CHAPTER 4

Building & comparing models

In this chapter, we introduce several important concepts.* First, we introduce the basic concepts and 'mechanics' for building models in **MARK**. Second, we introduce some of the concepts behind the important questions of 'model selection' and 'multi-model inference'. *How* to build models in **MARK** is 'mechanics' – *why* we build certain models, and what we do with them, is 'science'. Both are *critical* concepts to master in order to use **MARK** effectively, and are *fundamental* to understanding everything else in this book, so take your time.

We'll begin by re-visiting the male European dipper data we introduced in the last chapter. We will compare 2 different subsets of models: models where either survival or recapture (or both) varies with time, or models where either survival or recapture (or both) are constant with respect to time. The models are presented in the following table, using the notation suggested in Lebreton *et al*. (1992).

model	*explanation*
$\{\varphi_t\, p_t\}$	both survival and encounter probability time dependent
$\{\varphi_.\, p_t\}$	survival constant over time, encounter probability time dependent
$\{\varphi_t\, p_.\}$	survival time dependent, encounter probability constant over time
$\{\varphi_.\, p_.\}$	both survival and encounter probabilities constant over time

In the following, we will go through the steps in fitting each of these 4 models to the data. In fact, these models are the same as those we fit in Chapter 3. So why do them again? In Chapter 3, our intent was to give you a (very) gentle run-through of running **MARK**, using some of the standard options. In this chapter, the aim is to introduce you to the mechanics of model building, from the ground up. We will not rely on 'built-in' or 'pre-defined' models in this chapter (in fact, you're not likely to ever use them again). Since you already saw the 'basics' of getting **MARK** up and running in Chapter 3, we'll omit some of the more detailed explanations for each step in this chapter.

However, we must emphasize that before you actually use **MARK** (or any other program) to compare different models, you need to first confirm that your 'starting model' (generally, the most parameterized or most general model) adequately fits the data. In other words, you **must** conduct a goodness-of-fit (GOF) test for your 'starting model'. GOF testing is discussed in detail in Chapter 5, and periodically throughout the remainder of this book. For convenience, we'll assume in this chapter that the 'starting model' does adequately fit the data.

* Very important...

4.1. Building models – parameter indexing & model structures

As in Chapter 3, start **MARK** by double-clicking the **MARK** icon. We're going to use the same data set we analyzed in Chapter 3 (**ed_males.inp**). At this point, we can do one of 2 things: (1) we can start a completely new **MARK** session (i.e., create a completely new `*.DBF` file), or (2) we can re-open the `*.DBF` file we created in Chapter 3, and append new model results to it. Since you already saw in Chapter 3 how to start a 'new' project, we'll focus here on the second possibility – appending new model results to the existing `*.DBF` file.

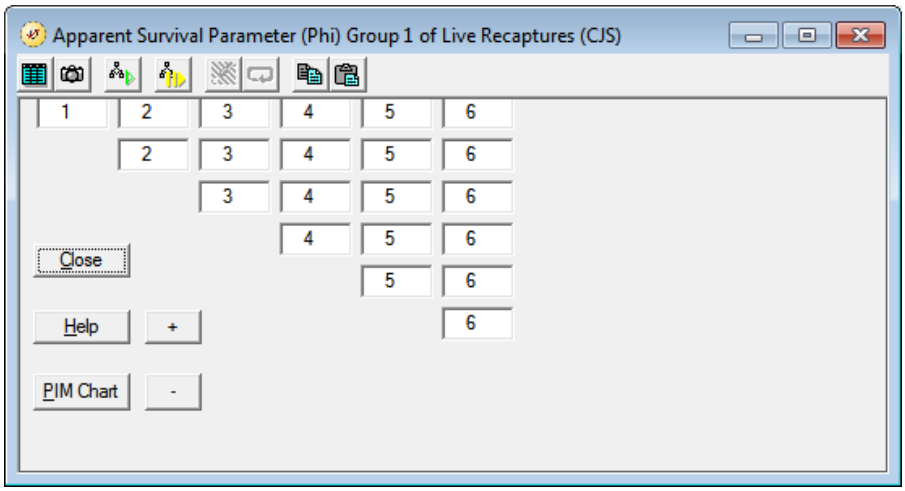
This is very easy to do – from the opening **MARK** 'splash screen', select '**Open**' from the '**File**' menu, and find the **ed_males.dbf** file you created in Chapter 3 (remember that **MARK** uses the prefix of the `*.INP` file – the file containing the encounter histories – as the prefix for the `*.DBF` file. Thus, analysis of **ed_males.inp** leads to the creation of **ed_males.dbf**). Once you've found the **ed_males.dbf** file, simply double-click the file to access it. Once you've double-clicked the file, the **MARK** 'splash screen' will disappear, and you'll be presented with the main **MARK** window, and the results browser. In the results browser, you'll see the models you already fit to these data in the last chapter (there should be 4 models), and their respective AIC and deviance values.

In this chapter, we want to show you how to build these models from scratch. As such, there is no point in starting with all the results already in the browser! So, take a deep breath and delete all the models currently in the browser! To do this, simply highlight each of the models in turn, and click the trash-can icon in the browser toolbar.

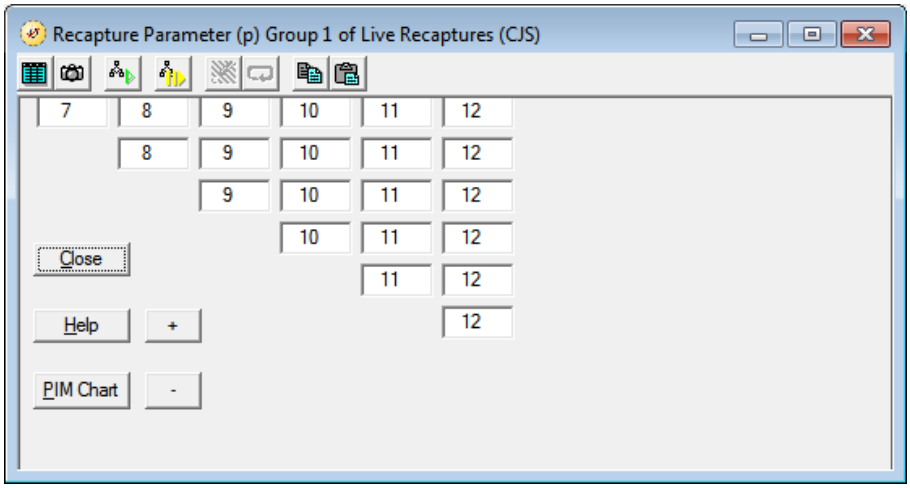
Next, bring up the *Parameter Index Matrices* (PIMs), which (as you may recall from Chapter 3), are fundamental to determining the structure of the model you are going to fit to the data. So, the first step is to open the PIMs for both the survival and recapture parameters. To do this, simply pull down the '**PIM**' menu, and select '**Open Parameter Index Matrix**'. This will present you with a dialog box containing two elements: '**Apparent Survival Parameter (Phi) Group 1**', and '**Recapture Parameter (p) Group 1**'. You want to select both of them. You can do this either by clicking on both parameters, or by simply clicking on the '**Select All**' button on the right-hand side of the dialog box.

Once you've selected both PIMs, simply click the '**OK**' button in the bottom right-hand corner. This will cause the PIMs for survival and recapture to be added to the **MARK** window.

Here's what they look like, for the survival parameters,



and the recapture parameters, respectively:



We're going to talk a **lot** more about PIMs, and why they look like they do, later on in this (and subsequent) chapters. For now, the only thing you need to know is that these PIMs reflect the currently active model. Since you deleted all the models in the browser, **MARK** reverts to the default model – which is **always** the fully time-dependent model. For mark-recapture data, this means the fully time-dependent CJS model.

OK, so now you want to fit a model. While there are some 'built-in' models in **MARK**, we'll concentrate at this stage on using **MARK** to manually build the various models we want to fit to our data. Once you've mastered this manual, more general approach, you can then proceed to using 'short-cuts' (such as built-in models). Using short-cuts before you know the 'general way' is likely to lead to one thing – you getting lost!

Looking back at the table on the first page of this chapter, we see that we want to fit 4 models to the data, $\{\varphi_t\, p_t\}$, $\{\varphi_t\, p_\cdot\}$, $\{\varphi_\cdot\, p_t\}$ and $\{\varphi_\cdot\, p_\cdot\}$. A quick reminder about model syntax – the presence of a 't' subscript means that the model is structured such that estimates for a given parameter are time-specific; in other words, that the estimates may differ over time. The absence of the 't' subscript (or, the presence of a 'dot') means the model will assume that the parameter is fixed through time (the use of the 'dot' subscript leads to such models usually being referred to as 'dot models' – naturally).

Let's consider model $\{\varphi_t\, p_t\}$ first. In this model, we assume that both apparent survival (φ) and recapture (p) can vary through time. How do we translate this into **MARK**? Pretty easy, in fact. First, recall that in this data set, we have 7 total occasions: the first occasion is the initial marking (or release) occasion, followed by 6 subsequent recapture occasions. Now, typically, in each of these subsequent recapture occasions 2 different things can occur.

Obviously, we can recapture some of the individuals previously marked. However, part of the sample captured on a given occasion is unmarked. What the investigator does with these individuals differs from protocol to protocol. Commonly, all unmarked individuals are given a unique mark, and released. As such, on a given recapture occasion, 2 types of individuals are handled and released: those individuals which have been previously marked, and those which are newly marked.

Whether or not the fate of these two 'types' of individuals is the same is something we can test (we will explore this in a later chapter). In some studies, particularly in some fisheries and insect investigations, individuals are only marked at the initial release (sometimes known as a 'batch mark'). There are no newly marked individuals added to the sample on any subsequent occasions. The distinctions between

these two types of mark-release schemes are important to understanding the structure of the parameter matrices **MARK** uses.

Consider our first model, the CJS model $\{\varphi_t\, p_t\}$ with full time-dependence in both survival and recapture probabilities. Let's assume there are no age effects (say, for example, all individuals are marked as adults – we deal with 'age' in a later chapter). In Chapter 3, we represented the parameter structure of this model as shown below:

$$\mathbf{1} \overset{\varphi_1}{\longrightarrow} \underset{p_2}{\mathbf{2}} \overset{\varphi_2}{\longrightarrow} \underset{p_3}{\mathbf{3}} \overset{\varphi_3}{\longrightarrow} \underset{p_4}{\mathbf{4}} \overset{\varphi_4}{\longrightarrow} \underset{p_5}{\mathbf{5}} \overset{\varphi_5}{\longrightarrow} \underset{p_6}{\mathbf{6}} \overset{\varphi_6}{\longrightarrow} \underset{p_7}{\mathbf{7}}$$

In fact, this representation is incomplete, since it does not record or index the fates of individuals newly marked and released at each occasion. These are referred to as 'cohorts' – groups of animals marked and released on a particular occasion.

We can do this easily by adding successive rows to our model structure, each row representing the individuals newly marked on each occasion. Since the occasions obviously occur sequentially, then each row will be indented from the one above it by one occasion. This is shown below:

$$\begin{array}{ll}
\textit{cohort 1} & \mathbf{1} \overset{\varphi_1}{\longrightarrow} \underset{p_2}{\mathbf{2}} \overset{\varphi_2}{\longrightarrow} \underset{p_3}{\mathbf{3}} \overset{\varphi_3}{\longrightarrow} \underset{p_4}{\mathbf{4}} \overset{\varphi_4}{\longrightarrow} \underset{p_5}{\mathbf{5}} \overset{\varphi_5}{\longrightarrow} \underset{p_6}{\mathbf{6}} \overset{\varphi_6}{\longrightarrow} \underset{p_7}{\mathbf{7}} \\
\textit{cohort 2} & \qquad \mathbf{2} \overset{\varphi_2}{\longrightarrow} \underset{p_3}{\mathbf{3}} \overset{\varphi_3}{\longrightarrow} \underset{p_4}{\mathbf{4}} \overset{\varphi_4}{\longrightarrow} \underset{p_5}{\mathbf{5}} \overset{\varphi_5}{\longrightarrow} \underset{p_6}{\mathbf{6}} \overset{\varphi_6}{\longrightarrow} \underset{p_7}{\mathbf{7}} \\
\textit{cohort 3} & \qquad\qquad \mathbf{3} \overset{\varphi_3}{\longrightarrow} \underset{p_4}{\mathbf{4}} \overset{\varphi_4}{\longrightarrow} \underset{p_5}{\mathbf{5}} \overset{\varphi_5}{\longrightarrow} \underset{p_6}{\mathbf{6}} \overset{\varphi_6}{\longrightarrow} \underset{p_7}{\mathbf{7}} \\
\textit{cohort 4} & \qquad\qquad\qquad \mathbf{4} \overset{\varphi_4}{\longrightarrow} \underset{p_5}{\mathbf{5}} \overset{\varphi_5}{\longrightarrow} \underset{p_6}{\mathbf{6}} \overset{\varphi_6}{\longrightarrow} \underset{p_7}{\mathbf{7}} \\
\textit{cohort 5} & \qquad\qquad\qquad\qquad \mathbf{5} \overset{\varphi_5}{\longrightarrow} \underset{p_6}{\mathbf{6}} \overset{\varphi_6}{\longrightarrow} \underset{p_7}{\mathbf{7}} \\
\textit{cohort 6} & \qquad\qquad\qquad\qquad\qquad \mathbf{6} \overset{\varphi_6}{\longrightarrow} \underset{p_7}{\mathbf{7}}
\end{array}$$

Notice that the occasions are numbered from left to right, starting with occasion 1. Survival probability is the probability of surviving between successive occasions (i.e., between columns). Each release cohort is listed in the left-hand column.

For example, some individuals are captured and marked on occasion 1, released, and potentially can survive to occasion 2. Some of these surviving individuals may survive to occasion 3, and so on. At occasion 2, some of the captured sample are unmarked. These unmarked individuals are newly marked and released at occasion 2. These animals comprise the second release cohort. At occasion 3, we take a sample from the population. Some of the sample might consist of individuals marked in the first cohort (which survived to occasion 3), some would consist of individuals marked in the second cohort (which survived to occasion 3), while the remainder would be unmarked. These unmarked individuals are newly marked, and released at occasion 3. These newly marked and released individuals comprise the third release cohort. And so on.

If we rewrite cohort structure, showing only the sampling occasion numbers, we get the structure shown at the top of the next page.

cohort 1	**1**	**2**	**3**	**4**	**5**	**6**	**7**
cohort 2		**2**	**3**	**4**	**5**	**6**	**7**
cohort 3			**3**	**4**	**5**	**6**	**7**
cohort 4				**4**	**5**	**6**	**7**
cohort 5					**5**	**6**	**7**
cohort 6						**6**	**7**

The first question that needs to be addressed is: does survival vary as a function of which cohort an individual belongs to, does it vary with time, or both? This will determine the indexing of the survival and recapture parameters. For example, assume that cohort does not affect survival, but that survival varies over time. In this case, survival can vary among intervals (i.e., among columns), but over a given interval (i.e., within a column), survival is the same over all cohorts (i.e., over all rows). Again, consider the following cohort matrix – but showing only the survival parameters:

cohort 1	**1**	$\xrightarrow{\varphi_1}$	**2**	$\xrightarrow{\varphi_2}$	**3**	$\xrightarrow{\varphi_3}$	**4**	$\xrightarrow{\varphi_4}$	**5**	$\xrightarrow{\varphi_5}$	**6**	$\xrightarrow{\varphi_6}$	**7**
cohort 2			**2**	$\xrightarrow{\varphi_2}$	**3**	$\xrightarrow{\varphi_3}$	**4**	$\xrightarrow{\varphi_4}$	**5**	$\xrightarrow{\varphi_5}$	**6**	$\xrightarrow{\varphi_6}$	**7**
cohort 3					**3**	$\xrightarrow{\varphi_3}$	**4**	$\xrightarrow{\varphi_4}$	**5**	$\xrightarrow{\varphi_5}$	**6**	$\xrightarrow{\varphi_6}$	**7**
cohort 4							**4**	$\xrightarrow{\varphi_4}$	**5**	$\xrightarrow{\varphi_5}$	**6**	$\xrightarrow{\varphi_6}$	**7**
cohort 5									**5**	$\xrightarrow{\varphi_5}$	**6**	$\xrightarrow{\varphi_6}$	**7**
cohort 6											**6**	$\xrightarrow{\varphi_6}$	**7**

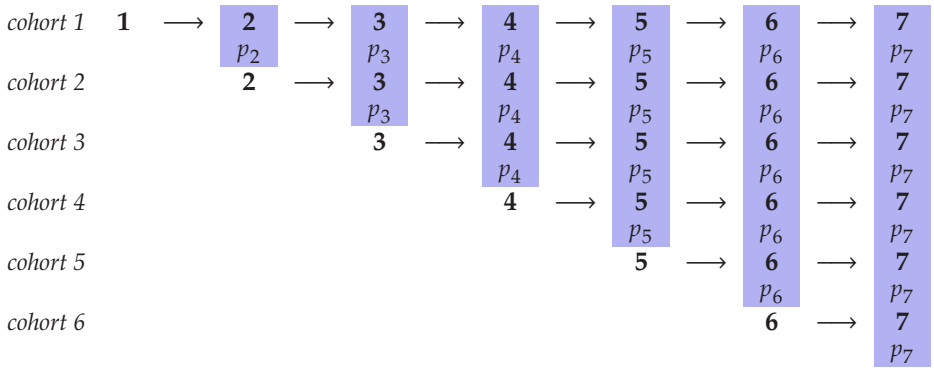
The shaded columns indicate that survival is constant over cohorts, but the changing subscripts in φ_i indicate that survival may change over time. This is essentially Table 7A in Lebreton *et al*. (1992). What **MARK** does to generate the parameter or model structure matrix is to reproduce the structure and dimensions of this figure, after first replacing the φ_i values with a simple numerical indexing scheme, such that φ_1 is replaced by the number 1, φ_2 is replaced by the number 2, and so forth. Thus, the preceding figure (above) is represented by a triangular matrix of the numbers 1 to 6 (for the 6 survival probabilities):

1	2	3	4	5	6
	2	3	4	5	6
		3	4	5	6
			4	5	6
				5	6
					6

This 'triangular matrix' (the PIM) represents the way that **MARK** 'stores' the model structure corresponding to time variation in survival, but no cohort effect (Fig. 7A in Lebreton *et al*. 1992). Notice that the dimension of this matrix is (6 rows by 6 columns), rather than (7 columns by 7 rows). This is because there are 7 capture occasions, but only 6 survival intervals (and, correspondingly, 6 recapture occasions). This representation is the basis of the PIMs which you see on your screen (it will also be printed in the output file). Perhaps most importantly, though, this format is the way **MARK** keeps

track of model structure and parameter indexing. It is essential that you understand the relationships presented in the preceding figures. A few more examples will help make them clearer.

Let's consider the recapture probability. If recapture probability is also time-specific, what do you think the model structure would look like? If you've read **and** understood the preceding, you should be able to make a reasonable guess. Again, remember that we have 7 sampling occasions – the initial marking event (occasion 1), and 6 recapture occasions. With time-dependence, and assuming no differences among cohorts, the model structure for recaptures would be:



Now, what are the corresponding index values for the recapture probabilities? As with survival, there are 6 parameters, p_2 to p_7 (corresponding to recapture on the second through seventh occasion, respectively). With survival probabilities, we simply looked at the subscripts of the parameters, and built the PIM. However, things are not quite so simple here (although as you'll see, they're not very hard). All you need to know is that the recapture parameter index values start with the first value after the survival values. Hmmm...let's try that another way. For survival, we saw there were 6 parameters, so our survival PIM looked like

1	2	3	4	5	6
	2	3	4	5	6
		3	4	5	6
			4	5	6
				5	6
					6

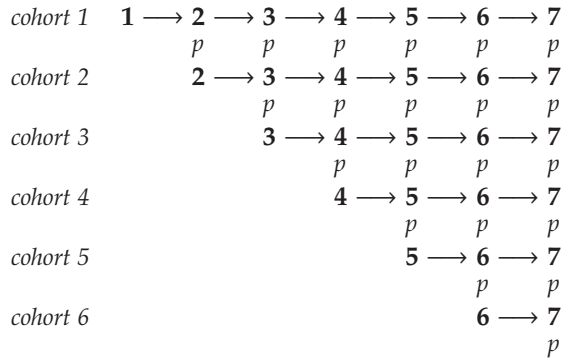
The last index value is the number '6' (corresponding to φ_6, the apparent survival probability between occasion 6 and occasion 7). To build the recapture PIM, *we start with the first value after the largest value in the survival PIM*. Since '6' is the largest value in the survival PIM, then the first index value used in the recapture PIM will be the number '7'. Now, we build the rest of the PIM. What does it look like?

If you think about it for a moment, you'll realize that the recapture PIM looks like:

7	8	9	10	11	12
	8	9	10	11	12
		9	10	11	12
			10	11	12
				11	12
					12

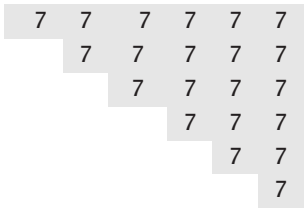
Do these look familiar? They might – look at the PIMs **MARK** has generated on the screen. In fact, we're now ready to 'run the CJS model' fully time-dependent model. We covered this step in Chapter 3, but let's go through it again (repetition is a good teacher). In fact, there are a couple of ways you can proceed. You can either (i) pull down the '**Run**' menu and '**Run the current model**' (the model defined by the PIMs is always the current model), or (ii) click the '**Run**' icon on the toolbar of either of the PIMs. This will bring up the '**Setup**' window for the numerical estimation, which you saw for the first time in Chapter 3. All you need to do is fill in a name for the model (we'll use `Phi(t)p(t)` for this model), and click the '**OK to run button**' (lower right-hand corner). Again, as you saw in Chapter 3, **MARK** will ask you about the 'identity matrix', and then spawn a numerical estimation window. Once it's finished, simply add these results to the results browser.

Now, let's consider model $\{\varphi_t\, p_.\}$ – time-dependent survival, but constant recapture probability. What would the PIMs for this model look like? The survival PIM would be identical to what we already have, so no need to do anything there. What about the recapture PIM? Well, in this case, we have constant recapture probability. What does the parameter structure look like? Look at the following figure:



Note that there are no subscripts for the recapture parameters – this reflects the fact that for this model, we're setting the recapture probability to be constant, both among occasions, and over cohorts.

What would the PIM look like for recapture probability? Recall that the largest index value for the survival PIM is the number '6', so the first index value in the recapture PIM is the number '7'. And, since the recapture probability is constant for this model, then the entire PIM will consist of the number '7':

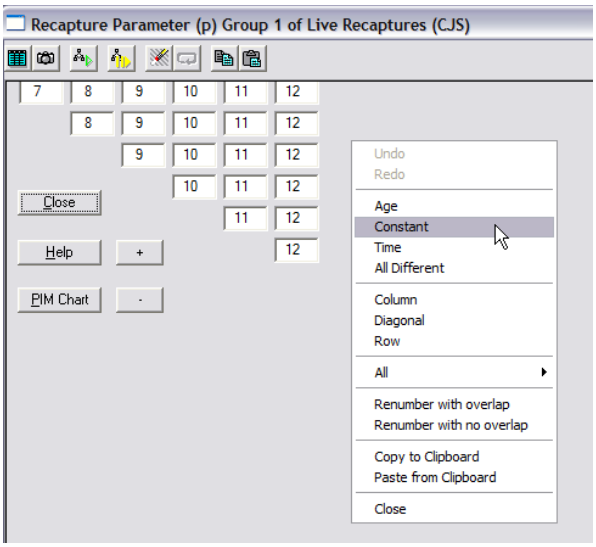


Now, how do we modify the PIMs in **MARK** to reflect this structure? As you'll discover, **MARK** gives you many different ways to accomplish the same thing. Modifying PIMs is no exception. The most obvious (and pretty well fool-proof) way to modify the PIM is to edit the PIM directly, changing each cell in the PIM, one at a time, to the desired value. For small PIMs, or for some esoteric model structures we'll discuss in later chapters, this is not a bad thing to try. Here, we'll use one of the built-in time-savers in **MARK** to do most of the work for us.

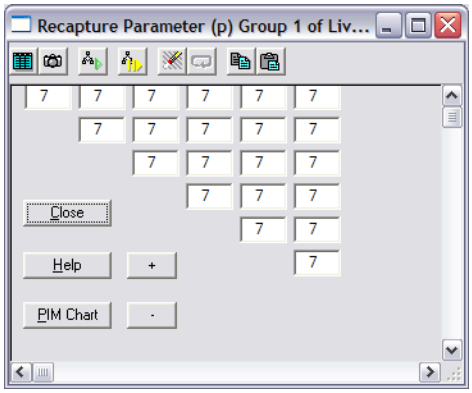
Remember, all we want to do here is modify the recapture PIM. To do this, make that PIM 'active' by clicking in the first 'cell' (upper left corner of the PIM). You can in fact make a window active by

clicking on it anywhere (it doesn't matter where – just remember not to click the '**X**' in the upper right-hand corner, since this will close the window!), but as we'll see, there are advantages in clicking in a specific cell in the PIM. When you've successfully selected a cell, you should see a vertical cursor in that cell.

Once you've done this, you can do one of a couple of things. You can pull down the '**Initial**' menu on the main **MARK** parent toolbar. When you do this, you'll see a number of options – each of them controlling the value (if you want, the initial value) of some aspect of the active window (in this case, the recapture PIM). Since we want to have a constant recapture probability, you might guess the '**Constant**' option on the '**Initial**' menu would be the right one. You'd be correct. Alternatively, you can right-click with the mouse anywhere in the recapture PIM window – this will generate the same menu as you would get if you pull down the '**Initial**' menu. Use whichever approach you prefer:



Once you've done this, you will see that all the values in the recapture PIM are changed to 7.



Believe it our not, you're now ready to run this model (model $\{\varphi_t\, p_.\}$). Simply go ahead and click the '**Run**' icon in the toolbar of either PIM. For a model name, we'll use '`Phi(t)p(.)`'. Once **MARK** is finished, go ahead and append the results from this run to the results browser.

What you might see is that model '`Phi(t)p(.)`' (representing model $\{\varphi_t\, p_.\}$) is listed first, and model

'Phi(t)p(t)' second, even though model 'Phi(t)p(t)' was actually run first. As you may recall from our discussions in Chapter 3, the model ordering is determined by a particular criterion (say, the AIC), and not necessarily the order in which the models were run.

Before we delve too deeply into the results of our analyses so far, let's finish our list of candidate models. We have model $\{\varphi_. \, p_t\}$ and model $\{\varphi_. \, p_.\}$ remaining. Let's start with model $\{\varphi_. \, p_t\}$ – constant survival, and time-dependent recapture probability. If you think about it for a few seconds, you'll realize that this model is essentially the 'reverse' of what we just did – constant survival and time-dependent recapture, instead of the other way around. So, you might guess that all you need to do is reverse the indexing in the survival and recapture PIMs. Correct again! Start with the survival PIM. Click in the first cell (upper left-hand corner), and then pull down the '**Initial**' menu and select '**Constant**', as you did previously. The survival PIM will change to a matrix of all '1's.

What about the recapture PIM? Again, click in the first cell. Since we're reusing the PIMs from our last analysis, the value of the first cell in the recapture PIM will be the number '7'. If we pull down the '**Initial**' menu and select '**Time**', we'd see the matrix change from all '7's to values from 7 to 12. Now, think about this for a minute. If we stop at this point, we'd be using the parameter index '1' for survival (constant survival), and indices 7 through 12 for recapture probability. What happened to indices 2 through 6?

In fact, nothing has really happened to them – but you don't know that yet. While it might make more sense to explicitly index the recapture PIM from 2 through 7, in fact, **MARK** will do this for you – but only during the numerical estimation itself. In fact, **MARK** more or less assumes that you've used '1' and '7 through 12' as the index values by mistake, and actually uses '1' and '2 through 7' when it does its calculations.

Let's prove this to ourselves. Leave the PIMs as they are now – all '1's for survival, and '7 through 12' for recapture, and press the '**Run**' icon on the PIM toolbar. You'll be dumped into the '**Numerical Estimation**' setup window. For a title, we'll use 'Phi(.)p(t)'. Then, simply click on the '**OK to Run**' button. Append the results to the browser. Now, before looking at the browser itself, have another look at both the survival and recapture PIMs. What you'll see is that **MARK** has 'corrected' the PIM indexing for the recapture PIM, such that it is now '2 through 7', rather than '7 through 12'.

Clever, eh? Yes, and no. It is nice that **MARK** does this for you, but you should **not** rely on software to do too much 'thinking' for you. It would have been better (albeit, somewhat slower) to simply index the recapture PIM correctly in the first place.

How would you do this? Aaah. . .this is where selecting the first cell in the PIM itself becomes important. Initially, the recapture PIM had all '7' values. You want to change it to time-dependent, but want the first value to be '2' (since the survival parameter index is '1'). To do this, simply change the value in the first cell of the recapture PIM from '7' to '2', and then select '**Time**' from the '**Initial**' menu. Let's put this into practice with the final model in our candidate model set – the constant survival and recapture model, $\{\varphi_. \, p_.\}$. For this model, the survival and recapture PIMs will be '1's and '2's, respectively.

begin sidebar

Why not '2's?

Why not use 2's for indexing survival, and 1's for recapture? In fact, **MARK** doesn't care at all – in **MARK**, the ordering or sequencing of PIM indexing is as arbitrary as which window you place where on the screen – it's entirely up to you. You would get the same 'results' using either '1' and '2' for survival and recapture, or '2' and '1', respectively.

end sidebar

In other words, for survival

1	1	1	1	1	1
	1	1	1	1	1
		1	1	1	1
			1	1	1
				1	1
					1

and for recapture

2	2	2	2	2	2
	2	2	2	2	2
		2	2	2	2
			2	2	2
				2	2
					2

One simple way to remember what the triangular matrix is telling you is to remember that time moves left to right, and cohort from top to bottom. If the numbers (indices) change in value from left to right, then the parameter changes with time. If they change from top to bottom, they change over cohort. Of course, the indices can change in either one or both directions simultaneously.

Once the PIMs are set, run the model (we'll use '`Phi(.)p(.)`' for the model name), and append the results to the browser. You now have 4 different model results in the browser, corresponding to each of the 4 models we're interested in. Of course, there are **many** other models we could have tried, but at this stage, we simply want to get comfortable building models in **MARK**. As we'll discuss shortly, the selection of the candidate set of models is crucial to our task. For now though, let's just consider these 4 as representative of models we have an interest in. Let's start by looking at the results browser itself:

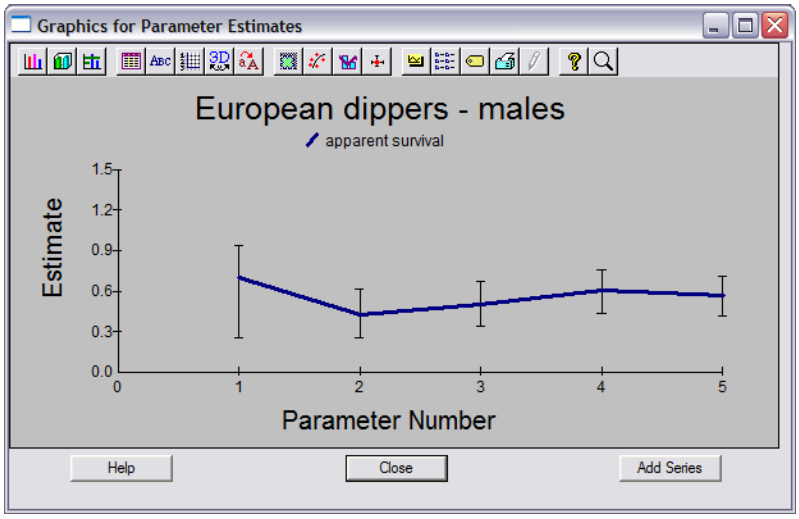
Results Browser: Live Recaptures (CJS)

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{phi(.)p(.)}	322.5527	0.0000	0.96003	1.0000	2	41.8147
{phi(t)p(.)}	330.0567	7.5040	0.02253	0.0235	7	38.8147
{phi(.)p(t)}	330.6794	8.1267	0.01650	0.0172	7	39.4374
{phi(t)p(t)}	336.4343	13.8816	0.00093	0.0010	11	36.4013

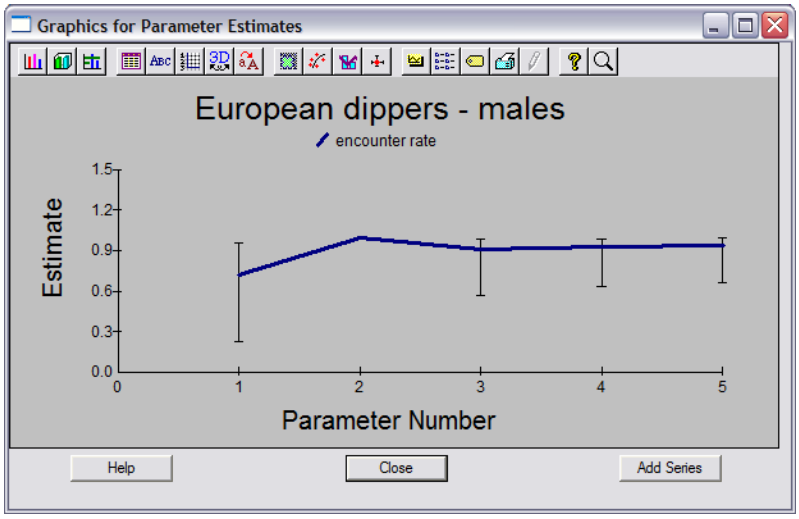
We see that the 4 models are listed, in order of ascending AIC, ranging from 322.55 for model $\{\varphi_. \, p_.\}$ to 336.43 for model $\{\varphi_t \, p_t\}$.

Before we evaluate the results from our 4 models, it is often a good starting point to take the estimates from the most fully parameterized model, and plot them (**MARK** has some basic 'built-in' plotting capabilities – simply click the '**`line graph`**' icon in the browser toolbar, and go from there. Fairly self-explanatory). Often, a sense of the underlying model structure is revealed by examination of the estimates from the most parameterized model. The reason is fairly straightforward – the more parameters in the model, the better the fit (smaller the deviance – more on model deviance later on). As

we will discuss shortly, this does not necessarily mean that it is the best model, merely the one that fits the best (this is a crucial distinction). However, the most parameterized model generally gives the most useful 'visual' representation of the pattern of variation in survival and recapture. In the case of our 4 models, the most parameterized is model $\{\varphi_t\, p_t\}$ – the CJS model. The parameter estimates (with 95% CI) for φ and p are plotted below – first, the estimated apparent survival probabilities ($\hat{\varphi}_i$),



and then, the estimated recapture probabilities ($\hat{p}_i$)



Note that in these figures we do not include all 6 estimates that **MARK** derives for both survival and recapture. Why? As it turns out, the final estimate for both survival and recapture is 0.7638. Is this just a coincidence? No! In fact, what **MARK** has done is estimate the square-root of the combined probability $\varphi_6 p_7$ (which Lebreton *et al.* (1992) refer to as β_7). For the time-dependent CJS model, the components of this product are not individually identifiable – without further information, we cannot separately estimate survival from recapture – we can only estimate the square-root of the product. We shall discuss this again in more detail later on. Since this $\varphi_6 p_7$ product term is not comparable to either survival or recapture probabilities separately, it is excluded from our plots. Of course, if you've

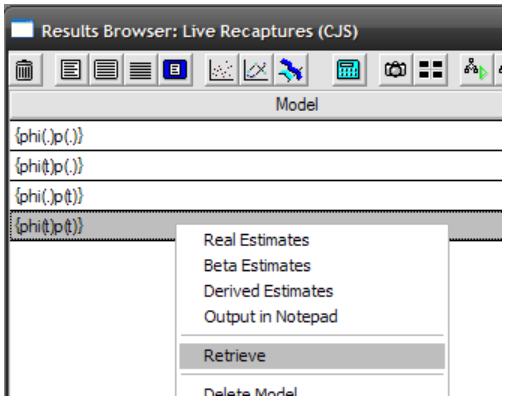
looked at the output listing already, you may have 'seen' that parameters 6 and 12 are not separately identifiable. However, as we've mentioned before, we do not favor unquestioning reliance on the ability of the software (be it **MARK** or any other application) to determine the number of estimable parameters – you should first develop an understanding of how it is done from first principals. This is covered in the Addendum at the end of this chapter. [We have placed it there to minimize disruption to the flow of the material on building models. However, you are strongly urged to read it carefully, at some point].

4.2. A quicker way to build models – the PIM chart

In the preceding example, we started our model building by opening the PIMs for both survival and recapture (the two primary parameters in the live encounter analysis of the male dipper data). We modified the PIMs to set the underlying parameter structure for our models. However, at this point, we want to show you another, more efficient way to do the same thing, by introducing one of the 'whiz-bang' (from the Latin) features of **MARK** – the *Parameter Index Chart* (hereafter referred to as the 'PIM chart'). Introducing the PIM chart, and demonstrating its utility is probably most effectively done by letting you have a look. We'll do so by considering two different numerical examples – one involving a single group of marked individuals, and the second involving multiple groups of marked individuals. Not only is the situation involving multiple groups quite common, but some of the notable advantages of using the PIM chart to speed model construction are even more obvious for analyses involving multiple groups of marked individuals.

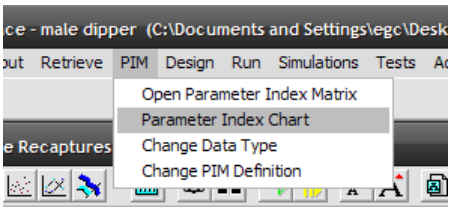
4.2.1. PIM charts and single groups – European dipper re-visited

Open up the male dipper data analysis – there should be 4 models in the results browser. We're going to replicate these 4 models again, this time using the PIM chart, rather than manually modifying the individual PIMs. We'll start with model $\{\varphi_t\, p_t\}$. Simply find that model in the results browser, right-click it and select '**`Retrieve`**' from the sub-menu. This will make the underlying PIM structure for this model active (you can always check this by opening up each of the individual PIMs and having a look).

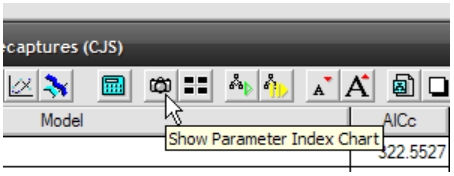


Recall that for model $\{\varphi_t\, p_t\}$, there are 6 intervals for survival, and 6 occasions for recapture – so, in theory, 12 total parameters that could be estimated: $1 \rightarrow 6$ for survival, and $7 \rightarrow 12$ for recapture (although we remember that for the fully time-dependent model, the final survival and recapture parameters are confounded – such that only 11 total parameters will be estimated). Recall that at this point, we're simply setting the underlying parameter structure for our models.

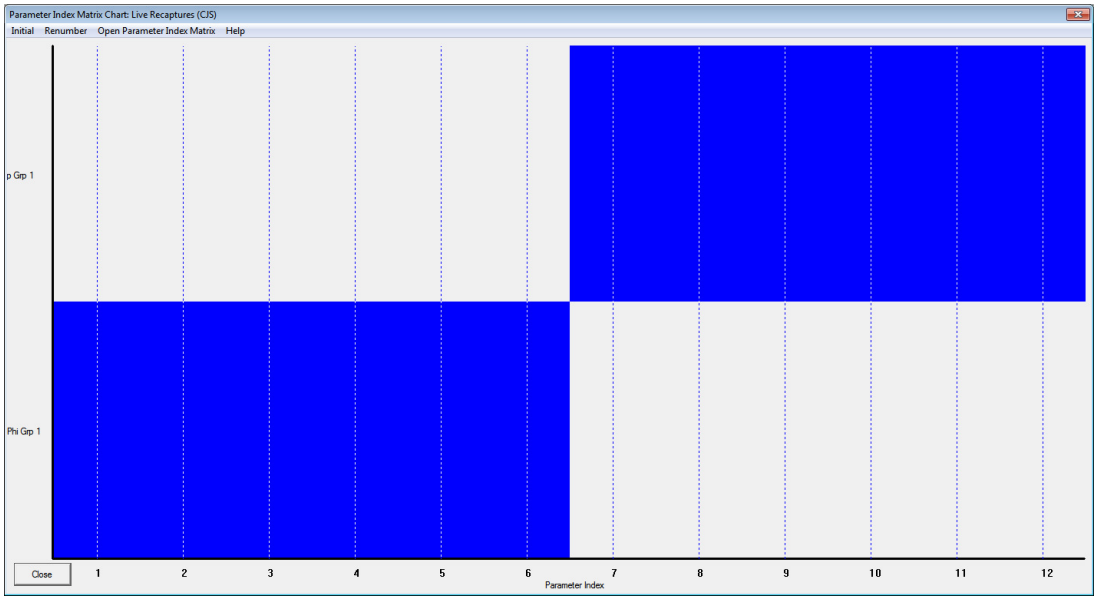
Now, let's have a look at this thing called the PIM chart. You can either (i) select '**`Parameter Index Chart`**' from the PIM menu



or (ii) click the appropriate icon in the main toolbar (for the PIM chart, this is the icon that looks like a 'camera', more or less).



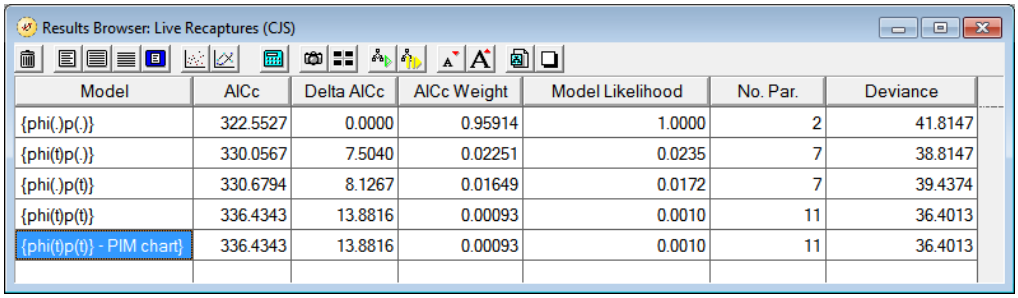
Go ahead and open up the PIM chart for model $\{\varphi_t\, p_t\}$:



Zowie! OK. . .now, what is it? The PIM chart is a simple, yet very useful visual tool for looking at the parameter indexing **MARK** uses for the various groups and parameters in the current model (in our case, model $\{\varphi_t\, p_t\}$). What you can see from the chart is that there are 2 main 'groupings' of parameters for this model: survival (φ) respectively, and recapture (p), respectively. Along the bottom axis is the parameter index itself, and along the vertical axis are the parameters. So, in this example, parameters 1 to 6 refer to the survival probabilities, and 7 to 12 correspond to the recapture parameters.

Now, at this point we haven't changed anything to the parameter structure – the PIM chart simply reflects the structure of the current model. You can confirm that in fact nothing has changed by running

this model, and adding the result to the browser. Make the title for the model '`Phi(t)p(t) - PIM chart`' – adding the extra label will help you identify which models in the browser were created using the PIM chart.


Results Browser: Live Recaptures (CJS)

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{phi(.)p(.)}	322.5527	0.0000	0.95914	1.0000	2	41.8147
{phi(t)p(.)}	330.0567	7.5040	0.02251	0.0235	7	38.8147
{phi(.)p(t)}	330.6794	8.1267	0.01649	0.0172	7	39.4374
{phi(t)p(t)}	336.4343	13.8816	0.00093	0.0010	11	36.4013
{phi(t)p(t)} - PIM chart}	336.4343	13.8816	0.00093	0.0010	11	36.4013

As you can see, the results for model '`Phi(t)p(t) - PIM chart`' are identical to those for model '`Phi(t)p(t)`'.

OK, so far, the PIM chart hasn't really done much for us, other than provide a convenient visual representation of the underlying parameter structure for our model. In fact, the greatest utility of the PIM chart is the ease with which you can use it to build other models. We'll demonstrate that now. Let's consider building model $\{\varphi_t\, p_.\}$. How can we build this model using the PIM chart? Recall that for this model, the underlying parameter structure for survival should look like

1	2	3	4	5	6
	2	3	4	5	6
		3	4	5	6
			4	5	6
				5	6
					6

and for recapture

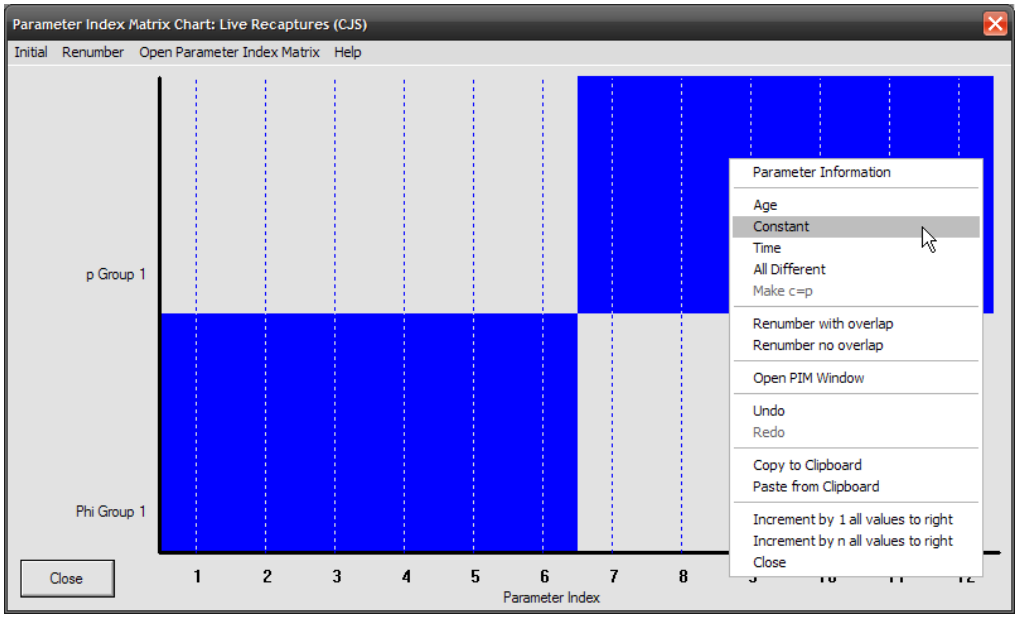
7	7	7	7	7	7
	7	7	7	7	7
		7	7	7	7
			7	7	7
				7	7
					7

However, the recapture PIM for our current model $\{\varphi_t\, p_t\}$ has the structure

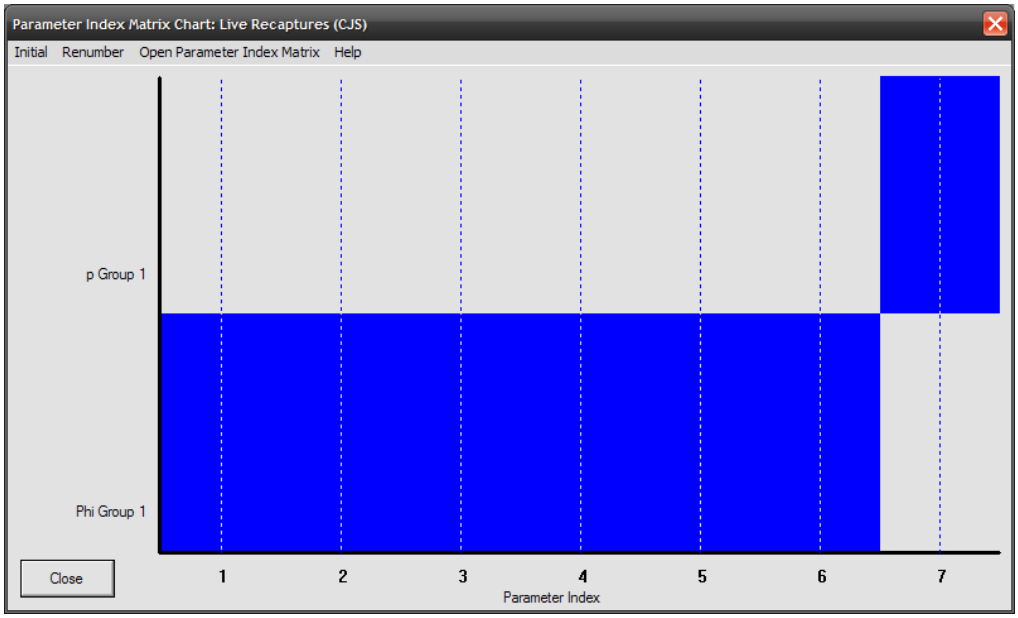
7	8	9	10	11	12
	8	9	10	11	12
		9	10	11	12
			10	11	12
				11	12
					12

So, we want to change the recapture PIM from time-dependent (index values $7 \rightarrow 12$) to time-invariant (constant; index values 7 only for all occasions).

How do we do this using the PIM chart? Easy! Simply open up the PIM chart, and right click on the 'blue-box' corresponding to the recapture parameter. This will spawn a sub-menu which list various options – the option we want to select is '**Constant**' (shown below):



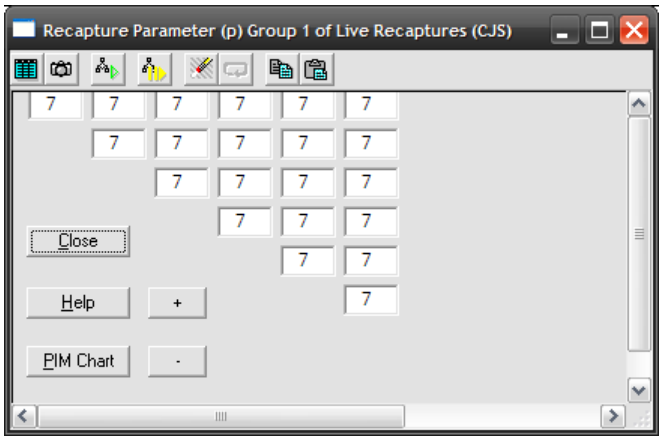
What this will do is change the parameter structure for the selected 'blue box' (which in this case represents the recapture parameter) and change it from the current structure (in this case, time-dependent) to constant. Go ahead and select '**Constant**':



Now, the right-most blue box has only one parameter index – '7'. The size of the box has changed

to reflect that we've gone from full time-dependence (6 parameters wide) to a constant 'dot' model (1 parameter wide).

We can confirm this by opening up the recapture PIM:

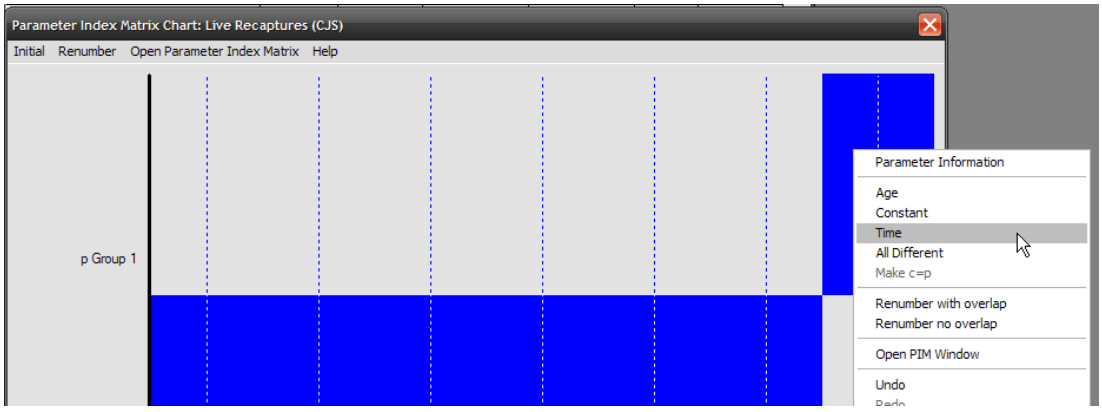


As expected, it consists entirely of the number '7'.

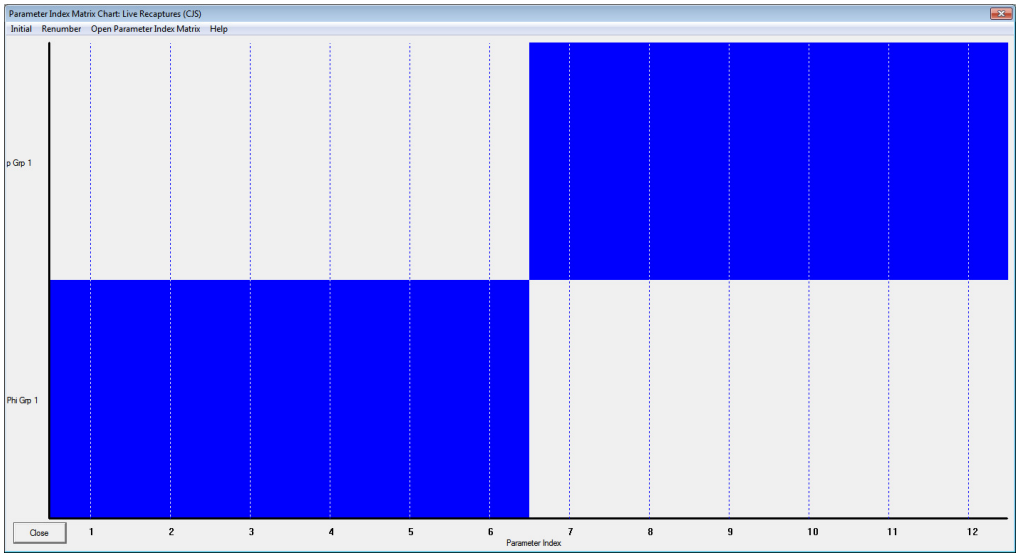
Now, wasn't that fast? To build model $\{\varphi_t\, p_.\}$ from model $\{\varphi_t\, p_t\}$, all we did was (i) open up the PIM chart for model $\{\varphi_t\, p_t\}$, (ii) right-click the 'blue box' corresponding to the recapture parameter, and (iii) select constant.

Go ahead and run this model – label it '`phi(t)p(.) - PIM chart`', and add the results to the browser. The results should be identical to those from model '`phi(t)p(.)`', which we fit by manually modifying the parameter-specific PIMs.

Now, what about model $\{\varphi_.\, p_t\}$? At this point we could either go back into the browser, retrieve model $\{\varphi_t\, p_t\}$, bring up the PIM chart, and repeat the steps we just took, except right-clicking on the survival 'blue box', rather than the recapture 'blue box'. Alternatively, we could simply bring up the PIM chart for the model we just fit $\{\varphi_t\, p_.\}$ and modify that. We'll use the second approach. Go ahead and bring back up the PIM chart. Now, we're going to right-click on the recapture 'blue-box', and change it from constant to time-dependent:

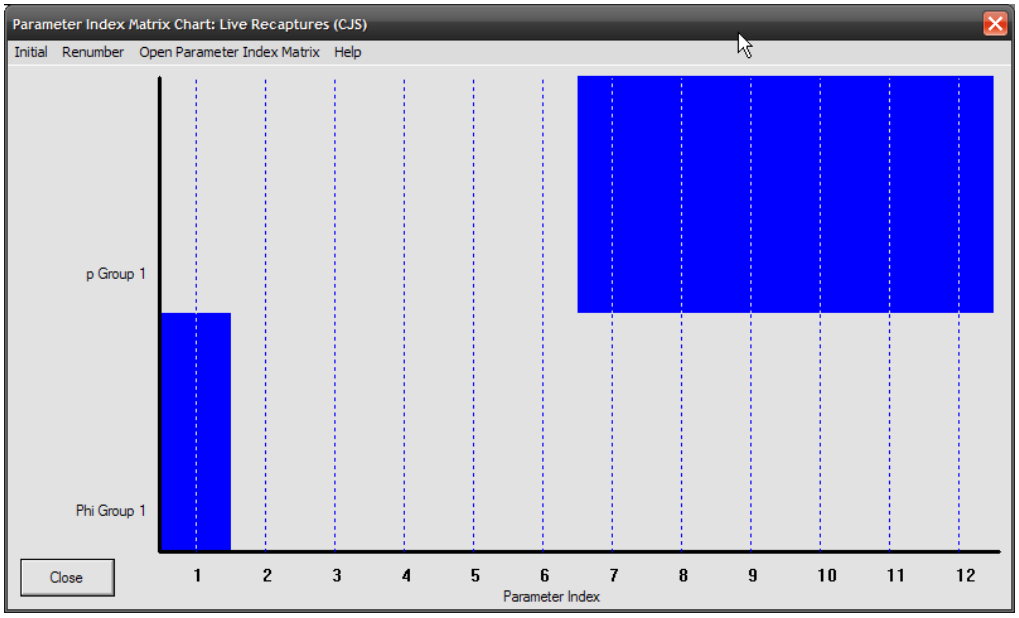


This will generate a PIM chart that looks like



Recognize it? You should – it's the PIM chart corresponding to the model we started with $\{\varphi_t\, p_t\}$ – which has 6 survival parameters, and 6 recapture parameters.

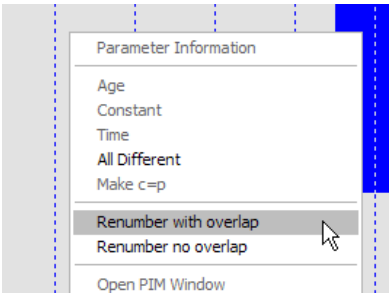
Now, right-click on the survival 'blue-box', and select '**`Constant`**'. Remember, we're trying to build model $\{\varphi_\cdot\, p_t\}$.



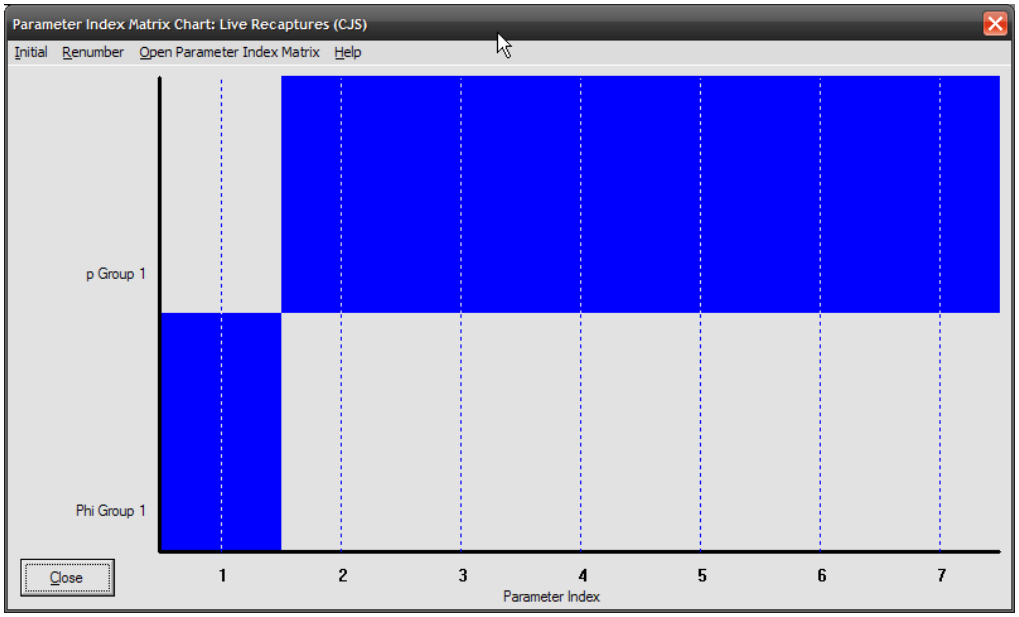
So, a couple of things to notice here. First, as intended, the 'blue box' for the survival parameter has 'shrunk', reflecting the fact that we've the structure for φ from 'time-dependent' (parameters $1 \rightarrow 6$) to 'constant' (parameter 1).

But, we also notice there is a substantial 'gap' between the two 'blue-boxes'. Parameter index values $2 \rightarrow 6$ don't correspond to anything. We want to eliminate the gap (i.e., remove the meaningless index values). You could do this in one of two ways. First, the PIM chart lets you manually 'drag' the 'blue-boxes' around. So, you could left-click the recapture 'blue-box' and, while holding down the left mouse button, drag the recapture blue box to the left, so that the left-most edge of the box corresponds to parameter index 2. Try it, it's pretty slick.

Alternatively, you can right-click anywhere on the PIM chart, and select either of the '**Renumber**' options you are given (the distinction between the two will become obvious in our next worked example):



Doing so will cause the PIM chart to change (shown at the top of the next page) – the gap between the two 'blue boxes' will be eliminated, and the structure will now reflect model $\{\varphi_{\cdot}\, p_t\}$.



Go ahead and run the model, label it '`phi(.)p(t) - PIM chart`', and add the results to the browser. Again, they should be identical to the results from fitting model '`phi(.)p(t)`' built by modifying the individual PIMs. Again, using the PIM chart is often much faster.

As a final test, try fitting model $\{\varphi_{\cdot}\, p_{\cdot}\}$. You'll know you've done it correctly if the results match those for '`phi(.)p(.)`' already in the browser.

4.2.2. PIM charts and multiple groups

This second worked example involves some data from two different nesting colonies of the swift (*Apus apus*), a small insectivorous bird. In addition to providing an opportunity to demonstrate the use of the PIM chart, this data set also introduces the general concept of comparing groups (an extremely common type of analysis you're likely to run into routinely). In fact, as we will see, it involves only slightly more work than the model comparisons we saw in the European dipper example we considered in the first part of this chapter.

The data consist of live encounter data collected over an 8 year study of 2 different swift colonies in southern France. One of the two colonies was believed to be of 'poorer' quality than the other colony for a variety of reasons, and the purpose of the study was to determine if these perceived differences between the two colonies (hereafter, **P** – 'poor', and **G** – 'good') were reflected in differences in either survival or recapture probability. The data for both the **P** and **G** colonies are both in **`aa.inp`** – the encounter frequencies are tabulated for the 'poor' and 'good' colonies, respectively. In this example, we will analyze the data in terms of the following 2 factors: colony (**G** or **P**), and time. Thus, this example is very similar to the European dipper example, except that we have added one more factor, colony. As such, the number of possible models is increased from $(2)^2 = 4$ models to $(4)^2 = 16$ models – survival and recapture could vary with colony, time or both. The candidate set of models is shown below:

$$\begin{array}{cccc}
\{\varphi_{c*t}\, p_{c*t}\} & \{\varphi_{c}\, p_{c*t}\} & \{\varphi_{t}\, p_{c*t}\} & \{\varphi_{.}\, p_{c*t}\} \\
\{\varphi_{c*t}\, p_{c}\} & \{\varphi_{c}\, p_{c}\} & \{\varphi_{t}\, p_{c}\} & \{\varphi_{.}\, p_{c}\} \\
\{\varphi_{c*t}\, p_{t}\} & \{\varphi_{c}\, p_{t}\} & \{\varphi_{t}\, p_{t}\} & \{\varphi_{.}\, p_{t}\} \\
\{\varphi_{c*t}\, p_{.}\} & \{\varphi_{c}\, p_{.}\} & \{\varphi_{t}\, p_{.}\} & \{\varphi_{.}\, p_{.}\}
\end{array}$$

With an increasing number of factors, the number of possible models that may need to be tested increases geometrically. Here we have an 8 year study, considering only 2 primary factors (colony and time), and there are at least 16 possible models to test (in fact, we will see in subsequent chapters there are potentially many more).

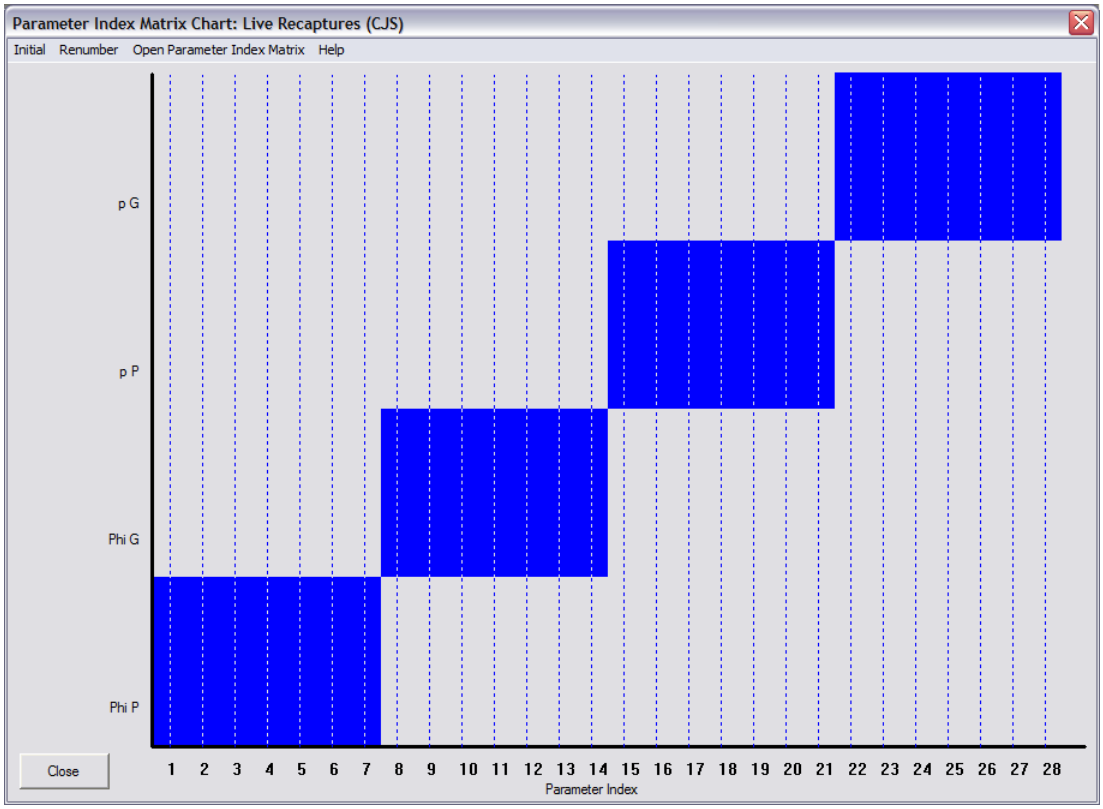
Two points to make before we go any further. First, we should make sure you understand the syntax of the model representations in the preceding table (which follow the approach recommended in Lebreton *et al*. 1992). Remember, that the subscripts for the two parameters (φ and p) reflect the structure of the model. The most general model in the table is model $\{\varphi_{c*t}\, p_{c*t}\}$. The '`c*t`' subscript means we have a 'full' model (for both survival and recapture), including both the main effects (`colony` and `time`) and the interaction of the two (i.e., '`c*t`' = `c + t + c.t + error`). By 'interaction', we are referring to the statistical meaning of the word – that `colony` and `time` interact, such that the relationship between survival or recapture and time can differ depending upon the colony (or conversely, the relationship between survival or recapture and `colony` can differ depending upon the time interval). This model is the most general, since any of the other models listed can be derived by removing one or more factors (a simple comparison of the 'complexity' of the subscripting for both survival and recapture among the various models will confirm this).

Second, by now you've no doubt noticed that we highlighted the word 'possible' repeatedly. We did so for a reason – to foreshadow discussion of 'model selection' and 'model uncertainty', presented later in this chapter. For the moment, we'll assume that we are particularly interested in whether or not there are differences in survival between the 2 colonies. We'll assume that there are no differences in recapture probability between the colonies. These assumptions are reflected in our 'candidate model set', which is a subset of the table presented above:

$$\begin{array}{ccc}
\{\varphi_{c*t}\, p_{t}\} & \{\varphi_{t}\, p_{t}\} & \{\varphi_{c}\, p_{t}\} \\
\{\varphi_{c*t}\, p_{.}\} & \{\varphi_{t}\, p_{.}\} & \{\varphi_{c}\, p_{.}\}
\end{array}$$

Note that this candidate model set reflects some 'prior thinking' about the data set, the analysis, the biology – different investigators might come up with different candidate model sets. However, for the moment, we'll use this particular candidate model set, and proceed to analyze the data. We will start by fitting the data to the most *general* approximating model in the model set $\{\varphi_{c*t}\, p_t\}$. It is the most general, because it has the most parameters of all the models we will consider. Start program **MARK**, and start a '**New**' project (i.e., from the '**File**' menu, select '**New**'). Pull in the data from **aa.inp** (2 groups, 8 occasions – the first frequency column represents the 'poor' colony, while the second frequency column represents the 'good' colony).

Next, either pull down the '**PIM**' menu, and select '**Parameter Index Chart**', or select the PIM chart icon on the toolbar. The resulting (default) PIM chart corresponds to model $\{\varphi_{c*t}\, p_{c*t}\}$ is shown at the top of the next page. Again, what you can see from the chart is that there are 4 main 'groupings' of parameters for this model: survival for good and poor colonies respectively, and recapture for the good and poor colonies, respectively. Along the bottom index is the parameter index itself, and along the vertical axis are the parameters and group labels. So, in this example, parameters 1 to 7 refer to the survival probabilities for the poor colony, 8 to 14 correspond to the survival parameters for the good colony, and so on. As mentioned in the European dipper example we just completed, the PIM chart allows you to quickly determine the structure of your model, in terms of the parameters indices.



However, before we proceed, think back for a moment to our candidate model set. In the model set, the most general model we want to fit is model $\{\varphi_{c*t}\, p_t\}$. However, the default model **MARK** starts with is always the fully structured model, in this case, model $\{\varphi_{c*t}\, p_{c*t}\}$. So, as a first step, we need to reconfigure the default model from $\{\varphi_{c*t}\, p_{c*t}\}$ to $\{\varphi_{c*t}\, p_t\}$. This involves changing the parameter structure for the recapture parameters, by eliminating the colony effect.

This should be reasonably straightforward. We have 8 occasions, and 2 groups. Thus, for a given group, we have 7 survival intervals, and 7 recapture occasions. For model $\{\varphi_{c*t}\, p_{c*t}\}$, the parameters would be numbered:

survival		*recapture*	
poor	good	poor	good
$1 \rightarrow 7$	$8 \rightarrow 14$	$15 \rightarrow 21$	$22 \rightarrow 28$

In other words, the PIMs would look like the following for survival:

1	2	3	4	5	6	7
	2	3	4	5	6	7
		3	4	5	6	7
			4	5	6	7
				5	6	7
		survival			6	7
		'poor'				7

8	9	10	11	12	13	14
	9	10	11	12	13	14
		10	11	12	13	14
			11	12	13	14
				12	13	14
		survival			13	14
		'good'				14

and for recapture

15	16	17	18	19	20	21
	16	17	18	19	20	21
		17	18	19	20	21
			18	19	20	21
				19	20	21
		recapture			20	21
		'poor'				21

22	23	24	25	26	27	28
	23	24	25	26	27	28
		24	25	26	27	28
			25	26	27	28
				26	27	28
		recapture			27	28
		'good'				28

Now, what we want to do is modify this structure to reflect model $\{\varphi_{c*t}\, p_t\}$ – in other words, we want to change the recapture PIMs so that they are the same between the two groups (the two colonies):

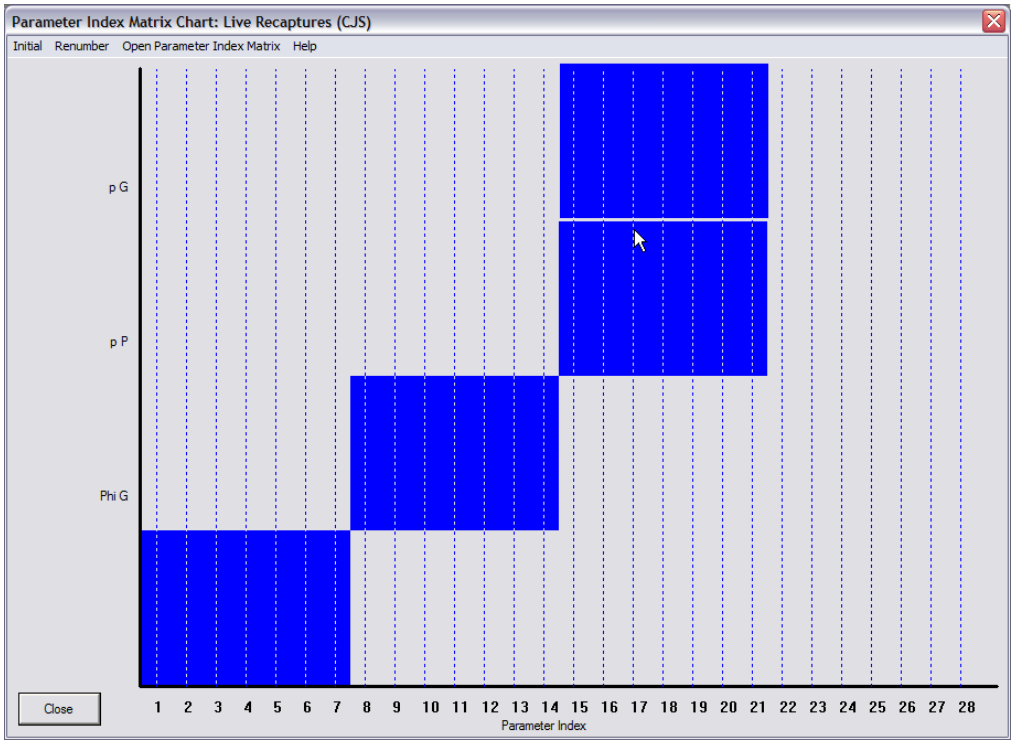
survival		*recapture*	
poor	good	poor	good
$1 \rightarrow 7$	$8 \rightarrow 14$	$15 \rightarrow 21$	$15 \rightarrow 21$

The recapture PIMs would now look like:

15	16	17	18	19	20	21
	16	17	18	19	20	21
		17	18	19	20	21
			18	19	20	21
				19	20	21
		recapture			20	21
		'poor'				21

15	16	17	18	19	20	21
	16	17	18	19	20	21
		17	18	19	20	21
			18	19	20	21
				19	20	21
		recapture			20	21
		'good'				21

While we could do this 'manually', by modifying the indexing for each individual PIM, **MARK** lets you accomplish this in a faster, more elegant way – by modifying the PIM chart directly. How? By simply selecting (left-click with the mouse) the 'good' colony 'blue box' in the PIM chart, and while holding down the left mouse button, dragging it to the left, so that it lines up with the recapture 'blue box' for the 'poor' colony, then releasing the mouse button (shown below). Compare this PIM chart with the original one.



Next, look at the PIMs – you'll see that the recapture PIMs are now identical for both groups, indexed from 15 → 21, just as they should be. Now isn't that easy? We're now ready to run our general, starting model $\{\varphi_{c*t}\, p_t\}$. Go ahead and run it, calling it model '`Phi(c*t)p(t)`'. Add the results to the browser. The model deviance is 107.563, with 20 estimated parameters (6 survival parameters for the 'poor' colony, 6 survival parameters for the 'good' colony, 6 recapture probabilities (the same for both colonies), and 2 β-terms (one for each colony). Make sure you understand the parameter counting!

Now, we simply run the other models in the candidate model set: $\{\varphi_{c*t}\, p_.\}$, $\{\varphi_t\, p_t\}$, $\{\varphi_c\, p_t\}$, $\{\varphi_t\, p_.\}$, and $\{\varphi_c\, p_.\}$. To reinforce your understanding of manipulating the PIM chart, and to demonstrate at least one other nifty trick with the PIM chart, we'll start with model $\{\varphi_t\, p_t\}$. For this model, the PIM structure would be:

For survival

1	2	3	4	5	6	7
	2	3	4	5	6	7
		3	4	5	6	7
			4	5	6	7
				5	6	7
	survival				6	7
	'*poor*'					7

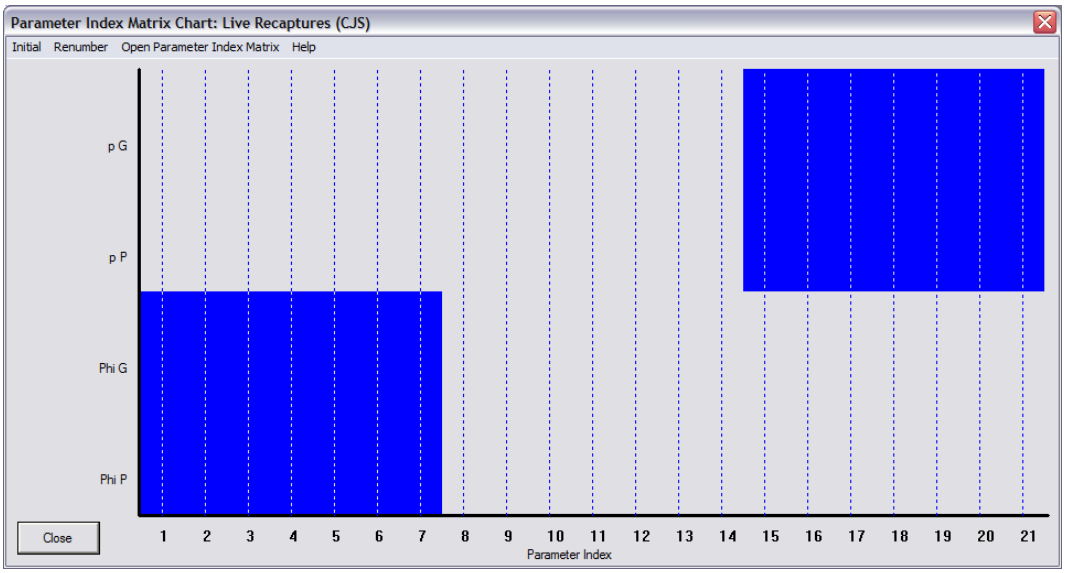
1	2	3	4	5	6	7
	2	3	4	5	6	7
		3	4	5	6	7
			4	5	6	7
				5	6	7
	survival				6	7
	'*good*'					7

and for recapture

8	9	10	11	12	13	14
	9	10	11	12	13	14
		10	11	12	13	14
			11	12	13	14
				12	13	14
		recapture			13	14
		'poor'				14

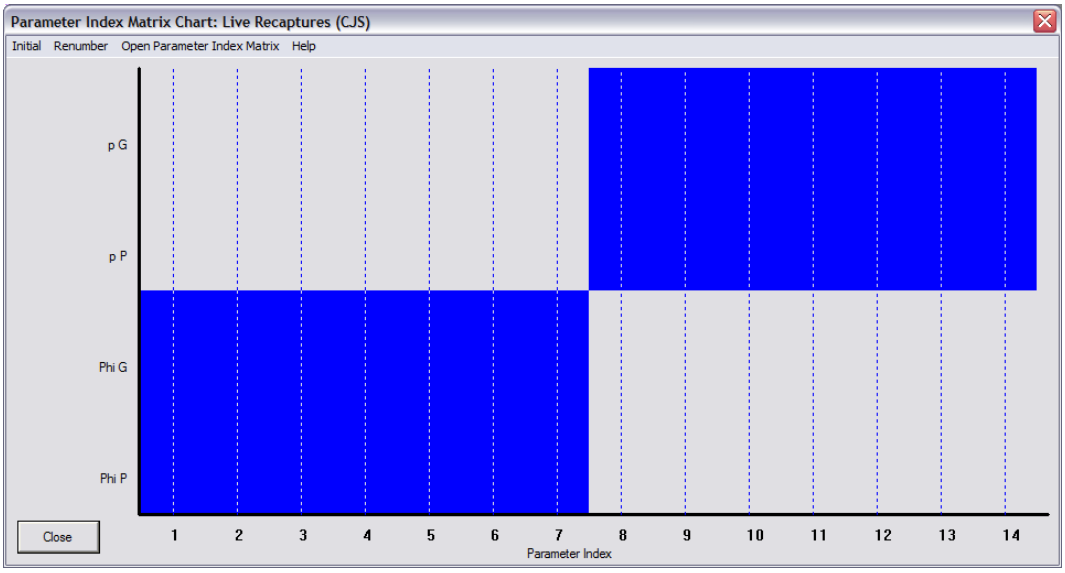
8	9	10	11	12	13	14
	9	10	11	12	13	14
		10	11	12	13	14
			11	12	13	14
				12	13	14
		recapture			13	14
		'good'				14

Now, if you understood our first attempt with manipulating the PIM chart, you might guess (correctly) that what you need to do is 'make the blue boxes for the survival parameters line up'. However, if you look at the chart, you see you could do this by grabbing the 'blue box' for survival for the poor colony and dragging it to the right (under the box for the good colony), or, in reverse, grabbing the box for the good colony, and dragging it to the left. We'll use the latter approach, because we want to point out another feature of the PIM chart that is worth noting. Here is the PIM chart:



Notice that now there is a gap between the 'stacked blue boxes' for the survival and recapture parameters – survival is indexed from 1 → 7, while recapture is indexed from 15 → 21. The index values for 8 → 13 don't correspond to any parameter. We want to eliminate the gap (i.e., remove the meaningless index values). You could do this manually, simply by dragging both recapture blue boxes to the left. Or, you could do this by right-clicking anywhere in the PIM chart. You'll be presented with a menu, which has '**Renumber with overlap**' as one of the options. '**Renumber with overlap**' means (basically), renumber to eliminate any gaps, but allow for the blue boxes for some parameters to overlap each other'. If you select the '**renumber with overlap**' option, the PIM chart will change to look like the chart shown at the top of the next page.

Pretty slick, eh? This corresponds to model $\{\varphi_t\, p_t\}$. Confirm this for yourself by checking the 4 individual PIMs (in fact, this is always a good idea, until you're 100% comfortable with **MARK**). Once you're sure you have the right model, go ahead and run it – call it '`phi(t)p(t)`', and add the results to the browser. This model has a much smaller AIC value than our starting model, although it has a larger deviance (which alone suggests that the time-dependent model does not fit the data as well as



the more general model which included colony effects). We'll defer discussion/interpretation of these 'model fitting' considerations to the next section.

There are still several other models in our candidate model set. Go ahead and run them – here are the results for all of the models in the candidate model set:

Results Browser: Live Recaptures (CJS)

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{phi(c)p(t)}	369.8080	0.0000	0.85650	1.0000	9	111.6644
{phi(c)p(.)}	373.5263	3.7183	0.13345	0.1558	3	128.1990
{phi(t)p(.)}	379.0123	9.2043	0.00859	0.0100	8	123.0601
{phi(t)p(t)}	382.8807	13.0727	0.00124	0.0014	13	115.7384
{phi(c*t)p(.)}	386.4962	16.6882	0.00020	0.0002	15	114.7094
{phi(c*t)p(t)}	391.4103	21.6023	0.00002	0.0000	20	107.5633

If your values for deviance and so on match those shown here, then that's a good clue that you've managed to build the models successfully (we'll be explaining what the various columns in the results browser are shortly).

begin sidebar

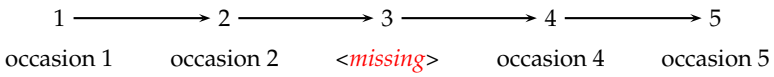
uneven time-intervals + missing sampling occasions

In the preceding, we have implicitly assumed that the interval between sampling occasions is identical throughout the course of the study (e.g., sampling every 12 months, or every month, or every week). But, in practice, it is not uncommon for the time interval between occasions to vary – either by design, or because of 'logistical constraints'. In the extreme, you might even miss a sampling occasion altogether. This has clear implications for how you analyze your data.

For example, suppose you sample a population each October, and again each May (i.e., two samples within a year, with different time intervals between samples; October → May, an interval of 7 months,

and May → October, an interval of 5 months). Suppose the *true* monthly survival rate is constant over all months, and is equal to 0.9. As such, the expected survival over the interval from October → May would be $0.9^7 = 0.4783$, while the expected survival rate over the interval from May → October would be $0.9^5 = 0.5905$. If you fit a model without accounting for these differences in time intervals, it is clear that there would 'appear' to be differences in survival between successive samples, when in fact the monthly survival does not change over time.

Alternatively, what if you're missing a sampling occasion altogether? For example, suppose you have a 5 occasion study, but for some reason, were unable to sample on the third occasion:



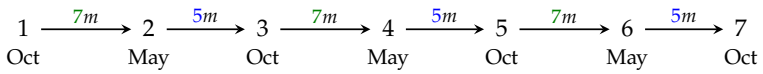
This situation has two implications. First, the encounter probability on the third occasion is logically 0. Second, in the absence of encounter data from the third occasion, the transition estimate for an individual alive and in the sample at occasion 2 would reflect the interval from occasion 2 → 4, where 4 is the next sampling occasion in the data, and not 2 → 3.

So, in effect, missing a sampling occasion altogether is equivalent to an unequal interval, at least with respect to estimating interval transition parameters, like survival.

However, it is quite different in terms of modeling the encounter probabilities, which represent (conceptually) ~instantaneous events at the end of the interval. For simple unequal intervals, where all occasions are sampled, there is an encounter parameter estimated for each encounter (sampling) occasion. For missing sampling occasions, you need to account for both the unequal interval that is generated as an artifact of the missing occasion, and the fact that the encounter probability is logically 0 for the missing occasion.

a. uneven time-intervals

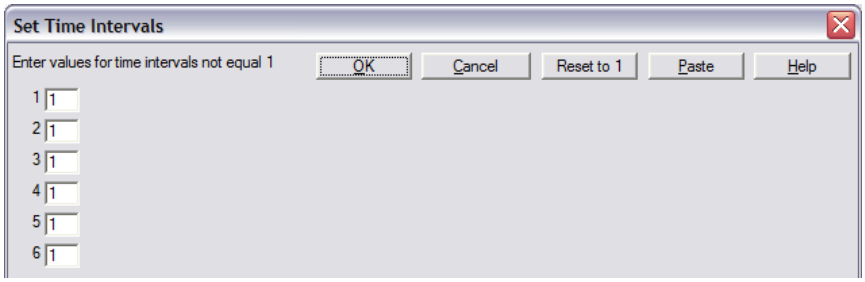
Here, we consider the situation where there are no missing sampling occasions, but where the interval between occasions varies. Imagine the following situation, where you sample a population each October, and again each May (i.e., two samples within a year, with different time intervals between samples; October→ May (7 months), and May→ October (5 months), for 7 occasions (assume the first sampling occasion is in October). Thus, the sampling intervals over the course of the study are



Suppose the 'monthly' survival probability is 0.95 (this was the value used to simulate the data – the recapture probability in the simulation was held constant at $p = 0.80$ at each occasion). Thus, the expected 'seasonal' survival probability for the May→ October season is $0.95^5 = 0.7738$, and $0.95^7 = 0.6983$ for the October → May season; in other words, the same *monthly* survival between seasons, but different expected *seasonal* survival probabilities. But, more importantly, since the monthly survival probability is the same, then if the seasons were the same length (say, both 6 months long),

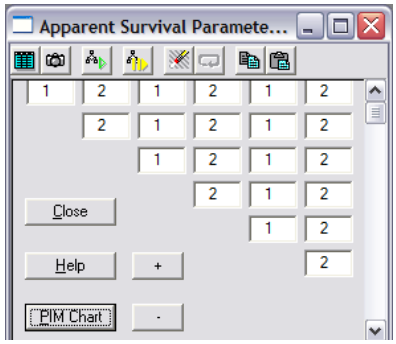
then we would expect that seasonal survival for both seasons would be the same, and that the best, most parsimonious model would likely be one that constrained survival to be the same between seasons.

What happens if you fit a model to these data where survival is constrained to be the same between seasons, without correctly specifying the different time intervals between sampling occasions? Start **MARK**, and read in the file **`variable_interval.inp`**. The simulated data represent a live mark-encounter study, which is the default data type in **MARK**. We specify 7 sampling occasions. If you click the button to the right of where you specify the number of encounter occasions, you'll see that **MARK** defaults to a common, constant interval of '1' time unit between each successive sampling occasion:



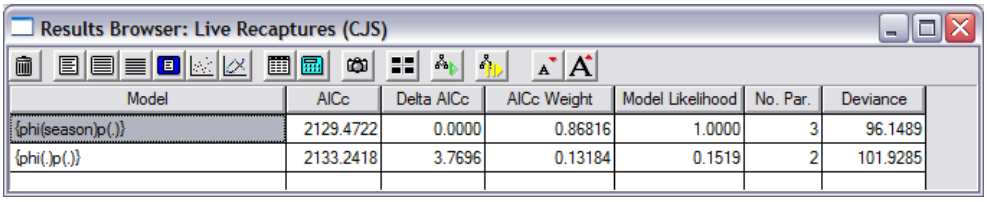
We know that for this example, these default intervals are incorrect, but to demonstrate what happens if you don't correctly specify the time interval we'll accept the default interval values of '1'. We'll fit 2 models to these data: model $\{\varphi_{.}\, p_{.}\}$, and model $\{\varphi_{(season)}\, p_{.}\}$, where the second model assumes there is a different survival probability between seasons (but that within season, the estimated survival probability is constant among years). How do we build model $\{\varphi_{(season)}\, p_{.}\}$?

Fairly simply – we can do this by using a common index value for each season in the survival PIM:



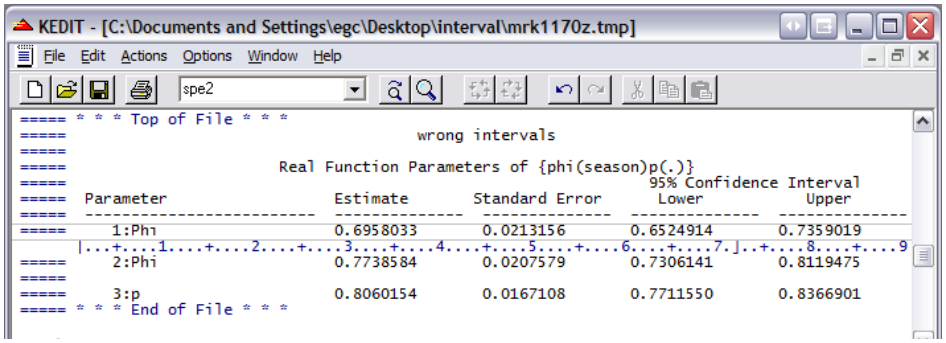
Here, the '1' index values correspond to the October → May season (recall that in this example, the first sampling occasion is assumed to be in October), and the '2' index values correspond to the May→ October season.

We see clearly (below) that model $\{\varphi_{(season)}\, p_{.}\}$ is not equivalent to model $\{\varphi_{.}\, p_{.}\}$:



Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{phi(season)p(.)}	2129.4722	0.0000	0.86816	1.0000	3	96.1489
{phi(.)p(.)}	2133.2418	3.7696	0.13184	0.1519	2	101.9285

We see from the parameter estimates for model $\{\varphi_{(season)}\, p_{.}\}$ (shown below) that the values for each season are very close to what we expected: 0.6958 is very close to $0.95^7 = 0.6983$, and 0.7739 is also very close to $0.95^5 = 0.7738$.



OK, all is well, right? Well, not quite. Suppose you wanted to test the hypothesis that *monthly* survival is the same between seasons. How would you do this? Well, you could derive an estimate of monthly survival from each of these seasonal estimates by taking the appropriate root of the estimated value. For example, for October → May, which is a 7 month interval, the estimated monthly survival probability is $\sqrt[7]{0.6958} = 0.9495$, and for May → October, which is a 5 month interval, the estimated survival probability is $\sqrt[5]{0.7739} = 0.9500$. While it is clear that both of these estimates are virtually identical in this instance, in practice you would need to derive SE's for these values, and use a formal statistical test to compare them (deriving the SE's for the nth roots – or any other function – of various parameter estimates involves use of the *Delta method* – see Appendix B).

How can we avoid these 'hand calculations'? Can we get **MARK** to give us the monthly estimates directly? In fact we can, by correctly specifying the time intervals. Obviously we do so by entering the appropriate intervals once we've specified the appropriate number of sampling occasions. The key, however, is in deciding what is the *appropriate* interval. Suppose we're really interested in the monthly survival value, and whether or not these values differ between seasons. How can we test this hypothesis in **MARK**, if the number of months in the two seasons differs?

In fact, it is quite straightforward*, but first you need to know something about how **MARK** handles time intervals. Consider the following example – suppose that 3 consecutive years of live trapping are conducted (with the first year capturing only new or unmarked animals), then a year is missed, then 3 more consecutive years are conducted. Then, the time intervals for these 5 encounter occasions would be specified as

```
1  1  2  1  1
```

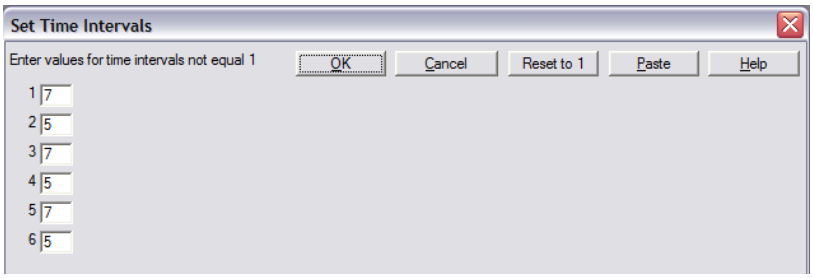
where the '2' indicates that the length of the time interval separating these 2 capture occasions is 2 years instead of 1 year like the other 4 intervals. The purpose of specifying the time intervals is to make the survival probabilities for each of the intervals comparable. Thus, the survival probability for all 5 of the above intervals will be an annual or 1 year probability, so that all can be constrained to be the same, even though the length of the time intervals to which they apply are not the same. The time interval is used as an *exponent* of the estimated survival probability to correct for the length of the time interval.

To explain in more detail, unequal time intervals between encounter occasions are handled by taking the length of the time interval as the exponent of the survival estimate for the interval, i.e., $S_i^{L_i}$. For the typical case of equal time intervals, all unity (1), this function has no effect (since raising anything to the power of 1 has no effect). However, suppose the second time interval is 2 increments in

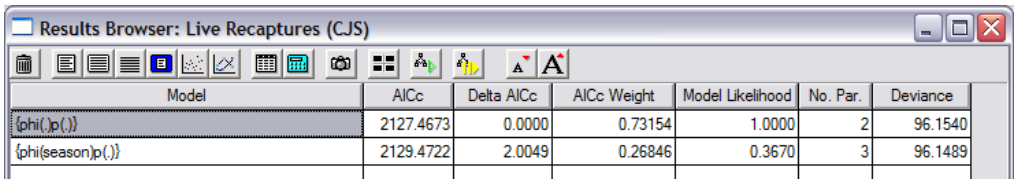
* At least, it is straightforward for simple live encounter models. Handling uneven intervals gets more complicated when we consider models where individuals 'move' among discrete states – these models are covered in later chapters.

length, with the rest 1 increment. This function has the desired consequences: the survival estimates for each interval are comparable, but the increased span of time for the second interval is accommodated. Thus, models where the same survival probability applies to multiple intervals can be evaluated, even though survival intervals are of different length. Moreover, you can use the exponent to derive estimates of survival for whatever interval you deem appropriate.

OK, back to our example – we're interested in monthly survival probabilities. To derive monthly survival probabilities, all you need to do is re-do the analysis, and enter the appropriate number of months:



Go ahead and re-run the analysis using the same two models. This time, however, we see that model $\{\varphi_{\cdot}\ p_{\cdot}\}$ is much better supported by the data than a model allowing for season differences:



Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{phi(.)p(.)}	2127.4673	0.0000	0.73154	1.0000	2	96.1540
{phi(season)p(.)}	2129.4722	2.0049	0.26846	0.3670	3	96.1489

Moreover, the estimated survival probability from this model (0.9497) is very close to the estimated true monthly survival probability used to simulate the data.

As a final test, suppose that instead of monthly estimates, you were interested in estimates calculated over 6 month intervals. Again, you could derive 6-month (i.e., half-year) survival estimates (and corresponding standard errors) by hand, but can you use **MARK** to do this for you directly? Yes. All you need to do is re-scale both seasonal intervals in terms of the desired season length. How? Simply by using the fact that a 7 month interval is in fact $(7/6) = 1.1\dot{6}$ times as long as a 6 month interval, and that a 5 month interval is $(5/6) = 0.8\dot{3}$ times as long as a 6 month interval. So, all you need to do is enter these re-scaled intervals into **MARK**. Note however that the interval input window in **MARK** does not expand 'visibly' to handle non-integer intervals (or even integer intervals that are very large). This is not a problem, however. Simply go ahead and enter the values (we'll use 1.167 and 0.833, respectively, so: `1.167 0.833 1.167 0.833 1.167 0.833`).

Since the true monthly survival probability is 0.95, then if the season lengths were actually the same (6 months), then we would expect the estimated seasonal survival probabilities for both re-scaled seasons to be the same, $(0.95)^6 = 0.7351$, and that model $\{\varphi_{\cdot}\ p_{\cdot}\}$ should be much better supported than competing model $\{\varphi_{(season)}\ p_{\cdot}\}$. In fact this is exactly what we see – the estimated 6-month survival probability from model $\{\varphi_{\cdot}\ p_{\cdot}\}$ is 0.7338, which is very close to the expected value of 0.7351.

b. missing sampling occasions

Now we consider the case where a sampling occasion is missed completely. As noted above, missing a sampling occasion altogether is equivalent to an unequal interval, at least with respect to estimating interval transition parameters, like survival. However, it is quite different in terms of modeling the encounter probabilities. For simple unequal intervals, where all occasions are sampled (at unequal intervals), there is a parameter estimated for each encounter occasion. In the case of 'missed sampling

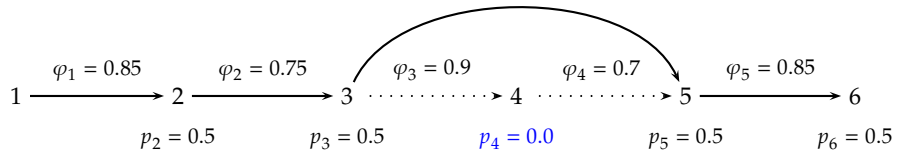
occasions', you need to account for both the unequal interval that is generated as an artifact of the missing occasion, and the fact that the encounter probability is logically 0 for the missing occasion.

There are several different approaches to handling a missing sampling occasion – the 'right' approach is the one that you find most convenient, and which best suits your purposes:

1. in the input file you can simply not include an encounter column in the encounter history for the occasion(s) where sampling did not occur. In this case, you then need to either (i) explicitly adjust the sampling interval to account for the missing occasion, or (ii) use a log link function, which can accommodate 'product estimates' over multiple (combined) intervals,
2. you can include a column of `0`'s in the encounter histories for the missing occasion. In other words, the encounter would be code '`0`' for every individual for that occasion. In this case, you need to explicitly set $p = 0$ for the missing occasion.
3. you can use a 'dot' (i.e., '`.`') instead of `0` in the encounter history (see Chapter 2). **MARK** recognizes the 'dot' as a 'missing occasion', and automatically fixes $p = 0$ for that missing occasion, without you having to do so explicitly.

To demonstrate these different approaches, we'll use a simulated data live encounter set – 6 occasions, where we assume that there was no sampling on occasion 4. We assumed that 500 new individuals were marked and released on each occasion, except for the missing occasion 4.* The true values for φ and p used to generate the encounter data are: $\varphi_1 = 0.85$, $\varphi_2 = 0.75$, $\varphi_3 = 0.9$, $\varphi_4 = 0.7$, $\varphi_5 = 0.85$, and p_2, p_3, p_5 and p_6 all equal 0.5, while $p_4 = 0$ (occasion 4 being the missing sampling occasion).

The basic structure and true parameter values are shown in the following diagram:



What is clear from this diagram is that in the absence of encounter data from occasion 4, then we are left with estimating a transition probability φ from occasion $3 \rightarrow 5$, which clearly would be a function of the product of $(\varphi_3 \times \varphi_4)$. **MARK** will then report estimates for φ_3 and φ_4 that are functions of this product (say, the square-root).

From the simulated data, we constructed 3 different `.INP` files: (i) **`missing_occasion_interval.inp`** (where the missing occasion 4 does not show up in the encounter history, and we explicitly set the interval for the transition from occasion $3 \rightarrow 5$ to '2'), (ii) **`missing_occasion_constrain.inp`** (where the missing occasion 4 is a column of `0`'s in the encounter history, and where we will need to *explicitly* fix $p_4 = 0$ in the estimation), and (iii) **`missing_occasion_dot.inp`** (where the missing occasion 4 is entered as a 'dot', which will cause **MARK** to *implicitly* assume $p_4 = 0$).

We'll start by considering the case where the missing occasion is not entered in the encounter history. Here, we have a couple of approaches we might try. We'll first consider an approach where we explicitly modify (specify) the default interval between sampling occasions. Start **MARK**, and specify a live encounter (CJS) data type. Select the **`missing_occasion_interval.inp`** input file, and specify 5 occasions (remember, in this `.INP` file, we do not have a column for the missing encounter occasion 4). Next, manually change the default interval between sampling occasions to '`1 1 2 1`'.

We'll fit 2 models to these data: $\{\varphi_t\ p_t\}$, and $\{\varphi_t\ p_.\}$. We see from the results browser (shown near the top of the next page), that model $\{\varphi_t\ p_.\}$ gets the most support in the data ($w = 0.775$). This is perhaps not surprising, since this model matches the overall structure of the generating model (above). Given

* It is possible to conceive of sampling situations where new individuals are marked and released on a particular occasion, but no reencounters with previously marked individuals occur. Such a situation might arise if the investigator is actively avoiding recapture of previously marked individuals, in order to concentrate field effort on capture and marking of new individuals.

the magnitude of the model selection uncertainty, we might consider *model averaging* the estimates of φ and p. However, since we don't introduce the concept and mechanics of model averaging until later in this chapter, we'll focus instead on estimates from the top model, $\{\varphi_t\ p_.\}$.

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance	-2Log(L)
{phi(t)p(.)}	5684.9430	0.0000	0.77547	1.0000	5	14.5900	5674.9226
{phi(t)p(t)}	5687.4219	2.4789	0.22453	0.2895	7	13.0512	5673.3838

Here are the estimates of the real parameters from this model:

Real Function Parameters of {phi(t)p(.)}

Parameter	Estimate	Standard Error	95% Confidence Interval Lower	Upper
1:Phi	0.8793586	0.0305467	0.8056530	0.9276228
2:Phi	0.7600228	0.0272857	0.7025736	0.8093860
3:Phi	0.7846389	0.0150980	0.7535779	0.8127576
4:Phi	0.8224165	0.0356383	0.7416440	0.8819557
5:p	0.4980605	0.0156225	0.4674858	0.5286497

We see that the estimates for $\hat{\varphi}_1 = 0.879$, $\hat{\varphi}_2 = 0.760$, and $\hat{\varphi}_4 = 0.822$ (here, corresponding to the interval from occasion $5 \rightarrow 6$) are quite close to the true parameters values (0.85, 0.75, and 0.85, respectively). Recall that true $\varphi_3 = 0.9$ and $\varphi_4 = 0.7$, meaning, the product probability of surviving the interval from $3 \rightarrow 5$ is $(0.9 \times 0.7) = 0.63$. As such, the square root of this value is $\sqrt{0.63} = 0.794$. The estimate $\hat{\varphi}_3 = 0.785$ is quite close to this value.

Alternatively, we could try a different 'link function' – specifically, the log link*. The topic of 'link functions' as a fundamental part of fitting models to encounter data, is covered in much detail in Chapter 6. For the moment, we simply need to remember some basic 'middle school' algebra involving 'logs of products'. If $a = b^l$, then $log(a) = l\log(b)$. Let l be the length of the interval between 2 sampling occasions, and let φ_0 be the survival over the base interval. If $l = 1$ (which is the default interval), then for some interval i, $\varphi_i = \varphi_0^1$, and $\log(\varphi_i) = (1)\log(\varphi_0)$. In this case, $\varphi_i \equiv \varphi_0$. Now, with a missing sampling occasion, $l > 1$. For our present example, with a missing occasion between occasions 3 and 5, the interval is $l = 2$. So, $\varphi_{3\rightarrow5} = \varphi_0^2$, and thus $\log(\varphi_{3\rightarrow5}) = 2\log(\varphi_0)$. So, if we could estimate $\hat{\varphi}_{3\rightarrow5}$ on the log scale (say, using the 'log link' in **MARK**), then our estimate for $\log(\hat{\varphi}_0)$ is simply $\hat{\varphi}_{3\rightarrow5}/2$. We would then back-transform to get our estimate of φ_0 on the real probability scale: $\hat{\varphi}_0 = \exp(\log(\hat{\varphi}_0))$.

All we need do in **MARK** is to (i) specify the 'log link' when we run the model (as opposed to the more common sin or logit links), and (ii) enter the length of the interval as a 'covariate' in the 'design matrix'. Step (i) is easy – you simply click the 'log link' radio button. Step (ii) is easy if you know how to build linear models and design matrices in **MARK** (which is the subject of Chapter 6). At this point, we'll simply show you (below) the design matrix for model $\{\varphi_t\ p_.\}$:

B1 interval 1	B2 interval 2	B3 interval 3-5	B4 interval 5-6	Parm	B5 p
1	0	0	0	1:Phi	0
0	1	0	0	2:Phi	0
0	0	2	0	3:Phi	0
0	0	0	1	4:Phi	0
0	0	0	0	5:p	1
0	0	0	0	6:p	1
0	0	0	0	7:p	1
0	0	0	0	8:p	1

* In fact, this was the approach that was used in later versions of program **SURGE** – Cooch *et al*. (1997).

Along the diagonal in columns 1 → 4 of the design matrix are the 'covariate' values for interval length between sampling occasions (i.e., the parameter l in the bits of algebra discussed above). For most of the intervals, $l = 1$, but between occasions 3 and 5, (with missing occasion 4), we enter $l = 2$ into the matrix.

After fitting this design matrix to the data, we get the following parameter estimates:

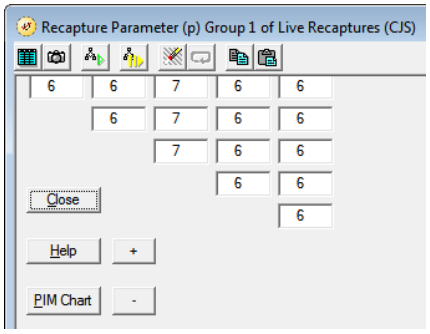
Parameter	Estimate	Standard Error	95% Confidence Interval Lower	Upper
1:Phi	0.8793587	0.0305468	0.8056530	0.9276229
2:Phi	0.7600228	0.0272857	0.7025736	0.8093860
3:Phi	0.6156583	0.0236930	0.5682969	0.6609239
4:Phi	0.8224165	0.0356383	0.7416439	0.8819557

We see that the estimates for φ_1, φ_2, and φ_5 are identical to those reported earlier when we explicitly changed the interval from the default '1 1 1 1' to '1 1 2 1'.

What about $\hat{\varphi}_{3\rightarrow5} = 0.616$ – where does this value come from? Recall that true $\varphi_3 = 0.9$ and $\varphi_4 = 0.7$, meaning, the true product probability of surviving the interval from 3 → 5 is $(0.9 \times 0.7) = 0.63$. When we explicitly changed the interval between sampling occasions to '1 1 2 1', and used the default logit or sin links, **MARK** returned $\hat{\varphi}_{3\rightarrow5} = 0.785$, which is quite close to $\sqrt{0.63} = 0.794$. So, our estimate of $\hat{\varphi}_4 = \hat{\varphi}_5 = 0.785$ (i.e., **MARK** reports the estimate as the square-root of the product of $\varphi_3\varphi_4$). Using the log link, however, the value reported is, in fact, the estimate of this product – the estimated value 0.616 is in fact the $(0.785)^2$. So, explicitly setting the interval – the estimate is the lth root of the product over the interval, while using the log link generates an estimate of that product.

Deciding which of these two approaches is 'better' or 'easier' – neither of which involve reformatting the encounter history data in any way – is ultimately a matter of personal preference. [It should be noted – and as discussed in Chapter 6 – that using the approach based on the log link does not change the overall fit of the model to the data, relative to the approach where we used, say, the sin or logit link but made explicit changes to the interval length at the outset.]

Next, we consider the situation where the missing occasion is 'coded' in the encounter history as a column of 0's. These encounter history data are found in **missing_occasion_constrain.inp**. Here, we must remember to explicitly fix $p_4 = 0$ in the numerical estimation. Start **MARK**, select the .INP file, and set the number of occasions to 6 (not 5 – remember, we have an encounter column for missing occasion 4, which was not the case in the previous analysis). Leave the intervals between sampling occasions at the default value of '1'. Again, we'll fit 2 models to these data: $\{\varphi_t\, p_t\}$, and $\{\varphi_t\, p_\cdot\}$. Note that for the second model, $\{\varphi_t\, p_\cdot\}$, we have to modify the PIM for the encounter probability p to allow us to fix one of the encounter parameter $p_4 = 0$. One way of setting up the PIM is as follows:



Here, parameter 7 corresponds to p_4 – we will fix parameter 7 to 0 during the numerical estimation. [Whereas for model $\{\varphi_t\, p_t\}$, we'll set parameter 8, which corresponds to p_4, to 0.]

Here are the results from fitting these two models to the data:

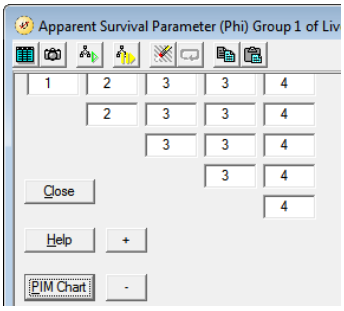
Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance	-2Log(L)
{phi(t)p(.) -- constraint on p4}	5684.9430	0.0000	0.90421	1.0000	5	14.5900	5674.9226
{phi(t)p(t) - constraint on p4}	5689.4328	4.4898	0.09579	0.1059	8	13.0512	5673.3838

We see that here, model $\{\varphi_t\, p_\cdot\}$ receives virtually all of the support in the data. Here are the parameter estimates from this model:

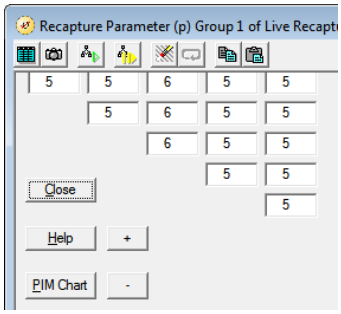
Parameter	Estimate	Standard Error	95% Confidence Interval Lower	Upper	
1:Phi	0.8793588	0.0305468	0.8056530	0.9276229	
2:Phi	0.7600227	0.0272857	0.7025735	0.8093859	
3:Phi	0.7804012	28.241000	0.1897496E-139	1.0000000	
4:Phi	0.7888997	28.548554	0.4496466E-145	1.0000000	
5:Phi	0.8224165	0.0356382	0.7416440	0.8819557	
6:p	0.4980605	0.0156225	0.4674858	0.5286497	
7:p	0.0000000	0.0000000	0.0000000	0.0000000	Fixed

Estimates for $\hat{\varphi}_1, \hat{\varphi}_2, \hat{\varphi}_5$ and p (parameter 6) are virtually identical to those from the previous analysis, and are close to the true parameter values used to simulate the data. Estimates for $\hat{\varphi}_3$ and $\hat{\varphi}_4$ are different from each other after the second decimal place, but their product is virtually identical to the square of the estimate from the preceding analysis: $(0.7804 \times 0.7889 = 0.6157 \approx (0.7846)^2 = 0.6156)$.

However, the SE reported for both parameters are clearly meaningless. Why? They're meaningless because for this example, we haven't imposed any sort of constraint on the survival estimates for the 'two pieces' of the interval from occasion $3 \rightarrow 5$ (i.e., φ_3, φ_4). As such, **MARK** has no way of estimating the covariance between the two reported values, which are 'evaluated' subject only to the constraint that their product is ≈ 0.6157). If, however, we modified the survival PIM, setting the estimates for the intervals from occasion $3 \rightarrow 4 \rightarrow 5$ equal to each other (for example, as shown below):



and re-index the PIM for the encounter probability



then, after fixing parameter 6 to 0, the estimates (shown below) for $\hat{\varphi}_3 = 0.7846$ and $\widehat{\mathrm{SE}} = 0.0151$ are identical to those reported in the first approach (above) where we dropped the 'missing occasion column' from the encounter history, and manually set the interval for that occasion to '2'.

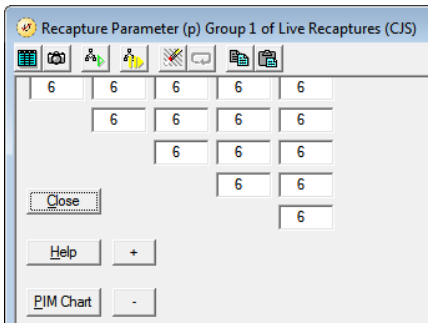
missing occasion -- fixing p

Real Function Parameters of {phi(t)p(.) -- constraint on p4 - fixed phi}

Parameter	Estimate	Standard Error	95% Confidence Interval Lower	Upper
1:Phi	0.8793605	0.0305469	0.8056539	0.9276247
2:Phi	0.7600234	0.0272858	0.7025740	0.8093867
3:Phi	0.7846388	0.0150980	0.7535777	0.8127575
4:Phi	0.8224160	0.0356381	0.7416439	0.8819549
5:p	0.4980598	0.0156225	0.4674851	0.5286490
6:p	0.2061796E-008	0.1584641E-005	-0.3103835E-005	0.3107958E-005

Finally, we consider this same analysis, but instead of a column of `0`'s, the missing occasion is entered into the encounter history as a 'dot'. These data are contained in **`missing_occasion_dot.inp`**. The potential advantage of using the 'dot' approach in coding missing sampling occasions is that it eliminates the need to explicitly set the encounter probability to 0 for the missing occasion. Again, start **MARK**, select the `.INP` file, and set the number of occasions to 6. Leave the intervals between sampling occasions at the default value of '1'. Again, we'll fit the same 2 models to these data: $\{\varphi_t\, p_t\}$, and $\{\varphi_t\, p_\cdot\}$.

However, unlike the previous analysis, where for model $\{\varphi_t\, p_\cdot\}$ we needed to modify the PIM for the encounter probability p to 'allow **MARK** to fix this missing occasion to 0', that is not the case here – **MARK** handles everything for us. We simply use a 'dot' (constant) PIM for p:



Here are the results of fitting the two models to the data:

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance	-2Log(L)
{phi(t)p(.)}	5684.9430	0.0000	0.90421	1.0000	5	14.5900	5674.9226
{phi(t)p(t)}	5689.4328	4.4898	0.09579	0.1059	8	13.0512	5673.3838

As with the earlier examples, we see that model $\{\varphi_t\, p_\cdot\}$ has most of the support in the data, so we'll look at estimates from this model only:

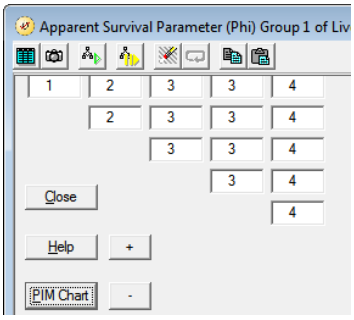
example of dot notation -- missing occasion

Real Function Parameters of {phi(t)p(.)}

Parameter	Estimate	Standard Error	95% Confidence Interval Lower	Upper
1:Phi	0.8793587	0.0305468	0.8056529	0.9276229
2:Phi	0.7600227	0.0272857	0.7025735	0.8093859
3:Phi	0.7845978	0.0000000	0.7845978	0.7845978
4:Phi	0.7846801	0.0000000	0.7846801	0.7846801
5:Phi	0.8224165	0.0356382	0.7416440	0.8819557
6:p	0.4980605	0.0156225	0.4674858	0.5286497

Again, estimates for $\hat{\varphi}_1$, $\hat{\varphi}_2$, $\hat{\varphi}_5$ and p (parameter 6) are nearly identical to those from the previous analysis, and are quite close to the true parameter values used to simulate the data. Estimates for $\hat{\varphi}_3$ and $\hat{\varphi}_4$ are again slightly different from each other. However, their product is identical to the products of the estimates from the preceding analysis: $(0.8084 \times 0.7616 = 0.6157 = 0.7804 \times 0.7889 = 0.6157)$. The SE reported for both parameters are clearly meaningless (for the same reason discussed above).

If, however, we modified the survival PIM as we did earlier, setting the estimates for the intervals from occasion $3 \rightarrow 4 \rightarrow 5$ equal to each other:



we end up with estimates which are identical to those we saw above:

example of dot notation -- missing occasion

Real Function Parameters of {phi(t)p(.)} - tweaked PIM}

Parameter	Estimate	Standard Error	95% Confidence Interval Lower	Upper
1:Phi	0.8793587	0.0305467	0.8056530	0.9276228
2:Phi	0.7600228	0.0272857	0.7025736	0.8093860
3:Phi	0.7846389	0.0150980	0.7535779	0.8127576
4:Phi	0.8224165	0.0356383	0.7416439	0.8819556
5:p	0.4980605	0.0156225	0.4674858	0.5286497

end sidebar

OK – so we've considered some of the basics of building some models in **MARK**. But, what model, or models, should we make inference from? How do we establish whether or not some factor 'significantly' influences survival, or some other parameter of interest? What parameter estimates are most appropriate to report? Of course, these are in fact *the* critical questions motivating the exercise of fitting models to data in the first place. We begin addressing them in the next section.

4.3. Model selection – the basics

In simplest terms, we might express our objective as trying to determine the best model from the set of approximating models we've fit to the data. How would we identify such a 'best model'? An intuitive answer would be to select the model that 'fits the data the best' (based on some statistical criterion - say smallest RSS, or equivalent).

However, there is a problem with this approach – the more parameters you put into the model, the better the fit (analogous to ever-increasing R^2 in a multiple regression as you add more and more terms to the model). As such, if you use a simple measure of 'fit' as the criterion for selecting a 'best' model, you'll invariably pick the one with the most parameters.*

* In the extreme, if you have 1 parameter for each data point, $R^2 \rightarrow 1.0$.

So, for our analysis of the European dipper data we would select model $\{\varphi_t\ p_t\}$ as our 'best' model, simply because it has the lowest deviance (36.4013), which of course it must since it has more parameters (11) than the other 3 models in the model set:

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{phi(.)p(.)}	322.5527	0.0000	0.96003	1.0000	2	41.8147
{phi(t)p(.)}	330.0567	7.5040	0.02253	0.0235	7	38.8147
{phi(.)p(t)}	330.6794	8.1267	0.01650	0.0172	7	39.4374
{phi(t)p(t)}	336.4343	13.8816	0.00093	0.0010	11	36.4013

Great, right? Don't we want to maximize the fit of our models to the data? Well – it's not quite that simple. While adding more parameters increases the *fit* of the model to the data, you pay a price in so doing – that price is parameter *uncertainty* (or variance).

Consider Fig. (4.1), shown below. This figure represents the trade-off between squared bias and variance versus the number of estimable parameters in the model. With an increasing number of parameters, the squared bias of the estimates of the individual parameters goes down.* In other words, the overall fit of the model to the data is better.

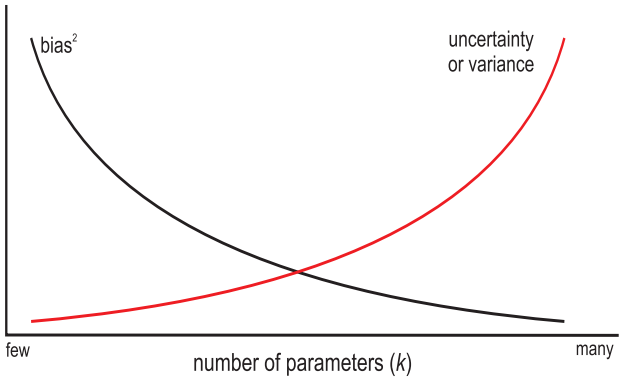


Figure 4.1: *Fundamental relationship between the number of parameters in a model (k), and the square of the bias (related to overall model fit to the data), and parameter uncertainty (precision of parameter estimates).*

But, this increase in fit comes at the cost of greater parameter uncertainty (i.e., larger and larger measures of parameter uncertainty – say, bigger and bigger SE for parameter estimates). In the extreme, if you have one parameter for each data point, the fit of the model to the data will be perfect (in other words, $R^2 = 1$). However, the SE for the estimates of each parameter will be on the interval $[-\infty, +\infty]$, which is clearly not particularly informative. How can we find a good, defensible compromise between the two? One approach (out of a growing number) is to make use of something called 'the AIC'.

* Formally, as a model becomes complex (more parameters), bias decreases, but variance of the estimates of parameter coefficients increases (as depicted in Fig. 4.1). Why?

Suppose that a response variable y can be modeled as $y = g(\mathbf{x}) + \epsilon$. The corresponding expected prediction error can be written as $E((y - \hat{g}(\mathbf{x}))^2)$. If $\hat{g}(\mathbf{x})$ is the prediction based on the within-sample data then

$$E\big((y - \hat{g}(\mathbf{x}))\big)^2 = \sigma^2 + (E(\hat{g}(\mathbf{x})) - g(\mathbf{x}))^2 + \text{var}(\hat{g}(\mathbf{x}))$$
$$= \text{irreducible error} + \text{bias}^2 + \text{variance}$$

The first term, *irreducible error*, represents the uncertainty associated with the true relationship that cannot be reduced by any model. In effect, the irreducible error is a constant in the expression. Thus, for a given *prediction error*, there is an explicit trade-off between minimizing the variance and minimizing the bias (i.e., if one goes down, the other goes up).

4.3.1. The AIC, in brief...

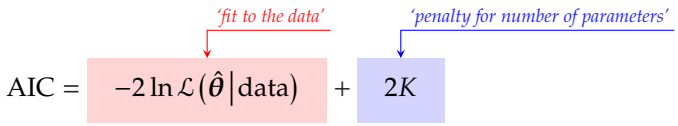
The AIC (which in fact is an acronym for 'another information criterion', but is almost universally referred to as 'Akaike's Information Criterion', after Hirotugu Akaike who first described it in 1973) comes to us from the world of information theory.

While the 'deep theory' underlying the AIC is somewhat dense (translation: not entirely trivial – see the relevant chapters in Burnham & Anderson (2002). For a somewhat more accessible introduction, see Hooten & Cooch 2019), in purely mechanical terms, the usual applications are straightforward. The AIC is calculated for a particular model as

$$\mathrm{AIC} = -2\ln\mathcal{L}\left(\hat{\boldsymbol{\theta}}\middle|\mathrm{data}\right) + 2K,$$

where $\mathcal{L}$ is the *model likelihood* ($\boldsymbol{\theta}$ represents the vector of the various parameter estimates given the data), and K is the number of parameters in the model.

The AIC can perhaps be best understood as a function of the 'fit' of the model to the data, and a 'penalty' term reflecting the number of parameters used to achieve that 'fit':



In general, the fit of the model to the data is 'represented' by the model likelihood (maximum likelihood estimation was introduced in Chapter 1). Thus, as the fit of the model goes up, the likelihood of the model (given the data) goes up (and thus $-2\ln(\mathcal{L})$ goes down). However, as indicated in the figure on the preceding page, the greater the number of parameters, the greater the parameter uncertainty or variance. Thus, as the fit of the model increases, the value of $-2\ln(\mathcal{L})$ goes down – and for a given number of parameters, the AIC declines. Conversely, for a given fit, if it is achieved with fewer parameters (lower K), then the calculated AIC is lower. The $2K$ term, then, is the *penalty* for the number of parameters. As K goes up, likelihood goes down, but this is balanced by the penalty of adding the term $2K$.*

So, one strictly utilitarian interpretation of the AIC is that the model with the lowest AIC is the 'best' model among those fit to the data because it is most parsimonious given the data – best fit with fewest parameters. However, more formally, and perhaps more importantly at least conceptually, the model with the lowest AIC within the candidate set of approximating models can be shown to be the model which is closest to 'full truth' – which is not known (and is not contained in the candidate model set).

Say, what? Start by imagining a model f which represents full truth. Such a model might exist in theory, but we will never be able to fully specify it. Consider an approximating model g. We use the term *approximating* for g since g (and in fact any model) is an approximation of truth. Our goal in model selection is (ultimately) to determine which of our models minimizes the difference (distance) between g and f. In the 1950's, Kullback and Leibler determined that if $I(f, g)$ represents the 'information' lost when model g is used to approximate full truth f, then $I(f, g)$, the distance between a model g and full truth f, is given as

$$I(f,g) = \int f(x)\ln\left(\frac{f(x)}{g(x\mid\boldsymbol{\theta})}\right)dx.$$

* The concept of penalizing complex models to account for optimism and improve predictive ability is much more general than how it is used in AIC – consult the literature on 'regularization' as a means of preventing overfitting (Hooten & Cooch 2019).

Here f and g are probability distributions. The verbal description of $I(f, g)$ is that it represents the distance from model g to model f. Alternatively, it is the information lost when using g to approximate f.* As above, $\boldsymbol{\theta}$ represents the vector of the various parameters used in the specification of g.

It might be helpful to consider the form of Kullback-Leibler (K-L) information for discrete probability models (since they are somewhat easier to grasp). Let the true state of the system be

$$f = \{p_1, p_2, \ldots, p_k\}.$$

Here there are k possible outcomes of the underlying random variable – the true probability of the ith outcome is given by p_i. Let the model *approximating* the state be

$$g = \{\gamma_1, \gamma_2, \ldots, \gamma_k\},$$

where γ_i represents the approximating probability distribution for the ith outcome. (Note that in the discrete case, $0 < p_i < 1, 0 < \gamma_i < 1$, and $\sum p_i = \sum \gamma_i = 1$).

The K-L information difference between models f and g is defined for discrete distributions to be

$$\begin{aligned} I(f, g) &= \sum_{i=1}^{k} p_i \ln\left(\frac{p_i}{\gamma_i}\right) \\ &= \sum_{i=1}^{k} p_i \ln p_i - \sum_{i=1}^{k} p_i \ln \gamma_i. \end{aligned}$$

(As an aside, you may recognize the first of the two terms in this difference as H, the Shannon-Weiner diversity index, another information-based measure.)

Back to the more general integral form, which for $I(f, g)$ can be written equivalently as a difference of integrals

$$\begin{aligned} I(f, g) &= \int f(x) \ln f(x) dx - \int f(x) \ln g(x \mid \boldsymbol{\theta}) dx \\ &= E_f[\ln f(x)] - E_f[\ln g(x \mid \boldsymbol{\theta})], \end{aligned}$$

where in the second line we make use of the fact that the form of each integral is that of an *expectation*.

$I(f, g)$ is known as the *Kullback-Leibler* (K-L) information, or distance. With a bit of thought, it is clear that a 'good' model is one where the distance between the model g and 'truth' f is as small as possible. In other words, a model which minimizes the K-L distance. But, if we don't (and can't ever) know truth, then how can we 'estimate' the K-L distance for a given model? In fact, we can't, but it turns out that doesn't matter. We can make use of *relative* K-L information instead.

What do we mean by 'relative' K-L information? Look at the RHS of the preceding equation:

$$I(f, g) = E_f[\ln f(x)] - E_f[\ln g(x \mid \boldsymbol{\theta})].$$

The first expectation $E_f[\ln(f(x))]$ is 'truth', which clearly must be a constant across models. Thus,

$$I(f, g) = \text{Constant} - E_f[\ln g(x \mid \boldsymbol{\theta})]$$

$$I(f, g) - \text{Constant} = -E_f[\ln g(x \mid \boldsymbol{\theta})].$$

* The negative of K-L information is Boltzmann's entropy, $H = -I(f, g)$, a fundamental concept in statistical thermodynamics.

The term $[I(f, g) - \text{Constant}]$ is called '*relative Kullback-Leibler information*' (or distance), and is the relative distance between truth (f) and the approximating model (g). Relative K-L information is measured on an interval scale, a scale without an absolute zero. (In fact, the absolute zero is truth and is no longer part of the expression.)

Why is the *relative* K-L information of interest? Suppose that in addition to our approximating model g we have a second approximating model h for the true state of nature f. The information lost in using model h to approximate f ('truth') is given by the following

$$\begin{aligned} I(f, h) &= \text{'approximating model'} - \text{'truth'} \\ &= \int f(x) \ln f(x) dx - \int f(x) \ln h(x \mid \boldsymbol{\theta}) dx \\ &= E_f[\ln f(x)] - E_f[\ln h(x \mid \boldsymbol{\theta})]. \end{aligned}$$

Observe that $E_f\left[\ln f(x)\right]$ is a common term in the expression for both model g and model h. Thus, if we want to compare model g to model h it makes sense to consider the difference $I(f, g) - I(f, h)$. If we do so then we find

$$\begin{aligned} I(f, g) - I(f, h) &= \left(E_f[\ln f(x)] - E_f[\ln g(x \mid \boldsymbol{\theta})]\right) - \left(E_f[\ln f(x)] - E_f[\ln h(x \mid \boldsymbol{\theta})]\right) \\ &= E_f[\ln h(x \mid \boldsymbol{\theta}] - E_f[\ln g(x \mid \boldsymbol{\theta})]. \end{aligned}$$

Note that the $E_f[\ln f(x)]$ terms for each approximating model have canceled out. Recall that $E_f[\ln f(x)]$ represents 'truth'! So, if our goal is to compare two models, 'truth cancels out' of the comparison. This is helpful since we cannot ever know what 'truth' is. Its absolute magnitude has no meaning. It is only useful for measuring how far apart two approximating models are *from each other*. This last expression represents the difference in *relative* Kullback-Leibler information between two models. So if our only goal is model *comparison* then our objective can be more limited. Rather than estimate K-L information we can estimate instead *relative* K-L information – the information lost when model g_i is used to approximate full reality (f) – in other words, the *distance* between model g_i and full reality (Fig. 4.2).

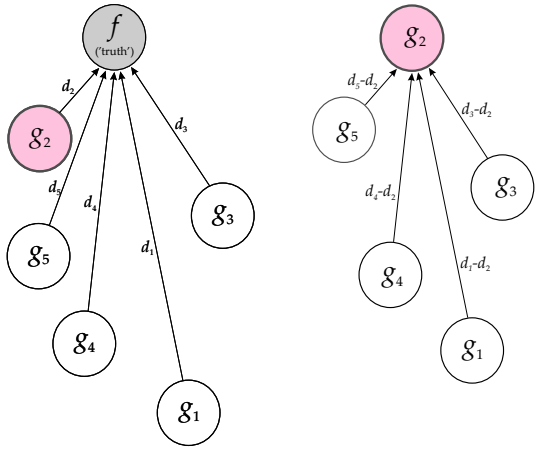


Figure 4.2: *Kullback-Leibler information is shown (at left) as the distances (d_i) between full reality (f) and the various models (g_i). The Δ values (right) provide the estimated distance of the various models to the best model (in this case, model g_2). These values are on the scale of information irrespective of the scale of measurement or type of data. From Burnham, Anderson & Huyvaert (2011).*

In either case, it seems compelling that one would want to select the model in the set of R models $(g_1, g_2, \ldots, g_R)$ that minimizes K-L information loss. That is, we want the model from within the model set that loses the least information about full reality, hence, the model that is closest to full reality in the current model set.

While all might seem well, recall that in our approximating model we typically won't know the exact value of $\boldsymbol{\theta}$. Instead we will have to use an estimate, $\hat{\boldsymbol{\theta}}$. To account for this additional uncertainty, Akaike suggested that what we should do is to calculate the *average* value of relative K-L information over all possible values of $\boldsymbol{\theta}$.

In terms of expectation we would call this quantity *expected* relative K-L information and write it as

$$E\left[E_f\left[\ln g(x \mid \boldsymbol{\theta})\right]\right].$$

Akaike showed that an asymptotically unbiased estimator of the relative expected K-L distance from 'truth' could be calculated as

$$\ln \mathcal{L}\left(\hat{\boldsymbol{\theta}} \middle| \text{data}\right) - K,$$

where $\mathcal{L}(\hat{\boldsymbol{\theta}} \mid \text{data})$ is the log likelihood function for an approximating model evaluated at the maximum likelihood estimate of the parameter set $\boldsymbol{\theta}$, and where K is the number of parameters estimated in maximizing the likelihood of the model.

Akaike then defined 'an information criterion' (AIC), by multiplying through by -2 (for 'historical reasons', it seems) to get the familiar

$$\text{AIC} = -2 \ln \mathcal{L}\left(\hat{\boldsymbol{\theta}} \middle| \text{data}\right) + 2K,$$

Thus, as suggested earlier, one should select the model that yields the smallest value of AIC among the models in the candidate model set, not simply because it provides some 'balance' between precision and fit, but because this model is estimated to be the 'closest' to the unknown reality that generated the sample data, from among the candidate approximating models being considered.

In other words, you should use the AIC to select the fitted approximating model that is estimated to be closest to the unknown truth (i.e., which minimizes the relative K-L distance). This, of course, amounts to selecting the model with the lowest AIC, among those models in the candidate model set. We emphasize here that the theory guarantees that the model with the lowest AIC has the smallest K-L distance amongst the models in the model set, conditional on the model set being specified *a priori*. It says nothing whatsoever about whether or not you have a 'good candidate model set' in the first place. Meaning, in addition, that the absolute magnitude of the AIC means nothing (i.e., is not interpretable based on the numerical value of the AIC alone) – it is the relative differences in the AIC values among the candidate models in the model set that is important.

Returning to the European dipper analysis, we note that even though model $\{\varphi_t\, p_t\}$ has the lowest deviance (best fit; 36.40), it also has the greatest number of parameters (11) and the highest AIC value. In contrast, the model deviance for model $\{\varphi_\cdot\, p_\cdot\}$ is the greatest (fits the least well), but because it uses only 2 estimated parameters, it in fact has the lowest AIC of the 4 models.

4.3.2. Some important refinements to the AIC

While Akaike derived an asymptotically unbiased estimator of K-L information, the AIC may perform poorly if there are too many parameters in relation to the size of the sample.

A small-sample (second order) bias adjustment led to a criterion that is called AIC_c (Sugiura 1978; Hurvich & Tsai 1989), that accounts for differences in effective sample size:

$$AIC_c = -2\ln\mathcal{L}(\hat{\theta}) + 2K + \left(\frac{2K(K+1)}{n-K-1}\right),$$

where n is the effective sample size*. Because AIC and AIC_c converge when the effective sample size is large, one can always use AIC_c. As such, the AIC values reported by **MARK** are by default based on this modified (corrected) version of the AIC.

We'll talk about additional modifications to the AIC, particularly to account for lack of fit (c), in the next chapter, but for the moment, conceptually at least, the AIC is simply the sum of 2 times the negative log of the model likelihood and 2 times the number of parameters, adjusted for sample size.

begin sidebar

Maximum likelihood, least-squares, and AIC

You may at this point be wondering what the connection is between 'AIC' and metrics for 'model fit' that you learned in some introductory statistics class (e.g., 'residual sums of squares, RSS').

We'll introduce the connection by means of a fairly familiar example – the MLE for the mean, variance and standard deviation of the normal distribution. The *pdf* (probability distribution function) for the normal distribution is

$$f(x) = \frac{1}{\sigma_x\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\bar{x}}{\sigma_x}\right)^2},$$

from which the likelihood is given as

$$\begin{aligned}\mathcal{L}\left(x_1, x_2, \ldots, x_N | \bar{x}, \sigma_x\right) &= \prod_{i=1}^{N}\left(\frac{1}{\sigma_x\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_i-\bar{x}}{\sigma_x}\right)^2}\right) \\ &= \frac{1}{\left(\sigma_x\sqrt{2\pi}\right)^N} e^{-\frac{1}{2}\sum_{i=1}^{N}\left(\frac{x_i-\bar{x}}{\sigma_x}\right)^2}.\end{aligned}$$

Then

$$\ln(\mathcal{L}) = -\frac{N}{2}\ln(2\pi) - N\ln\sigma_x - \frac{1}{2}\sum_{i=1}^{N}\left(\frac{x_i-\bar{x}}{\sigma_x}\right)^2.$$

Taking the partial derivatives of $\mathcal{L}$ with respect to each one of the parameters and setting them equal to zero yields,

$$\frac{\partial\mathcal{L}}{\partial\bar{x}} = \frac{1}{\sigma_x^2}\sum_{i=1}^{N}(x_i-\bar{x}) = 0,$$

* For many data types, the effective sample size is the number of Bernoulli trials. So, for the live encounter CJS model, the number of animals released and re-released is taken as the effective sample size, because these releases form Bernoulli trials. Similarly for dead recoveries (Chapter 8) and known fate data types (Chapter 17). Difficulties arise for models that have different types of parameters – what constitutes the 'effective sample size' for these data types is an open question.

For example, consider patch occupancy models (introduced in detail in Chapters 22 & 23) – ψ (the overall proportion of patches occupied) has a different sample size than the encounter probability, p: ψ is based on the number of patches, whereas p is based on the number of visits to patches. **MARK** has the capability to specify the effective sample size under the adjustments menu choice of the results browser – this can be useful if there is uncertainty about the effective sample size for a given data type (provided you know what you're doing).

and,

$$\frac{\partial \mathcal{L}}{\partial \sigma_x^2} = -\frac{N}{\sigma_x} + \frac{1}{\sigma_x^3}\sum_{i=1}^{N}(x_i - \bar{x})^2.$$

Solving these two equations simultaneously for $\bar{x}$ and σ_x yields

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad \sigma_x^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2.$$

Now, consider again the $\ln \mathcal{L}$ expression:

$$\ln(\mathcal{L}) = -\frac{N}{2}\ln(2\pi) - N\ln\sigma_x - \frac{1}{2}\sum_{i=1}^{N}\left(\frac{x_i - \bar{x}}{\sigma_x}\right)^2.$$

You might (should) remember from your statistics class that the residual sums of squares (RSS) is given as

$$RSS = \left(x_i - \bar{x}\right)^2.$$

Thus, we can rewrite the $\ln \mathcal{L}$ as

$$\begin{aligned}\ln(\mathcal{L}) &= -\frac{N}{2}\ln(2\pi) - N\ln\sigma_x - \frac{1}{2}\sum_{i=1}^{N}\left(\frac{x_i - \bar{x}}{\sigma_x}\right)^2 \\ &= -\frac{N}{2}\ln(2\pi) - N\ln\sigma_x - \frac{1}{2}\left(\frac{RSS}{\sigma_x^2}\right).\end{aligned}$$

We see clearly that minimizing the RSS is equivalent to minimizing the likelihood.

Finally, differentiating this expression with respect to σ^2 yields

$$\hat{\sigma^2} = \frac{RSS}{N},$$

which when substituted into the likelihood expression yields

$$\begin{aligned}\ln(\mathcal{L}) &= -\left(\frac{N}{2}\right)\ln(2\pi) - N\ln\sigma_x - \frac{1}{2}\left(\frac{RSS}{\sigma_x^2}\right) \\ &= C - \frac{N}{2}\ln\left(\frac{RSS}{N}\right) - \frac{N}{2},\end{aligned}$$

where C is the constant $-(N/2)\ln(2\pi)$. Thus, we can write the AIC in terms of RSS as

$$\begin{aligned}\text{AIC} &= -2\ln\mathcal{L} + 2K \\ &= N\ln\left(\frac{RSS}{N}\right) + 2K.\end{aligned}$$

end sidebar

4.3.3. BIC – an alternative to the AIC

While the AIC has been shown to be a good omnibus approach to model selection, there are some theoretical considerations which may justify consideration of an alternative model selection criterion.

One such measure is the BIC (Bayes Information Criterion), which can be used instead of the AIC in **MARK** – simply select '**File | Preferences | Display BIC instead of AIC**'.

BIC or QBIC are alternative model selection metrics to AIC_c or $QAIC_c$. The number of parameters in the model is K. The BIC depends on the number of parameters as

$$\text{BIC} = -2\ln\mathcal{L}\left(\hat{\boldsymbol{\theta}}\right) + K\ln\left(n_e\right),$$

as does the QBIC (*quasi*-BIC)

$$\text{QBIC} = \frac{-2\ln\mathcal{L}\left(\hat{\boldsymbol{\theta}}\right)}{\hat{c}} + K\ln\left(n_e\right),$$

where n_e is the effective sample size, and $\hat{c}$ is an adjustment for lack of fit of the general model to the data (this is introduced in the next chapter). If you select the BIC, model weights and model likelihood are also computed using BIC instead of AIC_c, so that model averaging is also conducted from the BIC.

When should you use BIC versus AIC? This is a very deep question, and we can only briefly describe some of the issues here. In general, recent research (much of it collated in Burnham & Anderson 2004) suggests there are distinct contexts (say, model sets consisting of simple versus complex models) for which BIC outperforms AIC (generally, when the approximating models in the model set are simple – relatively few 'main effect' factors), or where AIC outperforms BIC (when models are multi-factorial, and generally more complex). AIC is often claimed (equally often without much empirical support) to 'over-fit' – select models which are overly parameterized (relative to the true generating model), whereas the BIC has been suggested to 'under-fit' – select models which are less parameterized than the true generating model.*

Why? While the technical reasons for any difference in 'relative performance' are complex, there is a simple intuitive argument based on the fundamental difference in how the AIC and BIC are estimated.

Consider the differences in the following two equations:

$$\text{AIC} = -2\ln\mathcal{L}\left(\hat{\boldsymbol{\theta}}\right) + 2K$$

$$\text{BIC} = -2\ln\mathcal{L}\left(\hat{\boldsymbol{\theta}}\right) + K\ln\left(n_e\right).$$

In simplest terms, the difference between the AIC and the BIC is in terms of the multiplier for K in the 'penalty term' for the number of parameters: 2 for the AIC, versus $\ln(n_e)$ for the BIC. Clearly, $2 \neq \ln(n_e)$. But, more importantly, the multiplier for the AIC (2) is a constant scalar, whereas for the BIC it scales as a function of the effective sample size. Recall that the larger the penalty, the simpler the selected model (all other things being equal). As a result, AIC tends to perform well for 'complex' true models and less well for 'simple' true models, while BIC does just the opposite.

In practice the nature of the true model, 'simple' or 'complex', is never known. Thus a data driven choice of model complexity penalty would be desirable. This is an active area of research. It is important to remember that the AIC is an estimate of the expected Kullback-Leibler discrepancy (discussed earlier), while BIC is (in fact) an asymptotic Bayes factor (see Link & Barker 2006). Since each method was derived with different motivations, it is not surprising that they have quite different theoretical properties.

While a full discussion of these issues is beyond the scope of what we want to present here, it is important to note that focus should not be on 'which model selection criterion is best?', but remembering that 'model selection should be considered as the process of making inference from a set of models, not just a search for a single best model'. As such, whenever possible, use model averaging. Not only does this account for model selection uncertainty regarding estimated parameters and weight of evidence

* '*All generalizations are false, including this one...*' – Alexandre Dumas, or Mark Twain, depending on your source.

for each approximating model, but also, differences between inference under AIC_c versus BIC diminish under model averaging.

Note: why doesn't **MARK** allow you to show both the AIC and BIC values/weights in the same browser? Simple – to help discourage you from using a side-by-side comparison of the two to guide your model selection – doing so would amount to little more than *post hoc* data dredging.

4.4. Using the AIC for model selection – simple mechanics...

The basic mechanics of using AIC_c for model selection in **MARK** are straightforward. The AIC_c is computed for each of the models in the candidate set, and the models are automatically sorted in descending order based on the AIC_c (i.e., the most parsimonious model – the one with the smallest AIC_c value – is placed at the top of the results).

OK – so you run **MARK**, and calculate the AIC_c for each model. What do you do if, say, the model with the lowest AIC_c differs from the next-lowest by only a small amount? How much 'support' is there for selecting one model over the other? Note – we intentionally use the word *support*, rather than *statistical significance*. We'll deal with the issue of 'significance', and related topics, shortly.

As a first step, the models should be calibrated to provide an index of 'relative plausibility' (i.e., the likelihood of the model given the model set), using what are known as *normalized Akaike weights*. These weights (w_i) are calculated for each approximating model (i) in the candidate model set as

$$w_i = \frac{\exp\left(\frac{-\Delta AIC}{2}\right)}{\sum\left\{\exp\left(\frac{-\Delta AIC}{2}\right)\right\}}.$$

What are we doing here, and why? What is the basis for this expression?*

To help understand the basic idea behind normalized AIC weights, and how they are calculated, consider the concept of the likelihood of the *parameters* $\boldsymbol{\theta}$ given a model g_i, and some data x

$$\mathcal{L}\left(\hat{\boldsymbol{\theta}} \mid x, g_i\right).$$

We can extend this basic idea to the concept of the likelihood of the *model* given the data

$$\mathcal{L}\left(g_i \mid x\right) \propto e^{-\frac{1}{2}\Delta_i},$$

where Δ_i is the difference in the AIC value between the model i and the model with the lowest AIC.† So, the likelihood of a model, given the data, is proportional to the difference in AIC between that model, and the model in the model set with the lowest AIC. Normalizing them creates a set of positive values that sum to one (which lends to the interpretation of relative or proportional support in the data for a given model, among the models in the candidate model set).

OK, fine, but you might be asking 'why is the likelihood of a model, given the data, proportional to the difference in AIC between that model and the candidate model with the lowest AIC?'. The following might help. We note that for model i, $\Delta AIC = AIC_i - AIC_0$, where AIC_0 is the minimum AIC in the candidate model set (i.e., the most parsimonious model).

* If you have a background in 'neural nets' and 'reinforcement-learning', you might also recognize this as the 'softmax function'.

† Note that the $-1/2$ term simply cancels out the fact that Akaike multiplied through by -2 to define his AIC. And, by exponentiating the ΔAIC values, we're effectively eliminating the log scale in the AIC expression.

Given that AIC = $-2\ln\mathcal{L} + 2K$, then

$$\begin{aligned}\Delta\text{AIC} &= \text{AIC}_i - \text{AIC}_0 \\ &= \left(-2\ln(\mathcal{L})_i + 2K_i\right) - \left(-2\ln(\mathcal{L})_0 + 2K_0\right) \\ -\frac{\Delta\text{AIC}}{2} &= \ln(\mathcal{L})_i - K_i - \ln(\mathcal{L})_0 + K_0.\end{aligned}$$

Thus,

$$\exp\left(\frac{-\Delta\text{AIC}_i}{2}\right) = \left(\frac{\mathcal{L}_i}{\mathcal{L}_0}\right)\exp\left(K_0 - K_i\right).$$

So, the term $\exp(-\Delta_i/2)$ in the expression used to calculate AIC weights is in fact a *likelihood ratio*, corrected for the difference in the number of parameters estimated from the models.

How are these weights used? A given w_i is considered as the weight of evidence in favor of model i as being the actual K-L best model in the set. These are termed *model probabilities* (in fact, they are also formally Bayesian posterior model probabilities; Burnham & Anderson 2004). So, w_i is the probability that model i is the actual K-L best model in the set. The bigger the w_i value, the bigger the weight.

For example, consider the following set of models, with their ΔAIC_c values and Akaike weights.

Model	ΔAIC	Akaike weight (w_i)
1	1.6	0.278
2	0.0	0.619
3	7.0	0.084
4	13.5	0.001
5	4.0	0.084
total		1.000

Here, model 2 is clearly the best (largest AIC weight), but how much better is it than the next best model (model 1)? The Akaike weights let us state that the best model (model 2) is over twice as well supported as the next best model (model 1), since (0.619/0.278) = 2.23. The remaining models (3, 4 and 5) have essentially no support in the data, relative to models 1 and 2.

MARK calculates Akaike weights automatically. For our European dipper analysis, here again are the AIC values, the ΔAIC values, and their relative (normalized) weights:

Results Browser: Live Recaptures (CJS)

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{phi(.)p(.)}	322.5527	0.0000	0.96003	1.0000	2	41.8147
{phi(t)p(.)}	330.0567	7.5040	0.02253	0.0235	7	38.8147
{phi(.)p(t)}	330.6794	8.1267	0.01650	0.0172	7	39.4374
{phi(t)p(t)}	336.4343	13.8816	0.00093	0.0010	11	36.4013

In this case, the results are clear – model $\{\varphi_.\, p_.\}$ is much better supported than any other model – the AIC for the next best model differs by 7.50, and has approximately 43-times less support than the best model.

Consider again are the results from the analysis of the swift data (shown at the top of the next page). Again, the results are quite clear – model $\{\varphi_c\, p_t\}$ is much better supported by the data than any other model in the candidate model set. The AIC_c for the next best model (model $\{\varphi_c\, p_\cdot\}$) differs by 3.72, and has approximately 6-times less support than the best model. Note that in this example, unlike for the European dipper data, the model with the fewest parameters is not the most parsimonious model. Again, ranking based on the AIC balances fit and precision, given the data.

If you look closely, you'll notice there is a column in the results browser labeled '**`model likelihood`**'. Here, *likelihood* has a technical meaning, that can be quantified and should not be confused with probability.* For example, if person A holds five raffle tickets and person B has one, person A is five times more *likely* to win than person B. We do not know the absolute *probability* of either person winning without knowing the total number of raffle tickets. The reported 'model likelihood' is the AIC (or BIC) *weight* for the model of interest divided by the AIC (or BIC) weight of the best model in the browser.

Results Browser: Live Recaptures (CJS)

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{phi(c)p(t)}	369.8080	0.0000	0.85650	1.0000	9	111.6644
{phi(c)p(.)}	373.5263	3.7183	0.13345	0.1558	3	128.1990
{phi(t)p(.)}	379.0123	9.2043	0.00859	0.0100	8	123.0601
{phi(t)p(t)}	382.8807	13.0727	0.00124	0.0014	13	115.7384
{phi(c*t)p(.)}	386.4962	16.6882	0.00020	0.0002	15	114.7094
{phi(c*t)p(t)}	391.4103	21.6023	0.00002	0.0000	20	107.5633

For example, the model likelihood reported for model $\{\varphi_c\, p_\cdot\}$ for our swift analysis (shown above) is calculated as the ratio the AIC weight for model $\{\varphi_c\, p_\cdot\}$ and the AIC weight for the model with the smallest AIC, $\{\varphi_c\, p_t\}$: (0.13345/0.85650) = 0.1558. In other words, the odds of model $\{\varphi_c\, p_\cdot\}$ being the K-L best model, rather than model $\{\varphi_t\, p_t\}$, is given as (0.1558 : 1.000) – or, the likelihood that model $\{\varphi_t\, p_t\}$ is the K-L best model is (1/0.1558) = 6.42 times greater than model $\{\varphi_c\, p_\cdot\}$.

This likelihood value is the *strength of evidence* of this model relative to the model with the lowest AIC in the set of models considered, and is the reciprocal of the formal *evidence ratio* (discussed in the following `-sidebar-`).

`begin sidebar`

AIC weights, evidence ratios, and 'rules of thumb'

The likelihood of an approximating model, g_i, given the data, is computed as:

$$\mathcal{L}\left(g_i \mid \text{data}\right) \propto \exp\left(-\frac{1}{2}\Delta_i\right),$$

where Δ_i is the difference in AIC between model g_i and the model with the lowest AIC.

To better interpret the *relative* likelihood of a model, given the data and the candidate set of models, we normalize the model likelihoods to be a set of 'Akaike weights', w_i, which sum to 1:

$$w_i = \frac{\exp\left(\frac{-\Delta\text{AIC}}{2}\right)}{\sum\left\{\exp\left(\frac{-\Delta\text{AIC}}{2}\right)\right\}}.$$

* You can add a column to the browser showing the maximized likelihood for the model (i.e., $-2\ln\mathcal{L}$) by selecting that option in '**`File`** | **`Preferences`**'.

It is important to remember that the w_i depend on the entire model set – if a model is added or dropped, the w_i must be recomputed for all the models in the modified model set. A given w_i is considered as the weight of evidence in favor of model g_i being the actual K-L best model, given that one of the models in the model set must be the K-L best model of that set of models.

For the estimated K-L best model, g_{min}, $\Delta\text{AIC} = 0$. Thus, for that model,

$$\mathcal{L}\left(g_{min} \mid \text{data}\right) \propto \exp\left(-\frac{1}{2}\Delta_i\right) \equiv 1.$$

Thus, the odds for the *i*th model actually being the K-L best model are thus given by the ratio

$$\frac{1}{e^{-1/2\Delta_i}} \equiv e^{1/2\Delta_i} \equiv \frac{w_1}{w_i},$$

where w_1 is the normalized AIC weight of the model with the smallest AIC value among the models in the candidate model set. Such ratios are termed *evidence ratios*, and represent the evidence about fitted models as to which is 'better' in the information theoretic sense. Evidence ratios provide a measure of the *relative likelihood* of one hypothesis (model) versus another.

Evidence ratios are invariant to other models in the model set, whereas model weights depend on all the other models in the candidate model set. Inference should be about models and parameters, given data; however, we note that *P*-values are probability statements about *data*, given null models. Model probabilities and evidence ratios provide a means to make inference directly about models and their parameters, given data.

For example, if we delete the 3 lowest-ranked models from our analysis of the swift data,

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{phi(c)p(t)}	369.8080	0.0000	0.85775	1.0000	9	111.6644
{phi(c)p(.)}	373.5263	3.7183	0.13364	0.1558	3	128.1990
{phi(t)p(.)}	379.0123	9.2043	0.00860	0.0100	8	123.0601

we see that the AIC weights change, but not the model likelihoods, calculated relative to the model with the lowest AIC. Remember – AIC weights are calculated relative to all other models in the candidate model set, while model likelihoods are calculated for a given model relative to the model with the lowest AIC. And, the reciprocal of the model likelihood is the evidence ratio.

Model likelihoods and evidence ratios are continuous measures. It is important to understand that there is a nonlinearity in evidence ratios as a function of the ΔAIC_i values. If we consider the ratio

$$\frac{w_1}{w_j} \equiv \frac{1}{e^{-1/2\Delta_j}} \equiv e^{1/2\Delta_j},$$

as a comparison of the evidence for the best model (lowest AIC) compared with any other model j, then we can generate following table:

Δ_j	2	4	8	10	15
evidence ratio	2.7	7.4	54.6	148.4	1808.0
model likelihood	0.3704	0.1352	0.0183	0.0067	0.0006

It is just this nonlinearity in the relationship between ΔAIC and the evidence ratio which lead to the 'rules of thumb' introduced by Burnham & Anderson. They suggested that when the difference in AIC between two models (ΔAIC) is < 2, then we are reasonably safe is saying that both models have approximately equal weight in the data. If $2 < \Delta\text{AIC} < 7$, then there is considerable support for a real difference between the models, and if $\Delta\text{AIC} > 7$, then there is strong evidence to support the

conclusion of differences between the models. From the preceding table, we see clearly that when ΔAIC ≤ 4, the model likelihood is $\gg \alpha = 0.05$. Meaning, there is a strong probability that any model with a ΔAIC ≤ 4 is, in fact, the K-L best model. Conversely, if ΔAIC ≥ 7, then there is a decreasing probability that the model is in fact the K-L best model, and we would conclude that there is strong evidence of real differences between the models.

Consider again the results from the swift example. Given the available data, a model where survival is fixed (i.e., constant) among years, but which differs between colonies, and where encounter probability varies over time, but not between colonies $\{\varphi_c\, p_t\}$ is $(0.8565/0.1335) = 6.42$ times more likely than a model where survival is again fixed (i.e., constant) among years, but which differs between colonies, and where encounter probability does not vary between colonies or over time $\{\varphi_c\, p_.\}$.

From a practical standpoint, when reporting model selection results (in a paper, or report), it is useful to report both AIC weights and either model likelihoods or evidence ratios (reporting both would be redundant, since the evidence is simply the reciprocal of the model likelihood; for example, in the swift analysis, the relative likelihood of model $\{\varphi_c\, p_.\}$ to model $\{\varphi_c\, p_t\}$ is 0.1558, from which we calculate the evidence ratio as $(1/0.1558) = 6.42$).

end sidebar

However, while AIC weights, and model likelihoods, and 'rules of thumb' are convenient, they don't quantify the degree of uncertainty in our model selection, over all models in the model set.

What do we mean by 'uncertainty', in the context of model selection? In any analysis, there is uncertainty in terms of which model is the 'best model'. In our swift analysis, for example, we determined which model is the most parsimonious, but how far from 'truth' is this model? The most parsimonious model is merely the model which has the greatest degree of support in the data. It is not 'truth' – it merely does somewhat better at explaining variation in the data than do other proposed models (we add in passing that '*All models are wrong, some are useful*' – G. Box). There is 'uncertainty' in terms of which model is the 'best model'.

How can we measure, or at least account for, this uncertainty? One approach to this problem is to base the inference on the entire set of models – an approach termed multimodel inference, or model averaging. We cover this in the next section.

4.5. Model uncertainty: an introduction to model averaging

In the analysis of the swift data set we considered earlier in this chapter, we compared the survival probability of birds as a function of the quality of their nesting colony ('good' versus 'poor'). We came to the conclusion that there was some support in the data for a colony effect (the 2 most parsimonious models both had a colony effect in survival, and the sum of their respective normalized AIC weights was 0.985, indicating $\approx 98.5\%$ of the support in the data are for these 2 models). If we look at the estimates from the most parsimonious model in our model set (model $\varphi_c\, p_t$), we see that the estimate of survival for the good colony was 0.77 (SE 0.041), while the estimate for the poor colony was 0.58 (SE 0.082). Our previous analysis seems to support the contention that this was a meaningful difference between the two colonies.

However, suppose you are charged with drafting a conservation plan for this population, and want to condition your recommendations on the possibility of differences in survival between the 2 colonies. Perhaps you want to use the estimates of survival in some form of model, projecting the likely consequences of one or more proposed actions. While how you might do this is obviously beyond the scope of this book (since it has little to do with **MARK** directly), it does raise at least one issue which is worth noting at this point. What should we use as the estimates of survival?

The obvious answer would be to use the estimates from the most parsimonious model alone. However,

taking this approach clearly ignores one salient fact – the estimates of sampling variance from a given model do not include model selection uncertainty – they are conditional only on the model used for the estimation. In other words, using the estimates from a single model in the candidate model set, even if it is the most parsimonious model in the set, ignores model uncertainty. For example, for the swift analysis, model $\{\varphi_c\, p_t\}$ has a AIC_c weight of 0.8565, while model $\{\varphi_c\, p_\cdot\}$ has a AIC_c weight of 0.1335. While model $\{\varphi_c\, p_t\}$ is clearly better supported, there is still uncertainty – there is at least some chance (approximately 13% chance) that in fact model $\{\varphi_c\, p_\cdot\}$ is the correct model, relative to the other models in the candidate model set.

Since there is uncertainty in which model is the correct model, we might consider accommodating this uncertainty in the estimates we report (or use subsequently in some model). This is where 'model averaging' comes in. The simplest way to think of what model averaging is all about is to recall the concept of 'weighted averages' from your introductory statistics class. What we want to do is take the estimates from our various models, and weight them by the relative support for that model in the data.

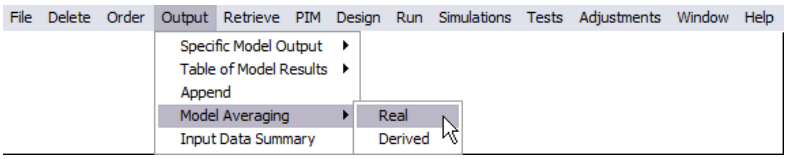
More precisely, we calculate an average value for a parameter θ by averaging over all models in the candidate model set with common elements in the parameter structure, weighted by normalized AIC model weights (*sensu* Buckland *et al*., 1997, Burnham & Anderson 2004):

$$\begin{aligned}\text{avg}(\hat{\theta}) &= \bar{\hat{\theta}} \\ &= \sum_{i=1}^{R} w_i \hat{\theta}_i,\end{aligned}$$

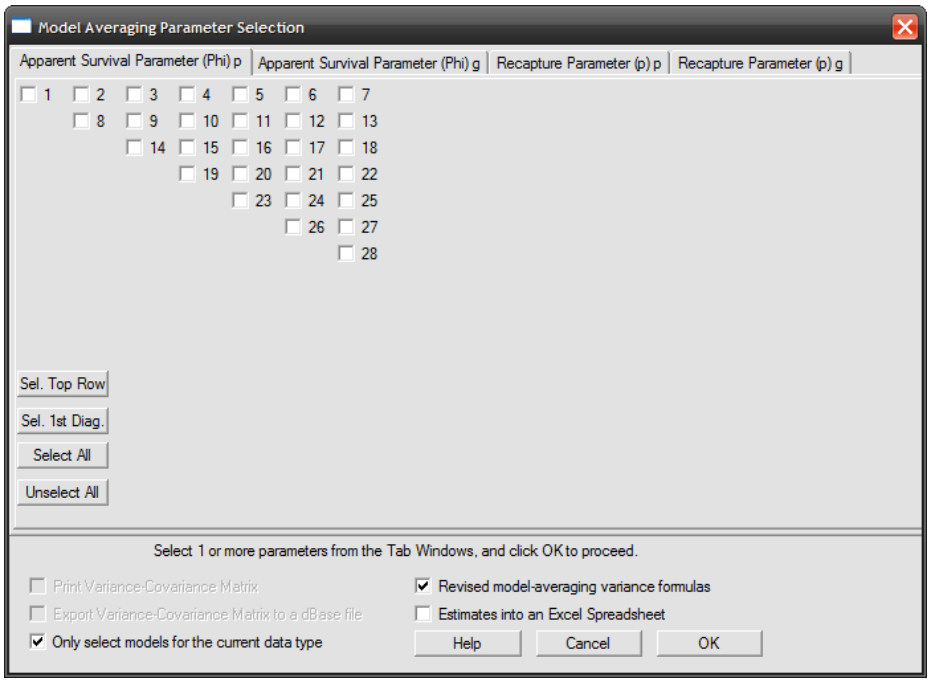
where w_i is the Akaike weight for model i. Hopefully, this makes intuitive sense – we weight the estimates of the various parameters by the model weights, which relate to how much support there is in the data for that model. We want to give higher weight to estimates from models with greater support in the data.

Let's see how we actually do this in **MARK**. Suppose for example we're interested in reporting the live encounter probabilities for each year in the swift study. What would our 'best' estimates be for annual encounter probability? We see from the results browser that the most parsimonious model, $\{\varphi_c\, p_t\}$, has time-dependence in p (i.e., p_t), while the next best supported model has constant p.

If we were simply to report the estimates from the most parsimonious model, we would be ignoring model section uncertainty. Instead, we want to average our estimates over the models in the candidate model set. To do this, pull down the '**Output**' menu, and select '**Model averaging**', and then '**Real parameters**'.



This will spawn the window shown at the top of the next page. Notice along the top there are 4 'tabs' – one tab for each of the 4 main parameters (survival for the poor colony, survival for the good colony, recapture for the poor colony, and recapture for the good colony). Also notice also that there is a triangular matrix of 'check boxes' which is structurally equivalent to the structure of the PIMs. Structurally equivalent, but...what do the numbers represent? Clearly, they're not equivalent to the indexing we used in our analysis of the swift data set – are they? No, as written, they're not, but...they're directly related. In fact, the numbering you see in the model averaging window corresponds to the index

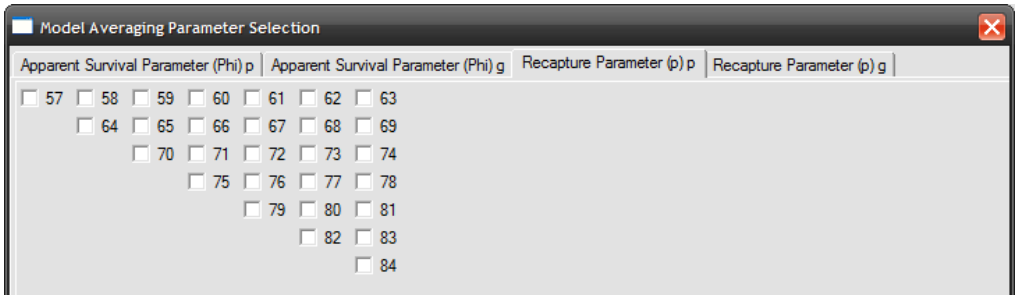


values that you would use in a PIM if the model had complete (`cohort` × `time`) dependence.

Say...what?? Well, cohort models and other extensions of the simple time-dependent model is something we'll get to in chapter 7. But, for now, simply think of the indexing you see in the model averaging window as corresponding to the possibility that there is a different estimate for each time period (time-dependence), and that within each time-period, the estimate might vary as a function of what year the organism was marked and released (cohort-dependence). Since there are 28 combinations of time and cohort in our swift data set, that is why the model averaging window has 28 cells, numbered 1 to 28. The numbering is left to right within each cohort in succession.

Again, do not get too concerned at this point if the distinction between 'time period' and 'cohort' is a bit confusing – by the time you reach the end of chapter 7, you'll fully understand the distinction.

First, we need to 'tell' **MARK** we want to average the recapture estimates across models. We'll start by selecting the '**`recapture parameter (p) P`**' tab, corresponding to the recapture probabilities for the poor (**P**) colony. Do this by clicking on that tab in the model averaging window:



Now, you'll see the triangular matrix is numbered 57 through 84. We're interested in annual estimates

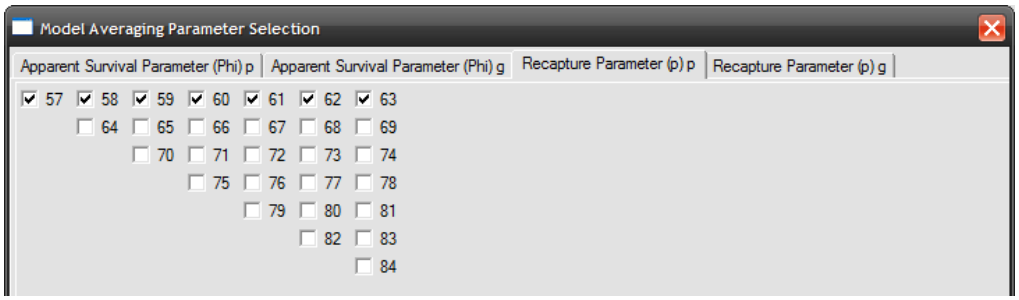
of p. As such, we need to 'tell' **MARK** to derive average values for each year. *Note*: make sure that the radio-button for '**`Revised model-averaging variance formulas`**' is checked.

How? Think back to what the PIM for a time-dependent model looks like. Recall from earlier in this chapter that the PIMs for time-dependent recapture, but no difference between colonies, looked like those shown at the top of the next page. Within each column (corresponding to a particular year) the index values in the PIM are the same. In other words, the survival probability in a given year does **not** depend on the year in which an individual was first marked and released (i.e., does not depend on its cohort). Thus, within the triangular matrix in the model averaging window, it doesn't matter which element of a given column you 'check', since all elements within a column are equivalent for our models!

15	16	17	18	19	20	21
	16	17	18	19	20	21
		17	18	19	20	21
			18	19	20	21
				19	20	21
		recapture			20	21
		'poor'				21

15	16	17	18	19	20	21
	16	17	18	19	20	21
		17	18	19	20	21
			18	19	20	21
				19	20	21
		recapture			20	21
		'good'				21

Thus, checking each of the elements in the first row (as shown below) will yield exactly the same results as clicking any one element in each of the columns. Having said that, it is probably a good idea to be somewhat systematic about things (in this case, perhaps by checking each of the elements in the first row of the matrix) – it will help you keep track of things later on.



Once you've specified the cells for the recapture parameter for the 'poor' colony, do the same for the 'good' colony, by clicking on the tab for that parameter, and again checking one element of each column. Once you've done so, you're ready to click the '**`OK`**' button to start the averaging. Before you do, note that in the lower left hand corner, an option to '**`only select models for the current data type`**' is checked by default. Ignore this for now (it will become important down the road).

So, go ahead and click the '**`OK`**' button. **MARK** will open up the editor, and present you with the 'averaged' results (the first 2 years from the output are shown at the top of the next page).

Note that the output is arranged by the index value you selected from the model averaging window. The first year corresponds to index value 57. Thus, you will need to 'keep track' of which index value corresponds to which year. For each parameter, each of the models in the candidate model set is listed, along with the model weight, the estimate from that model, and the standard error of the estimate.

At the bottom of the '`Estimate`' and '`Standard Error`' columns are the 'averaged' values. For parameter 57 (corresponding to the recapture probability for the poor colony on the first occasion), the model averaged value is 0.88047. The weighted SE is 0.08045. This is followed by something called

```
                    Recapture Parameter (p) p Parameter 57
Model                                       Weight    Estimate      Standard Error
------------------------------------------ -------   -------------- --------------
{Phi(g) p(t) PIM}                          0.85650   0.9088828      0.0855604
{Phi(g) p(.) PIM}                          0.13345   0.7071420      0.0494676
{Phi(t) p(.) PIM}                          0.00859   0.7434219      0.0484090
{Phi(t) p(t) PIM}                          0.00124   0.8811188      0.1099380
{Phi(g*t) p(.) PIM}                        0.00020   0.7446420      0.0482857
{Phi(g*t) p(t) PIM}                        0.00002   0.8821994      0.1091012
------------------------------------------ -------   -------------- --------------
Weighted Average                                     0.8804713      0.0804479
Unconditional SE                                                    0.1072292
95% CI for Wgt. Ave. Est. (logit trans.) is 0.4999712 to 0.9819060
Percent of Variation Attributable to Model Variation is 43.71%
```

the '`Unconditional SE`', which is somewhat larger (numerically) than the weighted SE. Below the unconditional SE is a 95% CI for the model weighted average, and a statement concerning the percentage of the variation attributable to model variation (43.71% for the present example). We'll deal with the distinction between the two SE's, and the variation due to model variation, in a moment. Note that the model averaged value of 0.88047 is somewhat lower than the estimate from the single most parsimonious model (0.9089). That is because it has been 'weighted down' by the other models in the candidate model set. Note also that only models with an AIC weight > 0 are shown (since only models with an AIC weight > 0 contribute to the weighted average).

There is one additional point we need to make here – have a look at the model averaged estimate for the final encounter probability (p_8):

```
                    Recapture Parameter (p) p Parameter 63
Model                                       Weight    Estimate      Standard Error
------------------------------------------ -------   -------------- --------------
{Phi(g) p(t) PIM}                          0.85650   0.4633089      0.0948795
{Phi(g) p(.) PIM}                          0.13345   0.7071420      0.0494676
{Phi(t) p(.) PIM}                          0.00859   0.7434219      0.0484090
{Phi(t) p(t) PIM}                          0.00124   0.5831049      70.9087680
{Phi(g*t) p(.) PIM}                        0.00020   0.7446420      0.0482857
{Phi(g*t) p(t) PIM}                        0.00002   0.5707244      114.6945500
------------------------------------------ -------   -------------- --------------
Weighted Average                                     0.4984620      0.1783423
Unconditional SE                                                    2.5472565
95% CI for Wgt. Ave. Est. (logit trans.) is 0.0000000 to 1.0000000
```

We see that the estimated SE is 2.547 (with an associated 95% CI of 0 → 1). Such a CI does not inspire much confidence (pun intended). Clearly, there is a problem here. If you look closely at the model averaging output for p_8 (above), you'll see that the 'culprit' is the extremely high conditional standard errors for models $\{\varphi_t\, p_t\}$ and $\{\varphi_{g*t}\, p_t\}$.

With a bit of thought – and perhaps a peek at the reconstituted estimates for both models – you might be able to guess the underlying reason. The problem is that both of these models are fully time-dependent for both parameters, and as such, the terminal φ and p parameters are confounded (i.e., not separately identifiable). One of the characteristics of such 'confounded' parameters is extremely large (i.e., implausible) SE's, which clearly have the potential to strongly influence the estimate of the unconditional standard error (especially if the model(s) with confounded parameters have some significant support in the data).

In such cases there is no absolute rule on how to best handle model averaging, but we suggest the following strategy: (i) if models with confounded parameters – or models with parameters which are poorly estimated given the data – having little → no support in the data, then it is generally acceptable to drop those models from the candidate model set, and re-run the averaging. However, (ii) if the 'problem models' have appreciable support in the data, you'll need to be more careful. You might choose simply to average only those 'well-estimated' parameters (for instance, in our example, $p_2 \rightarrow p_7$, leaving out p_8), but you need to first confirm that those models aren't well-supported simply because of the poorly estimated parameters. Again, there is no perfect solution.

What about survival? The model averaged estimates for survival are shown below:

year	*poor colony*	*good colony*
1	0.577354	0.768136
2	0.575062	0.765835
3	0.575294	0.766077
4	0.575638	0.766456
5	0.575610	0.766379
6	0.576261	0.766969
7	0.572892	0.763680

Recall that our 2 most parsimonious models (comprising ~ 99% of the support in the data) had a colony effect, but no time-dependence. We see from the model average values that there is little annual variation in the estimates – not surprising given that the models with any time-dependence had very little support in the data. However, there is a clear difference in the model averaged estimates between colonies.

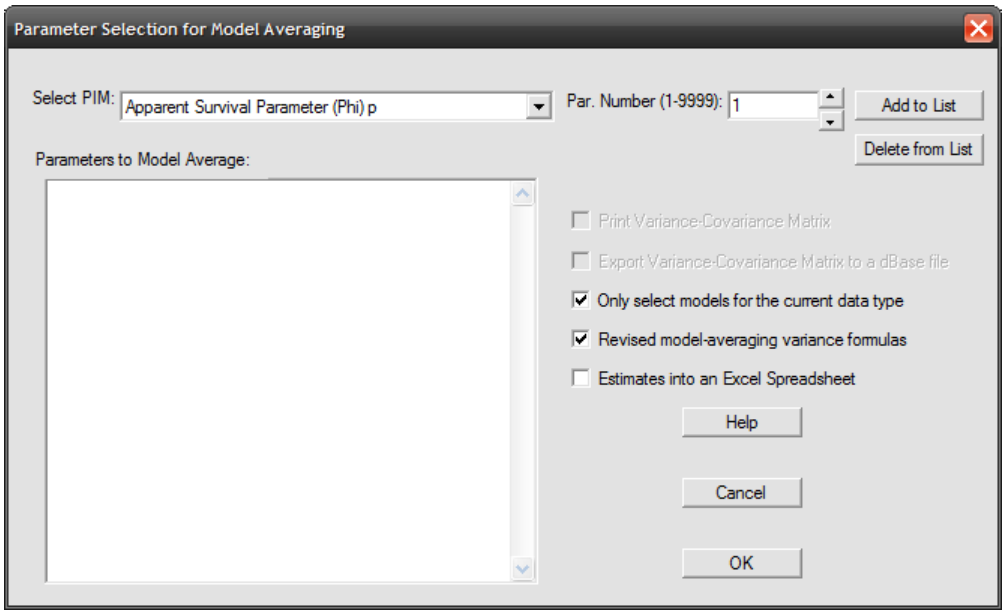
We will revisit model averaging again – it's a very important concept, since it relates directly to the important issue of 'model selection uncertainty'.

begin sidebar

Non-interactive model averaging

Some problems are too large for the interactive interface (which we just introduced) for specifying the parameters to be model averaged. An option is available in the '**`File | Set Preferences`**' window to change the default interface to a less interactive mode.

To see how it works, select the '**`non-interactive`**' option in the preferences window, and restart **MARK**. Pull up the same analysis we used to demonstrate the interactive model averaging window (the *Apus apus* analysis). Again, select '**`Output | Model averaging | Real`**'. This will bring up the '**`Non-interactive model averaging window`**', shown below:



As with the interactive model-averaging described earlier, make sure that the radio-button for '**Revised model-averaging variance formulas**' is checked.

Along the top of this window, you'll see that there is a pull-down menu, which lets you select which of the parameters you want to average (for this example, there are 4 parameters you could select among: $\varphi_g, \varphi_p, p_g$ and p_p). To the right of this pull-down menu is a box where you select the index number of the parameter you want to average. Note that it defaults to 1 → 9999. However, as soon as you click inside this box, it will update to indicate the range of index values used in the data set you're analyzing (in this example, 1 → 28).

Now, suppose you want to model average parameters 1 → 7, which correspond to $\varphi_1 \rightarrow \varphi_7$ for the good colony. You simply toggle through the numbers 1 → 7, selecting the '**Add to list**' button for each number in turn. As with the interactive model averaging window, you can output various aspects of the averaging (e.g., into an Excel spreadsheet), simply by selecting the appropriate radio button on the right-hand side. Note that two of the options (for printing and exporting the variance-covariance matrix) are greyed out until you select at least two parameters for averaging.

end sidebar

4.5.1. Model averaging: deriving SE and CI values

In the preceding section, we noted that two different SE's for a model averaged parameter estimate were given by **MARK**: a weighted average SE (in effect, the average of the individual model SE's weighted by their respective AIC weights), and the 'unconditional SE'. These in turn were followed by a 95% CI, and a statement concerning the proportion of the variation due to model selection uncertainty. Why two different SE's? Which one is the 'right one' to report? Where does the 95% CI come from? And what does 'model selection uncertainty' refer to in the context of model averaging?

In general, the precision of an estimator should ideally have 2 *variance components*: (1) the *conditional* sampling variance, $\widehat{\mathrm{var}}(\hat{\theta}_i \,|\, \mathcal{M}_i)$, given model i, and (2) variation associated with *model selection uncertainty*. Buckland *et al*. (1997) provide an effective method to estimate an estimate of precision that is **not** conditional on a particular model (their estimator was subsequently revised in Burnham & Anderson 2004). Assume that some scalar parameter θ is in common to all models being considered in the candidate model set. [If our focus is on a structural parameter that appears only in a subset of our full set of models, then we must restrict ourselves to that subset in order to make the sort of inferences considered here about the parameter of interest.]

So, estimates of the SE for a given model are *conditional* on that model. How do we get an *unconditional* estimate of the SE for the parameter averaged over models?

From Burnham & Anderson (2004), we will take the estimated *unconditional* variance of $\hat{\theta}$ as

$$\widehat{\mathrm{var}}(\bar{\hat{\theta}}) = \sum_{i=1}^{R} w_i \left[\widehat{\mathrm{var}}(\hat{\theta}_i \mid \mathcal{M}_i) + (\hat{\theta}_i - \bar{\hat{\theta}})^2 \right],$$

where

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

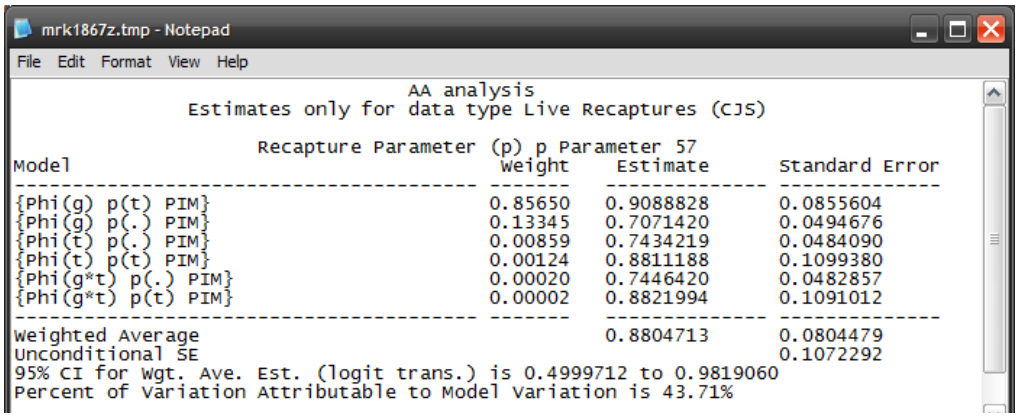
and the w_i are the Akaike weights (Δ_i) scaled to sum to 1. The subscript i refers to the i^{th} model. The value $\bar{\hat{\theta}}$ is a weighted average of the estimated parameter $\hat{\theta}$ over R models ($i = 1, 2, \ldots, R$).

This estimator of the *unconditional* variance is clearly the sum of 2 components: (1) the conditional sampling variance $\left(\widehat{\text{var}}(\hat{\theta}_i \,|\, \mathcal{M}_i)\right)$ and (2) a term for the variation in the estimates across the R models $\left(\hat{\theta} - \bar{\hat{\theta}}\right)^2$. The estimated *unconditional* SE is given as

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

It is this unconditional variance (and associated CI) that you would report, since it accounts for both conditional model-specific variation, as well as variation resulting from model selection uncertainty (i.e., among models in the candidate model set). **MARK** gives you an estimate of the proportion of the variance in the model averaged parameter estimate owing to model selection uncertainty.

Let's work through an example, using the model averaged estimates for p_2 for the poor colony:


mrk1867z.tmp - Notepad

File Edit Format View Help

AA analysis
Estimates only for data type Live Recaptures (CJS)

Recapture Parameter (p) p Parameter 57

Model	Weight	Estimate	Standard Error
{Phi(g) p(t) PIM}	0.85650	0.9088828	0.0855604
{Phi(g) p(.) PIM}	0.13345	0.7071420	0.0494676
{Phi(t) p(.) PIM}	0.00859	0.7434219	0.0484090
{Phi(t) p(t) PIM}	0.00124	0.8811188	0.1099380
{Phi(g*t) p(.) PIM}	0.00020	0.7446420	0.0482857
{Phi(g*t) p(t) PIM}	0.00002	0.8821994	0.1091012
Weighted Average		0.8804713	0.0804479
Unconditional SE			0.1072292

95% CI for Wgt. Ave. Est. (logit trans.) is 0.4999712 to 0.9819060
Percent of Variation Attributable to Model Variation is 43.71%

Start by taking a weighted average of the conditional (model-specific) SE's, weighting by the model-specific Akaike weights w_i:

$$\left[(0.08556 \times 0.8565) + (0.04947 \times 0.1335) + \ldots\right]/1.0 = 0.08045,$$

which is what is reported by **MARK** (as shown, above). Again, this is a simple weighted average, and should not be reported as the SE for the model averaged parameter estimate.

Now, let's calculate the *unconditional* SE, which **MARK** reports as 0.10723. We start by estimating the unconditional variance of the parameter as

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

To perform this calculation, we'll need the model-specific estimates of the variance (on the normal probability scale) for the parameter. These can be derived from the model averaging output by squaring the reported conditional SE for each model. Given the calculated model averaged value for the parameter, $\bar{\hat{p}}_{2,g} = 0.88047$, then

$$\widehat{\text{var}}(\bar{\hat{p}}_{2,g}) = \sum_{i=1}^{R} w_i \left[\widehat{\text{var}}(\hat{p}_{2,g} \mid \mathcal{M}_i) + \left(\hat{p}_{2,g_i} - \bar{\hat{p}}_{2,g}\right)^2\right]$$

$$
\begin{aligned}
&= 0.85650\left[0.007321 + (0.90888 - 0.88047)^2\right] \\
&\quad + 0.13345\left[0.002447 + (0.70714 - 0.88047)^2\right] \\
&\quad + \cdots \\
&\quad + 0.00002\left[0.011903 + (0.88220 - 0.88047)^2\right] \\
&= 0.011498.
\end{aligned}
$$

So, our estimate of the *unconditional* variance of the encounter probability p_2 for the poor colony is 0.011498. The standard error is estimated simply as the square-root of the variance: $\sqrt{0.011498} = 0.10723$, which is what is reported by **MARK** (to within rounding error). The 95% CI reported by **MARK** is [0.5000, 0.9819]. How the reported 95% CI is calculated is discussed in the following -sidebar-.

Confidence intervals can also be constructed using a profile likelihood approach, but this is beyond the scope of our discussion at this point. In addition, we will leave the consideration of the reported proportion of the variation due to model selection uncertainty to a later chapter.

begin sidebar

SE and 95% CI

The usual (familiar) approach to calculating 95% confidence limits for some parameter θ is $\hat{\theta} \pm (1.96 \times \widehat{\mathrm{SE}})$. Is this how **MARK** calculates the 95% CI on the *real* probability scale? Take the example we just considered, above – the estimated SE for $\tilde{\hat{\varphi}} = 0.88047$ was $\sqrt{0.011498} = 0.10723$. So, you might attempt to calculate the 95% CI on the real probability scale simply as $0.88047 \pm (1.96 \times 0.10723) = [0.67030, 1.09064]$. However, not only is this not what is reported by **MARK** ([0.5000, 0.98191]), but it isn't even 'reasonable', since the calculated UCL (1.09064) is > 1, which is clearly not possible for a $[0, 1]$ bounded parameter on the real probability scale.

Why the difference between what **MARK** reports and what we have calculated by hand using $\hat{\theta} \pm (1.96 \times \widehat{\mathrm{SE}})$? The difference is because **MARK** first calculates the model-averaged 95% CI on the *logit* scale, before back-transforming to the real probability scale. Doing so ensures that the back-transformed CI will be $[0, 1]$ bounded. The logit scale, and back-transforming from the logit scale to the normal probability scale, are discussed in detail in Chapter 6. But, to briefly demonstrate 'what **MARK** is doing', consider the following example.

Suppose that the candidate model set for the swift analysis is reduced to just two models: $\{\varphi_t\, p_t\}$ and $\{\varphi_g\, p_.\}$ (we do this only to make the demonstration simpler). Using the logit link for both models, the results for fitting these two candidate models to the data are shown below:

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{phi(g)p(t)}	369.8080	0.0000	0.86520	1.0000	9	111.6644
{phi(g)p(.)}	373.5263	3.7183	0.13480	0.1558	3	128.1990

If we run through the model-averaging routine for (say) $\hat{\varphi}_3$ for the 'good' colony, **MARK** reports the following:

```
                Apparent Survival Parameter (Phi) Group 1 Parameter 3
Model                                      Weight    Estimate      Standard Error
---------------------------------------- -------  -------------- --------------
{phi(g)p(t)}                             0.86520   0.5770598      0.0771524
{phi(g)p(.)}                             0.13480   0.5553164      0.0767936
---------------------------------------- -------  -------------- --------------
Weighted Average                                   0.5741287      0.0771040
Unconditional SE                                                  0.0774609
95% CI for Wgt. Ave. Est. (logit trans.) is 0.4201336 to 0.7149725
Percent of Variation Attributable to Model Variation is 0.92%
```

So, the model averaged estimate for $\bar{\hat{\varphi}}_{3,p}$ is 0.5741295, with an unconditional SE of 0.077461. The reported 95% CI for the weighted average estimated, back-transformed from the logit scale, is [0.420134, 0.714973].

Now, let's see if we can derive the reported 95% CI for the model-averaged estimate 'by hand'. We'll demonstrate 2 different approaches: one based on the *Delta method*, and one based on a straight application of the concepts of 'model averaging' introduced in the preceding.

We'll start with the approach based on the Delta method, since this approach is in fact the one used by **MARK**. In short, what we want to do is 'take the model-averaged estimate and SE, transform them onto the logit scale, calculate the 95% limits with $\pm 1.96 \times$ logit(SE), then back-transform the 95% limits from the logit scale $\rightarrow$ the real probability scale.

While this sounds easy, it is important to recall (from Chapter 1) that the variance for a parameter can be estimated from the likelihood based on the rate of change in the likelihood at the MLE for that parameter (i.e., the second derivative of the likelihood evaluated at the MLE). As such, you can't simply back-transform from the SE on the logit scale to the probability scale, since the different scalings influence the shape of the likelihood surface, and thus the estimate of the SE.

To get around this problem, we make use of the Delta method. The Delta method is particularly handy for approximating the variance of transformed variables (and clearly, that is what we are dealing with here). The details underlying the Delta method are beyond our scope at this point (the Delta method is treated more fully in Appendix B); here we simply demonstrate the application for the purpose of estimating the variance of the back-transformed parameter.

For example, suppose one has an MLE $\hat{\gamma}$ and an estimate of var($\hat{\gamma}$), but makes the transformation,

$$\hat{\theta} = e^{\hat{\gamma}^2}.$$

Then, using the Delta method, we can write

$$\widehat{\text{var}}(\hat{\theta}) \approx \left(\frac{\partial \hat{\theta}}{\partial \hat{\gamma}}\right)^2 \times \widehat{\text{var}}(\hat{\gamma}).$$

So, all we need to do is differentiate the transformation function for θ with respect to γ, which yields $2\gamma.e^{\gamma^2}$. We would simply substitute this derivative into our expression for the variance, yielding

$$\widehat{\text{var}}(\hat{\theta}) \approx \left(2\hat{\gamma}.e^{\hat{\gamma}^2}\right)^2 \times \widehat{\text{var}}(\hat{\gamma}).$$

Given values for $\hat{\gamma}$, and $\widehat{\text{var}}(\hat{\gamma})$, you could easily derive the estimate for $\widehat{\text{var}}(\hat{\theta})$.

What about the logit transform? Actually, it's no more difficult, although the derivative is a bit 'uglier'. Since the logit transformation is given as

$$\hat{\varphi} = \frac{e^{\hat{\beta}}}{1 + e^{\hat{\beta}}},$$

then

$$\begin{aligned}
\widehat{\text{var}}(\hat{\varphi}) &\approx \left(\frac{\partial \hat{\varphi}}{\partial \hat{\beta}}\right)^2 \times \widehat{\text{var}}(\hat{\beta}) \\
&= \left(\frac{e^{\hat{\beta}}}{1 + e^{\hat{\beta}}} - \frac{\left(e^{\hat{\beta}}\right)^2}{1 + \left(e^{\hat{\beta}}\right)^2}\right)^2 \times \widehat{\text{var}}(\hat{\beta}) \\
&= \left(\frac{e^{\hat{\beta}}}{\left(1 + e^{\hat{\beta}}\right)^2}\right)^2 \times \widehat{\text{var}}(\hat{\beta}).
\end{aligned}$$

Now, to the task at hand. First, we need the model-averaged estimate of $\bar{\hat{\varphi}}_{3,p}$, which from the **MARK** output (at the bottom of the preceding page), is reported as 0.5741287, with an unconditional SE of 0.0774609. So, we first transform the model-averaged estimate onto the logit scale: $\ln(\theta/(1-\theta)) = \ln(0.5741287)/(1-0.5741287) = 0.2987164$.

The next step is to take the estimate of the unconditional SE, square it to get the variance (in other words, the estimated variance is $0.0774609^2 = 0.00600019$), and then use the Delta method to approximate the variance on the logit scale. So, we're actually going in the opposite direction to the preceding demonstration of the Delta method (where the transformation was from the logit → real scale; here we want to go from real → logit).

Since the logit transform is $\beta = \ln(\theta/(1-\theta))$, then application of the Delta method yields

$$\begin{aligned}\widehat{\mathrm{var}}(\hat{\beta}) &\approx \left(\frac{\partial\hat{\beta}}{\partial\bar{\hat{\varphi}}}\right)^2 \times \widehat{\mathrm{var}}(\bar{\hat{\varphi}}) \\ &= \frac{(1-\varphi)^2\left(\frac{\varphi}{(1-\varphi)^2}+\frac{1}{1-\varphi}\right)^2}{\varphi^2} \times \widehat{\mathrm{var}}(\hat{\varphi}) \\ &= \frac{1}{(\varphi-1)^2\varphi^2} \times \widehat{\mathrm{var}}(\hat{\varphi}).\end{aligned}$$

So, substituting in our estimates for $\bar{\hat{\varphi}}_{3,p}$ and $\widehat{\mathrm{var}} = 0.0060019$, the Delta method approximation to the variance on the logit scale is

$$\begin{aligned}\widehat{\mathrm{var}}(\hat{\beta}) &\approx \frac{1}{(\varphi-1)^2\varphi^2} \times \widehat{\mathrm{var}}(\hat{\varphi}) \\ &= 0.10036674,\end{aligned}$$

and thus, the SE on the logit scale is $\sqrt{0.10036674} = 0.316807101$.

All we need to do now is derive the 95% confidence limits on the logit scale: $0.2987164 \pm (1.96 \times 0.31687101) = [-0.32222552, 0.919658318]$.

Final step – simply back-transform these 95% from the logit scale → real probability scale. So,

$$\frac{e^{-0.32222552}}{1+e^{-0.32222552}} = 0.42013347, \quad \text{and} \quad \frac{e^{0.919658318}}{1+e^{0.919658318}} = 0.71497248.$$

Thus, the back-transformed 95% CI is [0.42013347, 0.71497248], which is what is reported by **MARK** (couple of pages back), to within rounding error. As noted earlier, this is the approach that **MARK** uses, regardless of the actual link function used when fitting the models to the data – which is why the 95% CI reported by **MARK** is always labeled as '**`95% CI for Wgt. Ave. Est. (logit trans.)`**'.

Now, the second approach to deriving the 95% CI which we'll demonstrate uses a direct application of 'model averaging' as introduced in this section. To do this, we need to first have a look at the β estimates (which were estimated on the logit scale), and associated estimates of the SE (and thus, variances) of those estimates, reported by **MARK**. The estimates are shown in the following table:

model	$\hat{\beta}$	$\widehat{\mathrm{SE}}$	$\widehat{\mathrm{var}}$
$\{\varphi_g\, p_t\}$	0.3107252	0.3161188	0.0999311
$\{\varphi_g\, p_.\}$	0.2221574	0.3109805	0.0967089

From the preceding, we calculate the estimated *unconditional* variance of $\hat{\theta}$ – on the *logit* scale! – as

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

and where the w_i are the Akaike weights (Δ_i) scaled to sum to 1.

For our two candidate models, the model averaged estimate is

$$\begin{aligned} \bar{\hat{\beta}} &= (0.86520 \times 0.3107184) + (0.13480 \times 0.2221763) \\ &= 0.2987863. \end{aligned}$$

Thus, for our two candidate models,

$$\begin{aligned} \widehat{\text{var}}(\bar{\hat{\varphi}}_{3,p}) &= \sum_{i=1}^{R} w_i \left[\widehat{\text{var}}(\hat{\theta}_i \mid \mathcal{M}_i) + (\hat{\theta}_i - \bar{\hat{\theta}})^2 \right] \\ &= 0.86520 \left[0.0999311 + (0.3107184 - 0.2987829)^2 \right] \\ &\quad + 0.13480 \left[0.0967089 + (0.2221763 - 0.2987829)^2 \right] \\ &= 0.10041161. \end{aligned}$$

So the estimated variance – on the logit scale! – is 0.10041161, so the estimated SE is $\sqrt{0.10041161} = 0.31687791$. Thus, the estimated 95% CI – on the logit scale! – is $0.2987863 \pm (1.96 \times 0.31687791) = [-0.3222944, 0.9198670]$. Note the close similarity of this 95% CI and the one calculated above using the Delta method.

Now, the final step – back-transforming the CI from the logit scale → real probability scale. As will be covered in detail in Chapter 6, the back transform of $\hat{\theta}$ estimated on the logit scale to the real probability scale is

$$\frac{e^{\hat{\theta}}}{1 + e^{\hat{\theta}}}.$$

So,

$$\frac{e^{-0.3148382}}{1 + e^{-0.3148382}} = 0.4201167, \quad \text{and} \quad \frac{e^{0.912404}}{1 + e^{0.912404}} = 0.71501500.$$

Thus, the back-transformed 95% CI is [0.4201167, 0.71501500], which is what is reported by **MARK** (presented a few pages back), to within rounding error, and is thus also quite similar to the one calculated above using the Delta method.

A reasonable question at this point might be 'why doesn't **MARK** generate model-averaged estimates for the β estimates on the link scale?'. The answer is largely technical (and at some levels, philosophical), but the 'short-form' is because it is unclear how best to do this sort of averaging of β estimates in general (there are any number of technical issues: what is β if a particular term isn't included in a candidate model? Is the estimator of the unconditional variance robust on the transformed scale, which might be strongly non-linear? And so on.).

In fact, some of these issues may in part explain the slight discrepancy in the CI between the two approaches used above. While there is a 'lively' debate about the issue of 'model averaging β estimates', there is a fair consensus that you are 'safest' if you only model average the reconstituted 'real estimates', which is precisely what **MARK** does.

end sidebar

4.6. Significance?

OK, fine. What about 'significance'? We'd hazard a guess that at this point, some (many?) of you may be wondering – 'OK – we can select a model, and can talk about the relative support of this model versus another model, we can come up with good average values, but – based on structural differences in the models, can we say anything about the significance of one or more factors?'. Often, this is the question of primary importance to the analyst – is there a 'significant' sex difference? Do the colonies differ 'significantly'? Is there evidence for a 'significant' trend over time?

Clearly, any discussion of importance, or 'significance' (in a statistical or biological context) starts with calculating the magnitude of the 'effect' – the difference in some parameter between 2 groups, or time intervals, or some other axes over which you want to characterize differences in the parameter. The question we face is 'is the difference as estimated a 'significant' difference'? Note that for the moment we've repeatedly referred to 'significance' parenthetically, since it might mean 'biological significance', or 'statistical significance', or both. It is critical to think critically about which context is appropriate.

For example, if we simply look at the estimates for our most parsimonious model from our analysis of the swift data, we see that the estimated survival for the 'poor' colony is 0.577, and for the good colony is 0.770 – a difference of 20%. If the survival probabilities for both colonies were in fact constant over time, then we can estimate lifespan as $(1/-\ln(S))$. Thus, estimated lifespan in the good colony is 3.83 years, while in the poor colony, estimated lifespan is 1.82 years, less than 50% of the estimate for the good colony. Whether or not *x.xx* years (or whatever time unit is appropriate for the organism in question) is biologically significant is entirely dependent on the biology of the organism. Since this example is dealing with birds, you can rest assured that a 50% difference in expected lifespan is likely to be highly significant in the biological sense. But, this is where the biologist must use his/her judgment as to what is (or is not) a *biologically* meaningful difference. This is generally not the same as a statistically significant difference.

Since the effect size is 'estimated', it will have an associated uncertainty which we can specify in terms of a confidence interval (CI) (the theory and mechanics of the estimation of the effect size, and the SE for the effect, are covered in Chapter 6). The question then becomes – what are the plausible bounds on the true effect size, and are biologically important effect sizes contained within these bounds? Suppose we consider a relative difference in survival of 15% or greater to be biologically important.

Suppose the estimated effect size for the difference in survival between the colonies was 19.3%, with a CI of 1.7%-36.9% As such, we might consider the results as *statistically* 'significant', since the CI doesn't include 0, but *biologically* inconclusive, because the CI includes values *below* 15%.

It is just these sorts of questions of 'biological subjectivity' which undoubtedly contributed to the popularity of focussing on 'statistical significance', since it gives the appearance of being 'objective'. Putting aside the philosophical debate as to which type of 'significance' is more important, we'll introduce the basic mechanics for classical significance testing (in the statistical sense) as implemented in **MARK**.

4.6.1. Classical significance testing in MARK

The classical 'statistical' approach focusses on assessing the 'significance' of one or more factors on variation in a particular parameter of interest. You may recall from Chapter 1 that we can use the properties of ML estimates as the basis for a number of different 'statistical tests' (Wald, Fisher's score...) to compare the relative fit of different competing models to the data.

One such test (the *likelihood ratio test*; LRT) is available in **MARK**. To apply an LRT, you take some likelihood function $\mathcal{L}(\theta_1, \theta_2, \ldots, \theta_n)$, and derive the maximum likelihood values for the various

parameters θ_i. Call this likelihood $\mathcal{L}_f$. Then, fix some of the parameters to specific values, and maximize the likelihood with respect to the remaining parameters. Call this 'restricted' likelihood $\mathcal{L}_r$. The likelihood ratio test says that the distribution of twice the negative log of the likelihood ratio, i.e., $-2\ln(\mathcal{L}_r/\mathcal{L}_f)$ is approximately χ^2 distributed with r degrees of freedom (where r is the number of restricted parameters). The LRT compares a restricted model which is '*nested*' within the full model (i.e., as is generally true with 'classical hypothesis testing', the LRT compares a pair of models).

begin sidebar

a (slightly) more technical derivation of the LRT

Consider some likelihood maximized for some true parameter value θ. If we write a Taylor expansion around this value as

$$\mathcal{L}(\theta) = \mathcal{L}(\hat{\theta}) + \left(\frac{\partial \mathcal{L}}{\partial \theta}\right)_{\theta=\hat{\theta}} (\theta - \hat{\theta}) + \frac{1}{2}\left(\frac{\partial^2 \mathcal{L}}{\partial \theta^2}\right)_{\theta=\hat{\theta}} (\theta - \hat{\theta})^2 + O(\theta - \hat{\theta})^2.$$

By definition,

$$\left(\frac{\partial \mathcal{L}}{\partial \theta}\right)_{\theta=\hat{\theta}} = 0.$$

So, we can express the Taylor expansion as the difference between the true parameter and the estimated parameter, as follows:

$$\mathcal{L}(\theta) - \mathcal{L}(\hat{\theta}) = \frac{1}{2}\left(\frac{\partial^2 \mathcal{L}}{\partial \theta^2}\right)_{\theta=\hat{\theta}} (\theta - \hat{\theta})^2 + O(\theta - \hat{\theta})^2$$

$$-2\left[\mathcal{L}(\theta) - \mathcal{L}(\hat{\theta})\right] = \left(-\frac{\partial^2 \mathcal{L}}{\partial \theta^2}\right)_{\theta=\hat{\theta}} (\theta - \hat{\theta})^2 + O(\theta - \hat{\theta})^2.$$

Dropping off the residual term,

$$-2\left[\mathcal{L}(\theta) - \mathcal{L}(\hat{\theta})\right] \cong \left(-\frac{\partial^2 \mathcal{L}}{\partial \theta^2}\right)_{\theta=\hat{\theta}} (\theta - \hat{\theta})^2.$$

Now, consider a simple binomial process, where we aim to estimate the parameter $\hat{p}$. Then, we can write

$$-2\left[\mathcal{L}(p) - \mathcal{L}(\hat{p})\right] \cong \left(-\frac{\partial^2 \mathcal{L}}{\partial p^2}\right)_{p=\hat{p}} (p - \hat{p})^2.$$

Now, recall from Chapter 1 (and basic common sense) that the MLE for $\hat{p}$ is (n/N) (say, n successes in N trials). Also recall that $-\partial^2\mathcal{L}/\partial p^2$ is an estimate of the variance of $\hat{p}$.
Then,

$$\begin{aligned} -2\left[\mathcal{L}(p) - \mathcal{L}(\hat{p})\right] &\cong \left(-\frac{\partial^2 \mathcal{L}}{\partial p^2}\right)_{p=\hat{p}} (p - \hat{p})^2 \\ &= \frac{1}{\widehat{\mathrm{var}}(\hat{p})} \cdot \left(p - \frac{n}{N}\right)^2 = \frac{p - E(\hat{p})}{\widehat{\mathrm{var}}(\hat{p})^2}. \end{aligned}$$

Now, some classical results show that as $N \rightarrow \infty$, then

$$\hat{p} \rightarrow N\left(E(\hat{p}), \sqrt{\text{var}(\hat{p})}\right) \quad \text{and} \quad \frac{\left(p - E(\hat{p})\right)^2}{\widehat{\text{var}}(\hat{p})} \rightarrow N(0,1)^2 = \chi^2_1.$$

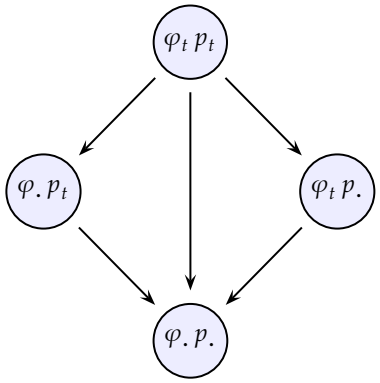
In other words, the parameter estimate is asymptotically normal convergent as sample size increases, and (more to the point here), that $-2(\mathcal{L}(p) - \mathcal{L}(\hat{p}))$ (i.e., the deviance) is χ^2 distributed.

This is a very convenient result – it says that as the sample size n approaches ∞, the test statistic $-2\ln(\Lambda)$ will be asymptotically χ^2 distributed with degrees of freedom equal to the difference in dimensionality (number of parameters) of the two models being compared. This means that for a great variety of hypotheses, a practitioner can take the likelihood ratio Λ, algebraically manipulate Λ into $-2\ln(\Lambda)$, compare the value of $-2\ln(\Lambda)$ given a particular result to the χ^2 value corresponding to a desired statistical significance, and create a reasonable decision based on that comparison.

end sidebar

In practical terms, the first step in using the LRT with models fit using **MARK** is to determine which models are nested. While this is not always as straightforward as it seems (see the `-sidebar-` a few pages ahead), it is relatively straightforward for our present example.

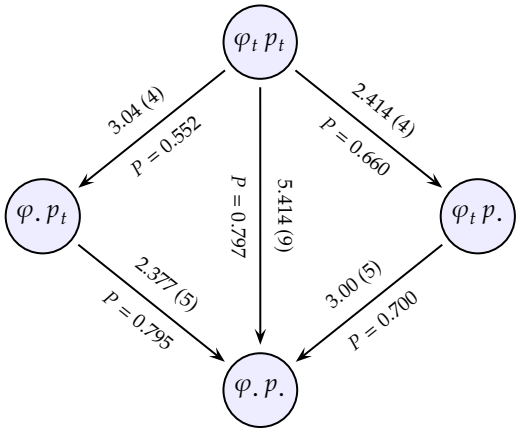
Consider the figure shown below, which represents the hierarchical relationship of the 4 models we fit to the male European dipper data. In this figure, 'nested' models are connected by the arrows. The direction of the arrows leads from a given model to the model 'nested' within it. Any two 'nested' models can be compared statistically using a likelihood ratio test. Provided that the reduced (less parameterized) model is satisfactory, the difference in deviances between two nested models is distributed as a χ^2 statistic with n degrees of freedom, where n is the difference in the number of parameters between the two models.



Now, we re-draw this figure (top of the next page), annotating it with the differences in deviance among 'nested' models, and the difference in the number of parameters, we obtained from our European dipper analysis.

The 'significance' of these differences (in the traditional sense) can be estimated from any standard χ^2 table, or directly using **MARK**. Recall from Chapter 3 that in **MARK** you can request a likelihood ratio test (LRT) between any two models, using the '**Tests**' menu. However, **MARK** doesn't 'know this', and performs LRT for **all** models with unequal numbers of parameters, and outputs results from all of these comparisons.

Clearly, the 'unequal number of parameters' criterion is not a particularly good one, so you'll need to pay attention. A significant difference between models means two things: (1) that there is a significant



increase in deviance with the reduction in the number of parameters, such that the reduced model fits significantly less well, and (2) the parameter(s) involved contribute significantly to variation in the data.

As we can see from the figure, there is no significant difference in model fit (difference in deviance) between the most parameterized model (the CJS model $\varphi_t\, p_t$) and any of the 3 other models. Thus, any of the 3 other models would be a 'better model' than the CJS model, since they fit the data equally well (statistically), but require fewer parameters to do so (i.e., are more parsimonious).

From the preceding figure and from the AIC$_c$ values tabulated in the results browser (shown at the top of the next page) we see that the most parsimonious model overall is model $\{\varphi_\cdot\, p_\cdot\}$ – i.e., the model where both survival and recapture probability are constant over years.

Results Browser: Live Recaptures (CJS)

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{phi(.)p(.)}	322.5527	0.0000	0.96003	1.0000	2	41.8147
{phi(t)p(.)}	330.0567	7.5040	0.02253	0.0235	7	38.8147
{phi(.)p(t)}	330.6794	8.1267	0.01650	0.0172	7	39.4374
{phi(t)p(t)}	336.4343	13.8816	0.00093	0.0010	11	36.4013

However, before we go any further, what hypotheses have we just tested? Consider the test of the CJS model $\{\varphi_t\, p_t\}$ versus $\{\varphi_\cdot\, p_t\}$. In this comparison we are testing the following 'hypothesis': that there is significant variation in survival over time. We are comparing the fit of a model where survival is allowed to vary over time $\{\varphi_t\, p_t\}$ to one where it doesn't $\{\varphi_\cdot\, p_t\}$. Since the fit of these two models is not 'significantly' different in the classical statistical sense ($\chi^2 = 3.05, df = 4, P = 0.552$), we might state that 'there is no evidence at a nominal $\alpha = 0.05$ level of significant annual variation in survival'.

begin sidebar

Which models are nested?

While the preceding example was simple enough, determining which models are nested is not always trivial. Here, we take a slightly more extended look at 'nestedness' and linear models.

Let's begin with addressing the question of why aren't models $\{\varphi_\cdot\, p_t\}$ and $\{\varphi_t\, p_\cdot\}$ in the preceding example nested? The easiest way to resolve which models of the models in this example are nested, and which aren't, is to try to answer the following question: 'would starting model **A** be equivalent to

reduced model **B** if you eliminated one or more of the factors from model **A**?' If so, then model **B** is 'nested' within model **A**.

For example, if we start with model $\{\varphi_t\ p_t\}$ (model **A**), we want to know if model $\{\varphi_t\ p_.\}$ (model **B**) is nested within it. So, what happens if you 'remove one or more of the factors from model **A**'? Well, in this case we see that if we eliminate 'time' from capture in model **A**, then model **A** is transformed into model **B**. Thus, we can say that model **B** $\{\varphi_t\ p_.\}$ is nested within model **A** $\{\varphi_t\ p_t\}$.

However, now compare models $\{\varphi_.\ p_t\}$ and $\{\varphi_t\ p_.\}$. If we consider these models as **A** and **B** respectively, we see that there is no simple transformation of model **A** into model **B**; we would have to drop the time-dependence from the recapture model, and add time to the survival model, to make models **A** and **B** equivalent. Since nesting requires only addition or subtraction of parameters (but not both), then these models are clearly not nested.

But, these examples are very simple. What about situations where 'nestedness' is not so obvious. For example, are the models $\{Y = x\}$ and $\{Y = \ln(x)\}$ nested? Clearly, we need a more general set of rules. Let's start by considering models which *are* nested.

> **nested models**: *Two models are nested if one model can be reduced to the other model by imposing a set of linear restrictions on the vector of parameters.*

For example, consider models f and g, which we'll assume have the same functional form and error structure. For convenience, we'll express the data as deviations from their means (doing so eliminates the intercept from the linear model, since it would be estimated to be 0). These two models differ then only in terms of their regressors.

In the following

$$f:\ Y = \beta_1 x_1 + \epsilon_0$$
$$g:\ Y = \beta_1 x_1 + \beta_2 x_2 + \epsilon_1,$$

the model f is nested within model g because by imposing the linear restriction that $\beta_2 = 0$, model g becomes model f.

What about non-nested models? Things get a bit more complex here, but we'll operationally define non-nested models as

> **non-nested models**: *Two models are non-nested, either partially or strictly* (discussed below), *if one model cannot be reduced to the other model by imposing a set of linear restrictions on the vector of parameters*

Examples of non-nested models include (but are not limited to):

- **No linear restriction possible to reduce on model to another**

  Consider the following two approximating models:

$$f:\ Y = \beta_1 x_1 + \beta_2 x_2 + \epsilon_0$$
$$g:\ Y = \beta_2 x_2 + \beta_3 x_3 + \epsilon_1.$$

  Models f and g are non-nested because even if we impose the restriction on model g that $\beta_3 = 0$, model g does not become model f.

  In fact, in this example, models f and g are *partially non-nested*, because they have one variable in common (x_2). If the two models didn't share x_2, then they would be *strictly non-nested*.

However, you need to be somewhat careful in defining models as strictly non-nested. There are, in fact, two cases where models with different sets of regressors may not be strictly non-nested.

Consider the following two models:

$$f: \; Y = \beta_1 x_1 + \epsilon_0$$
$$g: \; Y = \beta_2 x_2 + \epsilon_1.$$

If either β_1 or β_2 equals zero, then the models are nested. This is trivially true. Less obvious, perhaps, is the situation where one of more of the explanatory variables in one model may be written as a linear combination of the explanatory variables in the second model.

For example, given the two models

$$f: \; Y = \beta_1 x_1 + \epsilon_0$$
$$g: \; Y = \beta_2 x_2 + \epsilon_1,$$

consider a third model h where

$$h: \; Y = \beta_3 x_3 + \epsilon_2 = \beta_1 x_1 + \beta_2 x_2 + \epsilon_2.$$

Then, perform the following hypothesis tests: model h versus model f (i.e., $\beta_2 = 0$ versus $\beta_2 \neq 0$), and model h versus model g (i.e., $\beta_1 = 0$ versus $\beta_1 \neq 0$).

- **Different functional forms used in two models**

The following are clearly different function forms

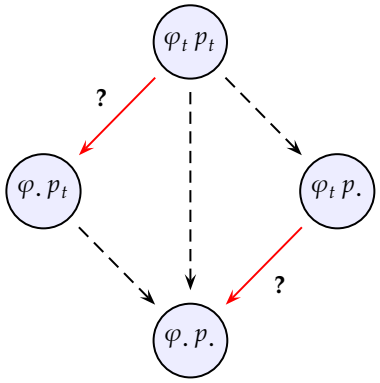
$$f: Y = X\beta + \epsilon$$
$$g: \ln(Y) = \ln(X)\gamma + \mu.$$

end sidebar

4.6.2. Some problems with the classical approach...

Seems straightforward, right? There are at least two 'mechanical' problems with this approach. First, the LRT is only strictly appropriate when the models being compared are nested. For non-nested models, you can use either the AIC, or approaches based on a resampling ('bootstrapping') approach.

Second, looking again at the hierarchial diagram (below), you should notice that there are in fact 2 different pairs of nested models (joined by red arrows) we could compare (using an LRT) to test for annual variation in survival: $\{\varphi_t \, p_t\}$ versus $\{\varphi_\cdot \, p_t\}$, or model $\{\varphi_t \, p_\cdot\}$ versus $\{\varphi_\cdot \, p_\cdot\}$.



In both cases, we are testing for significant annual variation in survival (i.e., φ_t vs $\varphi_.$). Do the two different sets of model comparisons give the same results? Do both tests lead to the same conclusion about annual variation in apparent survival, φ?

We'll briefly explore this question using live encounter data contained in the file **`LRT-demo.inp`** – 5 sampling occasions.

Here are the results of fitting models $\{\varphi_t\, p_t\}$, $\{\varphi_t\, p_.\}$, $\{\varphi_.\, p_t\}$ and $\{\varphi_.\, p_.\}$:

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance	-2Log(L)
{Phi(.) p(t) PIM}	849.3549	0.0000	0.52042	1.0000	5	13.3161	839.2382
{Phi(t) p(.) PIM}	850.5384	1.1835	0.28798	0.5534	5	14.4996	840.4217
{Phi(t) p(t) PIM}	852.2986	2.9437	0.11944	0.2295	7	12.1577	838.0798
{Phi(.) p(.) PIM}	853.3062	3.9513	0.07217	0.1387	2	23.3609	849.2830

Our interest lies in comparison of model $\{\varphi_t\, p_t\}$ with model $\{\varphi_.\, p_t\}$, versus a comparison of model $\{\varphi_t\, p_.\}$ with model $\{\varphi_.\, p_.\}$. Using '**`Tests | LR Tests`**', we generate the following test results:

Reduced Model	General Model	Chi-sq.	df	Prob.
{Phi(.) p(t) PIM}	{Phi(t) p(.) PIM}	-1.184	0	******
{Phi(.) p(t) PIM}	{Phi(t) p(t) PIM}	1.158	2	0.5604
{Phi(.) p(.) PIM}	{Phi(.) p(t) PIM}	10.045	3	0.0182
{Phi(t) p(.) PIM}	{Phi(t) p(t) PIM}	2.342	2	0.3101
{Phi(.) p(.) PIM}	{Phi(t) p(.) PIM}	8.861	3	0.0312
{Phi(.) p(.) PIM}	{Phi(t) p(t) PIM}	11.203	5	0.0475

The model comparisons of interest are shaded in 'orange'. For the comparison of the general model $\{\varphi_t\, p_t\}$ with nested (reduced) model $\{\varphi_.\, p_t\}$, the LRT yields a $\chi^2_2 = 1.158$, which is not close to significant at the nominal $\alpha = 0.05$ level ($P = 0.5604$). versus a comparison of model $\{\varphi_t\, p_.\}$ with model $\{\varphi_.\, p_.\}$. Taken alone, this would suggest no strong evidence for annual variation in apparent survival.

However, now consider the comparison of the general model $\{\varphi_t\, p_.\}$ with nested (reduced) model $\{\varphi_.\, p_.\}$. Here, the LRT yields a $\chi^2_3 = 8.861$, which is in fact significant at the nominal $\alpha = 0.05$ level ($P = 0.0312$). So, this comparison would suggest that there *is* evidence for annual variation in apparent survival, opposite to what we concluded in the first analysis!

The observation that the different model comparisons yielded very different results, and thus different conclusions, is clearly a problem. In fact, for any typical candidate model set, there are likely to be > 1 pairs of nested models which could be compared using a LRT to test the same hypothesis. And there is a fair likelihood that in many cases, these LR tests will yield different, often contradictory results (as in our present example).

Thus, the obvious question is: which of the possible nested comparisons for a given hypothesis is the 'right one' to use? A commonly suggested approach (which is not without its critics) is to start from the most parsimonious acceptable model still containing the effect you want to test, and then use the LRT to test the nested model without this factor. You can use AIC (or sequential LRT tests where appropriate) to identify the model which has the fewest parameters while still fitting the data and containing the factor you are interested in. The advantage of using this model is that tests are generally most powerful in a 'parsimonious context'. In our example, model $\{\varphi_t\, p_.\}$ is more parsimonious than model $\{\varphi_t\, p_t\}$, and thus we might conclude that the LRT between model $\{\varphi_t\, p_.\}$ and nested (reduced) model $\{\varphi_.\, p_.\}$ is the more powerful of the two comparisons, and thus, supporting a conclusion of 'significant' annual variation in apparent survival.

4.6.3. 'Significance' of a factor using AIC

Despite several difficulties with the classical approach to testing for 'statistical significance', there would seem to be one singular advantage relative to multimodel inference based on an information theoretic index like the AIC or BIC. Namely, that there is a relatively straightforward way (caveats notwithstanding) to give some sort of statement about the 'significance' (importance) of some factor(s). Is there something equivalent that can be done with the information theoretic approach?

Burnham & Anderson have noted that assessment of the relative importance of variables has often been based only on the best model (e.g., often selected using a stepwise testing procedure of some sort). Variables in that best model are considered 'important', while excluded variables are considered 'not important'. They suggest that this is too simplistic. Importance of a variable can be refined by making inference from all the models in the candidate set. Akaike weights are summed for all models containing predictor variable (i.e., factor) x_j, $j = 1, \ldots, R$. Denote these sums as $w_{+(j)}$. The predictor variable with the largest predictor weight, $w_{+(j)}$, is estimated to be the most important, while the variable with the smallest sum is estimated to be the least important predictor.

Can we be somewhat more 'formal' about this? Consider the following example – a simple linear model with three regressors, x_1, x_2 and x_3. The objective was to examine the eight possible models consisting of various combinations of these regressors.

The following tabulates each of the possible models, along with hypothetical AIC weights (in the table, a '1' indicates that x_i is in the model; otherwise, it is excluded).

x_1	x_2	x_3	w_i
0	0	0	0.00
1	0	0	0.10
0	1	0	0.01
0	0	1	0.05
1	1	0	0.04
1	0	1	0.50
0	1	1	0.15
1	1	1	0.15

The selected best model has a weight of only 0.5 (suggesting strong model selection uncertainty). However, the sum of the weights for variable x_1 across all models containing x_1 is 0.79. This is evidence of the importance of this variable, across all eight of the models considered. Variable x_2 was not included in the selected best model, but this does not mean that it is of no importance (which might be the conclusion if you made inference only on the K-L best model). Actually, its relative weight of evidence support is 0.35. Finally, the sum of AIC weights for x_3 is 0.85.

Thus, the evidence for the importance of variable x_3 is substantially more than just the weight of evidence for the best model. We can order the three predictor variables in this example be their estimated importance: x_3, x_1, x_2 with importance weights of 0.85, 0.79, and 0.35, respectively. This basic idea extends to subsets of variables. For example, we can judge the importance of a pair of variables, *as a pair*, by the sum of the AIC weights of all the models that include that pair of variables. Similar procedures apply when assessing the relative importance of interaction terms.

To demonstrate this in application to a mark-recapture analysis, consider the results from the swift analysis (shown at the top of the next page). We notice that the two most parsimonious models in the candidate model set have colony differences in the survival parameters – model $\{\varphi_c\, p_t\}$ and model

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{phi(c)p(t)}	369.8080	0.0000	0.85650	1.0000	9	111.6644
{phi(c)p(.)}	373.5263	3.7183	0.13345	0.1558	3	128.1990
{phi(t)p(.)}	379.0123	9.2043	0.00859	0.0100	8	123.0601
{phi(t)p(t)}	382.8807	13.0727	0.00124	0.0014	13	115.7384
{phi(c*t)p(.)}	386.4962	16.6882	0.00020	0.0002	15	114.7094
{phi(c*t)p(t)}	391.4103	21.6023	0.00002	0.0000	20	107.5633

$\{\varphi_c\ p_.\}$ have virtually all of the support in the data. Moreover, the top two models, comprising $(0.857 + 0.133) = 99\%$ of the support in data, both have φ_c in common for the survival parameter. Meaning, only models with a colony effect on the apparent survival rate have any appreciable support in the data.

At this point, we might conclude that there is considerable evidence of a difference in survival between the 2 colonies. What about using cumulative support over all models in the model set? The summed AIC weights for `colony`, `time`, and `colony.time` for survival are: 0.9896, 0.0098, and 0.0007, respectively. Clearly, there is very strong support for a `colony` effect.

As suggested by Burnham & Anderson (2002, 2004), summing support over models is regarded as superior to making inferences concerning the relative importance of variables based only on the best model. This is particularly important when the second or third best model is nearly as well supported as the best model or when all models have nearly equal support. While this approach seems to have some merit, there is by no means consensus that this is the 'best approach', or that it 'works' in all cases – see for example Murray & Conner (2009).

Further, there are 'design' considerations about the set of models to consider when a goal is assessing variable importance. For example, it seems to be particularly important is that the model set be 'symmetrical' or 'balanced' with respect to each factor of interest (i.e., that the model set has roughly the same number of models with, and without a particular factor). This is not always easy to accomplish (e.g., how do you balance models for interaction terms?). See Doherty, White & Burnham (2010), and Arnold (2010) for a discussion of these issues. The following `-sidebar-` is also relevant.

begin sidebar

The 'masquerading variable' issue...

Anderson (2007) and Arnold (2010) both discuss the 'pretending' or 'masquerading' variables problem where a variable with little or no information is included in the minimum AIC_c model to produce a model relatively close in weight to the minimum AIC_c model. Because the new model with the unimportant variable is close to the top model, the natural conclusion is that the additional variable is important. However, as Anderson (2007) and Arnold (2010) both discuss, such is not the case. The additional model has valid weight, but only because the model is identical to the top model with only the additional variable added. The variable is 'riding on the coat tails' of the minimum AIC_c model.

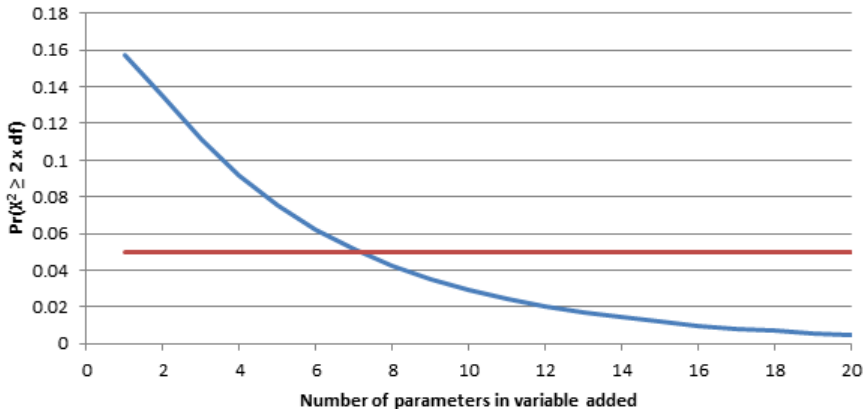
For the moment, just consider the theory for use with AIC (not AIC_c), and that the additional variable only adds 1 parameter to the new model. Likelihood theory shows that the $-2\ln(\mathcal{L})$ for the new model can never be smaller than the top AIC model, i.e., the addition of another variable can only improve the $-2\ln(\mathcal{L})$. Suppose that the new model has exactly the same $-2\ln(\mathcal{L})$, so that the AIC for the new model is exactly 2 units larger than the minimum AIC model because of the additional parameter (i.e., the additional variable increases K by 1). The AIC value for the new model can never be lower than 2 AIC units (unless there is a numerical optimization issue, always something to be aware of).

However, likelihood theory also tells us just how much the $-2\ln(\mathcal{L})$ is expected to change, because the difference between the two $-2\ln(\mathcal{L})$ values is a likelihood ratio test and is distributed as a χ^2_1 distribution, i.e., a χ^2 distribution with 1 degree of freedom. That is, a likelihood ratio test of the null

hypothesis of no effect for a single degree of freedom is distributed as χ^2_1 under the null hypothesis. Because the expected value (mean) of χ^2_1 is just the degrees of freedom, the mean difference in the $-2\ln(\mathcal{L})$ values is 1 under the null. Further, the probability that the difference in the two $-2\ln(\mathcal{L})$ values is > 2 (i.e., the addition of the covariate now generates a model with smaller AIC than the original model) is $\Pr(\chi^2_1 \geq 2) = 0.157299$.

Now extend this thinking to include variables with ≥ 1 parameter, e.g., time effect or a categorical variable requiring multiple parameters to model it. Again, the null distribution of the likelihood ratio test is just distributed as χ^2_{df} where df is the difference in the number of parameters of the 2 models (i.e., the number of parameters required to model the covariate).

Consider the curve $\Pr(\chi^2_{df} \geq (2 \times df))$ = the P-value of the likelihood ratio test when the AIC values of the 2 models are equal. The following graph shows what this curve looks like (blue line) for df up to 20 parameters in the added variable:



One important observation from this graph is that a variable with > 8 df is below the usual $\alpha = 0.05$ level shown with the orange flat line. What this means is that a likelihood ratio test of the variable would reject the null hypothesis, yet AIC would still select the model without the variable. The difference in conclusions between a likelihood ratio test and AIC model selection only increases as the df increases beyond 8. AIC will select the model without the variable even though the variable is statistically significant ($P < 0.05$) in a likelihood ratio test. In contrast, to the left of 8 parameters, AIC is selecting the covariate model even though $P > 0.05$.

Now return to using AIC_c instead of AIC. The AIC_c value is always larger than the AIC value, but the same impacts still occur. A lower bound of 2 for a single df variable is no longer fixed, but gets larger as the sample size gets smaller. But your interpretation of the importance of the variable added must still reflect this coat-tail effect of the variable 'riding on the strength' of the minimum AIC_c model.

end sidebar

4.7. LRT or AIC, or something else?

At this point, you're probably mulling over a few things. First, we've covered a lot of ground – both technically, and conceptually. We've seen how to build and fit various models to our data using **MARK**. We've also introduced the important topic of 'model selection' – the use of LRT and AIC (or BIC), counting parameters, and 'hypothesis testing'. We've also considered the important idea of 'model averaging'. You're also probably thinking (hopefully) thats it's about time for us to make broad, categorical suggestions about what to do – AIC/BIC or LRT? Significance test, or effect size?

Alas, prepare to be disappointed. While we have our personal opinions, **MARK** itself is 'politically

neutral' on this one – you can choose to adopt an 'information theoretic' approach, or invoke a classic LRT approach (but **not** combinations of the two – you have to pick either of the 'two roads', and not mix and match the two), and talk about the significance of an effect based on a nominal *P*-value.

The whole issue of whether or not to use *P*-values, and 'classical hypothesis testing' is (and has been for some time) the subject of much debate in the literature. The fairly recent advent of methods for model selection based on information theory, and model averaging, has added some additional nuance to the discussion – for example, Stephens *et al*. (2005), Lukacs *et al*. (2005).

The literature related to 'model selection' and 'multi-model inference', is large, and expanding rapidly. As a starting point, start by having a careful read of the seminal text by Burnham & Anderson:

Model Selection and Multi-Model Inference (2nd Edition) – Ken Burnham and David Anderson. (2002) Spring-Verlag. 496 pages.

Additionally, Hooten & Hobbs (2015) have recently published a very thorough treatment of both Bayesian and 'frequentist' approaches to model selection (including the AIC – see also Hooten & Cooch (2019) for a less technical overview of some of the same material).

4.8. Summary

In this chapter, we looked at the basic mechanics (and some of the theory) of using **MARK** to construct and evaluate models fit to 'encounter data'. We've looked at the problem of staggered entry of marked individuals into the population (and how this leads logically to the triangular parameters structures – the PIMs). We've also considered (at an introductory level) the mechanics and theory of model selection: the LRT and the AIC. In the next chapter, we'll consider goodness of fit testing – an essential step in evaluating models.

4.9. References

Akaike, H. (1973) Information Theory and an Extension of the Maximum Likelihood Principle. In: B. N. Petrov and F. Csaki, eds. *Second International Symposium on Information Theory*. Budapest: Akademiai Kiado, pp. 267-281.

Anderson, D. R. (2007) *Model Based Inference in the Life Sciences: A Primer on Evidence*. Springer. 184 pages.

Arnold, T. W. (2010) Uninformative parameters and model selection using Akaike's Information Criterion. *Journal of Wildlife Management*, **74**, 1175-1178.

Buckland, S. T., Burnham, K. P., and Anderson, D. R. (1997) Model selection: an integral part of inference. *Biometrics*, **53**, 603-618.

Burnham, K. P., and Anderson, D. R. (2002) *Model Selection and Multi-Model Inference* (2nd Edition) Spring-Verlag. 496 pages.

Burnham, K. P., and Anderson, D. R. (2004) Multimodel inference - understanding AIC and BIC in model selection. *Sociological Methods & Research*, **33**, 261-304.

Burnham, K. P., Anderson, D. R., and Huyvaert, K. P. (2011) AIC model selection and multimodel inference in behavioral ecology: some background, observations, and comparisons. *Behavioral Ecology and Sociobiology*, **65**, 23-35.

Cooch, E. G., Pradel, R., and Nur, N. (1997) *A Practical Guide to Mark-Recapture Analysis using SURGE*. (Second edition). CEFE/CNRS, Montpelier, France.

Doherty, P. F., White, G. C., and Burnham, K. P. (2010) Comparison of model building and selection strategies. *Journal of Ornithology*, DOI 10.1007/s10336-010-0598-5

Dongarra, J. J., Bunch, J. R., Moler, C. B., and Stewart, G. W. (1979) LINPACK User's Guide. Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania.

Hooten, M. B., and Hobbs, N. T. (2015) A guide to Bayesian model selection for ecologists. *Ecological Monographs*, **85**, 3-28.

Hooten, M. B., and Cooch, E. G. (2019) Comparing ecological models. In *Quantitative Analyses for Wildlife Science* (L. A. Brennan, A. N. Tri, and B. G. Marcott, eds). Johns Hopkins University Press.

Hurvich, C. M., and Tsai, C. L. (1989) Regression and time series model selection in small samples. *Biometrika*, **76**, 297-307.

Lebreton, J.-D., Burnham, K. P., Clobert, J., and Anderson, D. R. (1992) Modeling survival and testing biological hypotheses using marked animals: a unified approach with case studies. *Ecological Monographs*, **62**, 67-118. doi:10.2307/2937171

Link, W. A., and Barker, R. J. (2006) Model weights and the foundations of multimodel inference. *Ecology*, **87**, 2626-2635.

Lukacs, P. M., Thompson, W. L., Kendall, W. L., Gould, W. R., Doherty, P. F., Burnham, K. P., and Anderson, D. R. (2007) Concerns regarding a call for pluralism of information theory and hypothesis testing. *Journal of Applied Ecology*, **44**, 456-460.

Murray, K., and Conner, M. M. (2009) Methods to quantify variable importance: implications for the analysis of noisy ecological data. *Ecology*, **90**, 348-355.

Stephens, P. A., Buskirk, S. W., Hayward, G. D., and Martinez del Rio, C. (2005) Information theory and hypothesis testing: a call for pluralism. *Journal of Applied Ecology*, **42**, 4-12.

Sugiura, N. (1978) Further analysis of the data by Akaike's information criterion and the finite corrections. *Communications in Statistics A*, **7**, 13-26.

Addendum – counting parameters

Although **MARK** does a good job of counting parameters, it is important that you understand how the model structure determines the number of parameters that are theoretically estimable. What **MARK** does is indicate (report) how many parameters are estimable, given the model, and the data. **MARK** does not indicate how many parameters are theoretically estimable, given the structure of the model. On occasion, there are discrepancies between the two.

There are 2 reasons why a particular model parameter might not be estimable. The first is because the parameter may be confounded with 1 or more other parameters in the model. An example is the last φ and p parameters in a time-specific Cormack-Jolly-Seber model, where only the product of φ and p can be estimated, but not the unique values of each. In this case, the parameters are not identifiable because of the *structure* of the model. This is referred to as *intrinsic non-identifiability*. The second situation arises either because the data are inadequate, or as an artifact of the parameter being 'poorly estimated' near either the 0 or 1 boundaries. This is referred to as *extrinsic non-identifiability*. If a parameter is extrinsically non-identifiable because of 'problems' with the data, then you may need to manually increase the number of estimated parameters **MARK** reports (given the data) to the number that *should* have been estimated (if there had been sufficient data). While there has been significant progress in formal 'analytical' analysis of intrinsic identifiability, these methods are complex, and do not apply generally to problems related to inadequate data or parameters estimated near the boundary. A numerical approach based on '*data cloning*' which can be used generally to help identify parameters that are not estimable is available in **MARK** is presented in Appendix F.

intrinsically non-identifiable parameters – an ad hoc *approach*

While there are methods (formal, numerical) for identifying intrinsically non-identifiable parameters, it is important that you develop an 'intuitive understanding' of the how such parameters arise in the first place. Let's assume that our data are 'good' – there are no 'structural problems' and that the only remaining task is to determine which parameters are separately identifiable. We'll concentrate on the 4 models we've examined in this chapter. We'll introduce an approach which is generally useful, if a bit cumbersome. In future chapters, where we explore significantly more complex models, we'll comment as needed on how the number of parameters was determined. Our most complex model in this chapter is the CJS model – complete time-dependence in both survival and recapture. In many ways, the most fundamental difficulty in counting parameters in general is nicely contained in this model, so it is a good starting point.

However, before we dive in, consider a much simpler situation. Consider the case of only 2 occasions, a release occasion, where newly marked individuals are released, and a single recapture occasion. This situation is common in short-term studies. In general, under this sampling scheme, what is done is to express the proportion of the individuals marked and released on the first occasion captured on the second occasion as a measure of the 'survival probability'. This fraction, also known as the 'return probability', is still widely found in the literature.

Unfortunately, naïve use of return probability poses real problems, since, in fact, it does not necessarily estimate survival probability at all. As noted in Lebreton *et al*. (1992), the number of individuals seen on the second occasion is the result of 2 events, not one; the frequency of individuals seen again on the second occasion is defined by the product of the number released on occasion 1 (R_1) times the probability of surviving to occasion 2 (φ_1), times the probability of being seen at occasion 2 given that it is in fact alive at occasion 2 (the recapture probability, p_2). Since the value of φ_1 and p_2 can vary between 0 and 1, the observed number of individuals at occasion 2 could reflect an infinite set of different combinations of either survival or recapture probability. For example, suppose 100 individuals

are marked and released at occasion 1, and 50 of these marked individuals are seen subsequently at occasion 2. The return probability is (50/100) or 0.5. However, does this really mean that 'survival' is 50%? Not necessarily. What it means is that $(100 \times \varphi_1 p_2) = 50$, or $(\varphi_1 p_2) = 0.5$. As you quickly see, there is an infinite set of combinations of φ_1 and p_2 which, when multiplied together, lead to the product 0.5. Thus, we can't necessarily say that 'survival' is 0.5, merely that the combined probability of surviving and being recaptured is 0.5. In other words, with only 2 occasions, the survival and recapture probabilities are not 'individually identifiable' – we cannot derive estimates for both parameters separately.

What do we need to do? Well, in order to separately derive estimates for these parameters, we need more information. We need at least one additional recapture occasion. The reason is fairly obvious if you look at the capture histories. As per Lebreton *et al*. (1992), let '1' represent when an individual is captured at a particular occasion, and '0' represent when it is not captured. With only 2 occasions and individuals released marked only on the first occasion, only 2 capture histories are possible: 10 and 11. As we just observed, with only two captures we can estimate only the product of survival and recapture. What about three occasions? As noted in Chapter 1, under this sampling scheme, at least 4 capture histories are possible for individuals marked on the first occasion:

encounter history	*probability*
111	$\varphi_1 p_2 \varphi_2 p_3$
110	$\varphi_1 p_2 (1 - \varphi_2 p_3)$
101	$\varphi_1 (1 - p_2) \varphi_2 p_3$
100	$1 - \varphi_1 p_2 - \varphi_1 (1 - p_2) \varphi_2 p_3$

The capture histories are given with the probability statements which, when multiplied by the number released at occasion 1, define the number of individuals with a given capture history expected at occasion 3. Concentrate for the moment on the third capture history in the table: '101'. You can see that there is a fundamental difference in this capture history from the one preceding it (where individuals are seen on each occasion). For capture history '101', individuals were released on occasion 1, not seen on occasion 2, but were seen again on occasion 3.

What does this sort of individual tell us? Well, clearly, if the individual was seen on occasion 3, then it must have been alive on occasion 2. The fact that we didn't see the individual at occasion 2 allows us to estimate the recapture probability, since recapture probability is merely the probability of seeing an animal at a particular occasion given that it is alive. Thus, because we have information from the third occasion, we can separately estimate the survival and recapture probabilities φ_1 and p_2 respectively.

Specifically,

$$\begin{aligned}
\frac{N_{111}}{N_{101}} &= \frac{\varphi_1 p_2 \varphi_2 p_3}{\varphi_1 (1 - p_2) \varphi_2 p_3} \\
&= \frac{\cancel{\varphi_1} p_2 \cancel{\varphi_2} \cancel{p_3}}{\cancel{\varphi_1} (1 - p_2) \cancel{\varphi_2} \cancel{p_3}} \\
&= \frac{p_2}{1 - p_2}.
\end{aligned}$$

Of course, **MARK** shields you from the complexities of the actual estimation itself, but in a very broad sense, it is the presence of '101' individuals along with the other capture histories that allows us to estimate survival and recapture rate separately.

But, it is important to note that we can't separately estimate **all** the parameters. Consider for instance φ_2 and p_3. Can we separate them? No! In fact, the product of these two parameters is completely analogous to a return probability between occasions 2 and 3. If we wanted to separate these 2 parameters, we'd need a fourth occasion, and so on.

Thus, in such a model where both survival and recapture probability are time-dependent, the terminal parameters are not individually identifiable – all we can do is estimate the product of the 2. Lebreton *et al*. (1992) refer to this product term as β_3.*

Thus, we can re-write our table, and the probability statements, as:

encounter history	*probability*
111	$\varphi_1 p_2 \beta_3$
110	$\varphi_1 p_2 (1 - \beta_3)$
101	$\varphi_1 (1 - p_2) \beta_3$
100	$1 - \varphi_1 p_2 - \varphi_1 (1 - p_2) \beta_3$

Now, we come to the original question: how many parameters do we have? In this case, with 3 occasions, and time-dependence in both survival and recapture, we have 3 estimable parameters: φ_1, p_2, and β_3. Do we always have a 'beta' parameter – a terminal product that cannot be separated into its component survival and recapture elements? The answer is, 'no'. Whether or not you have a 'beta' term depends upon the structure of your model.

We can demonstrate this by going back to the 4 models used in this chapter. We start with the fully time-dependent CJS model. From the preceding discussion, you might expect that there is likely to be a 'beta' term, since we have time-dependence for both parameters. Your intuition is correct. How can we count them? While there are a number of possible schemes you could use to count parameters (including rote memory of certain algebraic relationships between the number of time units and the number of parameters for a given type of model – see Table 7 in Lebreton *et al*. 1992), we prefer a more cumbersome, but fairly fool-proof way of counting them without resorting to memorization.

To use this approach, simply do the following. For a given model, write out all the saturated capture histories, and their associated probability statements, for each cohort. A 'saturated capture history' is the capture history where the individual was seen on each occasion following its release. In our European dipper example, there are 7 occasions, so our table of saturated capture histories, and substituting $\beta_7 = \varphi_6 p_7$, the associated probability statements, would look like the table shown below:

encounter history	*probability*
1111111	$\varphi_1 p_2 \varphi_2 p_3 \varphi_3 p_4 \varphi_4 p_5 \varphi_5 p_6 \beta_7$
0111111	$\varphi_2 p_3 \varphi_3 p_4 \varphi_4 p_5 \varphi_5 p_6 \beta_7$
0011111	$\varphi_3 p_4 \varphi_4 p_5 \varphi_5 p_6 \beta_7$
0001111	$\varphi_4 p_5 \varphi_5 p_6 \beta_7$
0000111	$\varphi_5 p_6 \beta_7$
0000011	β_7

Now, all you need to do is count how many unique parameters there are. A parameter is unique if

* To some degree, using β was an unfortunate choice, since the β parameter takes on singular importance – and refer to something altogether different than 'parameter confounding' – in the context of linear models, which we introduce in depth in chapter 6.

it occurs at least once in any of the probability statements. If you count the unique parameters in this table, you will see that there are 11 of them: 5 survival probabilities (φ_1 to φ_5), 5 recapture probabilities (p_2 to p_6), and one 'beta' term, β_7, the product of $\varphi_6 p_7$. Note, that this is only a technique to help you count the number of 'potentially' identifiable parameters – this does **not** necessarily mean that all of them are estimable. That is determined by the data. We introduce an approach (based on 'data cloning') for handling this issue in Appendix F.

Now, a fair question is 'why do we need to write out the saturated capture histories and the probability statements for all cohorts, since we could have used just the first cohort to count unique parameters?'. Well, the answer is, in this case, you really didn't need to. However, as you will see, this approach is useful and necessary for more complicated models. We introduce it now just to get you in the habit.

Let's consider the next two models: $\{\varphi_t\, p_\cdot\}$ and $\{\varphi_\cdot\, p_t\}$. From the results browser, we see that both models have 7 parameters. Let's confirm this. Again, we use the 'saturated capture histories approach'.

Start with the model $\{\varphi_t\, p_\cdot\}$:

encounter history	*probability*
1111111	$\varphi_1 p \varphi_2 p \varphi_3 p \varphi_4 p \varphi_5 p \varphi_6 p$
0111111	$\varphi_2 p \varphi_3 p \varphi_4 p \varphi_5 p \varphi_6 p$
0011111	$\varphi_3 p \varphi_4 p \varphi_5 p \varphi_6 p$
0001111	$\varphi_4 p \varphi_5 p \varphi_6 p$
0000111	$\varphi_5 p \varphi_6 p$
0000011	$\varphi_6 p$

Now, in this case, we do not have a terminal β term. The terminal product is $\varphi_6 p_\cdot$. Are both parts separately estimable? Yes. Since the constant recapture probability occurs at each occasion, we can use the information from preceding occasions to estimate the value of p. And, if we know the recapture probability p, then we can estimate any of the survival probabilities, including φ_6.

Specifically, φ_6 is estimated as

$$\hat{\varphi}_6 = \frac{\hat{\beta}_7}{\hat{p}},$$

under the assumption that $p_7 = p$ (this is clearly an untestable assumption). Thus, we have 7 identifiable parameters: 6 survival rates (φ_1 to φ_6) and 1 recapture probability (p). For the model $\{\varphi_\cdot\, p_t\}$, we have the same situation (7 estimable parameters) but in reverse. Finally, for our model $\{\varphi_\cdot\, p_\cdot\}$ (constant survival and recapture), there are only two estimable parameters.

How does MARK count parameters?

Needless to say, **MARK** uses a more 'technical' approach to counting the number of estimable parameters than the *ad hoc* approach described above. Here, we summarize some of the 'technical details'.

In Chapter 1, we considered the derivation of the MLE and the variance for a simple example involving a model with only 2 parameters: φ and p. The likelihood for the particular example was given as

$$\ln \mathcal{L}(\varphi, p) = 7 \ln(\varphi p \varphi p) + 13 \ln(\varphi p (1 - \varphi p)) + 6 \ln(\varphi (1 - p) \varphi p) + 29 \ln(1 - \varphi p - \varphi (1 - p) \varphi p).$$

We first derived the Hessian $\mathbf{H}$ as the matrix of second partial derivatives of the likelihood $\mathcal{L}$ with respect to the parameters φ and p):

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 \mathcal{L}}{\partial \varphi^2} & \frac{\partial^2 \mathcal{L}}{\partial \varphi \partial p} \\ \frac{\partial^2 \mathcal{L}}{\partial p \partial \varphi} & \frac{\partial^2 \mathcal{L}}{\partial p^2} \end{bmatrix}.$$

Next, we evaluated the Hessian at the MLE for φ and p (i.e., we substituted the MLE values for our parameters – $\hat{\varphi} = 0.6648$ and $\hat{p} = 0.5415$ – into the Hessian), which yielded the information matrix $\mathbf{I}$:

$$\mathbf{I} = \begin{bmatrix} -203.06775 & -136.83886 \\ -136.83886 & -147.43934 \end{bmatrix}.$$

The negative inverse of the information matrix ($-\mathbf{I}^{-1}$) is the variance-covariance matrix of the parameters φ and p:

$$-\mathbf{I}^{-1} = -\begin{bmatrix} -203.06775 & -136.83886 \\ -136.83886 & -147.43934 \end{bmatrix}^{-1} = \begin{bmatrix} -0.0122 & 0.0181 \end{bmatrix}.$$

While deriving the variance-covariance matrix is obviously the basis for estimating parameter precision, there is further utility in the information matrix: skipping the theory, the *effective rank* of the information matrix is an estimate of the *maximum* number of estimable parameters (but this does not account for confounded parameters). The effective rank of

$$-\mathbf{I}^{-1} = \begin{bmatrix} 0.0131 & -0.0122 \\ -0.0122 & 0.0181 \end{bmatrix},$$

is 2, meaning, we have 2 estimable parameters (which by now we know to be true for this model).

What is the effective rank of a matrix? Technically, the rank of a matrix (or a linear map, to be complete) is the dimension of the range of the matrix, corresponding to the number of linearly independent rows or columns of the matrix (or to the number of nonzero singular values of the map). The details can be found in any introductory linear algebra text, but the basic idea is easily demonstrated. Consider the following (4×4) matrix:

$$\mathbf{A} = \begin{bmatrix} 2 & 4 & 1 & 3 \\ -1 & -2 & 1 & 0 \\ 0 & 0 & 2 & 2 \\ 3 & 6 & 2 & 5 \end{bmatrix}.$$

The second column ($\mathbf{c}_2$) is twice the first column ($2 \cdot \mathbf{c}_1$), and the fourth column ($\mathbf{c}_4$) equals the sum of the first and the third ($\mathbf{c}_1 + \mathbf{c}_3$). Thus, column 2 is *dependent* on column 1, column 4 is *dependent* on columns 1 and 3, while columns 1 and 3 are *linearly independent*. Therefore, the set of linearly independent columns is $\{\mathbf{c}_1, \mathbf{c}_3\}$, and all other columns are linear combinations of them. Thus, the rank of $\mathbf{A}$ (i.e., the dimension of the column space) is 2. **MARK** uses numerical methods (discussed briefly later in this addendum) to calculate the effective rank of a matrix.

What about a less obvious case? For example, suppose we re-write the likelihood in terms of time-specific φ and p parameters:

$$\begin{aligned} \ln \mathcal{L}(\varphi_1, \varphi_2, p_2, p_3) = {} & 7 \ln(\varphi_1 p_2 \varphi_2 p_3) + 13 \ln(\varphi_1 p_2 (1 - \varphi_2 p_3)) \\ & + 6 \ln(\varphi_1 (1 - p_2) \varphi_2 p_3) + 29 \ln(1 - \varphi_1 p_2 - \varphi_1 (1 - p_2) \varphi_2 p_3). \end{aligned}$$

We use **MARK** to find the MLE estimates as: $\hat{\varphi}_1 = 0.6753$, $\hat{p}_2 = 0.5385$, and $\hat{\varphi}_2 = \hat{p}_3 = 0.5916$. Now, what is important here is that the terminal β_3 term is estimated as the product of φ_2 and p_3 – in fact, the estimates of φ_2 and p_3 could be **any** values from $0 \rightarrow 1$, as long as the product $(\varphi_2 p_3) = (0.5916)^2 = 0.3500$, (where $0.3500 = \hat{\beta}_3 = \hat{\varphi}_2 \hat{p}_2$). This becomes important later on.

For now, though, let's concentrate on parameter counting. First, we derive the Hessian, which for this model $\{\varphi_t\, p_t\}$ is given as

$$\mathbf{H} = \begin{bmatrix} \dfrac{\partial^2 \mathcal{L}}{\partial \varphi_1^2} & \dfrac{\partial^2 \mathcal{L}}{\partial \varphi_1 \partial \varphi_2} & \dfrac{\partial^2 \mathcal{L}}{\partial \varphi_1 \partial p_2} & \dfrac{\partial^2 \mathcal{L}}{\partial \varphi_1 \partial p_3} \\ \dfrac{\partial^2 \mathcal{L}}{\partial \varphi_2 \partial \varphi_1} & \dfrac{\partial^2 \mathcal{L}}{\partial \varphi_2^2} & \dfrac{\partial^2 \mathcal{L}}{\partial \varphi_2 \partial p_2} & \dfrac{\partial^2 \mathcal{L}}{\partial \varphi_2 \partial p_3} \\ \dfrac{\partial^2 \mathcal{L}}{\partial p_2 \partial \varphi_1} & \dfrac{\partial^2 \mathcal{L}}{\partial p_2 \partial \varphi_2} & \dfrac{\partial^2 \mathcal{L}}{\partial p_2^2} & \dfrac{\partial^2 \mathcal{L}}{\partial p_2 \partial p_3} \\ \dfrac{\partial^2 \mathcal{L}}{\partial p_3 \partial \varphi_1} & \dfrac{\partial^2 \mathcal{L}}{\partial p_3 \partial \varphi_2} & \dfrac{\partial^2 \mathcal{L}}{\partial p_2 \partial p_3} & \dfrac{\partial^2 \mathcal{L}}{\partial p_3^2} \end{bmatrix}.$$

Again, the *variances* of the parameters are along the diagonal, and the *covariances* are off the diagonal.

Next, we substitute in the MLE estimate for our 4 parameters. While we have unique estimates for $\hat{\varphi}_1$ and $\hat{p}_2$, what about the terminal $\hat{\beta}_3$ term? If we use the values **MARK** reports ($\hat{\varphi}_2 = \hat{p}_3 = \hat{\beta}_3 = 0.5916$), then the resulting information matrix is singular (meaning: we can't invert it to derive the variance-covariance matrix). Is this a problem?

Well, yes and no. A problem clearly if we want to estimate the variance-covariance matrix for our parameters (which we obviously want to do for any model). But, if the information matrix is singular, what can you do? Well, what if instead of $\hat{\varphi}_2 = \hat{p}_3 = 0.5916$, we instead had used $\hat{\varphi}_2 = 0.3500$, $\hat{p}_3 = 1.0$ (such that $\hat{\beta}_3$ still equals 0.3500). Again, remember that the estimates of φ_2 and p_3 could be **any** value from $0 \rightarrow 1$, as long as the product $(\varphi_2 p_3) = (0.5916)^2 = 0.3500$ (as noted above).

Substituting these values into the Hessian yields the information matrix, from which the negative inverse yields the variance-covariance matrix:

$$\mathbf{I} = \begin{bmatrix} -108.12 & -48.14 & -67.80 & -16.85 \\ -48.14 & -147.03 & 22.87 & -51.46 \\ -67.80 & 22.87 & -117.25 & 8.01 \\ -16.85 & 51.46 & 8.01 & -18.01 \end{bmatrix} \rightarrow -\mathbf{I}^{-1} = \begin{bmatrix} 0.024 & -0.008 & -0.016 & -0.006 \\ -0.008 & 318.191 & 0.005 & -909.091 \\ -0.016 & 0.005 & 0.019 & 0.008 \\ -0.006 & -909.091 & 0.008 & 2597.417 \end{bmatrix}.$$

Obviously, the variances for φ_2 and p_3 are 'wonky' (from the Latin). We discussed earlier how this can (on occasion) be used as a rough diagnostic to when parameters are inestimable.

But, our main objective here is to determine how *many* parameters are estimable? If we take the rank of this information matrix, we get 4, which we know to be correct, because in effect we've 'manually separated' the elements of the $\hat{\beta}_3$ term.

What if we had calculated the rank of the matrix substituting $\hat{\varphi}_1 = 0.6753$, $\hat{p}_2 = 0.5385$, and $\hat{\varphi}_2 = \hat{p}_3 = 0.5916$? We noted already that the information matrix using these values is singular, but...what about the rank? In fact, if we take the rank of the information matrix, we get 3, which matches the number of estimable parameters.

But, how many parameters are actually estimated *given the data*? In **MARK**, computation of the Hessian is performed with a finite central difference approximation for the second derivatives (i.e., the second derivatives are estimated *numerically*, not *analytically*). How does this work? Well, first define $\mathcal{L}_{0,0}$ as the log likelihood computed for the maximum likelihood estimates of a β_i and β_j. Further define $\mathcal{L}_{1,0}$ as the log likelihood computed with β_i incremented by the amount h, and $\mathcal{L}_{2,0}$ as the log likelihood computed with β_i incremented by the amount $2h$. Similarly, $\mathcal{L}_{-2,0}$ is the log likelihood computed with decremented by the amount $2h$. Using this notation, the second partial of the log likelihood for β_i is computed as:

$$\frac{\partial^2 \mathcal{L}_{0,0}}{\partial \beta_i^2} = \frac{1}{12h^2}\left(-\mathcal{L}_{2,0} + 16\mathcal{L}_{1,0} - 30\mathcal{L}_{0,0} + 16\mathcal{L}_{-1,0} - \mathcal{L}_{-2,0}\right),$$

and the joint partial of the log likelihood for β_i and β_j is computed as:

$$\frac{\partial^2 \mathcal{L}_{0,0}}{\partial \beta_i \partial \beta_j} = \frac{1}{4h^2}\left(-\mathcal{L}_{1,1} + \mathcal{L}_{1,-1} - \mathcal{L}_{-1,1} + \mathcal{L}_{-1,-1}\right).$$

Given the number of function evaluations needed to compute these derivatives, it is obvious why the computation of the variance-covariance matrix takes so long to calculate once the optimizations have completed*. However, a precise calculation of the information matrix is needed, not only to provide an estimate of the variance-covariance matrix of the β estimates, but also to compute the estimated number of parameters.

To invert and also compute the rank of the Hessian, a numerical approach based on a singular-value decomposition is computed (for you techno-philes – using the DSVDC algorithm of Dongarra *et al*. (1979), as implemented in **LINPACK**). This algorithm returns a vector of *singular values* (the **S** vector) of the same length as the number of rows or columns of the Hessian, sorted into descending order.

The trick at this point is to determine whether the smallest value(s) of the singular values in the **S** vector is (are) actually zero. Two rules are applied to make this decision. First, a *threshold value* is computed (and printed in the full output file) that is, in effect, a guess at what the minimum singular value would be smaller than if there were more betas than can be estimated. This threshold is based on the number of parameters used in the optimization and the value of h used to compute the Hessian. The precision of the numerical estimates in the Hessian is a function of h, as well as the number of columns in the design matrix (the number of β values). In **MARK**, the threshold is estimated from the gradient (of the likelihood) as the maximum value of **G** times 2.

Using this threshold value, all values of the conditioned singular values vector that are smaller than the threshold are considered to be parameters that have not been estimated. Conversely, all values of the conditioned singular value vector that are greater than the threshold are considered to be parameters that were estimated, and are part of the parameter count.

The threshold condition may suggest that all of the β values were parameters that were estimated, i.e., the smallest conditioned singular value is greater than the threshold. An additional test is performed to evaluate whether some of the β parameters were not actually estimated. The ratio of consecutive values in the sorted singular value array is used to identify large jumps in the singular values. Typically, the ratios of consecutive values decline slowly until a large gap is reached where parameters are not estimated.

As an example, consider the following portion of the full output for the global model (i.e., model

* All experienced **MARK** users have learned to be patient as the variance-covariance matrix is calculated, especially for complex models

$\{\varphi_t\ p_t\}$) fit to the male European dipper data, where the last φ and last p is identifiable only as a product (i.e., they are not separately estimable – a point we've made before). Thus, instead of the 12 parameters that you might have initially assumed are estimable, only 11 are actually estimated.

Here are the relevant sections from the full **MARK** output. First, the threshold is reported as:

```
 Threshold {phi(t)p(t) - sin - standard optimization} = 0.6680064E-005
```

Where does this value come from? As noted earlier, the numerical threshold value is estimated from the gradient as the maximum value of **G** times 2. For the male dippers, using a sin link with model $\{\varphi_t\ p_t\}$, the gradient and the maximum value of the gradient, **G**, are given as:

```
Gradient {phi(t)p(t) - sin - standard optimization}:
0.000000     -0.3304314E-05   0.000000      0.3125295E-05 -0.3340032E-05
0.000000       0.000000       0.000000     -0.1947290E-05 -0.1883578E-05
0.000000      0.2452112E-05

Maximum ABS(G) {phi(t)p(t) - sin - standard optimization} = 0.3340032E-05
```

The threshold from the gradient for this model is thus calculated as $(2{\times}0.3340032E{-}05) = 0.6680064E{-}005$, as reported by **MARK** (above).

Next, the sorted **S** vector:

```
S Vector {phi(t)p(t) - sin link}:
44.93890      42.21556      38.37998      34.75334      27.85340
15.31132      13.39506      10.90580      10.07123      2.918919
2.862238     0.1477205E-05
```

We see that only the final singular value ($0.1477205E{-}05$) is less than the threshold ($0.6680064E{-}005$), and so **MARK** concludes that there are 11 estimable parameters:

```
Number of Estimated Parameters {phi(t)p(t) - sin link} = 11
```

As discussed earlier, **MARK** also evaluates the ratio of consecutive values in the sorted singular value array to identify large jumps in the singular values. For the male dipper data, using the sin link, the 'gaps' (ratios) reported by **MARK** are:

```
Ratio Threshold = 50.00000  Max Gap (11/12) = 1937604.  Next Max Gap (9/10) = 3.450328
```

Only the final ratio (of singular values 11 and 12) is greater than the threshold of 50, whereas the next largest ratio of singular values 9 and 10 is smaller than the threshold. The singular value ratio which is greater than the threshold involves singular values 11 and 12. The preceding ratio, involving parameter 10 and 11 is not (in this case, it isn't even the second largest ratio – the second largest ratio involves singular values 9 and 10). So, this indicates that singular value 12 is the unique singular value which is not estimable (i.e., $(10/11) < 50$, $(11/12) > 50$, so singular value 12 is not estimated).

To make it easier to identify which model parameters correspond to the singular values, **MARK** prints a vector indicating the likely ordering of parameters by estimability (from most to least):

```
A Attempted ordering of parameters by estimatibility:
  5  4  3 12  2 11 10  9  1  7  8  6
Beta number 6 is a singular value.
```

This suggests that the 'least estimable' parameter (singular value 12) is parameter 6, which corresponds to $\hat{\varphi}_6$, which we know to be confounded with p_7 (and therefore not separately estimable).

So, both the 'threshold' and 'gap value' criteria lead to the same conclusion, and so **MARK** reports that 11 parameters (out of 12 structural parameters) are estimable, given the model structure, and the data, and places that value (11) in the browser for the column in the browser showing number of parameters.

An important point is how the link function can play into this process. In the above example, the *sin* link was used, so that parameters on their boundaries were still considered estimable. In contrast, with the *logit* link, parameters on their boundary often appear to have not been estimated (the underlying reason for this statement is discussed at length in Chapter 6). The following output is for the identical model, $\{\varphi_t\, p_t\}$, but now run using a logit link.

For the logit link, the threshold is estimated as:

```
Threshold {phi(t)p(t) - logit link} = 0.7471357E-005
```

Now, if we look at the **S** vector

```
S Vector {phi(t)p(t) - logit link}:
10.71643      9.017176      8.331518      7.603417      6.789932
2.093137     0.9428915     0.9341091     0.9204661     0.5904492
0.7234874E-06 0.5507724E-08
```

we see that the final 2 singular values are smaller than the threshold, so **MARK** reports 10 parameters, not 11:

```
Number of Estimated Parameters {phi(t)p(t) - logit link} = 10
```

If we consider the 'gap' (ratio) approach, we see (below) that the maximum 'gap' (ratio) in the ordered **S** vector occurs for singular values 10 and 11 (~ 816,115), while the next maximum gap, for singular values 11 and 12, is much smaller (~ 131):

```
Ratio Threshold = 50.00000   Max Gap (10/11) = 816115.4  Next Max Gap (11/12) = 131.3587
Gap Method for Num. of Estimated Parameters {phi(t)p(t) - logit link} = 10
```

Because both ratios are greater than the threshold of 50, the parameters corresponding to those singular value ratios are deemed to be not estimated – in this case singular values 11, and 12.

To make it easier to identify which model parameters correspond to the singular values, **MARK** prints a vector indicating the likely ordering of parameters by estimability (from most to least):

```
Attempted ordering of parameters by estimatibility:
  5  4  3  2  6  1 11 10  9  7 12  8
Beta number 8 is a singular value.
```

This suggests that the 'least estimable' parameter is parameter 8 (singular value 12), which corresponds to $\hat{p}_3$. The next 'least estimable' parameter is parameter 12 (singular value 11), which is $\hat{p}_7$, which we know to be confounded with $\hat{\varphi}_6$.

So, both the 'threshold' and 'gap value' criteria lead to the same conclusion, and so **MARK** reports that 10 parameters (out of 12 structural parameters) are estimable, given the model structure, and the data, and places that value (10) in the browser for the column in the browser showing number of parameters. However, we know that the value 10 is incorrect – it should be 11.

The difference between the results using the logit and sin links can be traced to $\hat{p}_3$. If you look at the real parameter estimates, you'll see that p_3 is estimated at its upper boundary of 1. For parameters estimated near the 0 or 1 boundaries, the estimates often appear to be singular, and not estimable. For this example, the β estimate for this parameter (p_3) was 21.011, which appears numerically (for the log likelihood) to have almost a zero first and second derivative for this parameter. In fact, a graph of the likelihood over possible β values in the range of, say, 19 to 23 would suggest the log likelihood is flat. As a result, this parameter is considered by **MARK** to not have been estimated, even though it actually was estimated. The user must correct the parameter count manually.

Alternatively, use the sin link to avoid problems with highly parameterized models where one or more parameters might be estimated near the 0 or 1 boundaries (we'll talk a lot more about link functions in Chapter 6).

When 'threshold' ≠ 'gaps'...

In the preceding, both the 'threshold' and 'gap' approaches yielded identical results, in terms of how many parameters each approach 'concluded' were estimable, given the model structure, and the data. What happens if they differ? Here, we consider the output from model $\{\varphi_t\, p_t\}$ for the male dippers, using a logit link, but specifying an 'alternative optimization' routine (something called *simulating annealing*, which is introduced in Chapter 10). Looking at the full output indicates a variety of 'problems' to consider.

First, **MARK** reports that the numerical convergence (when optimizing the likelihood) is suspect:

```
* *  WARNING  * *   Numerical convergence suspect.
```

In contrast to the preceding examples (using standard optimization), the estimated threshold is quite large:

```
Threshold {phi(t)p(t) - logit - SA optimization} = 12.502350
```

If we next consider the ordered **S** vector

```
S Vector {phi(t)p(t) - logit - SA optimization}:
11.96782      10.58541      9.014868      8.330652      6.789937
2.093127     0.9435693     0.9341522     0.9204858     0.5904339
0.2303222E-07 0.8980657E-08
```

we see that all 12 of the singular values are less than the threshold value. So, **MARK** concludes that none of the parameters are estimable, which is clearly problematic:

```
Number of Estimated Parameters {phi(t)p(t) - logit - SA optimization} = 0
```

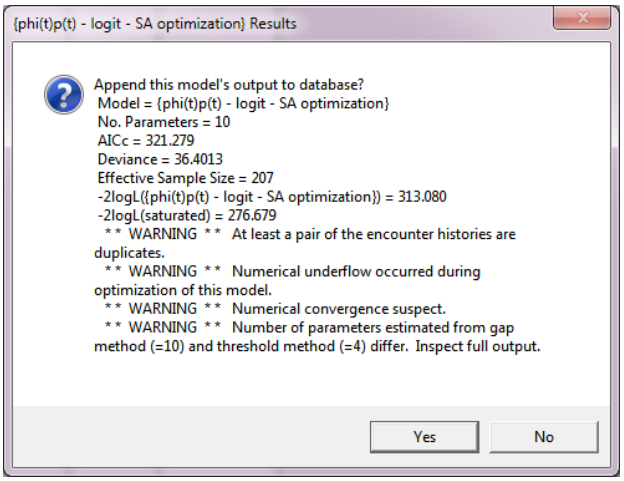
Looking next at the 'gap' (ratio) results, we see that this approach does a bit better. In fact, it returns a value of 10 estimable parameters, as in the preceding analysis using the standard optimization routine and the logit link:

```
Ratio Threshold = 50.00000  Max Gap (10/11) = 0.2563513E+08  Next Max Gap (5/6) = 3.243920

Gap Method for Num. of Estimated Parameters {phi(t)p(t) - logit - SA optimization} = 10
```

Now, at this point, you're faced with an obvious problem. The two approaches **MARK** uses to 'count parameters' yield very different outcomes (and in addition, we know that 0 parameters as determined by the 'threshold' approach has to be wrong). So, what to do?

Even though **MARK** will output the larger of the two values to the browser, you should spend some time evaluating the models and your data carefully if the two estimates differ. **MARK** will actually 'flag' this difference so you're aware of it, in several ways. First, when numerical evaluation of the likelihood has completed, **MARK** responds with a popup window, which indicates (near the bottom) that the '`number of parameters estimated from gap method (=10) and threshold method (=4) differ. Inspect full output`':



In addition, **MARK** will indicate in the results browser that there is a potential issue with the number of parameters estimated, by changing the color of the model name (white letters on blue background), with '`Check Par Cnt`' (*'check parameter count'*)*

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.
{phi(t)p(t) - logit - standard optimization}	334.2029	0.0000	0.42961	1.0000	10
Check Par. Cnt. {phi(t)p(t) - logit - SA optimization}	334.2029	0.0000	0.42961	1.0000	10
{phi(t)p(t) - sin - standard optimization}	336.4343	2.2314	0.14078	0.3277	11

Both 'warnings' prompt you to more examine the 'full output' (shown on the next page), and will also print out the full vector of successive ratios of the singular values, as one approach to help you try to figure out why the reported number of estimable parameters might differ between the two approaches.

```
 * *  WARNING  * *   Number of parameters estimated from gap method (=10) and
                      threshold method (=0) differ.  Inspect full output.

Ratios of S Vector {phi(t)p(t) - logit - SA optimization}:
 1/2  1.130595    2/3  1.174217    3/4  1.082132    4/5  1.226912    5/6   3.243920
 6/7  2.218308    7/8  1.010081    8/9  1.014847    9/10  1.558999   10/11 0.2563513E+08
 11/12  2.564648
```

* You can directly edit this modified model name – say, after checking the full output to diagnose the difference between the two parameter counts – by double-clicking the model name in the browser. Doing so will also change the color scheme back to the default (black text on white background).

Finally, you may notice that **MARK** gives you the option of choosing between two different procedures to estimate the variance-covariance matrix of the estimates. The first is the inverse of the Hessian matrix obtained as part of the numerical optimization of the likelihood function. This approach is not reliable, and should only be used when you are not interested in the standard errors, and already know the number of parameters that were estimated. The only reason for including this method in the program is that it is the fastest – no additional computation is required for the method.

The second method (the default) computes the information matrix directly using central difference approximations to the second partial derivatives (introduced a few pages back). This method (labeled the **2ndPart** method) provides the most accurate estimates of the standard errors, and is the default and preferred method.

Because the rank of the variance-covariance matrix is used to determine the number of parameters that were actually estimated, using different methods will sometimes result in a different number of parameters estimated, which can have important implications for model selection.