'s and *c*'s refer to specific sampling occasions.

It is perhaps easiest to introduce the various models and parameters indicated in the preceding figure, by associating them with the different data types available in **MARK**. When you select '**`closed captures`**' in the data type specification window, **MARK** presents you with a popup window allowing

you to select among these 12 different data types (top of the next page).



The first data type is labeled '**`Full Likelihood p and c`**'. These are the models of Otis *et al*. (1978). They are based on the full likelihood parametrization with three types of parameters; p_i, c_i, and f_0 (the number of individuals in the population, but not encountered).

The second data type is labeled '**`Huggins p and c`**'. These are the models of Huggins (1989). In this model, the likelihood is conditioned on the number of animals detected and f_0 therefore drops out of the likelihood. These models contain only p_i and c_i; the abundance N is estimated as a derived parameter. As noted earlier, the primary advantage of the Huggins data type is that individual covariates can be used to model p and c.

The next 4 model types are *heterogeneity* models. These models incorporate a *finite mixture* as an approximation to individual heterogeneity in the p_i parameter. In this model,

$$p_i = \begin{cases} p_{i,A} & \text{with Pr}(\pi) \\ p_{i,B} & \text{with Pr}(1-\pi), \end{cases}$$

for the case with two mixtures A and B, although the model can be extended to >2 mixtures. As written (above), the parameter π is the probability that the individual occurs in mixture A. For >2 mixtures, additional π parameters must be defined (i.e., π_A, π_B,...), but constrained to sum to 1.

Note that the '**`heterogeneity models`**' for both full likelihood closed captures and the Huggins' models come in one of two forms, differentiated by the presence of either (i) the mixture parameter, π, and both the p_i and c_i parameters, or (ii) the mixture parameter, π, and a single encounter parameter, p, only. The latter parameterizations (with only the π and p parameters) represent simple individual heterogeneity models, with parameters π, $p_{i,A} = p_A$, and $p_{i,B} = p_B$, and assume no temporal or behavioral variation. In contrast, the *full* parametrization models (including π, p and c parameters) provide for all three effects of time, behavior, and heterogeneity. Of course, any of the reduced models can be run from the full parameterizations if the appropriate constraints are applied.

The next six data types generalize the previous six data types to handle uncertainty in identification of individuals, typically from genotyping error (Lukacs & Burnham 2005). These models include an additional parameter, α, that is the probability that the individual was correctly identified on its first observation. In these models, N is estimated as a derived parameter. While it is possible to construct models for every data type using only the '**`Full Likelihood heterogeneity pi, p, and c with mis-identification`**' or '**`Huggins heterogeneity pi, p, and c with mis-identification`**' models, the other data types are included to allow the user a less cumbersome set of parameters for building more constrained models.

Finally, there is a '**Huggins p and c with Random Effects**' model, which uses numerical integration to integrate out continuous, individual random differences in latent encounter probability. This approach is conceptually somewhat 'outside' the simple 'full likelihood' versus 'conditional likelihood' models split introduced earlier.

The heterogeneity, misidentification and random effects models will be treated in more detail later in this chapter.

15.3.1. Constraining the final **p**

A subtlety of the closed population models is that the last p parameter is not identifiable unless a constraint is imposed. When no constraint is imposed on the last p_i , the likelihood is maximized with the last $p = 1$, giving the estimate $\hat{N} = M_{t+1}$. Why?

Consider a simple 2 occasion study. For this study, there are 4 possible encounter histories: '11', '10', '01', and '00'. Under the full likelihood model, the probabilities of observing each history are:

history	*probability*
11	$p_1 c_2$
10	$p_1(1-c_2)$
01	$(1-p_1)p_2$
00	$(1-p_1)(1-p_2)$

Our interest concerns the final p parameter, p_2. We see that p_2 is a term in the probability expression for the '01' and '00' histories only. Taking the ratio of the *observed* frequency of '00' individuals to the *observed* frequency of '01' individuals (which is an *ad hoc* way of estimating p_2; see Chapter 1), then

$$\frac{f_{\{00\}}}{f_{\{01\}}} = \frac{\cancel{(1-p_1)}(1-p_2)}{\cancel{(1-p_1)}p_2}$$

$$= \frac{(1-p_2)}{p_2}.$$

Focus on the LHS of this expression. The numerator, $f_{\{00\}}$, must be 0. Why? This must be true since the '00' history refers to individuals not seen. So, the observed frequency of animals not seen, $f_{\{00\}}$, is 0 (obviously), and thus the LHS of our equation is $0/f_{\{01\}} = 0$.

Thus, we solve for p_2 as

$$\frac{f_{\{00\}}}{f_{\{01\}}} = \frac{(1-p_2)}{p_2}$$

$$0 = \frac{(1-p_2)}{p_2}$$

$$= 1 - p_2$$

$$\therefore \quad \hat{p}_2 = 1.$$

OK, so the final encounter probability p_2 is estimated at 1. But, why is that a problem? If you think about it, with 'perfect' first detection at *any* occasion, the number of individuals never encountered throughout the study (f_0) would be 0, which in turn makes $\hat{N} = M_{t+1}$.

Let's confirm this 'logical conclusion' by means of some simple algebra. Recall that the canonical estimator for $\hat{N}$ is the count statistic (in this case, M_{t+1}) divided by the encounter probability. For a two occasion study,

$$\hat{N} = \frac{M_{t+1}}{\left(1 - \left[(1-\hat{p}_1)(1-\hat{p}_2)\right]\right)}.$$

If $\hat{p}_2 = 1$, then

$$\begin{aligned}
\hat{N} &= \frac{M_{t+1}}{\left(1 - \left[(1-\hat{p}_1)(1-\hat{p}_2)\right]\right)} \\
&= \frac{M_{t+1}}{\left(1 - \left[(1-\hat{p}_1)(1-1)\right]\right)} \\
&= \frac{M_{t+1}}{(1-0)} \\
&= M_{t+1}.
\end{aligned}$$

Thus, unless a constraint is placed on the last p, then the estimated abundance N will simply be M_{t+1}. Thus, it is diagnostic to check to see whether $\hat{N} = M_{t+1}$, and if so, to see if the last p_i estimate equals 1. If they are, then you've forgotten to constrain p.*

So, in model M_t, the constraint of $p_i = c_i$ is imposed, providing an estimate of the last p from the last c. Likewise, under model M_b, the constraint of $p_i = p_.$ is imposed, so that the last p is assumed equal to all the other p values. A similar constraint is used for model M_{bh}, i.e., $p_{i,A} = p_A$, $p_{i,B} = p_B$, and so on. Under model M_{tb}, the p_i and c_i are modeled as a constant offset (O_{beh}) of one another, i.e., $c_i = (p_i + O_{beh})$. This relationship will depend on the link function used, but the last p_i is still obtained as c_i minus the offset (where the offset is estimated from the data on the other p_i and c_i).

Under model M_{tbh}, the offset between the p_i and c_i is applied, with an additional offset(s) included to model the relationship among the mixtures, i.e., $p_{i,B} = (p_{i,A} + O_B)$, $p_{i,C} = (p_{i,A} + O_C)$, with a different offset applied to each succeeding mixture. Similarly, $c_{i,B} = (p_{i,B} + O_{beh}) = (p_{i,A} + O_B + O_{beh})$, with the resulting relationship depending on the link function used. With this model, the relationship between the mixtures of the p_i is maintained, i.e., the ordering of the mixtures is maintained across occasions. Model M_{th} can also be modeled as an additive offset between the mixtures, although other relationships are possible because the last p_i for each mixture is estimated from the corresponding last c_i.

Although other relationships than those of the preceding paragraph can be proposed to provide identifiability, the proposed models must provide identifiability of all the initial capture probabilities.

* The diagnostics of the final $\hat{p} = 1$ and $\hat{N} = M_{t+1}$ strictly apply only to the *full likelihood* model. Under the *conditional likelihood* (Huggins) model, the final $\hat{p} \neq 1$, and $\hat{N} \neq M_{t+1}$. However, the reported estimates of $\hat{p}$ and $\hat{N}$ from the unconstrained Huggins model will be meaningless, since the SE will generally be unrealistically large.

15.4. Encounter histories format

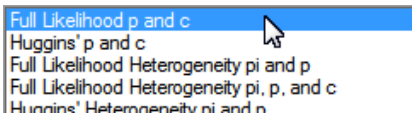
All of the closed capture-recapture models use the LLLL encounter histories format (see chapter 2 for more detail). By the definition of a closed population, animals are not dying, therefore a dead encounter is not possible. On the same line of reasoning, time between sampling occasions is not relevant because there is no survival or movement process to consider. Encounter histories are followed by group frequencies. For the Huggins models, group frequencies can be followed with individual covariates. All encounter histories end with the standard semicolon.

```
/* Closed capture-recapture data for a Huggins model.
   tag #, encounter history, males, females, length */
 /* 001 */   1001    1   0   22.3;
 /* 002 */   0111    1   0   18.9;
 /* 003 */   0100    0   1   20.6;
```

If you wish to analyze a data set that contains covariates in the input with both full and conditional likelihoods, you must initially import that data set by selecting a '**Huggins**' data type. The '**Closed Captures**' data type will not allow individual covariates to be specified. In this case, it is likely best to create two separate **MARK** files for the analysis because the AIC_c values are not comparable between the '**Closed Captures**' and '**Huggins**' data types.

15.5. Building models

Now it is time to move on to the actual mechanics of closed population abundance estimation in **MARK**. We will analyze some simulated data contained in (**simple_closed1.inp**). In this simulated data set (which consists of 6 encounter occasions), true $N = 350$. The total number of individuals encountered was $M_{t+1} = 339$ (so, 11 individuals were never seen). Open **MARK** and create a new database using the '**File | New**' option. Select the '**Closed Captures**' radio-button. When you click on the '**Closed Captures**' radio-button, a window will open that allows you to select a model type, shown earlier in this chapter. To start, select '**Full Likelihood p and c**'.



Enter a title, select the input file, and set the number of encounter occasions to 6.

To start, we'll construct some of the 'standard' closed capture models, as originally described in Otis *et al*. (1978). Model notation for the closed capture-recapture models in the literature often still follows that of Otis *et al*. (1978). Now that more complex models can be built, it seems appropriate to use a notation that is similar to the notation used for other models in **MARK**. Thus, our notation in this chapter will be based on a description of the parameters in the models.

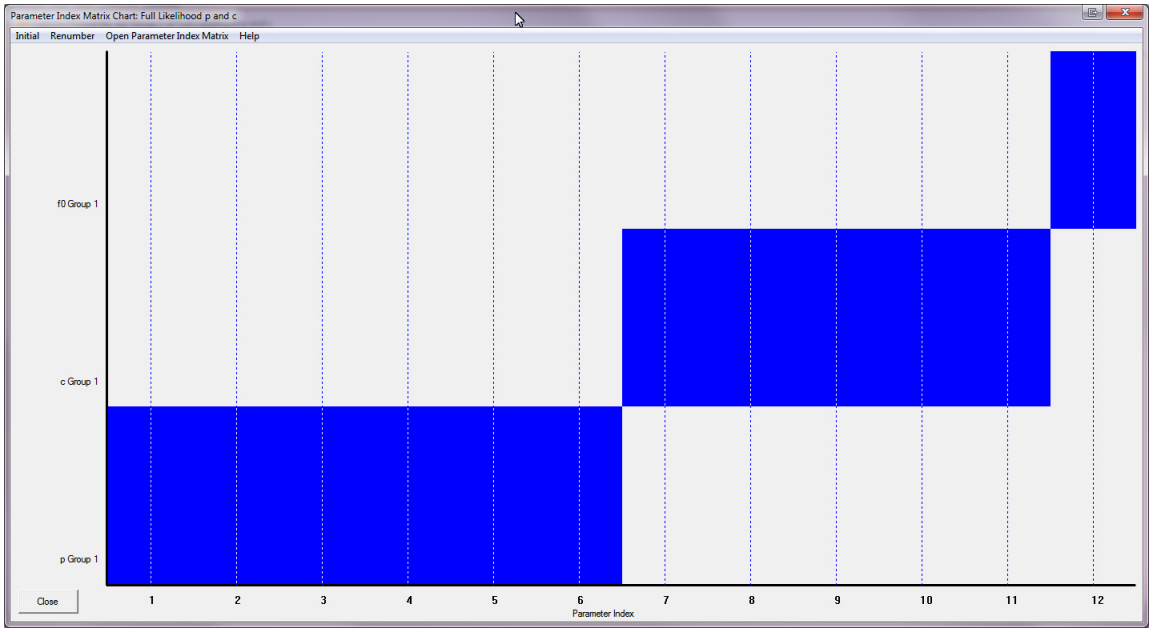
At the top of the next page, we present a table contrasting model notation based on Otis *et al*. (1978) and expanded notation based on a more explicit (and informative) description of the parameters. Combinations of the models described are possible.

Otis notation	*Expanded notation*	*Description*
M_0	$\{f_0, p(\cdot) = c(\cdot)\}$	Constant p
M_t	$\{f_0, p(t) = c(t)\}$	Time varying p
M_b	$\{f_0, p(\cdot), c(\cdot)\}$	Behavioral response
M_h or M_{h2}	$\{f_0, p_a(\cdot) = c_a(\cdot), p_b(\cdot) = c_b(\cdot), \pi\}$	Heterogeneous p

If you look closely at the 'expanded notation', you'll see that models are differentiated based on relationships between the p and c parameters. This is important – the closed capture-recapture models are one of the model types in **MARK** where different types of parameters are modeled as functions of each other. In this case p and c are commonly modeled as functions of one another. This makes intuitive sense because both p and c relate to catching animals.

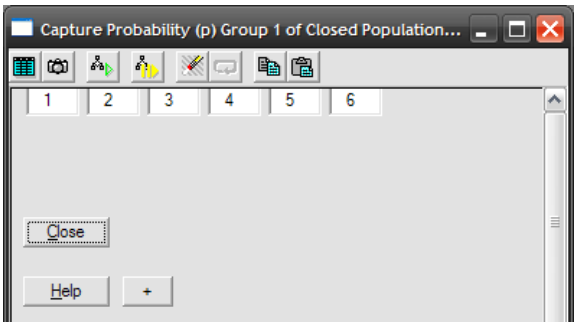
With that said, let's begin building a few models to learn the mechanics of using **MARK** to estimate abundance. We'll start with models $\{f_0, p(\cdot) = c(\cdot)\}$, $\{f_0, p(t) = c(t)\}$, and $\{f_0, p(\cdot), c(\cdot)\}$ (i.e., models M_0, M_t and M_b).

Let's first examine the default PIM chart for the '**`Full Likelihood p and c`**' models:



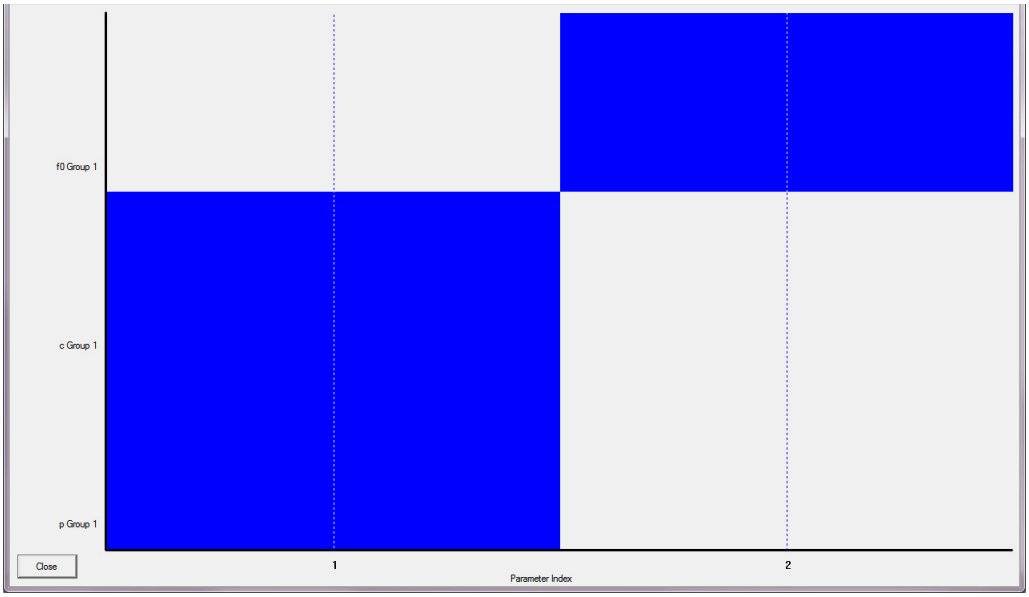
MARK defaults to a time-varying parameter structure where there is a different p and c for each occasion. Recall from section 14.3.1 that abundance is not estimable with this model structure because no constraint is imposed to estimate p_6. If this default, fully time-dependent model is fit to the data, $\hat{N} = M_{t+1}$ and $\hat{p}_6 = 1.0$ regardless of the data. Therefore, in every model we build we must put some constraint on p_i for the last encounter occasion so that this parameter is estimated.

If we open the PIM windows, we'll notice that the p's and c's have only a single row of text boxes. For example, for p (shown at the top of the next page):



In the closed capture models, every individual is assumed to be in the population and at risk of capture on every occasion. Therefore, there is no need for cohorts (expressed as multiple rows in the PIM window) as there is for many of the open-population models.

We'll start with $\{f_0, p(\cdot) = c(\cdot)\}$ – for this model, there is no temporal variation in either p or c, and the two parameters are set equal to each other. This model is easily constructed using the PIM chart:



Go ahead and run this model, and add the results to the browser. Couple of important things to note. First, it is common for AIC_c values to be negative for the full likelihood closed captures models. Negative AIC_c values are legitimate and interpreted in the same way as positive AIC_c values. The negative AIC arises due to the portion of the multinomial coefficient that is computed. Recall that for the full likelihood for the 2-sample situation, the multinomial coefficient was written as

$$\frac{N!}{m_2!(n_1-m_1)!(n_2-m_2)!(N-r)!} \equiv \frac{(f_0+M_{t+1})!}{m_2!(n_1-m_1)!(n_2-m_2)!f_0!},$$

which, after dropping terms that did not include N (or f_0), simplifies to

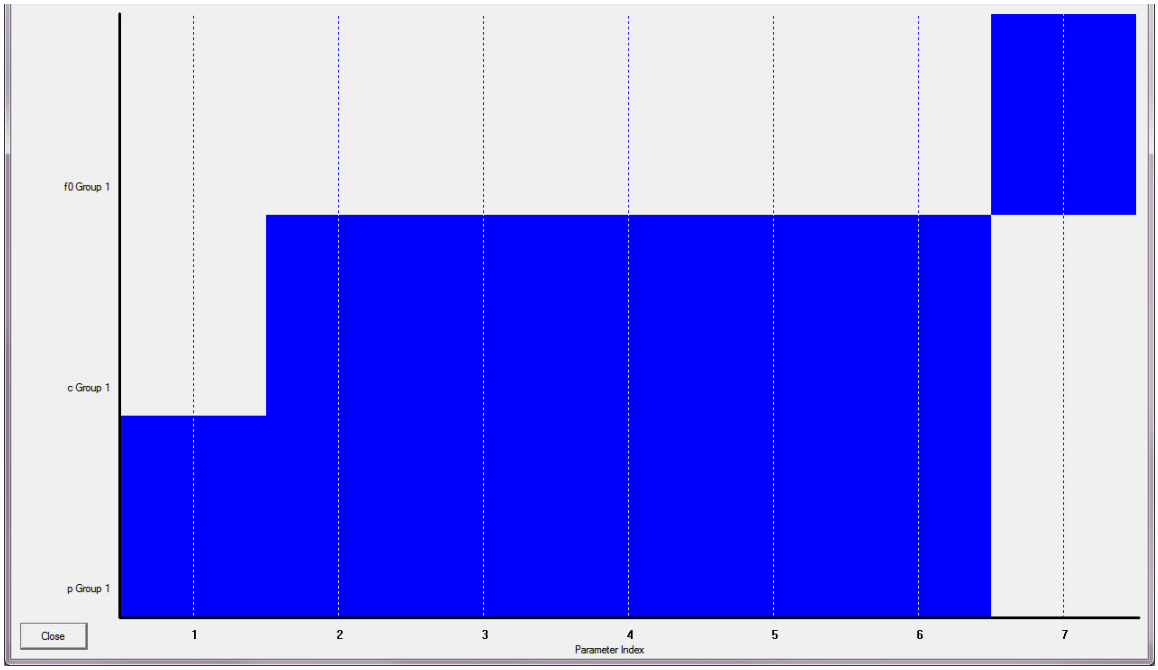
$$\frac{(f_0+M_{t+1})!}{\cancel{m_2!}\,\cancel{(n_1-m_1)!}\,\cancel{(n_2-m_2)!}\,f_0!} \propto \frac{(f_0+M_{t+1})!}{f_0!},$$

which is frequently negative (which results in a negative AIC_c). In contrast, AIC_c values from the conditional likelihood models are typically positive. Regardless, the model with the 'most negative' AIC_c, i.e., the one furthest from zero, is the most parsimonious model.

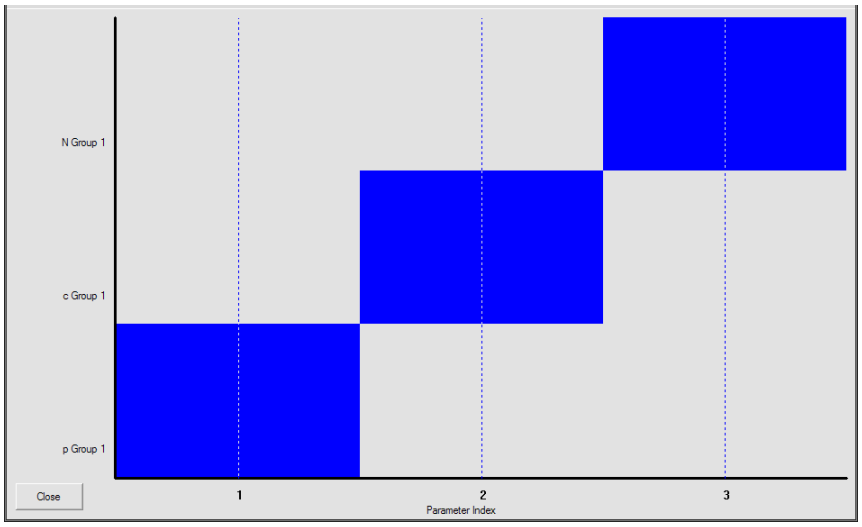
Also, note that **MARK** defaults to a sin link, just as it does with all other data types when an identity design matrix is specified. In the case of the closed models, the sin link is used for the p's and c's, but a log link is used for f_0. The log link is used because f_0 must be allowed to be in the range of $[0 \rightarrow \infty]$. Therefore, no matter what link function you select, a log link will be used on f_0. If you choose the '**Parm-Specific**' option to set different link functions for each parameter, be sure you choose a link that does not constrain f_0 to the $[0, 1]$ interval. Choose either a log or identity link (log is preferable).

Now, we'll build model $\{f_0, p(t) = c(t)\}$ (i.e., model M_t). Remember, there is no c parameter for the first occasion because it is impossible for an animal to be recaptured until it has been captured once. **MARK** offers an easy way to assure that the correct p's line up with the correct c's: under the '**Initial**' menu select '**make c=p**' and renumber with overlap. The constraint on p_6 in this model is that $p_6 = c_6$.

Here is the PIM chart:



Finally, we'll build model $\{f_0, p(\cdot), c(\cdot)\}$ (i.e., model M_b). Here, we're accounting for possible differences in 'behavior' between the first encounter, and subsequent encounters. Such a 'behavioral' effect might indicate some permanent 'trap effect' (trap 'happiness' or trap 'aversion'). For model $\{f_0, p(\cdot), c(\cdot)\}$, shown at the top of the next page, there is a 'behavior' effect, but no temporal variation:



Note that there is no 'overlap' (i.e., no function relating p and c) for this model – this is analogous to the default model $\{f_0, p(t), c(t)\}$, shown earlier. However, in this instance, all parameters are estimable because of the constraint that p and c are constant over time – the lack of estimability for the final p occurs if p is time dependent. As such, model $\{f_0, p(\cdot), c(t)\}$ would be estimable, while model $\{f_0, p(t), c(\cdot)\}$ would not (for this model $\hat{N} = M_{t+1}$). You might want to confirm this for yourself.

begin sidebar

simple extension – removal models

Now let's consider a *removal* model. These are commonly used in fisheries work where the researcher does not want to subject a fish to multiple passes of electricity. Therefore, the fish that are encountered are held aside until all sampling has occurred.

To accomplish this in **MARK**, build an $\{f_0, p(t), c(\cdot)\}$ or $\{f_0, p(\cdot), c(\cdot)\}$ model. Then click '**Run**' to open the run window. Click the '**fix parameters**' button. A window will open listing all of the real parameters in the model. Simply fix $c = 0$, and run the model.

Note, however, that a special requirement of removal data is that there has to be a general downward trend in the number of animals removed on each occasions, i.e., there has to be some depletion of the population. Seber & Whale (1970) showed that N and p can be estimated from data when the following "failure criterion" is satisfied:

$$\sum_{j=1}^{t}(t+1-2j)u_j > 0,$$

where t is the number of sampling (removal) occasions, and u_j is the number of animals captured and removed on occasion j.

end sidebar

15.6. Closed population models and the design matrix

In the preceding, we constructed 3 simple models using the PIM chart. While using the PIM chart was very straightforward for those models, through the design matrix **MARK** allows models to be fit that were not possible with the PIM chart. For example, it is possible to build an $\{f_0, p(t) = c(t) + b\}$ model

where capture probability and recapture probability are allowed to vary through time, but constrained to be different by an additive constant on the logit scale. It is also worth noting that these extended models are not available in program **CAPTURE** (one of several reasons that **CAPTURE** is no longer preferred for fitting closed population abundance models).

As introduced in Chapter 6, one approach to doing this is to first build a general model using PIMs, and then construct the design matrix corresponding to this general model. Then, once you have the general model constructed using the design matrix, all other models of interest can be constructed simply by modifying the design matrix. In this case, the most general model we can build is $\{f_0, p(t), c(t)\}$. As noted earlier, we know before the fact that this particular model is not a useful model, but it is convenient to build the design matrix for this model as a starting point.

To do this we need the PIMs in the full time varying setup (as shown earlier). Go ahead and run this model, and add the results to the browser. Look at the real and derived parameter estimates – note that (i) $\hat{p}_5 = 1.0$, and (ii) $\hat{N} = M_{t+1} = 339$. Note as well that the reported SE's for both $\hat{p}_5$ and $\hat{N}$ are impossibly small – a general diagnostic that there is 'something wrong' with this model. As discussed earlier, this is not a useful model without imposing some constraints since the estimate of $\hat{N} = M_{t+1}$.

Now, the design matrix. Recall that there are 12 parameters specifying this model: $1 \rightarrow 6$ for p, $7 \rightarrow 11$ for c, and parameter 12 for abundance, N. Thus, our design matrix will have 12 columns. Now, if you select '**`Design | Full`**', **MARK** will respond with the default DM shown below:

B1: p Int	B2: p t1	B3: p t2	B4: p t3	B5: p t4	B6: p t5	Parm	B7: c Int	B8: c t1	B9: c t2	B10: c t3	B11: c t4	B12: f0 Int
1	1	0	0	0	0	1:p	0	0	0	0	0	0
1	0	1	0	0	0	2:p	0	0	0	0	0	0
1	0	0	1	0	0	3:p	0	0	0	0	0	0
1	0	0	0	1	0	4:p	0	0	0	0	0	0
1	0	0	0	0	1	5:p	0	0	0	0	0	0
1	0	0	0	0	0	6:p	0	0	0	0	0	0
0	0	0	0	0	0	7:c	1	1	0	0	0	0
0	0	0	0	0	0	8:c	1	0	1	0	0	0
0	0	0	0	0	0	9:c	1	0	0	1	0	0
0	0	0	0	0	0	10:c	1	0	0	0	1	0
0	0	0	0	0	0	11:c	1	0	0	0	0	0
0	0	0	0	0	0	12:f0	0	0	0	0	0	1

Here, we see a DM which is strictly analogous to what we might have expected for 3 parameters – each parameter (in this case, p, c and f_0) has a separate 'block' within the matrix: p in the upper-left, c in the middle, and f_0 in the lower-right. If you go ahead and run this model, you'll see (below) that it gives you exactly the same model deviance as the general model built with PIMs.

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{f0,p(t),c(t) - DM)}	-530.1030	0.0000	0.53930	1.0000	10	41.6249
{f0,p(t)=c(t)}	-528.6779	1.4251	0.26447	0.4904	7	49.1035
{f0,p(t),c(t) - PIM}	-528.0811	2.0219	0.19624	0.3639	11	41.6249
{f0,p(.),c(.)}	-501.8661	28.2369	0.00000	0.0000	3	83.9587
{f0,p(.)=c(.)}	-494.4003	35.7027	0.00000	0.0000	2	93.4304

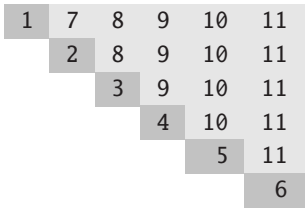
You'll also note, however, that the AIC_c reported for this DM-built general model is not the same as the AIC_c reported for the general model built with PIMs (−530.1030 versus −528.0812). If the model

deviances are the same, but the reported AIC_c values are different, then this implies that the number of estimated parameters is different. In fact, we see that the number estimated for the 'full default DM' model is 10, whereas for the model built with PIMs, the number reported is 11. In fact, for this model, the difference in the number reported isn't particularly important, since this is not a 'reasonable' model in the first instance (as mentioned several times earlier in this chapter). The fact that the model deviances 'match' indicates that the DM is correct.

However, while this is 'technically' true, the default DM assumes that there is no interest in creating a functional relationship between any of the parameters. While normally this is a reasonable assumption (e.g., in a CJS live encounter study, there is no plausible reason to create a functional relationship between φ and p), this is clearly not the case for closed population abundance models, where many of the models of interest are specified by imposing a particular relationship between p and c. For example, model $\{f_0, p(t) = c(t)\}$ imposes a relationship between p and c at each sampling occasion t.

How do we accommodate our interest in specifying these relationships between p and c in the DM? In fact, it is very easy, with a simple conceptual 'trick' – we're going to treat the two parameters p and c as if they were levels of some putative 'treatment' – in precisely the same way (structurally) that we handled age (TSM) effects for individuals marked as young in age (TSM) models (Chapter 7 – section 7.2). As a reminder, recall how we would construct the design matrix to correspond to the PIM for survival for a simple age model, with 2 age classes, and time-dependence in each age class. Assume that we have 7 occasions.

Recall that the PIM for this model looks like:



So, based on the number of indexed parameters in the PIM, we know already that our design matrix for survival would need to have 11 rows and 11 columns.

What does the linear model look like? Again, writing out the linear model is often the easiest place to start. In this case we see that over a given time interval, we have, in effect, 2 kinds of individuals: *juveniles* (individuals in their first year after marking), and *adults* (individuals at least 2 years after marking). Thus, for a given `TIME` interval, there are 2 groups: juvenile and adult. If we call this group effect `AGE`, then we can write out our linear model as

$$\begin{aligned}
\text{'survival'} &= \texttt{AGE + TIME + AGE.TIME} \\
&= \beta_1 \\
&\quad + \beta_2(\texttt{AGE}) \\
&\quad + \beta_3(\texttt{T}_1) + \beta_4(\texttt{T}_2) + \beta_5(\texttt{T}_3) + \beta_6(\texttt{T}_4) + \beta_7(\texttt{T}_5) \\
&\quad + \beta_8(\texttt{AGE.T}_2) + \beta_9(\texttt{AGE.T}_3) + \beta_{10}(\texttt{AGE.T}_4) + \beta_{11}(\texttt{AGE.T}_5).
\end{aligned}$$

Again, recall from Chapter 7 that there is no $(\texttt{AGE.T}_1)$ interaction term. Also remember, we're treating the two age classes as different groups – this will be the key 'conceptual step' in seeing how we apply the same idea to closed population abundance models.

The design matrix corresponding to this linear model is:

B1 int	B2 age	B3 t1	B4 t2	B5 t3	B6 t4	B7 t5	B8 a.t2	B9 a.t3	B10 a.t4	B11 a.t5	Parm	B12 p2	B13 p3
1	1	1	0	0	0	0	0	0	0	0	1:Phi	0	0
1	1	0	1	0	0	0	1	0	0	0	2:Phi	0	0
1	1	0	0	1	0	0	0	1	0	0	3:Phi	0	0
1	1	0	0	0	1	0	0	0	1	0	4:Phi	0	0
1	1	0	0	0	0	1	0	0	0	1	5:Phi	0	0
1	1	0	0	0	0	0	0	0	0	0	6:Phi	0	0
1	0	0	1	0	0	0	0	0	0	0	7:Phi	0	0
1	0	0	0	1	0	0	0	0	0	0	8:Phi	0	0
1	0	0	0	0	1	0	0	0	0	0	9:Phi	0	0
1	0	0	0	0	0	1	0	0	0	0	10:Phi	0	0
1	0	0	0	0	0	0	0	0	0	0	11:Phi	0	0
0	0	0	0	0	0	0	0	0	0	0	12:p	1	0
0	0	0	0	0	0	0	0	0	0	0	13:p	0	1

So, column `B2` in this design matrix indicates a putative 'age group' – for a given cohort, and a given time step, is the individual young (indicated with the dummy '1') or adult (indicated with the dummy '`0`'). If you don't recall this connection, go back and re-read section 7.2.

Now, what does this have to do with building design matrices for closed abundance estimation models? The connection relates to the idea of creating a 'logical group'. For age models, we used the age of an individual for a given cohort and time step as a grouping variable. For closed population abundance models, we do the same thing – except that instead of age, we're going to 'group' as a function of whether or not the individual has been captured at least once or not. In other words, we're going to treat the parameters p (caught for the first time) and c (caught subsequently) as levels of a putative 'encounter' group (analogous to young and adult, respectively).

This will make more sense when you see how we set up the DM. Here it is – note that it is *identical* to the age (TSM) model (above):

B1 intercept	B2 enc grp	B3 t1	B4 t2	B5 t3	B6 t4	B7 t5	Parm	B8 eg*t2	B9 eg*t3	B10 eg*t4	B11 eg*t5	B12 f0
1	1	1	0	0	0	0	1:p	0	0	0	0	0
1	1	0	1	0	0	0	2:p	1	0	0	0	0
1	1	0	0	1	0	0	3:p	0	1	0	0	0
1	1	0	0	0	1	0	4:p	0	0	1	0	0
1	1	0	0	0	0	1	5:p	0	0	0	1	0
1	1	0	0	0	0	0	6:p	0	0	0	0	0
1	0	0	1	0	0	0	7:c	0	0	0	0	0
1	0	0	0	1	0	0	8:c	0	0	0	0	0
1	0	0	0	0	1	0	9:c	0	0	0	0	0
1	0	0	0	0	0	1	10:c	0	0	0	0	0
1	0	0	0	0	0	0	11:c	0	0	0	0	0
0	0	0	0	0	0	0	12:f0	0	0	0	0	1

Column `B1` is the common intercept – this is a necessary step (and a key difference from the default DM) in order to allow us to specify a functional relationship between p and c. Column `B2` is the column which specifies the putative 'encounter group' – first encounter (corresponding to parameter p) or subsequent encounter (corresponding to parameter c). Note that there are 6 '1's; for p, but only 5 '`0`'s' for c (since there is no c parameter for occasion 1).

This is *entirely analogous* to having no adults in the first occasion for individuals marked as young. Columns B3 → B7 correspond to the time steps – again, note that for parameter c, there is no time coding for interval 1. These are followed by the interaction columns B8 → B11. Again, there is no logical interaction of p and c for occasion 1 (since there is no parameter c_1), so the interaction columns start with time interval 2. Finally, column B12 for the parameter f_0.

Go ahead, run this model, and add the results to the browser:

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{f0,p(t),c(t) - DM)}	-530.1030	0.0000	0.35035	1.0000	10	41.6249
{f0,p(t),c(t) - DM - common intercept}	-530.1030	0.0000	0.35035	1.0000	10	41.6249
{f0,p(t)=c(t)}	-528.6779	1.4251	0.17181	0.4904	7	49.1035
{f0,p(t),c(t) - PIM}	-528.0811	2.0219	0.12748	0.3639	11	41.6249
{f0,p(.),c(.)}	-501.8661	28.2369	0.00000	0.0000	3	83.9587
{f0,p(.)=c(.)}	-494.4003	35.7027	0.00000	0.0000	2	93.4304

We see that the model deviances for the general model constructed with (i) PIMs, (ii) the default DM (which used a separate intercept for each parameter), and (iii) the modified DM which used a common intercept, are all identical.

Now, let's use the DM to build the 3 models we constructed previously using PIMs. First, model $\{f_0, p(\cdot) = c(\cdot)\}$. We see that (i) there is no temporal variation (meaning, we simply delete the columns corresponding to time and interactions with time from the DM – columns B3 → B11), and (ii) $p = c$ (meaning, we delete the column specifying difference between the 'encounter groups' – column B2):

B1: intercept	Parm	B2: f0
1	1:p	0
1	2:p	0
1	3:p	0
1	4:p	0
1	5:p	0
1	6:p	0
1	7:c	0
1	8:c	0
1	9:c	0
1	10:c	0
1	11:c	0
0	12:f0	1

Run this model and add the results to the browser:

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{f0,p(t),c(t) - DM)}	-530.1030	0.0000	0.35035	1.0000	10	41.6249
{f0,p(t),c(t) - DM - common intercept}	-530.1030	0.0000	0.35035	1.0000	10	41.6249
{f0,p(t)=c(t)}	-528.6779	1.4251	0.17181	0.4904	7	49.1035
{f0,p(t),c(t) - PIM}	-528.0811	2.0219	0.12748	0.3639	11	41.6249
{f0,p(.),c(.)}	-501.8661	28.2369	0.00000	0.0000	3	83.9587
{f0,p(.)=c(.)}	-494.4003	35.7027	0.00000	0.0000	2	93.4304
{f0,p(.)=c(.) - DM coding}	-494.4003	35.7027	0.00000	0.0000	2	93.4304

We see the model results match those of the same model constructed using PIMs.

What about model $\{f_0, p(\cdot), c(\cdot)\}$? Here, we again delete all of the time and interaction columns, but retain the column coding for the 'encounter group' term in the model:

B1: intercept	Parm	B2: enc grp	B3: f0
1	1:p	1	0
1	2:p	1	0
1	3:p	1	0
1	4:p	1	0
1	5:p	1	0
1	6:p	1	0
1	7:c	0	0
1	8:c	0	0
1	9:c	0	0
1	10:c	0	0
1	11:c	0	0
0	12f0	0	1

Again, we see (below) that the results of fitting this model constructed using the DM approach exactly match those from the same model constructed using PIMs:

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{f0,p(t),c(t) - DM)}	-530.1030	0.0000	0.35035	1.0000	10	41.6249
{f0,p(t),c(t) - DM - common intercept}	-530.1030	0.0000	0.35035	1.0000	10	41.6249
{f0,p(t)=c(t)}	-528.6779	1.4251	0.17181	0.4904	7	49.1035
{f0,p(t),c(t) - PIM}	-528.0811	2.0219	0.12748	0.3639	11	41.6249
{f0,p(.),c(.)}	-501.8661	28.2369	0.00000	0.0000	3	83.9587
{f0,p(.),c(.) - DM coding}	-501.8661	28.2369	0.00000	0.0000	3	83.9587
{f0,p(.)=c(.)}	-494.4003	35.7027	0.00000	0.0000	2	93.4304
{f0,p(.)=c(.) - DM coding}	-494.4003	35.7027	0.00000	0.0000	2	93.4304

Finally, model $\{f_0, p(t) = c(t)\}$. Here, we have no 'encounter group' effect, but simple temporal variation in p and c. We simply delete the interaction and 'encounter group' columns:

B1: intercept	Parm	B2: t1	B3: t2	B4: t3	B5: t4	B6: t5	B7: f0
1	1:p	1	0	0	0	0	0
1	2:p	0	1	0	0	0	0
1	3:p	0	0	1	0	0	0
1	4:p	0	0	0	1	0	0
1	5:p	0	0	0	0	1	0
1	6:p	0	0	0	0	0	0
1	7:c	0	1	0	0	0	0
1	8:c	0	0	1	0	0	0
1	9:c	0	0	0	1	0	0
1	10:c	0	0	0	0	1	0
1	11:c	0	0	0	0	0	0
0	12f0	0	0	0	0	0	1

We see (below) that the model deviances are identical, regardless of whether or not the PIM or DM approach was used.

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{f0,p(t),c(t) - DM)}	-530.1030	0.0000	0.29898	1.0000	10	41.6249
{f0,p(t),c(t) - DM - common intercept}	-530.1030	0.0000	0.29898	1.0000	10	41.6249
{f0,p(t)=c(t)}	-528.6779	1.4251	0.14662	0.4904	7	49.1035
{f0,p(t)=c(t) - DM}	-528.6779	1.4251	0.14662	0.4904	7	49.1035
{f0,p(t),c(t) - PIM}	-528.0811	2.0219	0.10879	0.3639	11	41.6249
{f0,p(.),c(.)}	-501.8661	28.2369	0.00000	0.0000	3	83.9587
{f0,p(.),c(.) - DM coding}	-501.8661	28.2369	0.00000	0.0000	3	83.9587
{f0,p(.)=c(.)}	-494.4003	35.7027	0.00000	0.0000	2	93.4304
{f0,p(.)=c(.) - DM coding}	-494.4003	35.7027	0.00000	0.0000	2	93.4304

Now, let's consider a model which we can't build using the PIM-only approach (or, as noted, if we'd relied on the default DM) – a model with an additive 'offset' between p and c. As we introduced in Chapter 6, to build such additive models, all you need to do is delete the interaction columns from the DM – this 'additive' model is shown below:

B1: intercept	B2: enc grp	Parm	B3: t1	B4: t2	B5: t3	B6: t4	B7: t5	B8: f0
1	1	1:p	1	0	0	0	0	0
1	1	2:p	0	1	0	0	0	0
1	1	3:p	0	0	1	0	0	0
1	1	4:p	0	0	0	1	0	0
1	1	5:p	0	0	0	0	1	0
1	1	6:p	0	0	0	0	0	0
1	0	7:c	0	1	0	0	0	0
1	0	8:c	0	0	1	0	0	0
1	0	9:c	0	0	0	1	0	0
1	0	10:c	0	0	0	0	1	0
1	0	11:c	0	0	0	0	0	0
0	0	12:f0	0	0	0	0	0	1

Remember that this model constrains time-specific estimates of p and c to parallel each other by a constant offset. In effect, this is a 'behavior+time' model. Whether or not this is a 'meaningful' model is up to you.

15.7. Heterogeneity models

If one were to take a strict random sample from a closed population (i.e., such that all individuals have the same probability of being included in the sample), one would expect the estimate of abundance to be unbiased (although a particular estimator might lead to bias under some circumstances). In the case of individual heterogeneity in the probability of encounter, where in the limit each individual i in the population has a unique encounter probability, p_i, negative bias in estimates of abundance is expected. Individuals with high detection probabilities would tend to appear in the encountered sample in greater proportion than they occur in the population (in effect, this is analogous to censoring individuals with low encounter probabilities out of the sample). This results in the estimated average encounter probability of encountered individuals tending to be greater than the true population average.

Thus, the denominator in the canonical estimator for abundance, $M_{t+1}/\hat{p}^*$, is generally biased high in the presence of individual heterogeneity, and the resulting estimate of abundance is biased low. This will be especially true in situations where the overall mean encounter probability is low; when the overall mean encounter probability is high, then even individuals with relatively low encounter probability are likely to be encountered at least once during the sampling period.

Largely because of concerns of the effect of individual encounter heterogeneity on bias in abundance estimate, deriving models for estimating abundance from closed populations that account for individual heterogeneity has been of historical interest, and has generally proven difficult.

There have been 2 fairly distinct, but not mutually exclusive, approaches to development of such intermediate models. The first considers models where the source of the heterogeneity among individuals is assumed *observable*, and can potentially be accounted for by ≥ 1 individual, measurable covariates – some individual trait that influences the detection probability of that individual. The individual covariate approaches have the advantage of potentially informing as to the underlying causes of the heterogeneity in encounter probability, in addition to reducing bias in abundance estimators. **MARK** allows individual heterogeneity to be approximated with finite mixtures (as above) or with individual covariates (using Huggins' conditional likelihood models).

Alternatively, there is a class of models where individual heterogeneity in encounter probability is *unobservable* (i.e., not modeled as a function of ≥ 1 factors or covariates), and is modeled as an individual random effect. Such models are very general because they do not require specification of the possible source(s) of the heterogeneity. Instead, they posit a parametric probability distribution for $\{p_i\}$ (i.e., the set $\{p_i\}$ is a random sample of size N drawn from some probability distribution), and use appropriate methods of parametric inference.

Such unobservable heterogeneity models can be broadly classified in terms of whether the distribution of individuals is modeled as either a discrete- or continuous-mixture, where the population is implicitly a mixture of individuals with different probabilities of encounter. Norris & Pollock (1996) and Pledger (2000, 2005) proposed discrete-mixture models where each individual p_i may belong to one of a discrete set of classes (reviewed by Pledger & Phillpot 2008); because the discrete set of classes is finite, these models are often referred to as finite-mixture models. Alternatively, the mixture distribution can be continuous infinite (Dorazio & Royle 2003). A commonly used distribution is the logit-normal, where individuals p_i are drawn from a normal distribution (on the logit scale) with specified mean μ and variance σ^2, that is, $\text{logit}(p_i) \sim \text{N}(\mu, \sigma^2)$.

MARK allows you to fit a class of models which are parameterized based on what are known as 'finite mixtures'. In these models,

$$p_i = \begin{cases} p_{i,A} & \text{with Pr}(\pi) \\ p_{i,B} & \text{with Pr}(1-\pi), \end{cases}$$

for the case with two mixtures A and B, although the model can be extended to >2 mixtures. As written (above), the parameter π is the probability that the individual occurs in mixture A. For >2 mixtures, additional π parameters must be defined (i.e., π_A, π_B,...), but constrained to sum to 1.* In practice, most data sets generally support no more than 2 mixtures. Note that the π parameter is assumed to be constant over time (i.e., an individual in a given mixture is always in that particular mixture over the sampling period). This has important implications for constructing the DM, which we discuss later.

Alternatively, **MARK** also allows the fitting of a continuous mixture, based on the logit-normal, using numerical integration of individual heterogeneity (modeled as a random effect), using *Gaussian-Hermite*

* In practice, this means that you should use the multinomial logit link function, MLogit, to ensure that the estimates do sum to 1. The MLogit link was introduced in Chapter 10.

quadrature. As discussed by Gimenez & Choquet (2010), integration by Gaussian-Hermite quadrature is very robust under the assumption that the random effect is Gaussian (or nearly so), and computationally is much more efficient than approaches based on Monte Carlo (MCMC) sampling. Further, because Gaussian-Hermite integration can be embedded in a classical likelihood-based modeling framework, we can use established methods for goodness-of-fit testing and model selection to evaluate the relative performance of different heterogeneity models in estimating abundance from closed population encounter data (White & Cooch 2017).

Here, we introduce both the discrete- and continuous-mixture approaches, as applied to closed population abundance estimation.* Although it remains to be determined how well this approach would work if the distribution of encounter rates was strongly asymmetric, the underlying model of normally distributed individual random effects on the logit scale for p provides a more realistic biological model of heterogeneity than discrete-mixture models *when individual heterogeneity is thought to occur over a continuous scale rather than a discrete set of mixtures*.

There are clearly cases, however, where the main source of individual heterogeneity might be better modeled assuming discrete classes (say, in cases where the main source of difference in encounter probability is an underlying discrete attribute, which may not be observable; e.g., sex, in cases where the sex of the organism is not observable given the data. Or situations where the disease 'state' (infected, not infected) cannot be determined easily on encounter. And so on.).

Our purpose here isn't to fully compare and contrast the two approaches in terms of relative bias and precision – it is more than likely that the performance of the two models will differ depending on the true underlying distribution of the heterogeneity (which, of course, is not known). Instead, we focus on the mechanics of the two approaches in **MARK**.

15.7.1. Finite, discrete mixture models

Before we demonstrate the 'mechanics' of fitting finite mixture models to the data, let's first consider the encounter histories (there are 2^k possible encounter histories for a k-occasion study), and their probabilities, for a 4-occasion case for the '**`Full likelihood p and c`**' data type:

history	*cell probability*
`1000`	$p_1(1-c_2)(1-c_3)(1-c_4)$
`0100`	$(1-p_1)p_2(1-c_3)(1-c_4)$
`0010`	$(1-p_1)(1-p_2)p_3(1-c_4)$
`0001`	$(1-p_1)(1-p_2)(1-p_3)p_4$
`1100`	$p_1c_2(1-c_3)(1-c_4)$
`1010`	$p_1(1-c_2)c_3(1-c_4)$
`1001`	$p_1(1-c_2)(1-c_3)c_4$
`1110`	$p_1c_2c_3(1-c_4)$

history	*cell probability*
`1101`	$p_1c_2(1-c_3)c_4$
`1011`	$p_1(1-c_2)c_3c_4$
`0110`	$(1-p_1)p_2c_3(1-c_4)$
`0101`	$(1-p_1)p_2(1-c_3)c_4$
`0011`	$(1-p_1)(1-p_2)p_3c_4$
`0111`	$(1-p_1)p_2c_3c_4$
`1111`	$p_1c_2c_3c_4$
`0000`	$(1-p_1)(1-p_2)(1-p_3)(1-p_4)$

* In fact, 'finite mixture models' and 'individual random effects' models (based on Gaussian-Hermite quadrature) are available for a number of additional data types in **MARK** – see Addendum 2 to this chapter.

If we want to add a *finite mixture* to the cell probability (i.e., for '**`Full Likelihood Heterogeneity with pi, p, and c`**' data type, with two mixtures), we modify the probability expressions as follows:

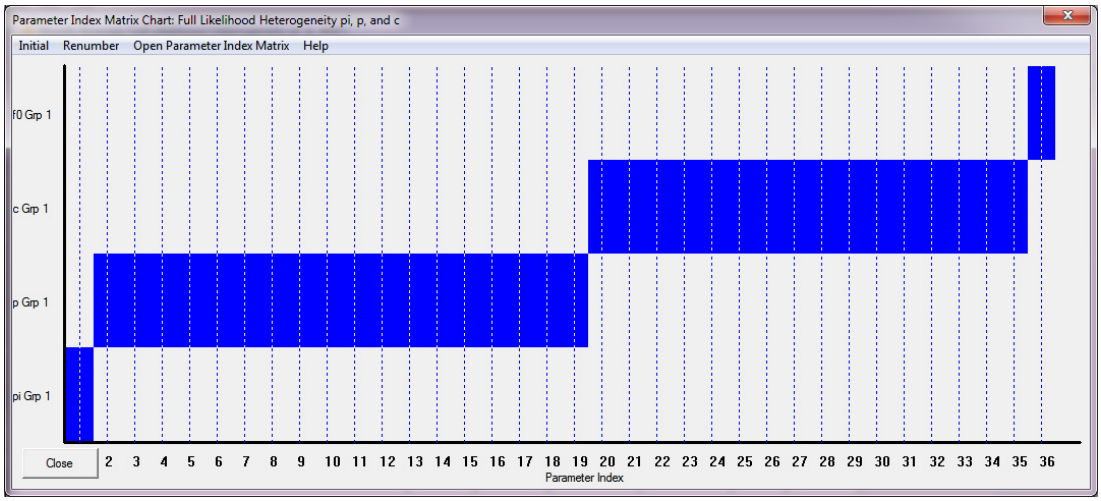
history	*cell probability*
1000	$\sum_{a=1}^{2}(\pi_a p_{a1}(1-c_{a2})(1-c_{a3})(1-c_{a4}))$
0100	$\sum_{a=1}^{2}(\pi_a (1-p_{a1})p_{a2}(1-c_{a3})(1-c_{a4}))$
0010	$\sum_{a=1}^{2}(\pi_a (1-p_{a1})(1-p_{a2})p_{a3}(1-c_{a4}))$
0001	$\sum_{a=1}^{2}(\pi_a (1-p_{a1})(1-p_{a2})(1-p_{a3})p_{a4})$
1100	$\sum_{a=1}^{2}(\pi_a p_{a1}c_{a2}(1-c_{a3})(1-c_{a4}))$
1010	$\sum_{a=1}^{2}(\pi_a p_{a1}(1-c_{a2})c_{a3}(1-c_{a4}))$
⋮	⋮

Note: the finite mixture models have a separate set of p's and c's for each mixture.

We will demonstrate the fitting of finite mixture ('heterogeneity') models to a new sample data set (**`mixed_closed1.inp`**). These data were simulated assuming a finite mixture (i.e., heterogeneity) using the generating model $\{f_0, \pi, p(\cdot) = c(\cdot) = constant\}$ – 9 occasions, 2 mixtures, $N = 2{,}000$, $\pi = 0.40$, and $p_{\pi_A} = 0.25, p_{\pi_B} = 0.75$. In other words, two mixtures, one with an encounter probability of $p = 0.25$, the other with an encounter probability of $p = 0.75$, with the probability of being in the first mixture $\pi = 0.40$.

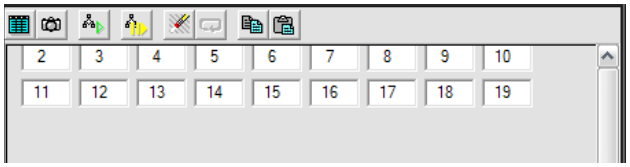
Start a new project, select the input data file, set the number of occasions to 9, and specify the '**`Full Likelihood Heterogeneity with pi, p, and c`**' data type. Once we've selected a closed data type with heterogeneity, you'll see that an option to specify the number of mixtures is now available in the '**`specification window`**' (lower-right side). We'll use 2 mixtures for this example.

Once you have specified the number of mixtures, open the PIM chart for this data type (when you switch data types, the underlying model will default to a general time-specific model):



Notice that there are twice as many p's and c's as you might have expected given there are 9 occasions represented in the data. This increase represents the parameters for each of the two mixture groups.

The PIM for the p's now has two rows defaulting to parameters 2 → 10 and 11 → 19.



Parameters 2 → 10 represent the p's for the first mixture, and 11 → 19 the p's for the second mixture. It becomes more important with the mixture models to keep track of which occasion each c corresponds to because now *both* parameter 2 and 11 relate to occasion 1 which has no corresponding c parameter.

We'll follow the approach used in the preceding section, by first fitting a general model based on PIMs to the data. You might consider model $\{f_0, \pi, p(t), c(t)\}$ as a reasonable starting model. However, there are two problems with using this as a general, starting model. First, you'll recall that there are estimation problems (in general) for a closed abundance model where both p and c are fully time-dependent. Normally, we need to impose some sort of constraint to achieve identifiability. However, even if we do so, there is an additional, more subtle problem here – recall we are fitting a heterogeneity 'mixture' model, where the parameter π is assumed to be constant over time. As such, there is no interaction among mixture groups possible over time. Such an interaction would imply time-varying π. Thus, the most general *meaningful* model we could fit would be an additive model, with additivity between the mixture groups, and interaction of p and c within a given mixture group. Recall that we can't construct this model using PIMs – building an additive model requires use of the design matrix.

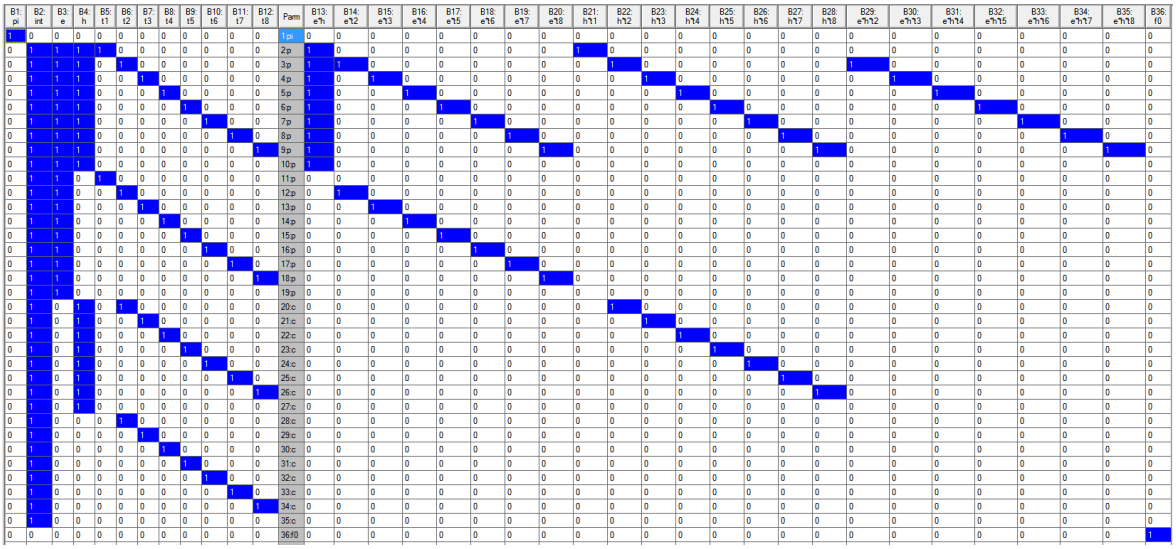
We see from the PIM chart (shown on the previous page) that the default model structure has 36 columns (i.e., 36 total structural parameters). *Note*: if you select '**`Design | Full`**', **MARK** will respond with an error message, telling you it can't build a default fully time-dependent DM. Basically, for heterogeneity models, you'll need to build the DM by hand – meaning, starting with a reduced DM. So, we select '**`Design | Reduced`**', and keep the default 36 columns.

Now, how do we build the DM corresponding to the PIM chart on the preceding page? We start by first writing out the linear model. To do so, we need to first consider the 'groups' in our model. Here, we have in fact 2 groups: (i) the putative 'encounter group' (`ENC`) representing the p and c parameters (as we saw in the preceding section), and (ii) a new 'heterogeneity' group (`HET`) representing what we might for convenience think of as the 'π' and '$1 - \pi$' groups. So, 2 ' encounter groups', 2 'heterogeneity groups', 9 occasions (`TIME`), and the various *plausible* interactions among them.

Here is our linear model (which we write only in terms of parameters p and c. Parameters π and f_0 are simple scalar constants):

$$
\begin{aligned}
f &= \texttt{ENC+HET+TIME+(ENC.HET)+(ENC.TIME)+(HET.TIME)+(ENC.HET.TIME)} \\
&= \beta_1 \\
&\quad + \beta_2(\texttt{ENC}) \\
&\quad + \beta_3(\texttt{HET}) \\
&\quad + \beta_4(\texttt{ENC.HET}) \\
&\quad + \beta_5(\texttt{T}_1) + \beta_6(\texttt{T}_2) + \beta_7(\texttt{T}_3) + \beta_8(\texttt{T}_4) + \beta_9(\texttt{T}_5) + \beta_{10}(\texttt{T}_6) + \beta_{11}(\texttt{T}_7) + \beta_{12}(\texttt{T}_8) \\
&\quad + \beta_{13}(\texttt{HET.T}_1) + \beta_{14}(\texttt{HET.T}_2) + \beta_{15}(\texttt{HET.T}_3) + \cdots + \beta_{20}(\texttt{HET.T}_8) \\
&\quad + \beta_{21}(\texttt{ENC.T}_2) + \beta_{22}(\texttt{ENC.T}_3) + \beta_{23}(\texttt{ENC.T}_4) + \cdots + \beta_{27}(\texttt{ENC.T}_8) \\
&\quad + \beta_{28}(\texttt{ENC.HET.T}_2) + \beta_{29}(\texttt{ENC.HET.T}_3) + \cdots + \beta_{34}(\texttt{ENC.HET.T}_8)
\end{aligned}
$$

So, 34 parameters in this linear model. If we add 2 (one each for π and N, respectively), we get 36 total. The design matrix corresponding to this model is shown on the next page (although you might need to put on some 'special reading glasses' to see it all):



Now, some important things to note from the linear model and corresponding DM. First, the two 'groups' (encounter and heterogeneity; `ENC` and `HET`, respectively) are each coded by a single column (single β) – columns `B3` for `ENC` and `B4` for `HET`. 9 sampling occasions, so 8 columns for time (`B5` $\rightarrow$ `B12`). The remaining columns code for the two-way interactions between `ENC (E)`, `HET (H)` and time `(T)`, and the three-way interaction (`H.E.T`$_x$).

Now, if you run this model constructed using the DM, you'll see that the model deviance is identical to the model constructed using PIMs (indicating that our DM is correct). However, if you look at the parameter estimates, you'll quickly notice that, as expected, quite a few of the parameters aren't identifiable. In particular, the final $\hat{p}$ estimates for the two mixture groups have problems, and the derived estimate of $\hat{N}$ is simply M_{t+1} (the SE of the abundance estimate is clearly wrong).

Why the problems? Simple – despite the fact we have 2 mixture groups, this is still model $\{p(t), c(t)\}$, which we know is not identifiable – and thus, is not a useful model to fit to the data – without constraints. One possible constraint is to model p and c as additive functions of each other. How can we modify the DM to apply this constraint?

Simple – by eliminating the interactions between `ENC` and `TIME`. In other words, deleting columns `B14` $\rightarrow$ `B20` (coding for the interaction of `ENC` and `TIME`), and columns `B29` $\rightarrow$ `B35` (coding for the 3-way interaction of `HET`, `ENC`, and `TIME`) from the DM shown on the previous page. This model allows time variation, behavioral variation and individual heterogeneity in capture probability, yet does so in an efficient and parsimonious (and estimable) manner.

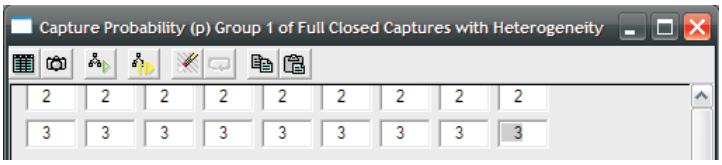
We can use this DM to create additional, reduced parameter models. For example, we could build model $\{f_0, p_a(t) = c_a(t) = p_b(t) + z = c_b(t) + z\}$ representing capture probability varying through time and additive difference between mixture groups, but with no interaction between p and c over time (no behavior effect). We do this simply by deleting the `ENC` column from the DM.

As a final test – how do we modify the DM to match the true generating model, which for these data was model $\{f_0, \pi, p_A = c_A, p_B = c_B\}$? To build this model from our DM, we simply delete (i) all the time columns, (ii) any interactions with time, and (iii) the encounter group column (`ENC`). We

delete the encounter group column because we're setting $p = c$. We retain the heterogeneity (mixture) group column (HET) since we want to allow for the possibility that encounter probability differs between mixtures (which of course is logically necessary for a mixture model!).

Both the real and derived parameter estimates ($\hat{\pi} = 0.607, \hat{p}_{\pi_A} = 0.250, \hat{p}_{\pi_B} = 0.754, \hat{N} = 1{,}995.494$) are quite close to the true parameter values used in the generating model. [But, what about $\hat{\pi}$? The true value we used in the simulation was $\pi = 0.40$. The estimated value $\hat{\pi} = 0.607$ is simply the complement.]

We can confirm that this corresponds to model $\{f_0, \pi, p_A = c_A, p_B = c_B\}$ by comparing the model fit with that from the PIM-based equivalent. We can do this in one of two ways – we can either (i) stay within the '**Full Likelihood Heterogeneity with pi, p, and c** ' data type, and build the appropriate PIMs, or (ii) change data type to the simpler '**Full Likelihood Heterogeneity Pi and p**', which defaults to our desired model. If we take the first approach, all we need to do is modify the two encounter probability PIMs as follows, for p and c, respectively, so they both have the following structure:



So, constant over time and no behavior effect (i.e., $p = c$) within mixture group. If you run this model, you'll see that it yields an identical model deviance (555.1792) as the model built earlier using the modified DM.

What about changing data types? Well, you might think that you need to restart **MARK**, and begin a new project after first specifying the new data type. In fact, you don't need to – you can simply 'tell' **MARK** that you want to switch data types (something **MARK** lets you do within certain types of models – in this instance, closed population abundance estimators). All you need to do is select '**PIM | change data type**' on the main menu bar, and then select '**Full Likelihood Heterogeneity Pi and p**' from the resulting popup window. As noted earlier, the default model for this data type is the model we're after – it is simply a reduced parameter version of the full model.

Interpreting $\hat{\pi}$ from finite mixture models

So, you do an analysis using a closed population heterogeneity abundance model, based on finite mixtures, and derive an estimate of $\hat{\pi}$. Perhaps you've built several such models, and have a model averaged estimate of $\bar{\hat{\pi}}$. So, what do you 'say' about this estimate of $\hat{\pi}$?

Easy answer – *generally nothing*. The estimate of $\hat{\pi}$ is based on fitting a finite mixture model, with a (typically small) number of *discrete* states. When we simulated such data (above), we used a discrete simulation approach – we simply imagined a population where 40% of the individuals had one particular detection probability, and 60% had a different encounter probability. In that case, because the distribution of individuals in the simulated population was in fact discrete, then the real estimate of $\hat{\pi}$ reflected the true generating parameter.

However, if in fact the variation in detection probability was (say) continuous, then the estimate of $\hat{\pi}$ reflects a 'best estimate' as to where a discrete 'breakpoint' might be (breaking the data into a set of discrete, finite mixtures). Such an estimate is not interpretable, by and large. Our general advice is to avoid *post hoc* story-telling with respect to $\hat{\pi}$, no matter how tempting (or satisfying) the story might seem.

15.7.2. Continuous mixture models using numerical integration

Now, we'll consider models where we assume that the individual heterogeneity is continuous logit-normal. The basic ideas underlying continuous mixture models are relatively simple. First, we assume a population where individual encounter probabilities were randomly drawn from a logit-normal distribution, specified by a known μ_p and σ_p^2. The continuous mixture model is implemented in **MARK** for using the Huggins estimator, extended by including an individual random effect for the encounter probability (p_{ik}) of each individual i constant across occasions $k = 1, \ldots, t$ on the logit scale following McClintock *et al*. (2009) (see also Chapter 19), Gimenez & Choquet (2010), and White & Cooch (2017):

$$\text{logit}(p_{ik}) = \beta_k + \epsilon_i,$$

with β_k a fixed effect modeling time, and ϵ_i a normally distributed random effect with mean zero and unknown variance σ_p^2. Hence

$$p_{ik} = \frac{1}{1 + exp(-(\beta_k + \sigma_p Z_i))},$$

where $Z_i \sim N(0., 1)$. Therefore, individual i on occasion k has the probability of being encountered

$$p_{ik} = \int_{-\infty}^{+\infty} \frac{1}{1 + exp\left(-(\beta_k + \sigma_p Z_i)\right)} \varphi(Z_i) dZ_i,$$

where $\varphi(Z_i)$ is the probability density function of the standard normal distribution. The estimate of population abundance, $\hat{N}$, is obtained following Huggins (1989) as the summation across animals encountered ≥ 1 time,

$$\hat{N} = \sum_{i=1}^{M_{t+1}} \left(\frac{1}{p_i^*}\right),$$

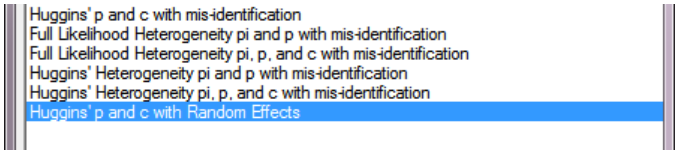
where

$$p_i^* = 1 - \int_{-\infty}^{+\infty} \prod_{i=1}^{k} \left(1 - \frac{1}{1 + \exp\left(-(\beta_k + \sigma_p Z_i)\right)}\right) \varphi(Z_i) dZ_i.$$

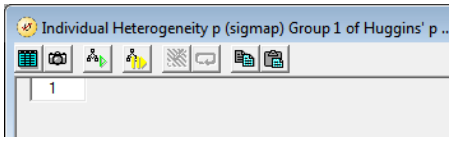
Because this integral does not have a closed form, the likelihood must be integrated numerically – in program **MARK**, this is accomplished using *Gaussian-Hermite quadrature* (McClintock *et al*. 2009, Gimenez & Choquet 2010, White & Cooch 2017).

To demonstrate the mechanics, we'll start with the same simulated data set we used in the preceding section where we introduced discrete-mixtures (**`mixed_closed1.inp`**). Recall that these encounter data were simulated using the generating model $\{f_0, \pi, p(\cdot) = c(\cdot)\}$ – 9 occasions, 2 mixtures, N = 2,000, $\pi = 0.40$, and $p_{\pi_A} = 0.25$, $p_{\pi_B} = 0.75$. In other words, the data do in fact consist of two discrete classes of individuals, one with an encounter probability of $p = 0.25$, the other with an encounter probability of $p = 0.75$, with the probability of being in the first mixture $\pi = 0.40$. With a bit of thought, you should realize that this data set is not symmetrical around some 'mean' encounter probability.

Start a new project – 9 occasions. We'll now select '**`Closed captures | Huggins' p and c with Random Effects`**' as the data type:

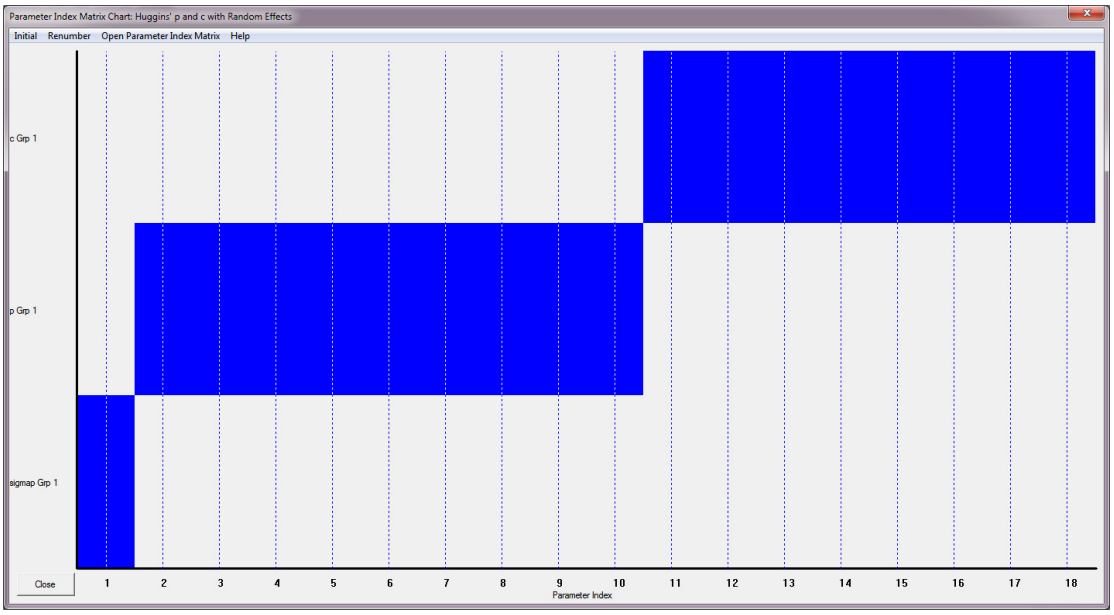


Once you click the '**OK**' button, **MARK** will present you with the PIM for the first parameter – in this case, the parameter is σ_p ('`sigmap`'), the standard deviation of the continuous distribution of the individual variation in encounter probability p:



As you can see, there is only a single 'cell' in the PIM – meaning, we assume that σ_p is constant over all sampling intervals. The is analogous to assuming the π is a constant over time in discrete mixture models.

If we look at the default PIM chart

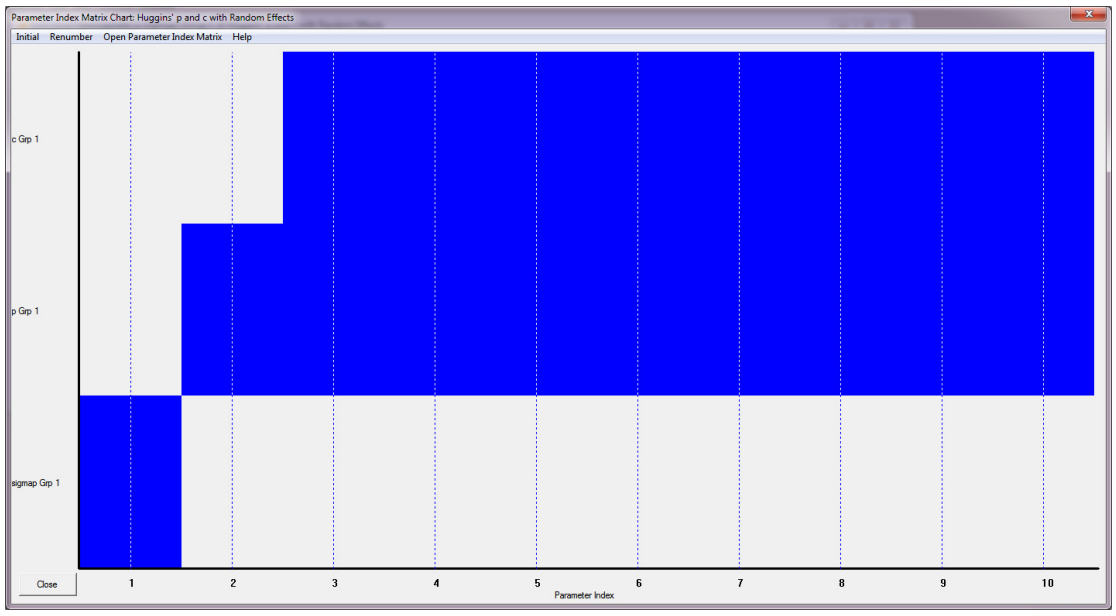


we see that we have the standard Huggins p and c parameters, and only a single 'blue box' for the new '`sigmap`' parameter. Since σ_p is a single parameter, and is assumed to be a constant over all sampling intervals, then the structure of the (linear) models used for p and c is identical to standard Huggins models. So, we could very easily fit models $\{\sigma_p, p_\cdot = c_\cdot\}, \{\sigma_p, p_\cdot, c_\cdot\}, \{\sigma_p, p_t = c_t\}$ (i.e., models M_0, M_b, M_t, but with an additional parameter, σ_p). For example, for model $\{\sigma_p, p_t = c_t\}$, we could simply modify the PIM chart as shown at the top of the next page.

Now, if you set this model up, and run it, the first thing you'll notice is that the model takes a lot longer to run that does a simple $\{p_t = c_t\}$ model. Numerically integrating out the individual heterogeneity as an individual random effect takes some computation time.

We'll run this model, and models $\{\sigma_p, p_\cdot = c_\cdot\}$ and $\{\sigma_p, p_\cdot, c_\cdot\}$ as well, and add the results to the browser:

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance	-2Log(L)
{sigmap,p(.)=c(.)}	21948.6987	0.0000	0.57194	1.0000	2	26194.8960	21944.6980
{sigmap,p(.),c(.)}	21949.4524	0.7537	0.39236	0.6860	3	26193.6500	21943.4510
{sigmap,p(t)=c(t)}	21954.2466	5.5479	0.03570	0.0624	10	26184.4320	21934.2340



If we look at the real parameter estimates, for, say, model $\{\sigma_p, p_t = c_t\}$, we see estimates for $p_t = c_t$, and for σ_p:

mixture using GH

Real Function Parameters of {sigmap,p(t)=c(t)}

Parameter	Estimate	Standard Error	95% Confidence Interval Lower	Upper
1:sigmap	1.3168239	0.0420543	1.2369439	1.4018625
2:p	0.5631512	0.0154952	0.5325810	0.5932489
3:p	0.5473311	0.0155815	0.5166513	0.5776554
4:p	0.5605177	0.0155113	0.5299259	0.5906565
5:p	0.5598591	0.0155152	0.5292622	0.5900079
6:p	0.5592005	0.0155191	0.5285984	0.5893593
7:p	0.5407277	0.0156102	0.5100165	0.5711327
8:p	0.5559059	0.0155378	0.5252795	0.5861133
9:p	0.5420488	0.0156048	0.5113432	0.5724383
10:p	0.5893904	0.0152981	0.5591095	0.6190087

Of course, over a candidate set of models, we might be interested in model averaged estimates of our real parameters, but for closed population abundance models, our usual interest concerns the derived estimate of abundance, $\hat{N}$.

mixture using GH
Estimates only for data type Huggins' p and c with Random Effects

Model	Derived Parameter N Population Size 1 Weight	Estimate	Standard Error
{sigmap,p(.)=c(.)}	0.57194	2031.2477443	14.0220548
{sigmap,p(.),c(.)}	0.39236	2025.4524572	14.3221365
{sigmap,p(t)=c(t)}	0.03570	2031.2485045	14.0229500
Weighted Average		2028.9739212	14.1398274
Unconditional SE			14.4209405

95% CI for Weighted Average Estimate is 2000.7088779 to 2057.2389646
Percent of Variation Attributable to Model Variation is 3.86%

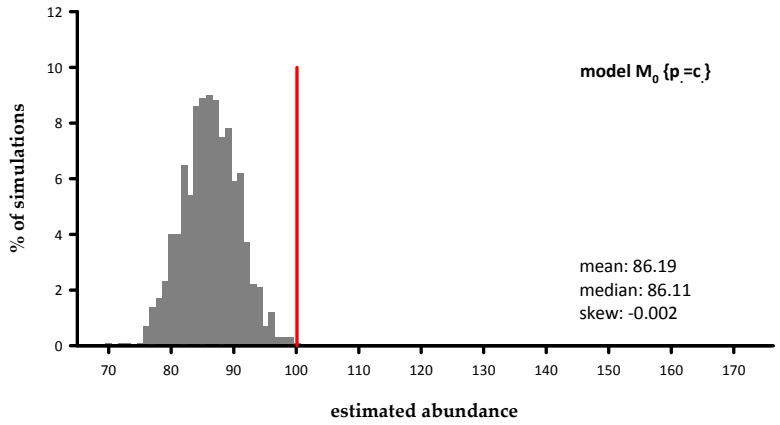
Here we see the model-averaged estimate of $\bar{\hat{N}} = 2028.974$.

How does this compare to the model-average estimate if we ignore heterogeneity, or if we use a discrete-mixture approach? If you fit models $\{p_\cdot = c_\cdot\}$, $\{p_\cdot, c_\cdot\}$, and $\{p_t = c_t\}$ (i.e., models M_0, M_b, M_t,

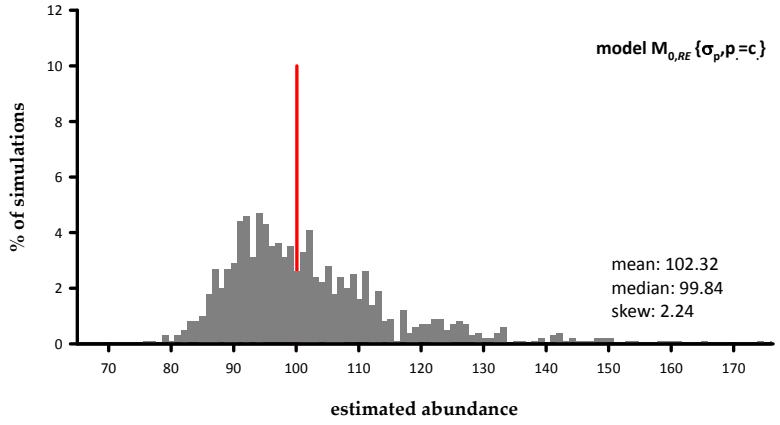
without the additional parameter, σ_p), the model averaged estimate is $\bar{\hat{N}} = 1941.475$, which is quite a bit different than the estimate from models fit using the individual random effect approach. If instead we fit mixture models $\{\pi, p_\cdot = c_\cdot\}$, $\{\pi, p_\cdot, c_\cdot\}$, and $\{\pi, p_t = c_t\}$ (i.e., models M_0, M_b, M_t, but with an additional mixture parameter, π), the model averaged estimate is $\bar{\hat{N}} = 1994.152$, which is only a bit different than the models fit without any structure to account for heterogeneity. Recall that true $N = 2{,}000$ for this data set, so it appears as if, in this case, modeling heterogeneity as an individual random effect has performed a bit less well than either using finite mixtures, or ignoring the heterogeneity altogether.

To emphasize the fact that results of using different approaches to heterogeneity can be 'twitchy' (from the Latin), here are some summary results from a large series of simulations (1,000) with true $N = 100$, $\sigma_p = 1.0$, $p_{\text{real}} = 0.35$, $k = 5$ occasions, where the encounter data were generated under true model $M_{0,RE}$ (i.e., $p_\cdot = c_\cdot$, with logit-normal variation in p_i for each individual i). To these simulated data, we fit 3 models to the data: $\{p_\cdot = c_\cdot\}$, $\{\sigma_p, p_\cdot = c_\cdot\}$, and $\{\pi, p_\cdot = c_\cdot\}$.

For model $\{p_\cdot = c_\cdot\}$, we see (below) that the distribution of estimates of N from each simulation (and various moments of the distribution of the estimates), as negatively biased with respect to the true value of $N = 100$ (vertical red line). This is entirely as expected:

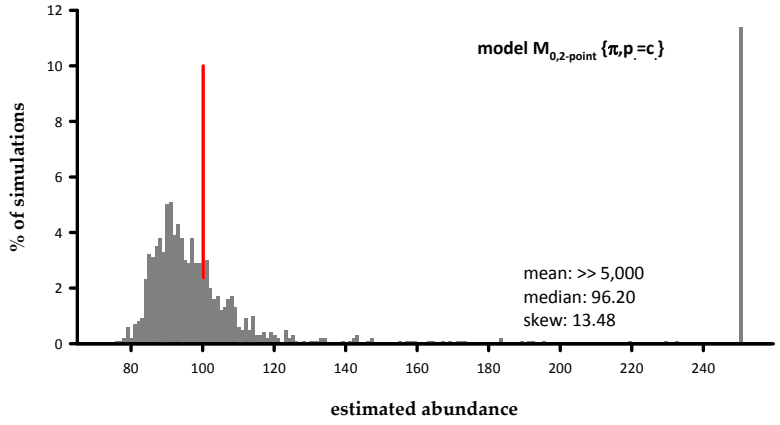


For model $\{\sigma_p, p_\cdot = c_\cdot\}$, where we use GH quadrature to integrate out the individual heterogeneity,



both the mean and median are unbiased (meaning, the expectation from this model is unbiased with respect to true $N = 100$).

On the other hand, if we fit a 2-point discrete mixture model $\{\pi, p_\cdot = c_\cdot\}$,



the mean of the distribution is impressively 'silly': $>> 5,000$, entirely driven by a few very large, and very spurious estimates for N (shown in aggregate by the large frequency bar at ~ 250). While there are approaches for handling these spurious values (say, by penalizing the likelihood), they don't always seem to work (especially for low p, and low number of sampling occasions). But the standard finite-mixture formulation clearly has some issues for these (and potentially other) situations.

begin sidebar

A convenient short-cut: pre-defined closed population models

It is fair to argue that the main objective for fitting closed population abundance models is to come up with the best estimate of abundance. Generally, this will involve averaging over multiple models (model-averaging for closed population abundance estimation is covered in section 14.10).

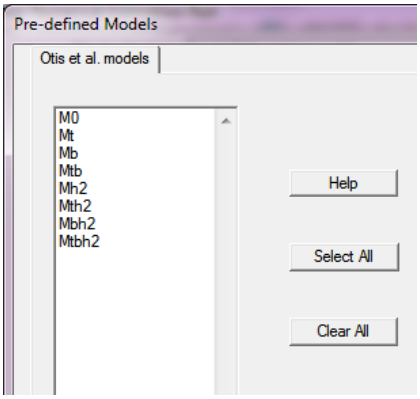
As part of this process, we will typically fit a candidate set of approximating models to the data – using either the full or conditional (Huggins) likelihood approach. In many cases, the model set will consist at minimum of what are commonly referred to as the 'Otis models' – described by Otis *et al*. (1978). In general this minimal model set consists of some or all of the following 8 models (for either the full or conditional likelihood approach):

Otis notation	*Expanded notation*	K
M_0	$\{f_0, p(\cdot) = c(\cdot)\}$	2
M_t	$\{f_0, p(t) = c(t)\}$	$t+1$
M_b	$\{f_0, p(\cdot), c(\cdot)\}$	3
M_{tb}	$\{f_0, p(t) = c(t) + z\}$	$t+2$
M_{h2}	$\{f_0, p_a(\cdot) = c_a(\cdot), p_b(\cdot) = c_b(\cdot), \pi\}$	4
M_{th2}	$\{f_0, p_a(\cdot) = c_a(\cdot) + t, p_b(\cdot) = c_b(\cdot) + t, \pi\}$	$t+3$
M_{bh2}	$\{f_0, p_a(\cdot) = c_a(\cdot) + z, p_b(\cdot) = c_b(\cdot) + z, \pi\}$	5
M_{tbh2}	$\{f_0, p_a(\cdot) = c_a(\cdot) + t + z, p_b(\cdot) = c_b(\cdot) + t + z, \pi\}$	$t+4$

At this point in the chapter, building these models 'by hand', using a design matrix, is not overly difficult. But, it can be somewhat time-consuming.

However, there is a time-saving option in **MARK** which will let you build all or some of these 8 models as 'pre-defined' models. From the browser, simply select '**`Run | Pre-defined models`**'. You will then be presented with the '**`Setup Numerical Estimation Run`**' window. Now, though, instead of a button for 'fixing parameters', you'll see a button to '**`Select Models`**'.

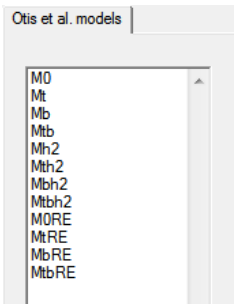
If you click this button, you will be presented with the following:



Note that the Otis model naming conventions are used (while perhaps not particularly informative of the underlying model structure, they are compact). All you need to do is select the models you'd like to fit. Although not indicated explicitly, all of the models are constructed using a design matrix (for some models, especially the heterogeneity models, this point might be implicit).

What is not immediately obvious, though, is that if you pick all 8 models, then **MARK** will fit all 8 models, even if the underlying data types when you started the analysis seems different than one of the pre-defined models. For example, suppose you start an analysis using the '**`Full likelihood p and c`**' data type. Recall that for this data type, the 3 structural parameters are: p, c, f_0. There is no π parameter for finite mixture heterogeneity models. Nonetheless, if you include heterogeneity models from the pre-defined models list (e.g., model M_{th2}), then **MARK** will go ahead and (i) internally change the data type from '**`Full likelihood with p and c`**' to '**`Full likelihood heterogeneity with pi, p and c`**', and then (ii) fit the pre-defined model to the encounter data.

Related to the preceding, if you want unconditional (Huggins) data types, then you have to have set the data type to Huggins, and vice versa for full likelihood models. The PIM structure at the time you hit the '**`Run | Pre-defined Models`**' dictates whether you get the full or Huggins likelihoods. For example, if you have set the data type to Huggins conditional likelihood, then '**`Run | Pre-defined Models`**' present the same 8 models as for the full likelihood approach, plus 4 additional models including individual random effects:



While in general pre-defined models should be used cautiously – since there isn't a lot of 'thinking'

involved with fitting them to the data – being able to build some of the canonical closed population abundance models with only a few clicks can be a real time-saver.

end sidebar

Closing comment: individual heterogeneity – the bane of abundance estimation

It is perhaps reasonable to speculate that in preceding example, where the underlying heterogeneity simulated in the data was truly continuous logit-normal, that GH integration will perform particularly well, better than a finite-mixture approach which models heterogeneity as if there were in fact discrete 'groups' of individuals. However, in the absence of strong prior information (or, 'expectation') about the form of the heterogeneity (at the least, continuous versus discrete), it is probably somewhat misleading to imply that one approach or the other will be the most robust in modeling heterogeneity.

While the approaches discussed in the preceding can be effective, the single best way to minimize the bias caused by individual heterogeneity is to get p as high as possible – the 'big law' of capture-recapture design. When p is high there is little room for variation and little chance that an individual is not detected.

Bill Link (2003, 2004) demonstrated that different models of the form of individual heterogeneity can lead to very different estimates of abundance and fit the data equally well. The magnitude of the differences in abundance estimates is related to p; when p is small the differences can be large. Therefore, to have much hope of estimating abundance with little bias, capture probability must be relatively high. In addition, the primary issue with using the individual random effects estimator (and likely discrete-mixture approaches as well) in adequately fitting complex models is obtaining an adequate number of occasions to be able to estimate σ_p (or π) with any precision. A lower level of > 5 occasions is likely necessary to achieve reasonable performance for either the continuous- or discrete-mixture approaches (White & Cooch 2017).

Individual heterogeneity is a common source of bias, typically causing capture-mark-reencounter estimates of population abundance to be biased low. Continuous- and finite-mixture models may be an effective approach to eliminate this bias when an adequate number of capture occasions are available and detection probabilities are relatively high. To have much hope of estimating abundance with little bias, we reiterate the 'big law' – encounter probability must be relatively high.

In sampling situations with low encounter probabilities ($p \leq 0.2$), and a low number of samples ($K \leq 5$), the investigator will need to be aware of the potential for biased estimates, and evaluate whether or not the direction of bias (which reflects the estimator – generally negative for models that do not account for heterogeneity, and potentially positive for models that do account for heterogeneity) is important in the applied context. For example, for management of small, threatened populations, the potential ramifications of under-estimating (negative bias) or over-estimating (positive bias) abundance are clearly of some importance.

15.8. Misidentification models

The likelihoods and cell probabilities get more complicated when we want to include the possibility of *misidentification* into the cell probabilities. In order to do this we must assume that (i) an individual encountered more than once is correctly identified (i.e., individuals captured on multiple occasions are correctly identified – owing to the greater amount of information gathered on which to base the identification), and (ii) individuals encountered only once may or may not be correctly identified.

First, we consider the closed capture cell probabilities without finite mixtures. We will add the possibility of *misidentification* (where α is the probability of correctly identifying the individual) to

the probabilities for a 4-occasion full likelihood closed population capture-recapture model:

history	*cell probability*
1000	$p_1\alpha(1-c_2)(1-c_3)(1-c_4)+p_1(1-\alpha)$
0100	$(1-p_1)\left[p_2\alpha(1-c_3)(1-c_4)+p_2(1-\alpha)\right]$
0010	$(1-p_1)(1-p_2)\left[p_3\alpha(1-c_4)+p_3(1-\alpha)\right]$
0001	$(1-p_1)(1-p_2)(1-p_3)\left[p_4\alpha+p_4(1-\alpha)\right]$
1100	$p_1\alpha c_2(1-c_3)(1-c_4)$
1010	$p_1\alpha(1-c_2)c_3(1-c_4)$
1001	$p_1\alpha(1-c_2)(1-c_3)c_4$
1110	$p_1\alpha c_2c_3(1-c_4)$
1101	$p_1\alpha c_2(1-c_3)c_4$
1011	$p_1\alpha(1-c_2)c_3c_4$
⋮	⋮

In the encounter histories for individuals encountered only once their probability expression is a summation across the two possible ways the history could have occurred; for example, consider history '0100'; captured for the first time, marked and released alive at occasion 2. Conditional on being alive and in the sample (i.e., available for capture) over the entire sampling period, then the probability of observing encounter history '0100' is $(1-p_1)$ (the probability of not being captured at the first occasion), times the sum of (1) the probability the individual was correctly identified and not seen again ($p_2\alpha(1-c_3)(1-c_4)$, or (2) the individual was misidentified and therefore unable to be seen again $p_2(1-\alpha)$.

When misidentification occurs, the constraint that $\hat{N} \geq M_{t+1}$ no longer holds. It is possible that enough animals are misidentified such that the number detected is greater than the number that actually exist in the population. Second, this increase in the numbers of animals supposedly encountered causes the estimated probability of detection to be smaller than it should be. The effect of these two factors is to cause the estimated abundance $\hat{N}$ to be too high.

To account for these problems, the sum $\hat{f}_0 + M_{t+1}$ must be adjusted for mis-identification error, $\hat{\alpha}$:

$$\hat{N} = \hat{\alpha}\left(\hat{f}_0 + M_{t+1}\right).$$

Therefore, in these models where misidentification is possible **MARK** presents $\hat{f}_0$ in the real parameter output and $\hat{N}$ in the derived parameter output as it is a function of more than one parameter.

15.8.1. Joint heterogeneity and misidentification models

Both the simple and complex heterogeneity models are available for the misidentification closed capture models (i.e., they are available data types). However, incorporation of both misidentification and heterogeneity typically leads to inconclusive results, in that misidentification is somewhat (almost totally) confounded with heterogeneity. Intuitively, misidentification is detected by too many animals only appearing once in the encounter histories. Thus, a large amount of individual heterogeneity may appear as misidentification, and vice versa, misidentification may appear as individual heterogeneity.

So, you can build models with both heterogeneity and misidentification, but there is a very good chance you won't be able to do much with the results.

Claudendo dicta: The issue of 'misidentification' in the broad sense has been largely 'solved' in an important, albeit non-trivial paper by Bill Link and colleagues (Link *et al*. 2010). The methods described in this paper are not at present implemented in **MARK**. Until then, the misidentification methods as available in **MARK** (as described above) might be helpful, if the level of misdentification rate is low.

15.9. Goodness-of-fit

In general, testing model fit in the closed-population capture-recapture models remains an unresolved issue, even more so than in other capture-recapture model types. A central component of the problem stems from the fact that there often is no unique way to compute a saturated model. If one was only concerned about time variation in capture probability, then goodness-of-fit is fairly straightforward. When individual heterogeneity is added into the problem there is an infinite set of possible models for heterogeneity. Thus, no unique goodness-of-fit exists.

In past, several tests of model assumptions have been developed for the closed-population capture-recapture models (Otis *et al*. 1978: 50-67, White *et al*. 1982: 77-79). The seven tests examine the fit of specific model forms relative to other specific models or vague alternatives (i.e., the model fails to fit for unspecified reasons). These tests are available in **MARK** through **CAPTURE** by selecting the '**Appropriate**' check box in the **CAPTURE** window. The tests were developed largely as a means of model selection in the absence of another method. Now that **MARK** employs AIC_c as a selection criterion and that it has been shown the model averaged estimates of N have better properties than single-model estimates (Stanley & Burnham 1998), the tests of Otis *et al*. (1978) have fallen out of use.

Typically, estimating over- dispersion (c) from the observed number of individuals associated with each possible encounter history is complicated by the large number of encounter histories with very low expected frequencies, especially when the average encounter probability is low. Commonly, an estimate of overdispersion is based on Pearson's χ^2 lack-of-fit statistic.

Recently, David Fletcher (Fletcher 2012) and colleagues (Afroz *et al*. 2020) have proposed a new approach which appears to work very well, at least for models (with or without heterogeneity), based on the Huggins conditional likelihood:

$$\hat{c} = \frac{\hat{c}_x}{\bar{r}}, \text{where } \bar{r} = \frac{1}{H}\sum_{i=1}^{H}\frac{y_i}{\hat{\eta}_i}.$$

Here, $\hat{c}_x$ is the estimator of overdispersion based on the Pearson χ^2 statistic (i.e., the Pearson χ^2 statistic divided by the degrees of freedom, where σ_p is included in the parameter count for the random effects models because it is an estimated parameter – D. J. Fletcher, University of Otago, *personal communication*), y_i and η_i are the observed and expected number of individuals with encounter history i, and $H = 2^t - 1$ is the number of observable histories over t occasions. One of the problems with using Pearson's statistic for sparse data is that the ith term involves dividing by η_i , which will often be very small. The new estimator makes an allowance for this because the ith term in the denominator also involves dividing by $\hat{\eta}_i$. Simulations suggest that this new estimator also performs better than those based on the deviance.

White & Cooch (2017) showed that the Fletcher $\hat{c}$ showed very good potential to detect individual heterogeneity, which would generally be of some importance in the context of abundance estimation. There are a couple of issues which you need to take into account in applying this approach. First, losses on capture or 'dots' in the encounter history will create encounter histories that are not considered in the total number of possible encounter histories. That is, the total number of possible encounter histories is based on no missing data. Second, parameter values that cause a reduction in the total number of encounter histories will bias the $\hat{c}$ estimate. Examples of such reductions are an occasion with $p = 0$, or

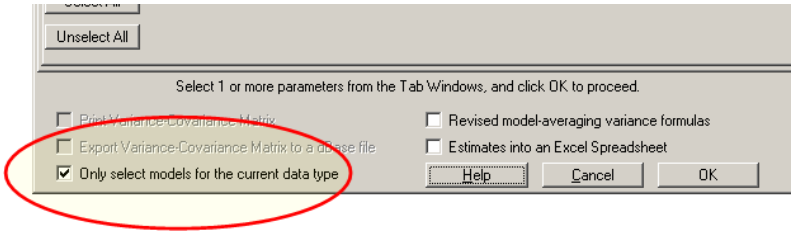
transition probabilities fixed to 0 or 1 in the multi-state data types.

For application of the Fletcher $\hat{c}$ to other data types (e.g., open Cormack-Jolly-Seber models, multi-state models...), see Chapter 5.

15.10. Model averaging and closed models

Model averaging is particularly important in the closed models because selecting a single model tends to be especially problematic when a parameter, in this case N, is in the multinomial coefficient. Typically, abundance would be the only parameter for which we're interested in a model averaged estimate. The basic concepts and mechanics of model averaging were introduced in earlier chapters.

To compute a model averaged estimate for abundance, select '**`Output | Model Averaging`**' then either '**`Real`**' or '**`Derived`**' from the menu. Select the appropriate parameter by checking the box from the PIM window that opens. Here, it will be especially important to note the check box in the lower left-hand corner of the model averaging window (highlighted in the red oval, below).



The highlighted 'check box' selects whether model averaging is performed across multiple data types. It is legitimate to model average across data types that are based on the same likelihood, but not across those based on different likelihoods.

What do we mean by 'different likelihoods'? Well, if you look back at the figure at the top of p. 4 in this chapter, you'll see that closed population abundance models are broadly dichotomized based on whether 'f_0, is included in the likelihood' (referred to as 'full likelihood' models), or not (referred to as 'conditional likelihood' or 'Huggins' models). Also recall that within either the full or conditional likelihood models, there are 2 discrete classes of models, depending on whether or not heterogeneity in encounter probability is being modeled using a finite mixture approach. In a moment, we'll discuss why this is important.

First, why is it not legitimate to average over models with different likelihoods? Recall that model averaging is based on an average of parameters over a candidate model set, where the conditional estimates from each individual model are weighted by normalized AIC weights. Also recall that the AIC is calculated as the sum of $-2\ln(\mathcal{L}) + 2K$ parameters. If the underlying models have different likelihoods, then it would clearly not be correct to model average parameters based on AIC weights normalized over those models.

However, while it is not possible to model average between different models based on conditional or unconditional likelihoods, there are two fairly simply approaches which allow you to accommodate the additional problem of averaging over models with and without finite mixtures. The approaches are based on the simple observation that all models are in fact mixture models – but, simply, some of those models have only a single mixture group. These models are, in fact, entirely equivalent conceptually to standard models without mixtures.

We demonstrate model averaging by considering analysis of some simulated data (contained in

N_avg.inp): true $N = 2{,}000$, 9 sampling occasions. We'll begin by assuming no heterogeneity in p or c, and will use the '**Full likelihood p and c**' data type (i.e., f_0 is included in the likelihood) for our analysis of these data.

To start, we'll fit 2 simple models: $\{f_0, p_t = c_t\}$ and $\{f_0, p_t = c_t + z\}$, where the latter model allows for an additive constant z between the two encounter types (recall that this model is equivalent to $M(bt)$, specifying both a 'behavior' effect, and a 'time' effect). The DM for the more general of our 2 candidate models, $\{f_0, p_t = c_t + z\}$ is shown below:

B1 intcpt	B2 encgrp	Parm	B3 t1	B4 t2	B5 t3	B6 t4	B7 t5	B8 t6	B9 t7	B10 t8	B11 f0
1	1	1:p	1	0	0	0	0	0	0	0	0
1	1	2:p	0	1	0	0	0	0	0	0	0
1	1	3:p	0	0	1	0	0	0	0	0	0
1	1	4:p	0	0	0	1	0	0	0	0	0
1	1	5:p	0	0	0	0	1	0	0	0	0
1	1	6:p	0	0	0	0	0	1	0	0	0
1	1	7:p	0	0	0	0	0	0	1	0	0
1	1	8:p	0	0	0	0	0	0	0	1	0
1	1	9:p	0	0	0	0	0	0	0	0	0
1	0	10:c	0	1	0	0	0	0	0	0	0
1	0	11:c	0	0	1	0	0	0	0	0	0
1	0	12:c	0	0	0	1	0	0	0	0	0
1	0	13:c	0	0	0	0	1	0	0	0	0
1	0	14:c	0	0	0	0	0	1	0	0	0
1	0	15:c	0	0	0	0	0	0	1	0	0
1	0	16:c	0	0	0	0	0	0	0	1	0
1	0	17:c	0	0	0	0	0	0	0	0	0
0	0	18:f0	0	0	0	0	0	0	0	0	1

For DM corresponding to the simpler, nested model $\{f_0, p_t = c_t\}$ – we simply delete the column in the DM corresponding to the 'encounter type' (encgrp).

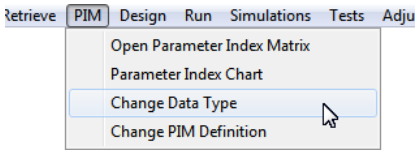
Here are the model fit results for these 2 models:

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{f0,p(t)=c(t)+z}	-1439.0830	0.0000	0.61518	1.0000	11	566.3322
{f0,p(t)=c(t)}	-1438.1447	0.9383	0.38482	0.6255	10	569.2729

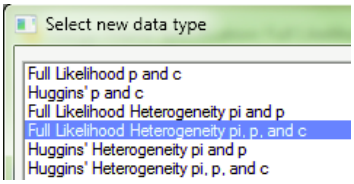
If we stopped here, and model averaged abundance, our model averaged estimate (based on these 2 models) would be $\bar{\hat{N}} = 1{,}996.97$, with an unconditional $\widehat{\text{SE}} = 2.40$.

Let's re-analyze these data using a model which assumes heterogeneity in encounter probability, using a finite mixture approach. Our purpose here is to consider model averaging over models with and without mixtures (in other words, based on different data types). In order to do this, we need to build the mixture models within the same '**MARK** project' (since we can only average across models within a given results browser). To do this, we're going to tell **MARK** that we want to 'change the data type' within our current analysis, from '**Full likelihood p and c**' to something else (a mixture model).

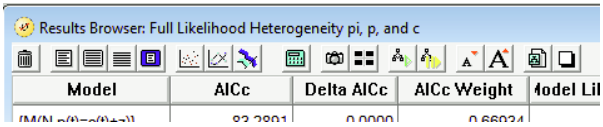
We do this by selecting '**`PIM | Change Data Type`**':



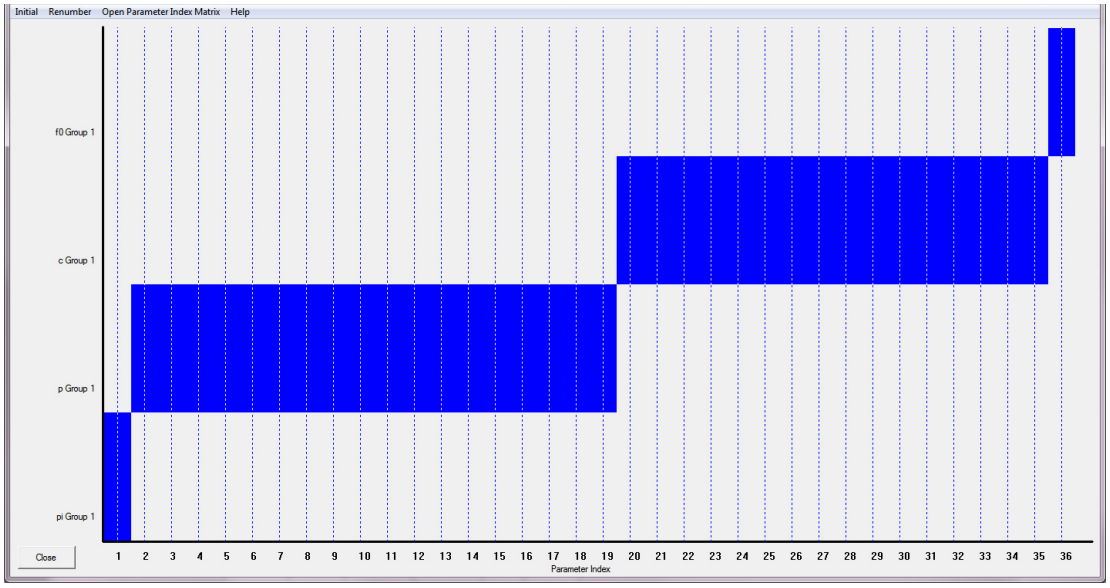
MARK will then present all the available data types which are consistent with your data, letting you select the one you want to change to. Here, we select '**`Full likelihood heterogeneity pi, p and c`**':



Once we've selected the new data type, **MARK** will ask you how many finite mixtures you want to model. We'll accept the default of 2 mixture groups. **MARK** will then drop you back into the browser – the only indication that the underlying data type has been changed is that the title of the results browser now says '`Full likelihood heterogeneity pi, p, and c`'.



The PIM chart (below) is another indication that the underlying data type has changed:



We see (above) that the default model now has the mixture parameter, π, with full time dependence for both encounter parameters, p and c.

Here, we'll fit model $\{f_0, \pi, p_{A,t} = c_{A,t} + z_A, p_{B,t} = c_{B,t} + z_B\}$ to the data (i.e., $\{p_t = c_t + z\}$, but separately within each of the 2 mixture groups). The DM for this model is shown below:

B1 pi	B2 int	B3 encgrp	B4 hetgrp	Parm	B5 t1	B6 t2	B7 t3	B8 t4	B9 t5	B10 t6	B11 t7	B12 t8	B13 f0
1	0	0	0	1:pi	0	0	0	0	0	0	0	0	0
0	1	1	1	2:p	1	0	0	0	0	0	0	0	0
0	1	1	1	3:p	0	1	0	0	0	0	0	0	0
0	1	1	1	4:p	0	0	1	0	0	0	0	0	0
0	1	1	1	5:p	0	0	0	1	0	0	0	0	0
0	1	1	1	6:p	0	0	0	0	1	0	0	0	0
0	1	1	1	7:p	0	0	0	0	0	1	0	0	0
0	1	1	1	8:p	0	0	0	0	0	0	1	0	0
0	1	1	1	9:p	0	0	0	0	0	0	0	1	0
0	1	1	1	10:p	0	0	0	0	0	0	0	0	0
0	1	1	0	11:p	1	0	0	0	0	0	0	0	0
0	1	1	0	12:p	0	1	0	0	0	0	0	0	0
0	1	1	0	13:p	0	0	1	0	0	0	0	0	0
0	1	1	0	14:p	0	0	0	1	0	0	0	0	0
0	1	1	0	15:p	0	0	0	0	1	0	0	0	0
0	1	1	0	16:p	0	0	0	0	0	1	0	0	0
0	1	1	0	17:p	0	0	0	0	0	0	1	0	0
0	1	1	0	18:p	0	0	0	0	0	0	0	1	0
0	1	1	0	19:p	0	0	0	0	0	0	0	0	0
0	1	0	1	20:c	0	1	0	0	0	0	0	0	0
0	1	0	1	21:c	0	0	1	0	0	0	0	0	0
0	1	0	1	22:c	0	0	0	1	0	0	0	0	0
0	1	0	1	23:c	0	0	0	0	1	0	0	0	0
0	1	0	1	24:c	0	0	0	0	0	1	0	0	0
0	1	0	1	25:c	0	0	0	0	0	0	1	0	0
0	1	0	1	26:c	0	0	0	0	0	0	0	1	0
0	1	0	1	27:c	0	0	0	0	0	0	0	0	0
0	1	0	0	28:c	0	1	0	0	0	0	0	0	0
0	1	0	0	29:c	0	0	1	0	0	0	0	0	0
0	1	0	0	30:c	0	0	0	1	0	0	0	0	0
0	1	0	0	31:c	0	0	0	0	1	0	0	0	0
0	1	0	0	32:c	0	0	0	0	0	1	0	0	0
0	1	0	0	33:c	0	0	0	0	0	0	1	0	0
0	1	0	0	34:c	0	0	0	0	0	0	0	1	0
0	1	0	0	35:c	0	0	0	0	0	0	0	0	0
0	0	0	0	36:f0	0	0	0	0	0	0	0	0	1

If we fit this model to the data, and add the results to the browser (below), we see that this new 'heterogeneity model' gets roughly 84% of the support in the data among our 3 candidate models:

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{f0,pi,p(A,t)=c(A,t)+z,p(B,t)=c(B,)+z}	-1443.3683	0.0000	0.83981	1.0000	13	558.0413
{f0,p(t)=c(t)+z}	-1439.0830	4.2853	0.09855	0.1173	11	566.3322
{f0,p(t)=c(t)}	-1438.1447	5.2236	0.06164	0.0734	10	569.2729

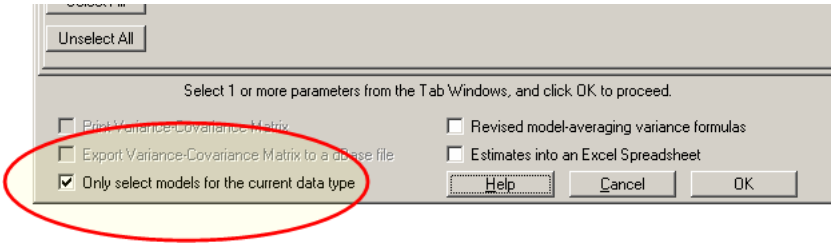
But, our interest here concerns model averaging. If at this point, having just fit the heterogeneity model, $\{f_0, \pi, p_{A,t} = c_{A,t} + z_A, p_{B,t} = c_{B,t} + z_B\}$, we run through the (by now) familiar mechanics of model averaging for N, we would see only one model reported in the model averaging output (shown at the top of the next page).

```
                                model averaging
    Estimates only for data type Full Likelihood Heterogeneity pi, p, and c

                                Derived Parameter 1
Model                                        Weight    Estimate      Standard Error
---------------------------------------- -------   -------------- --------------
{f0,pi,p(A,t)=c(A,t)+z,p(B,t)=c(B,)+z}   1.00000   1998.3164945   2.8151126
---------------------------------------- -------   -------------- --------------
Weighted Average                                   1998.3164945   2.8151126
Unconditional SE                                                  2.8151126
```

Why only one model, and not all three? Simple – at present there is only one model in the browser based on the 'currently active' data type (i.e., full likelihood with 2 finite mixtures). **MARK** knows that the other 2 models in the current model set were constructed using the a different data type ('full likelihood without mixtures'), and thus doesn't try to average over them. Alternatively, if you select (by right-clicking and retrieving) either of the other two models we constructed using the **'Full likelihood p and c'** data type (i.e., $\{f_0, p_t = c_t\}$ or $\{f_0, p_t = c_t + z\}$), and then model average, the model averaging will be based on these 2 models only (since they share a common data type).

Note: Not only is **MARK** 'smart enough' to recognize which models in the browser are based on the same data type, but it is also smart enough to re-calculate AIC weights during the averaging to include only those models with the common (active) likelihood structure. So, the model averaged estimated is correctly reported as $\bar{\hat{N}} = 1{,}996.97$, with an unconditional $\widehat{\text{SE}} = 2.40$ (identical to what we reported earlier for these 2 models, before we changed the data type).

Back to the problem at hand. Remember at the outset of this section we alerted you to the default (selected) option in the model averaging procedure in **MARK**, to '**only select models for the current data type**' (as circled in red, below).



This is the option which 'tells **MARK**' to average only over models of the current data type.

However, our apparent inability to model average over the complete model set represented in the browser seems unfortunate, since we might imagine a full candidate model set with and without heterogeneity models, over which we'd like to derive a model averaged estimate for abundance, $\hat{N}$. What can we do?

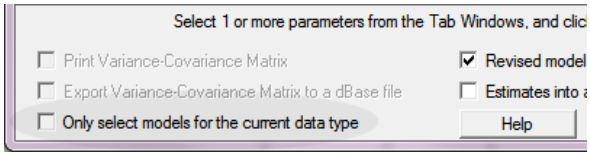
We suggested earlier that there are 2 related approaches you can adopt to average over all 3 models – both of which are based on the same assumption. For either approach, the key conceptual step is to realize that *any* model constructed using the **'Full likelihood p and c'** data type is simply

a heterogeneity model constructed using the **'Full likelihood heterogeneity pi, p, and c'** data type, with one important constraint – fixing $\pi = 1$. (Similarly, any model constructed using '**Huggins p and c**' is simply a '**Huggins heterogeneity p and c**' model, again after fixing $\pi = 1$).

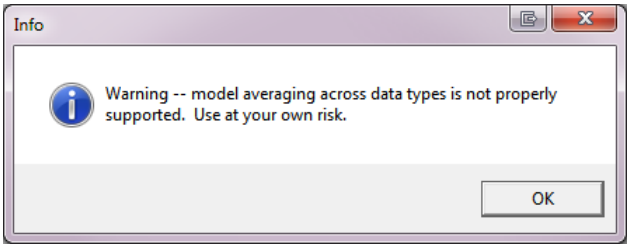
If you think about it for a moment, this should make sense – the **'Full likelihood p and c'** data type is simply a heterogeneity model with only one mixture group (i.e., where $\pi = 1$). So, you could, if you wanted to, force the **'Full likelihood heterogeneity pi, p, and c'** data type to fit models for the **'Full likelihood p and c'** data type, simply by fixing the mixture parameter π to 1. We'll consider this approach in a moment.

The quickest approach to handling model averaging in this case is to 'tell **MARK**' to ignore the fact that, structurally, there are two different data types in the browser. We can do this here because, in fact, the **'Full likelihood p and c'** data type is simply a full likelihood heterogeneity model where π is fixed to 1. In other words, although the models represent two different data types, they have the same underlying likelihood structure. In fact, one data type is equivalent to the other, subject to a particular constraint on one of the parameters (i.e., fixing $\pi = 1$).

So, we run through the mechanics of model averaging, except that this time, we 'turn off' the option to restrict averaging to only models of the current data type, by unchecking the appropriate check-box, as shown below:



Now, when you uncheck this option, **MARK** will respond with the following rather ominous warning message when you try to average over models:



In this instance, we'll assume our underlying logic is correct, and so we can proceed with the final steps of model averaging abundance N.

Here are the estimates:

model averaging

Derived Parameter 1

Model	Weight	Estimate	Standard Error
{f0,pi,p(A,t)=c(A,t)+z,p(B,t)=c(B,)+z}	0.83981	1998.3164945	2.8151126
{f0,p(t)=c(t)+z}	0.09855	1997.5435131	2.4970104
{f0,p(t)=c(t)}	0.06164	1996.0406631	1.9051863
Weighted Average		1998.1000288	2.7276734
Unconditional SE			2.7972971

We see that now, **MARK** averages over all 3 of the models in the browser – the model averaged estimate for abundance is $\bar{\hat{N}} = 1{,}998.10$, with an unconditional $\widehat{\text{SE}} = 2.80$

However, are these estimates correct? Did we in fact 'know what we were doing' when we overrode **MARK**'s warning about averaging over data types? Was our underlying logic that in fact these models have the same underlying likelihood structure correct? We can prove to ourselves that 'we got things right' (and confirm that **MARK** has given us the correct estimates using the preceding approach) by (i) reconstructing the model set using the same data type for all three models, and (ii) manually fixing $\pi = 1$ for two of them. While this is easy enough in principle, in practice this approach will require some thought, since you're going to need to think through carefully which columns in the **'Full likelihood heterogeneity pi, p, and c'** data type DM you need to keep, or modify, when you are reducing the number of heterogeneity groups to 1 (i.e., single mixture group).

To start, have another look at the DM for model $\{f_0, \pi, p_{A,t} = c_{A,t} + z_A, p_{B,t} = c_{B,t} + z_B\}$, shown on p. 40. Notice that there is a column for 'hetgrp', to account for the 2 mixture groups in this model. If we want to force this model to be equivalent to a model without heterogeneity, without switching the underlying data type, we need to do 2 things: (1) delete the 'hetgrp' column from the DM, and (2) fix $\pi = 1$ before starting the numerical estimation run.

Go ahead and delete the 'hetgrp' column from the DM. What is the model represented by this DM? If you look closely, and think about it a bit, you'll realize that without the 'hetgrp' column, you're left with model $\{p_t = c_t + z\}$. Go ahead and run this model – call it 'f0,pi=1,p(t)=c(t)+z' (we'll use 'pi=1' in the model name to indicate we built this model using only a single mixture group). Remember to fix $\pi = 1$ before starting the numerical estimation.

When finished, add the results to the browser:

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{f0,pi,p(A,t)=c(A,t)+z,p(B,t)=c(B,)+z}	-1443.3683	0.0000	0.76447	1.0000	13	558.0413
{f0,p(t)=c(t)+z}	-1439.0830	4.2853	0.08971	0.1173	11	566.3322
{f0,pi=1,p(t)=c(t)+z}	-1439.0830	4.2853	0.08971	0.1173	11	566.3322
{f0,p(t)=c(t)}	-1438.1447	5.2236	0.05611	0.0734	10	569.2729

The deviances for model 'f0,pi=1,p(t)=c(t)+z' and model 'f0,p(t)=c(t)+z' are identical (meaning, they are the same model!).

Next, how would we build model $\{p_t = c_t\}$, using the heterogeneity model approach? Simple – in addition to deleting the 'hetgrp' column, we now also delete the 'encgrp' column (leaving only 'pi', 'incpt', the time columns ('t1' → 't9'), and N). Go ahead and delete the 'encgrp' column, fix $\pi = 1$, and add the results to the browser (call this model 'pi=1,p(t)=c(t)').

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{f0,pi,p(A,t)=c(A,t)+z,p(B,t)=c(B,)+z}	-1443.3683	0.0000	0.72386	1.0000	13	558.0413
{f0,p(t)=c(t)+z}	-1439.0830	4.2853	0.08494	0.1173	11	566.3322
{f0,pi=1,p(t)=c(t)+z}	-1439.0830	4.2853	0.08494	0.1173	11	566.3322
{f0,p(t)=c(t)}	-1438.1447	5.2236	0.05313	0.0734	10	569.2729
{f0,pi=1,p(t)=c(t)}	-1438.1447	5.2236	0.05313	0.0734	10	569.2729

Again, we see that fits for model 'f0,pi=1,p(t)=c(t)' and model 'f0,p(t)=c(t)' are identical (meaning, once again, that they are the same model!).

OK, now for the big moment. We've proven to ourselves that we can build models for the **'Full**

likelihood p and c' data type using the **'Full likelihood heterogeneity pi, p, and c'** data type, simply by fixing $\pi = 1$, and making appropriate modifications to the DM (paying particular attention to terms involving the 'heterogeneity group' column). So, in fact, we could have built all 3 candidate models ($\{p_t = c_t\}$, $\{p_t = c_t + z\}$ and $\{\pi, p_{A,t} = c_{A,t} + z_A, p_{B,t} = c_{B,t} + z_B\}$), using the **'Full likelihood heterogeneity pi, p, and c'** data type – meaning, a single common data type. Meaning, we can model average over all 3 models without overriding the default option in **MARK** that prevents averaging over models built using different data types.

Go ahead and delete models 'p(t)=c(t)+z' and 'p(t)=c(t)' from the browser, leaving only those models built using the **'Full likelihood heterogeneity pi, p, and c'** data type (i.e., all 3 models in the browser are based on the same underlying data type).

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{f0,pi,p(A,t)=c(A,t)+z,p(B,t)=c(B,)+z}	-1443.3683	0.0000	0.83981	1.0000	13	558.0413
{f0,pi=1,p(t)=c(t)+z}	-1439.0830	4.2853	0.09855	0.1173	11	566.3322
{f0,pi=1,p(t)=c(t)}	-1438.1447	5.2236	0.06164	0.0734	10	569.2729

Go ahead and derive a model averaged estimate for $\bar{\hat{N}}$, based on these 3 models – without unchecking the '**Only select models for the current data type**' (since these models are all of the same data type):

model averaging
Estimates only for data type Full Likelihood Heterogeneity pi, p, and c

Model	Derived Parameter 1 Weight	Estimate	Standard Error
{f0,pi,p(A,t)=c(A,t)+z,p(B,t)=c(B,)+z}	0.83981	1998.3164945	2.8151126
{f0,pi=1,p(t)=c(t)+z}	0.09855	1997.5434147	2.4970007
{f0,pi=1,p(t)=c(t)}	0.06164	1996.0408300	1.9052311
Weighted Average		1998.1000294	2.7276752
Unconditional SE			2.7972924

Using this approach, the model averaged estimate for abundance is $\bar{\hat{N}} = 1{,}998.10$, with an unconditional $\widehat{\text{SE}} = 2.80$, which are identical to the estimates we derived earlier.

Given the preceding, there is a fair argument to be made that you should only use the **'heterogeneity pi, p, and c'** data types (for either the full or conditional likelihoods), since it allows you to model average over all the candidate models. However, keeping track of 'encounter groups' and 'heterogeneity groups' does require more work to get things right. As long as you understand what you're doing, simply forcing **MARK** to average over both data types is decidedly quicker. But, remember – you can only average over models with a common likelihood structure: full likelihood (with and without mixtures), or Huggins' conditional likelihood (with and without mixtures).

On the other hand, the situation is mechanically much simpler if you (are willing to) assume the heterogeneity among individuals is continuous logit normal – in other words, use the **'Huggins' p and c with Random Effects'** data type. In that case, you would simply apply the constraint $\sigma_p = 0$ (analogous to using the $\pi = 1$ constraint for the finite mixture models approach just described, above). In this way, you could build an entire symmetrical model set, where each model has a version where 'heterogeneity is turned off' (i.e., by fixing $\sigma_p = 0$), along with the same model where heterogeneity is 'left turned on' (i.e., by not fixing σ_p). We can demonstrate this by fitting models $\{p(t) = c(t), \sigma_p > 0\}$,

$\{p(t) = c(t), \sigma_p = 0\}$, $\{p(t) = c(t) + z, \sigma_p > 0\}$, and $\{p(t) = x(t) + z, \sigma_p = 0\}$. For the second and fourth models, respectively, we fix $\sigma_p = 0$.

The results of fitting these four models to the encounter data in **`N_avg.inp`** are below:

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance	-2Log(L)
{sigmap,p(t)=c(t)}	24854.2743	0.0000	0.67136	1.0000	10	26861.6920	24834.2620
{sigmap,p(t)=c(t)+z}	24855.8297	1.5554	0.30846	0.4595	11	26861.2450	24833.8150
{sigmap=0,p(t)=c(t)+z}	24862.1213	7.8470	0.01327	0.0198	10	26869.5390	24842.1090
{sigmap=0,p(t)=c(t)}	24863.4280	9.1537	0.00691	0.0103	9	26872.8480	24845.4180

We see that models $\{p(t) = c(t), \sigma_p > 0\}$ and $\{p(t) = c(t) + z, \sigma_p > 0\}$ have most of the support in the data, with the simpler model (without an additive effect of recapture) having twice as much support as the next best model. The results clearly support the hypothesis that there is heterogeneity in the capture probability among individuals in the sample.

Model averaging estimates of N is straightforward, since all of our candidate models are clearly of the same data type. We simply need to remember that for the Huggins conditional likelihood, N is estimated as a derived parameter. So, we need to select '**`Output | Model Averaging | Derived`**'. The resulting estimate is $\bar{\hat{N}} = 1998.795$, with an unconditional SE of $\widehat{SE} = 2.640$, both very close to the model averaged estimates derived using the finite mixture approach (above). But, arguably with a fair bit less 'work' involved in building the models.

So, should you conclude that you should routinely use the **`'Huggins' p and c with Random Effects'`** data type? While there are clear advantage in adopting this approach, there are at least a couple of reasons why you might not want to. First, you might not 'believe' that individual heterogeneity in capture probability is continuous logit normal distributed. Second, and of much more practical consideration, the **`'Huggins' p and c with Random Effects'`** models, which are based on Gauss-Hermite quadrature to numerically integrate out the random effect, are much more time consuming to run. A model set which might take a minute or so to run in its entirety using finite mixtures to model the heterogeneity might take many more minutes (sometimes, many, *many* more) to complete.

15.10.1. Estimating CI for model averaged abundance estimates

The usual (simplest) approach to estimating the confidence interval for a given parameter makes use of asymptotic variances, covariances – typically, these can be generated from the information matrix for models with maximum likelihood estimates (this is discussed elsewhere). However, there is a basic problem with applying this 'classical' approach to estimates of abundance – specifically, the classical approach requires asymptotic normality of point estimates $\hat{N}$, and this assumption is frequently not met for any number of reasons.

An alternative approach is to focus on the number of animals that are not caught (f_0), where $f_0 = (N - M_{t+1})$ (this relation was introduced earlier in this chapter). On the assumption that this quantity follows a log-normal distribution (which has been generally confirmed by various authors), then lower and upper CI interval bounds for $\hat{N}$ are given by*

$$\left[M_{t+1} + (\hat{f}_0/C), M_{t+1} + (\hat{f}_0 \times C)\right],$$

where

* There is a typographical error in the equation for C in the Williams, Nichols & Conroy book (p. 304, section 14.2.4). The version presented here is correct.

$$\hat{f}_0 = \hat{N} - M_{t+1} \quad \text{and} \quad C = \exp\left\{1.96\left[\ln\left(1 + \frac{\widehat{\text{var}}(\hat{\bar{N}})}{\hat{f}_0^2}\right)\right]^{1/2}\right\}.$$

Note that since $\hat{N} = M_{t+1} + \hat{f}_0$, then $\widehat{\text{var}}(\hat{N})$ is exactly the same as the variance of $\hat{f}_0$, because M_{t+1} is a known constant.

As such,

$$\begin{aligned}\frac{\widehat{\text{var}}(\hat{N})}{\hat{f}_0^2} &= \frac{\widehat{\text{var}}(\hat{f}_0)}{\hat{f}_0^2} \\ &= \widehat{\text{CV}}(\hat{f}_0)^2.\end{aligned}$$

Commonly in these kinds of calculations, the square of the CV (coefficient of variation) of f_0 is embedded in the formula.

It is important to note that the lower bound of this confidence interval cannot be smaller than M_{t+1}, but the upper bound frequently is larger than the upper bounds computed with the information matrix under the assumption of normality. This is the approach used by **MARK** to derive the CI for $\hat{N}$ (regardless of whether N is a derived or real parameter).

Now, how do we handle the calculation of the CI for the model averaged estimate of abundance, $\hat{\bar{N}}$?

From Buckland *et al*. (1997), the estimated unconditional (i.e., model averaged) variance $\widehat{\text{var}}(\bar{\hat{\theta}})$, calculated over models $\{M_1, M_2, \ldots, M_R\}$ is given as

$$\widehat{\text{var}}(\bar{\hat{\theta}}) = \sum_{i=1}^{R} w_i\left(\widehat{\text{var}}(\hat{\theta}_i \mid M_i) + (\hat{\theta}_i - \bar{\hat{\theta}})^2\right), \quad \text{where } \bar{\hat{\theta}} = \sum_{i=1}^{R} w_i\hat{\theta}_i.$$

Here, the w_i are the Akaike weights (Δ_i) scaled to sum to 1. The subscript i refers to the i^{th} model. The value $\bar{\hat{\theta}}$ is a weighted average of the estimated parameter θ over R models ($i = 1, 2, \ldots, R$).

This estimator of the *unconditional* variance is clearly the sum of 2 components: (i) the *conditional* sampling variance $\widehat{\text{var}}(\hat{\theta}_i|M_i)$ (i.e., conditional on model M_i), and (ii) a term for the variation in the estimates across the R models, $(\hat{\theta}_i - \bar{\hat{\theta}})^2$. The sum of these terms is then merely weighted by the Akaike weights w_i.

Thus, the unconditional standard error would be given as

$$\widehat{\text{SE}}(\bar{\hat{\theta}}) = \sqrt{\widehat{\text{var}}(\bar{\hat{\theta}})}.$$

OK – given all this, back to the original question – how do you estimate the confidence interval for model averaged abundance estimates?

We'll demonstrate the mechanics by means of a worked example. Suppose you fit 3 different full likelihood models ($\{p_t = c_t, f_0\}, \{p_\cdot, c_\cdot, f_0\}, \{p_\cdot = c_\cdot, f_0\}$) to some closed capture data (**`bbsample.inp`** - 8 capture occasions), where $M_{t+1} = 43$.

Here is a tabulation of the relevant results of fitting these models to the data:

model	QAIC_c	w_i	$\hat{N}$	$\widehat{\text{var}}(\hat{N})$
$\{p_\cdot = c_\cdot, f_0\}$	115.364	0.676	53.604	25.737
$\{p_\cdot, c_\cdot, f_0\}$	117.201	0.270	50.867	43.398
$\{p_t = c_t, f_0\}$	120.395	0.055	53.117	24.257

Now, we first need to calculate the unconditional variance of $\hat{N}$. Since our model averaged estimate of $\hat{\theta}$ is given as

$$\bar{\hat{\theta}} = \sum_{i=1}^{R} w_i \hat{\theta}_i,$$

then $\bar{\hat{N}}$ is given as

$$\begin{aligned}\bar{\hat{N}} &= \sum_{i=1}^{R} w_i \hat{N}_i \\ &= (0.676 \times 53.604) + (0.270 \times 50.867) + (0.055 \times 53.117) \\ &= 52.839,\end{aligned}$$

and

$$\begin{aligned}\widehat{\text{var}}\left(\bar{\hat{N}}\right) &= \sum_{i=1}^{R} w_i \left(\widehat{\text{var}}\left(\hat{N}_i \mid M_i\right) + \left(\hat{N}_i - \bar{\hat{N}}\right)^2\right) \\ &= 0.676\left[25.737 + \left(53.604 - 52.839\right)^2\right] + 0.270\left[43.398 + \left(50.867 - 52.839\right)^2\right] \\ &\quad + 0.055\left[24.257 + \left(53.117 - 52.839\right)^2\right] \\ &= \left(17.794 + 12.751 + 1.329\right) = 31.867.\end{aligned}$$

In fact, **MARK** (correctly) handles the calculation of the unconditional variance for you – you would simply need to take the reported unconditional SE and square it to get the unconditional variance. But you need to calculate the CI by hand.

To do so, we first calculate

$$C = \exp\left\{1.96\left[\ln\left(1 + \frac{\widehat{\text{var}}\left(\bar{\hat{N}}\right)}{\hat{f}_0^2}\right)\right]^{1/2}\right\}.$$

Since $M_{t+1} = 43$ for this data set, and since $\bar{\hat{N}} = 52.839$, then

$$\begin{aligned}\bar{\hat{f}}_0 &= \bar{\hat{N}} - M_{t+1} \\ &= (52.839 - 43) \\ &= 9.839,\end{aligned}$$

and thus

$$C = \exp\left\{1.96\left[\ln\left(1 + \frac{\widehat{\mathrm{var}}(\bar{\hat{N}})}{\hat{f}_0^2}\right)\right]^{1/2}\right\}$$

$$= \exp\left\{1.96\left[\ln\left(1 + \frac{31.867}{(9.839)^2}\right)\right]^{1/2}\right\} = 2.845.$$

Last step. Now that we have a value for C, we can derive the 95% CI as

$$\left[43 + \left(9.839/2.845\right), 43 + \left(9.839 \times 2.845\right)\right] = [46.458, 70.992].$$

OK, this seems like a lot of work, but in this particular example, it was necessary. If we had simply used the 'automatic' model averaging in **MARK**, the CI reported by **MARK** for $\bar{\hat{N}}$ is [41.775, 63.905]. There is clearly a fundamental problem with this CI, since the lower bound is less than M_{t+1} (41.775 < 43). Clearly, this makes no sense whatsoever. In contrast, the CI we derived 'by hand' does not bound M_{t+1}. Not only was the reported lower-limit of the CI too low, but the upper limit was as well.

Now, in the preceding example, there was an *obvious* 'problem' with the simple model-averaged CI for $\bar{\hat{N}}$ reported by **MARK**. However, even if the lower bound of the reported CI is $\geq M_{t+1}$, don't take this as evidence that the reported CI is correct.

For example, consider fitting models $\{f_0, p(\cdot) = c(\cdot)\}$ and $\{f_0, p(\cdot), c(\cdot)\}$ to the 'Carothers A' data set (found in the `\examples` subdirectory created when you installed **MARK**).

Here is a tabulation of the relevant results of fitting these models to the data:

model	QAIC_c	w_i	$\hat{N}$	$\widehat{\mathrm{var}}(\hat{N})$
$\{f_0, p(\cdot) = c(\cdot)\}$	−99.7370	0.63460	368.128	212.944
$\{f_0, p(\cdot), c(\cdot)\}$	−98.6330	0.36540	392.480	1234.986

If we had used the model averaging option in **MARK**, the model averaged estimate is $\bar{\hat{N}} = 377.027$, and the reported 95% CI is [324.292, 429.761]. For this data set, $M_{t+1} = 283$, so, in one sense at least, the reported CI for the model average abundance estimate *seems* reasonable, since the lower limit of the CI is greater than M_{t+1} (i.e., 324.292 > 283). How does the reported CI compare with the one derived using the calculations presented above?

Again, we start by first deriving an estimate of the variance of the model averaged abundance:

$$\widehat{\mathrm{var}}(\bar{\hat{N}}) = \sum_{i=1}^{R} w_i\left(\widehat{\mathrm{var}}(\hat{N}_i \mid M_i) + \left(\hat{N}_i - \bar{\hat{N}}\right)^2\right)$$

$$= 0.63460\left(212.944 + \left(368.128 - 377.027\right)^2\right)$$

$$+ 0.36540\left(1234.986 + \left(392.480 - 377.027\right)^2\right)$$

$$= 723.910.$$

Note that if we were to fit these models in **MARK**, the unconditional SE for the model averaged abundance would be reported as 26.9045. If we square this value, we get $(26.9045)^2 = 723.901$.

Again, the unconditional SE – and thus the variance – reported by **MARK** is correct (i.e., you do not need to calculate the SE – or variance – by hand. We are simply demonstrating the underlying calculations).

However, the CI as reported by **MARK** is not correct – this, you need to do by hand.

As in the first example, we first calculate

$$C = \exp\left\{1.96\left[\ln\left(1 + \frac{\widehat{\text{var}}(\bar{\hat{N}})}{\hat{f}_0^2}\right)\right]^{1/2}\right\}.$$

Since $M_{t+1} = 283$ for this data set, and since $\bar{\hat{N}} = 377.027$, then

$$\begin{aligned}\hat{f}_0 &= \bar{\hat{N}} - M_{t+1} \\ &= (377.027 - 283) = 94.027.\end{aligned}$$

Thus,

$$\begin{aligned}C &= \exp\left\{1.96\left[\ln\left(1 + \frac{723.910}{(94.027)^2}\right)\right]^{1/2}\right\} \\ &= 1.733.\end{aligned}$$

Final step. Now that we have a value for C, we can construct the 95% CI around the model averaged estimate $\bar{\hat{N}} = 377.027$ as

$$\Big[283 + \big(94.027/1.733\big), 283 + \big(94.027 \times 1.744\big)\Big] \Longrightarrow \big[337.26, 445.94\big].$$

Recall that if we had used the model averaging option in **MARK**, the reported model averaged 95% CI was [324.292, 429.758]. Again, these reported lower- and upper-limits of the CI are both different than the ones we just calculated 'by hand'.

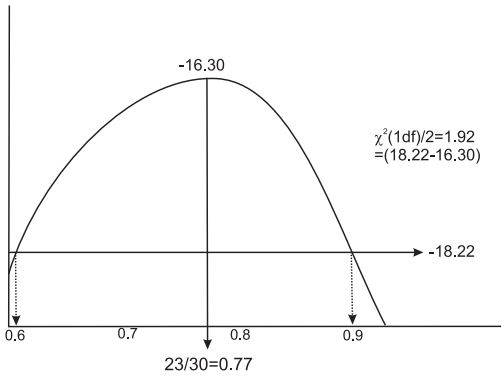
The general recommendation, then, is to calculate the 95% CI for the model averaged abundance 'by hand', using the procedure outlined above.

begin sidebar

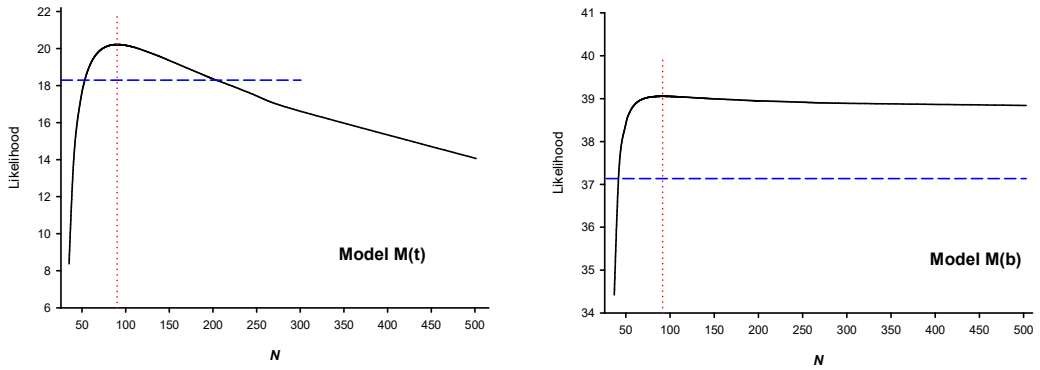
Profile confidence intervals – careful!

In chapter 1, we introduced the profile likelihood approach to constructing confidence intervals. Typically, to construct a CI based on the profile likelihood, you take the value of the log likelihood at the maximum (−16.30 in the example, shown in the figure at the top of the next page), add 1.92 to it (preserving the sign), and look to see where the line corresponding to this sum (−18.22 = −[16.30 + 1.92]) intersects with the *profile* of the log likelihood function. The two intersection points of this line and the profile correspond to the upper- and lower-bounds of the CI.

For closed population abundance estimators, there is need to be cautious in using profile likelihoods to generate CI, having to do with the fact that abundance estimates are not [0, 1] bounded parameters. The maximum bound (if in fact one exists) is determined by the likelihood. There are situations for some closed models where the upper bound of the likelihood profile $\rightarrow \infty$.



For example, take the likelihood for models $\{f_0, p(t) = c(t)\}$ (i.e., model M_t) and model $\{f_0, p(\cdot), c)(\cdot)\}$ (i.e., model M_b), fit to some data. The likelihood profiles for both models are shown below:



We see that the likelihood profile for model $\{f_0, p(t) = c(t)\}$ rises to the MLE (vertical dotted line), and then falls, such that the horizontal dashed line corresponding to the MLE−1.92 intersects the likelihood at 2 points (which represent the two bounds of the 95% CI). On the other hand, for model $\{f_0, p(\cdot), c)(\cdot)\}$ the likelihood rises, but then never falls to <2 units from the MLE – and, as such, there is no upper bound for the profile likelihood!

end sidebar

15.11. Parameter estimability in closed models

It is important to examine the real parameter results to see if $p_t = 1.0$ and $\hat{N} = M_{t+1}$. This would indicate that the model you constructed was not estimable. Be careful – incorrectly built models may appear very good in terms of AIC_c. If you don't know what M_{t+1} is for a particular data set, it can be found in the full model output labeled as '`M(t+1)`'.

In addition, it has been noted several times that a constraint must be placed on p_t in order to properly estimate N. It is straightforward to demonstrate that an estimate of p_t is necessary to get an estimate of N. We've already done it once. We'll do it again here to make sure you don't forget.

Consider the following estimator of N from a $t = 3$ occasion capture-recapture study,

$$\hat{N} = \frac{M_{t+1}}{1 - \left[(1-\hat{p}_1)(1-\hat{p}_2)(1-\hat{p}_3)\right]}.$$

Now if $\hat{p}_3 = 1$, then the denominator in the estimator above equals 1. Thus, the estimate of $\hat{N} = M_{t+1}$.

Let's consider the estimability of the p's, now that we know we need $\hat{p}_t$ to get $\hat{N}$. The first p is estimable because we have information in the subsequent capture occasions about the proportion of marked and unmarked animals captured. This goes for each p until we get to p_t. On the last occasion, there are no future occasions from which to pull information. Thus, we must place a constraint of p_t. The constraint can be in the form of modeling p_t as a function of previous p's or as a function of the recaptures, or by constraining estimates to be functions of one or more covariates. Recall that constraining parameters as linear function of a covariate can often 'solve' identifiability issues.

15.12. Other applications

Closed population capture-recapture models have been used for other applications beyond estimating the number of individuals in a population. There is a natural extension to estimating the number of species in an area. In this case, encounter histories represent detections of species rather than individuals. Heterogeneity in detection probability among species is virtually guaranteed.

Closed capture-recapture models and modifications thereof are widely used in human demography. There they are typically referred to as multiple list sampling. Several lists containing people from a population of interest, for example drug users in a city, act as sampling occasions. Individuals are matched across lists to estimate abundance.

The closed population capture-recapture models underpin the secondary sampling periods in a robust design (Kendall *et al*. 1997; see Chapter 16). It is therefore essential to understand the closed captures models in order to fully understand the robust design

15.13. Summary

Despite a seemingly simple goal, estimating abundance can be quite difficult. The closed capture-recapture models contain numerous, subtle complications. **MARK** offers a framework for a variety of models addressing different assumptions, compares models and most importantly model averages estimated abundance.

An additional advantage of **MARK** is the ability to combine data from multiple study sites. It is too often argued in the ecological literature that capture-recapture is not useful because the sample size at any one trapping grid is too small. Through the use of groups, **MARK** allows data from multiple grids to be used to jointly estimate detection probability. While this may bias the estimate of N somewhat for each individual grid, it remains a better solution than using minimum number known alive as an index. Moreover, **MARK** handles all of the covariances among the N's estimated from common data.

15.14. References

Afroz, F., Parry, M., and Fletcher, D. (2020) Estimating overdispersion in sparse multinomial data. *Biometrics*, **76**, 834-842.

Boulanger, J., White, G. C., McLellan, B. N., Woods, J., Proctor, M., and Himmer, S. (2002) A meta-analysis of grizzly bear DNA mark-recapture projects in British Columbia, Canada. *Ursus*, **13**, 137-152.

Carothers, A. D. (1973) Capture-recapture methods applied to a population with known parameters. *Journal of Animal Ecology*, **42**, 125-146.

Dorazio, R. M., and Royle, J. A. (2003) Mixture models for estimating the size of a closed population when capture rates vary among individuals. *Biometrics*, **59**, 351-364.

Fletcher, D. (2012) Estimating overdispersion when fitting a generalized linear model to sparse data. *Biometrika*, **99**, 230-237.

Gimenez, O., and Choquet, R. (2010) Individual heterogeneity in studies on marked animals using numerical integration: capture-recapture mixed models. *Ecology*, **91**, 951-957.

Huggins, R. M. (1989) On the statistical analysis of capture experiments. *Biometrika*, **76**, 133-140.

Kendall, W. L., Nichols, J. D., and Hines, J. E. (1997) Estimating temporary emigration using capture-recapture data with Pollock's robust design. *Ecology*, **78**, 563-578.

Link, W. A. (2003) Non-identifiability of population size from capture-recapture data with heterogeneous detection probabilities. *Biometrics*, **59**, 1125-1132.

Link, W. A. (2004) Individual heterogeneity and identifiability in capture-recapture models. *Animal Biodiversity and Conservation*, **27**, 441-449.

Link, W. A., Yoshizaki, J., Bailey, L. L., and Pollock, K. H. (2010) Uncovering a latent multinomial: Analysis of mark-recapture data with misidentification. *Biometrics*, **66**, 178-185.

Liu, Q., and Pierce, D. (1994) A note on Gauss-Hermite quadrature. *Biometrika*, **81**, 624-629.

Lukacs, P. M., and Burnham, K. P. (2005) Estimating population size from DNA-based closed capture-recapture data incorporating genotyping error. *Journal of Wildlife Management*, **69**, 396-403.

McClintock, B. T., White, G. C., Burnham, K. P., and Pryde, M. A. (2009) A generalized mixed effects model of abundance for mark-resight data when sampling is without replacement. Pages 271-289 in D. L. Thomson, E. G. Cooch, and M. J. Conroy, editors. *Modeling Demographic Processes in Marked Populations – Environmental and Ecological statistics 3*. Springer, New York, New York, USA.

Norris, J. L., and Pollock, K. H. (1996) Nonparametric MLE under two closed capture-recapture models with heterogeneity. *Biometrics*, **52**, 639-649.

Otis, D. L., Burnham, K. P., White, G. C., and Anderson, D. R. (1978) Statistical inference from capture data on closed animal populations. *Wildlife Monographs*, **62**.

Pledger, S. (2000) Unified maximum likelihood estimates for closed capture-recapture models using mixtures. *Biometrics*, **56**, 434-442.

Pledger, S. (2005) The performance of mixture models in heterogeneous closed population capture-recapture. *Biometrics*, **61**, 868-873.

Pledger, S., and Phillpot, P. (2008) Using mixtures to model heterogeneity in ecological capture-recapture studies. *Biometrical Journal*, **50**, 1022-1034.

Powell, L. A., and Gale, G. A. (2015) *Estimation of Parameters for Animal Populations*. Caught Napping Publications, Lincoln, NE. 256 pp.

Seber, G. A. F., and Whale, J. F. (1970) The removal method for two and three samples. *Biometrics*, **26**,

393-400.

Stanley, T. R., and Burnham, K. P. (1998) Information-theoretic model selection and model averaging for closed-population capture-recapture studies. *Biometrical Journal*, **40**, 475-494.

Stanley, T. R., and Burnham, K. P. (1999) A closure test for time-specific capture-recapture data. *Environmental and Ecological Statistics*, **6**, 197-209.

White, G. C., Anderson, D. R., Burnham, K. P., and Otis, D. L. (1982) Capture-recapture and removal methods for sampling closed populations. Los Alamos National Laboratory Publication LA-8787-NERP. Los Alamos, NM.

White, G. C., and Cooch, E. G. (2017) Population abundance estimation with heterogeneous encounter probabilities using numerical integration. *Journal of Wildlife Management*, **81**, 322-336.

Addendum 1 – testing equality of estimated abundance between groups

In section 15.2.1, we noted that because **MARK** reports $\hat{N}$ as a *derived* parameter (this is true whether you use models based on either the full or conditional likelihoods), comparing abundance estimates between (say) two groups is not as simple as building a model where you set $N_1 = N_2$. Remember, you can only apply constraints to parameters that are included in the likelihood. Since abundance N isn't in the likelihood, you can't build models that explicitly constrain N.

However, we also recall that $\hat{N} = \hat{f}_0 + M_{t+1}$, and since M_{t+1} is a constant, then $\hat{N} \propto \hat{f}_0$. So we might wonder if constraining $\hat{f}_0$ (which *is* in the likelihood for the *full* likelihood models) be equivalent to constraining $\hat{N}$? As noted in section 15.2.1, the answer is 'no', conceptually, since it makes little logical sense to ask 'is the number never caught (f_0) is the same in the 2 locations?', which is exactly what you'd be asking if you set (say) $\hat{f}_{0,\text{grp 1}} = \hat{f}_{0,\text{grp 2}}$ in the design matrix.

However, we can in fact address the question of comparing abundance estimates, by making use of the Markov Chain Monte Carlo (MCMC) capabilities in **MARK**. If we were using a specialized MCMC application, like **JAGS**, or **BUGS**, we could simply create a derived parameter as a function of other parameters in the model (which could be either real, or derived), and then analyze the posterior samples for this derived parameter (this ability to explicitly code functions of parameters is one of the real conveniences of using MCMC, typically in but not confined to a Bayesian framework).

The MCMC capabilities in **MARK** (which are discussed in much more detail than we show here in Appendix E) do not allow the explicit construction of a user-specified derived parameter. However, we can accomplish the same thing, albeit in a slightly more 'brute-force' way, but simply (i) taking the individual sample chains from the MCMC simulations, (ii) deriving the function of these parameters over the chains – in this case, calculating the difference between the derived estimates of abundance, and (iii) evaluating this difference as the posterior distribution for the difference (which it is). In fact, this is equivalent to what **JAGS** or **BUGS** does, except that instead of calculating the difference in the derived abundance estimates at each step of the sampler, we simply do it *post hoc* – after the samplers are finished.

More specifically for our problem:

1. We'll first fit a closed population abundance estimation model – we'll use the Huggins conditional likelihood – to simulated data where there are 2 groups (with true $N_1 = 100$ and $N_2 = 108$. So, the true difference in abundance between the two is 8).
2. We'll then re-run this approximating model, using the MCMC capabilities in **MARK**. The `MCMC.BIN` file that will get created will have the posterior estimates for the two derived parameters (being, estimated abundance), for each group (i.e., $\hat{N}_1$, $\hat{N}_2$), which are referred to in `MCMC.BIN` as '`derived1`' and '`derived2`', respectively.
3. We will import the `MCMC.BIN` into a statistical program (for this demonstration, we'll use **R**), and 'post-process' the data in `MCMC.BIN`, creating a derived variable we might call '`diff`', which is the difference between `derived1` and `derived2`, the derived abundance estimates for the two groups. [So, in fact, '`diff`' is a derived variable of two other derived variables.]
4. Finally, we'll look at the distribution of this derived '`diff`' variable, and if it doesn't bound 0, you might safely conclude there is a real difference in estimated abundance between the two groups.

OK, let's see how this is done. For this demonstration, we'll consider 3 separate datasets, all simulated under the same generating model, $\{p = c\}$, using a Huggins conditional likelihood: **`diff_highp.inp`**,

diff_medp.inp, and **diff_lowp.inp**. These 3 simulated datasets differ from each other only in terms of the underlying detection probabilities (high, medium, and low). Each consists of 2 groups (where true abundance for group 1 was 100, and true abundance for group 2 was 108), 5 sample occasions. To add a bit of uncertainty to the simulated data, we set the encounter probability for group 2 to be somewhat lower than for group 1: for **diff_highp.inp**, $p_1 = 0.6, p_2 = 0.5$; for **diff_medp.inp**, $p_1 = 0.4, p_2 = 0.3$; and for **diff_lowp.inp**, $p_1 = 0.2, p_2 = 0.1$).

For any of the 3 simulated data sets (we'll compare results from all 3 later), we first fit model {grp, $p = c$} (i.e., the generating model) to the data. For purposes of demonstrating the steps, we'll consider **diff_highp.inp**. We'll use a logit link (for reasons discussed in Appendix E). At this point, building and fitting this simple model should be straightforward. Here is the results browser after fitting this model to the data:

Model	AICc	Delta AICc	AICc Weight	No. Par.	Deviance	-2Log(L)
{grp, p=c}	1382.7932	0.0000	1.00000	2	1552.1224	1378.7814

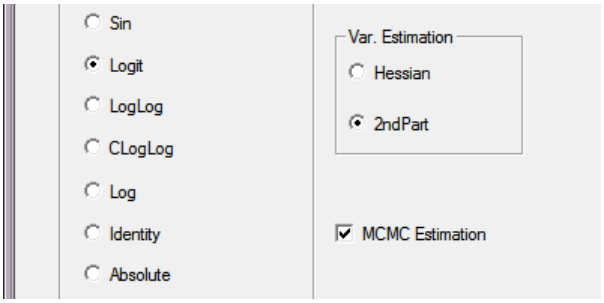
and the derived estimates for the two groups:

Estimates of Derived Parameters
Population Estimates of {grp, p=c}

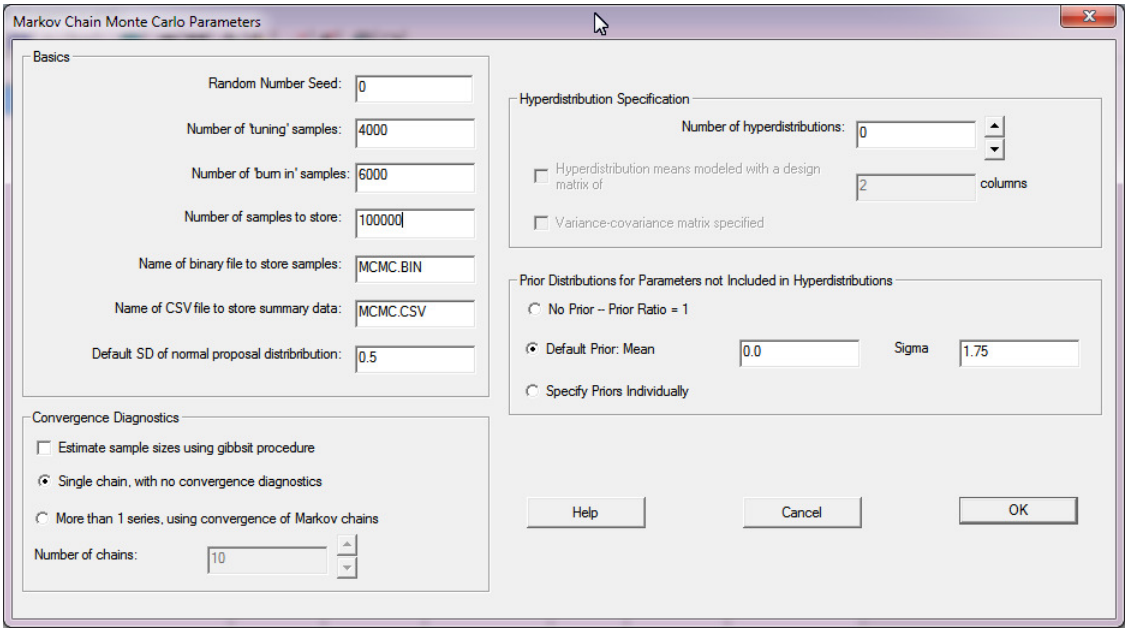
Group	N-hat	Standard Error	95% Confidence Interval Lower	Upper
1	100.93016	1.0059396	100.16574	105.22026
2	108.06908	2.2719347	105.46582	115.29569

We see that the derived estimates are both quite close to the true values used in simulating the data. The arithmetic difference between the two derived estimates is 7.14. The question might be – 'is this difference larger than 0, such that we could conclude that there is a difference between the 2 groups?'. We observe that both derived estimates are reported with estimates of the uncertainty for each parameter, and as such, we could try applying some *post hoc* statistical approach to comparing them (any standard biometry text is likely to have at least one approach to estimating the significance of a difference of two values, where each value has an associated uncertainty). We might also consider an approach which explicitly accommodates covariance between the estimates – one such approach, based on the 'Delta method', is discussed at length in Appendix B.

Here, though, we will simply apply some *post hoc* processing of MCMC samples to directly address the question of whether or not the uncertainty of the difference between the two derived estimates of abundance bounds 0. To start, we simply re-run the same model we just ran, but only after we first check the '**MCMC Estimation**' box in the '**Setup Numerical Estimation Run**' window:



Once you click the '**OK to Run**' button, **MARK** will respond with a window (below) where you specify the MCMC parameters that will specify aspects of the numerical estimation (see Appendix E for a complete discussion of these parameters):



What is generally important is that we want a sufficient number of samples (at all stages) to ensure that the samplers have converged on the stationary joint distribution. For this example we've used 4,000 'tuning' samples, 6,000 'burn in' samples, and 100,000 samples from the posterior distribution. We've also specified only a single chain, with no convergence diagnostics. We click the '**OK**' button, and let it run (for this problem, the estimation run is very fast – this will not always be the case for MCMC analysis!).

Once finished, **MARK** will output the results to the editor. If you scroll down to the bottom of the output listing, you'll see various macro values that can be used for post-processing of the chains for each parameter. These macro values are copied into **SAS** or **R** programs that are provided in the **MARK** helpfile. We'll demonstrate the mechanics using **R**.

For our simulated data, fit using model $\{\texttt{grp}, p = c\}$, the **R** macro values are:

```
ncovs <- 2;  # Number of beta estimates
nmeans <- 0;  # Number of mean estimates
ndesigns <- 0; # Number of design matrix estimates
nsigmas <- 0; # Number of sigma estimates
nrhos <- 0; # Number of rho estimates
nlogit <- 2; # Number of real estimates
nderived <- 2; # Number of derived estimates
filename <- "C:\\USERS\\USER\\DESKTOP\\MCMC.BIN";  # path to MCMC.BIN file
```

So, all we do is copy this into the appropriate section at the top of the **R** script provided in the **MARK** helpfile. The script is fairly lengthy, so we won't reproduce it in full here. Instead we'll focus on the additional steps you'll need to execute in order to derive an estimate of the variance for the product of the first 3 survival estimates.

First, copy the macro variables (above) into the **R** script, and execute it 'as is'. This will create an MCMC 'object', called 'mcmcdata', that is compatible with one of several **R** packages (e.g., coda). This object contains each of the individual Markov chains, for each parameter.

Normally, what you'd do at this point is use some package, like coda, to post-process the chains, and generate various descriptive statistics and associated graphics. However, what we want to do here is estimate the difference between the two derived estimates of abundance. To do this, we will (i) extract the chains for the derived abundance estimates, which are labeled 'derived1' and 'derived2' from the mcmcdata object, (ii) calculate the difference between the two, and (iii) generate various descriptive statistics for this difference.

While there are any number of ways you might do this in **R**, the following works well enough. The first thing we do is convert the MCMC 'object' (mcmcdata) to a dataframe. We'll call this new dataframe 'chaindata':

```
chaindata <- as.data.frame(mcmcdata);
```

Next, we'll add a column to this new dataframe for the difference ('derived2'-'derived1'), and label this new column 'diff'. Note, in the dataframe, these parameters are referred to (by their column names, which are explicitly set by the preceding **R** script) as 'derived1' and 'derived2', respectively:

```
chaindata$diff <- chaindata$derived2-chaindata$derived1;
```

All that's left is to look at the summary statistics for this difference. The following snippet of **R** code will probably suffice (here, we use the 95% HPD – highest posterior density – to evaluate whether or not the posterior distribution includes 0 – see Appendix E for discussion of HPD.) You need to revert chaindata back to an MCMC object, using the as.mcmc command to do so, however.):

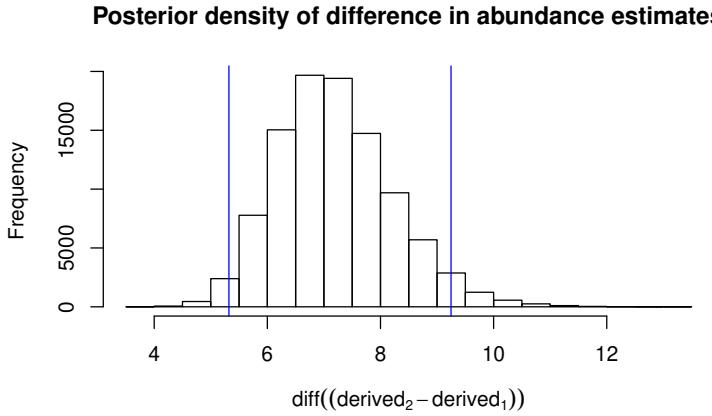
```
cat("basic summary stats of diff in derived abundance estimates \nn")
print(summary(chaindata$diff));
cat("variance of difference...\n")
print(var(chaindata$diff));

hold <- as.mcmc(chaindata$diff)
hpd <- HPDinterval(hold,prob=0.95)
print(hpd)

hist(chaindata$diff,xlab = expression(paste(diff((derived[1] - derived[2])))),
   main="Posterior density of difference in abundance estimates")
abline(v=hpd[1],col="blue")
abline(v=hpd[2],col="blue")
```

Here are the resulting summary statistics for **diff_highp.inp** (your results might differ, slightly – MCMC is a 'Monte Carlo' procedure, and each run will yield a slightly different chain):

```
basic summary stats of diff in derived abundance estimates
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.887   6.481   7.113   7.201   7.827  13.017
variance of difference...
   [1] 1.042797
```

```
 95% HPD on difference...
        lower    upper
 var1 5.320796 9.247095
```

So, the mean difference between our two abundance estimates is 7.2, with a 95% credibility interval (based on the HPD) of [5.321, 9.247], that does not bound 0. The posterior distribution and the 95% cutoffs based on the HPD (vertical blue lines) is shown in the following histogram generated by the preceding code:



Based on the summary results, it might be quite reasonable to conclude that there the difference in population sizes between the two groups is significantly greater than 0.

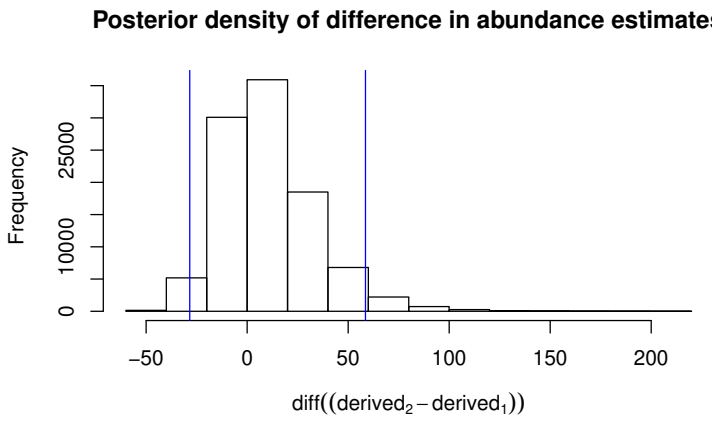
On the other hand, if we repeat all of these steps, but instead analyze the encounter data in **diff_lowp.inp** instead (where the encounter probabilities are much lower),

```
  basic summary stats of diff in derived abundance estimates
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-56.905  -5.198   7.315  10.701  22.800 219.939
variance of difference...
 [1] 541.7442

 95% HPD on difference...
        lower    upper
 var1 -28.41012 58.57325
```

we observe that even though the median/mean difference is similar to the value from our first analysis, the estimated variance is much larger (541.7 vs 1.04). This is reflected in the much wider credibility interval (again based on the HPD) of [−28.41, 58.57]. This interval clearly bounds 0 (as shown in the histogram shown at the top of this page), so here, we would not be comfortable concluding there is a real difference in population size between the two groups.

For completeness, let's also consider the results using the simulated data in **diff_medp.inp**, where the encounter probabilities were 'intermediate' between the two data sets we've just considered. Here, the variance of the estimate difference is 24.7, with a 95% credible interval based on the HPD of [−0.205, 18.959].

Comparing the credible intervals among the 3 data sets re-emphasizes a critical point – precision of our estimates increases with increasing detection probability. The 'big law' for abundance estimation



(and 'mark-reencounter'-type analyses in general) is that you should do your best to increase detection probabilities as much as you can.

Quick aside here – the simulation capabilities in **MARK** (Appendix A) are extremely helpful in planning your study – should you mark more new individuals, or should you work harder to detect the individuals you have already marked? In this example, it is clear that you would need to have fairly high encounter probabilities in order to detect (based on some criterion – say, with 95% probability) a true difference in abundance as small as was simulated in the encounter data.

So there you go...using MCMC in **MARK** to assess the difference in abundance estimates between two groups. In fact, the MCMC approach in **MARK** has a lot of value added beyond this particular application (for example, calculating the variance of a product of parameters – see Addendum to Appendix E). Its fairly easy to use it to generate mean/variances of all sorts of functions of derived parameters. As noted earlier, the only difference between what **MARK** does, and what (say) **JAGS** or **OpenBUGS** does it that the latter calculates the derived values at each step of the chain, whereas with **MARK**, you post-process the chain. The former (i.e., what **JAGS** or **OpenBUGS** does, for instance) is computationally more convenient, but the latter approach (that **MARK** uses) yields identical results.

model averaging?

One final point – the preceding steps were applied to a single model (being, model $\{\texttt{grp}, c = p\}$). Typically, you will (and should) have more than one model to fit to the encounter data. In the context of model averaging abundance estimates (and by extension, model averaging the derived differences in abundance estimates), the simplest approach* would be to:

1. derive MCMC estimate of the difference in abundance, for each model in the candidate model set. You could then model average these differences (using normalized AIC weights) in the usual fashion. This is straightforward.
2. then, using Buckland's expression for the unconditional variance from Chapter 4 (and as illustrated here in this chapter – section 15.10.1) derive a model averaged estimate for the variance of the difference between estimates of abundance, and then construct the appropriate unconditional CI for that (again, as discussed in section 15.10.1).

* meaning...without resorting to 'fancy' things like reversible jump MCMC and related technologies (which are unlikely to be implemented in **MARK** – and are not easy to implement even in more general 'languages' like **JAGS** or **OpenBUGS**).

Addendum 2 – heterogeneity modeling for other data types

Heterogeneity is a common, verging on ubiquitous issue, regrdless of the data type, or type of analysis you're working with. When individual animals do not behave identically with the same exact detection or survival probability, extra-binomial variation or overdispersion occurs. Each animal has its own parameter value, leading to parameter heterogeneity. The extra-binomial variation causes standard errors of parameter estimates to be too small, i.e., too precise, because this extra variation is not recognized in the model.

Two approaches are available in **MARK** for some data types to model individual heterogeneity. While there are no perfect solutions to handling heterogeneity in general, use of discrete- and continuous-mixture models has proven to be a robust approach in many cases. In both cases, more than ≥ 5 occasions are required to be able to detect and account for individual heterogeneity in any reasonable way. That is, the animal (or plot for occupancy) must be observed multiple times to be able to detect and estimate the heterogeneity.

Finite-mixture models

Pledger (2000) and Pledger *et al*. (2003) proposed using finite-mixture models to model heterogeneity. This approach has been implemented for the 'encounter probability' parameter in **MARK** for a variety of data types, and for both 'encounter' and 'survival', or 'survival' only for others:

data type	*parameter(s) with mixture option*	*chapter(s)*
closed captures	p, c	15
robust design	p, c	16
Cormack-Jolly-Seber	φ, p	$3 \rightarrow 8$
Pradel	p	13
robust-design Pradel	p	–
Link-Barker	p	12
Seber dead recovery	S	8
occupancy (single season)	p	22
occupancy (multi-season)	p	23

Continuous-mixture models

The second method of modeling individual heterogeneity is to add a normally-distributed random error with standard deviation σ to the logit value of the parameter, then numerically integrate out this random error. The first data type in **MARK** that did this was the logit-normal mark-resight estimator (McClintock & White 2009, McClintock *et al*. 2009, Chapter 18). Gimenez & Choquet (2010) proposed the same approach for Cormack-Jolly-Seber models. Because of the relative simplicity and 'biological realism' of the approach, it has been incorporated with a number of **MARK** data types.

Because the random effect is numerically integrated out of the model using Gaussian-Hermite quadrature (this chapter, Gimenez & Choquet 2010, White & Cooch 2017), the computer time to run these models is significantly greater than the corresponding model without the random effect. The default

in **MARK** is to integrate over 101 nodes, so roughly 101 times the computing time is needed compared to the data type that does not include the random effect. Further, for data types that have multiple random effects, such as the Cormack-Jolly-Seber or Link-Barker, the integration loops are nested. So for the Cormack-Jolly-Seber data type with random effects being estimated for both φ and p, the CPU time will be approximately $(101 \times 101) = 10{,}201$ times longer than a normal Cormack-Jolly-Seber model.

The number of nodes for Gaussian-Hermite quadrature can be set in the '**`File | Preferences`**' dialog window. The minimum value allowed is 15, and the max 505. For 505 nodes, the weights are zero for at least 15 nodes in each tail, so more nodes doesn't actually improve precision in the tails. The default (101) tends to work very well for most purposes.

A nice feature of this approach is that the estimate of the σ parameter modeling the individual heterogeneity is available. Thus, you can get some idea of how much individual heterogeneity is provided around the mean value on the logit scale. Further, the σ parameter can be set to zero to evaluate the impact of individual heterogeneity. In all cases, fixing the σ parameter to zero will give exactly the same likelihood value as the model not including the random effect(s).

Data types in **MARK** where the individual random effects methodology has been implemented are shown in the following:

data type	*parameter(s) with Gauss-Hermite option*	*chapter(s)*
mark-resight logit	p	19
closed abundance, Huggins	p, c	15
robust design, Huggins	φ, p	$3 \rightarrow 8$
Cormack-Jolly-Seber	φ, p	$3 \rightarrow 8$
multi-state	S, p, ψ	10
robust-design Pradel	p	–
Link-Barker	φ, p, f	12
Burnham live-dead	S, F, p, r^*	10
Seber dead recovery	S, r^*	9
occupancy	p	22
known-fate	$S^\dagger$	17
density	p	21

(∗) Parameters marked with a (∗) can have a random effect specified mainly for simulation purposes. Because these parameters are only observed once in the probability of an encounter history, the random effect σ is not identifiable.

(†) S in the 'known fate' data type can be modeled with the saturated model in **MARK**. That is, you can create the saturated model and hence not be able to estimate the σ_S parameter because you have made it not identifiable.