CHAPTER 16

Known-fate models

In previous chapters, we've spent a considerable amount of time modeling situations where the probability of encountering an individual is less than 1. However, there is at least one situation where we do not have to model detection probability – *known-fate data*, so-called because we *know* the fate of each marked animal with certainty. In other words, encounter probability is 1.0 (which must be true if we know the fate of a marked individual with certainty). This situation typically arises when individuals are radio-marked, although certain kinds of plant data can also be analyzed with the known fate data type. In such cases, known-fate data are important because they provide a theory for estimation of survival probability and other parameters (such as emigration). The focus of known fate models is the estimation of survival probability S, the probability of surviving an interval between sampling occasions. These are models where it can be assumed that the sampling probabilities are 1. That is, the status (dead or alive) of all tagged animals is known at each sampling occasion. For this reason, precision is typically quite high, as precise as the binomial distribution allows, even in cases where sample size is often fairly small. The only disadvantages might be the cost of radios and possible effects of the radio on the animal or its behavior. The model is a product of simple binomial likelihoods. Data on egg mortality in nests and studies of sessile organisms, such as mollusks, have also been modeled as known fate data.

In fact, the known fate data type is exactly the same as logistic regression in any statistical package. The main advantage of using **MARK** for known fate analysis is the convenience of model selection, and the capabilities to model average survival estimates easily, and compute random effects estimates.

16.1. The Kaplan-Meier Method

The traditional starting point for the analysis of known-fate data is the Kaplan-Meier (1958) – we'll discuss it briefly here, before introducing a more flexible approach that will serve as the basis for the rest of this chapter.

The Kaplan-Meier (hereafter, K-M) estimator is based on observed data at a series of occasions, where animals are marked and released only at occasion 1. The K-M estimator of the survival function is

$$\hat{S}_t = \prod_{i=1}^{t} \left(\frac{n_i - d_i}{n_i} \right)$$

where n_i is the number of animals alive and at risk of death at occasion i (given that their fate is known at the end of the interval), d_i is the number known dead at occasion i, and the product is over i up to the tth occasion (this estimator is often referred to as the product-limit estimator). Critical here is that n_i is the number known alive at the start of occasion i and whose fate (either alive or dead) is known at the end of the interval. Thus, the term in parentheses is just the estimated survival for interval i. Note that n_i does not include individuals censored during the interval. It is rare that a survival study will observe the occasion of death of every individual in the study. Animals are 'lost' (i.e., censored) due to radio failure or other reasons. The treatment of such censored animals is often important, but often somewhat subjective. These K-M estimates produce a survival function (see White and Garrott 1990); the cumulative survival up to time t. This is a step function and is useful in comparing, for example, the survival functions for males vs. females.

If there are no animals that are censored, then the survival function (empirical survival function or ESF) is merely,

$$\hat{S}_t = \left(\frac{\text{number alive longer than } t}{n}\right) \quad \text{for } t \geq 0$$

This is the same as the intuitive estimator where no censoring is occurring: $\hat{S}_t = n_{t+1}/n_t$; for example, $\hat{S}_2 = n_3/n_2$. The K-M method is an estimate of this survival function in the presence of censoring. Expressions for the variance of these estimates can be found in White and Garrott (1990).

A simple example of this method can be illustrated using the data from Conroy *et al*. (1989) on 48 radio-tagged black ducks. The data are

	survived to occasion							
week	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
number alive at start	48	47	45	39	34	28	25	24
number dying	1	2	2	5	4	3	1	0
number alive at end	47	45	39	34	28	25	24	24
number censored	0	0	4	0	2	0	0	0

Here, the number alive at the start of an interval are to known be alive at the start of sampling occasion j. This is equivalent to being alive at the start of interval j. For example, 47 animals are known to be alive at the beginning of occasion 2. Forty-five are alive at the start of interval 3, but 4 are censored from these 45 because their fate is unknown at the end of the interval, so that $n_3 = 41$. A further example is that 34 ducks survived to the start of occasion 5. Thus, the MLEs are

$$\hat{S}_1 = 47/48 = 0.979$$
$$\hat{S}_2 = 45/47 = 0.957$$
$$\hat{S}_3 = 39/41 = 0.951 \quad (\textit{note}\text{: only 41 because 4 were censored})$$
$$\hat{S}_4 = 34/39 = 0.872$$
$$\hat{S}5 = 28/32 = 0.875 \quad (\textit{note}\text{: only 32 because 2 were censored})$$
$$\hat{S}_6 = 25/28 = 0.893$$
$$\hat{S}_7 = 24/25 = 0.960$$
$$\hat{S}_8 = 24/24 = 1.00$$

Here one estimates 8 parameters – call this model `S(t)`. One could seek a more parsimonious model in several ways. First, perhaps all the parameters were nearly constant; thus a model with a single survival probability might suffice (i.e., `S(.)`) If something was known about the intervals (similar to the flood years for the European dipper data) one could model these with one parameter and denote the other periods with a second survival parameter.

Finally, one might consider fitting some smooth function across the occasions and, thus, have perhaps only one intercept and one slope parameter (instead of 8 parameters). Still other possibilities exist for both parsimonious modeling and probable heterogeneity of survival probability across animals. These extensions are not possible with the K-M method and K-L-based (i.e., AIC) model selection is not possible. To do this, we need an approach based on maximum likelihood estimation – as it turns out, the simple binomial model will do just that for known-fate data.

16.2. The Binomial Model

In the K-M approach, we estimated the survival probability by

$$\hat{S}_t = \prod_{i=1}^{t}\left(\frac{n_i - d_i}{n_i}\right)$$

where d_i is the number dying over the ith interval, and n_i is the number alive ('at risk') at the start of the interval and whose fate is also known at the end of the interval (i.e., not censored). Here, we use the equivalence (under some conditions) of the K-M estimator, and a binomial estimator, to recast the problem in a familiar likelihood framework.

Consider the situation for the case in which all animals are released at some initial time $t = 0$, and there is no censoring. If we expand the product term from the preceding equation, over the interval $[0, t]$,

$$\hat{S}_t = \left(\frac{n_0 - d_0}{n_0}\right)\left(\frac{n_1 - d_1}{n_1}\right)\ldots\left(\frac{n_t - d_t}{n_t}\right)$$

We notice that in the absence of censoring (which we assume for the moment), the number of animals at risk at the start of an interval is always the previous number at risk, minus the number that died the previous interval.

Thus, we can re-write the expanded expression as

$$\begin{aligned}
\hat{S}_t &= \left(\frac{n_0 - d_0}{n_0}\right)\left(\frac{n_1 - d_1}{n_1}\right)\ldots\left(\frac{n_t - d_t}{n_t}\right)\\
&= \left(\frac{n_0 - d_0}{n_0}\right)\left(\frac{n_0 - (d_0 + d_1)}{n_0 - d_0}\right)\times\left(\frac{n_0 - (d_0 + d_1 + d_2)}{n_0 - (d_0 + d_1)}\right)\times\ldots\\
&\quad\times\left(\frac{n_0 - (d_0 + d_1 + \cdots + d_t)}{n_0 - (d_0 + d_1 + \cdots + d_{t-1})}\right)
\end{aligned}$$

OK, while this looks impressive, its importance lies in the fact that it can be easily simplified to

$$\begin{aligned}
\hat{S}_t &= \left(\frac{n_0 - d_0}{n_0}\right)\left(\frac{n_0 - (d_0 + d_1)}{n_0 - d_0}\right)\times\cdots\times\left(\frac{n_0 - (d_0 + d_1 + \cdots + d_t)}{n_0 - (d_0 + d_1 + \cdots + d_{t-1})}\right)\\
&= \left(\frac{n_0 - (d_0 + d_1 + \cdots + d_t)}{n_0}\right)
\end{aligned}$$

If you look at this expression closely, you'll see that the numerator is the number of individuals from the initial release cohort (n_0) that remain alive (i.e., which do not die – the number that die is given by $(d_0 + d_1 + \cdots + d_t)$), divided by the number initially released. In other words, the estimate of survival to time t is simply the number surviving to time t, divided by the number released at time 0.

Now, this should sound familiar – hopefully you recognize it as the usual binomial estimator for survival as (in this case) number of survivors ('successes', in a literal sense) in n_0 trials. Thus, if

$$\hat{S}_i = \frac{y_i}{n_i}$$

where y_i is the number surviving to time i (on the interval $[i-1, i]$), and n_i is the number alive ('at risk') at the start of the interval (i.e., at time i), then we can write

$$\hat{S}_t = \prod_{i=1}^{t}\left(\frac{n_i - d_i}{n_i}\right) = \prod_{i=1}^{t}\left(\frac{y_i}{n_i}\right)$$

If you recall the brief introduction to likelihood theory in Chapter 1 (especially the section discussing the binomial), it will be clear that the likelihood expression for this equation is

$$\mathcal{L}(\theta \mid n_i, y_i) = \prod_{i=1}^{t} S_i^{y_i}(1 - S_i)^{(n_i - y_i)}$$

where θ is the survival model for the t intervals, n_i is the number of individuals alive (at risk) during each interval, y_i is the number surviving each interval, and S_i is the MLE of survival during each interval.

As suggested at the start of this section, the binomial model allows standard likelihood-based estimation and is therefore similar to other models in program **MARK**. To understand analysis of known-fate data using the binomial model in **MARK**, we first must understand that there are 3 possible scenarios under the known fate model. In a known-fate design, each tagged animal either:

1. survives to end of study (detected at each sampling occasion so fate is known on every occasion)
2. dies sometime during the study (its carcass is found on the first occasion after its death so that its fate is known)
3. survives up to the point where its fate is last known, at which time it is censored → the fate is known

Note, for purposes of estimating survival probabilities, there is no difference between animals seen alive and then removed from the population at occasion k and those censored due to radio failure or for other reasons. The binomial model assumes that the capture histories are mutually exclusive and that animals are independent, and that all animals have the same underlying survival probability when individuals are modeled with the same survival parameter (homogeneity across individuals). Known fate data can be modeled by a product of binomials.

Let us reconsider the black duck data (seen previously), using the binomial model framework : $n_1 = 48$, and $n_2 = 44$; the likelihood is

$$\mathcal{L}(S_1 \mid n_1, n_2) = \binom{n_1}{n_2} S_1^{n_2}(1 - S_1)^{(n_1 - n_2)} = \binom{48}{44} S_1^{44}(1 - S_1)^{(48-44)}$$

Clearly, one could find the MLE, $\hat{S}_1$, for this expression (e.g., $\hat{S}_1 = 44/48 = 0.917$). We could also easily derive an estimate of the variance (see section 1.3.1 in Chapter 1). Of course, the other binomial terms are multiplicative, assuming independence. The survival during the second interval is based on $n_2 = 44$ and $n_3 = 41$,

$$\mathcal{L}(S_1 \mid n_1, n_2) = \binom{n_1}{n_2} S_2^{n_2}(1 - S_2)^{(n_1 - n_2)} = \binom{41}{44} S_2^{41}(1 - S_2)^{(44-41)}$$

As noted above, the likelihood function for the entire set of black duck data (modified to better make some technical points below) is the product of these individual likelihoods.

16.3. Encounter Histories

Parameterization of encounter histories for a known-fate data is critical, and is structurally analogous to the `LDLD` format used in some other analyses (e.g., Burnham's live encounter-dead recovery model) – these are discussed more fully in Chapter 2. For the encounter histories for known-fate data, each entry is paired, where the first position (`L`) is a `1` if the animal is known to be alive at the start of occasion j; that is, at the start of the interval. A '`0`' in this first position indicates the animal was not yet tagged or otherwise not known to be alive at the start of the interval j or else its fates is not known at the end of the interval (and thus the animal is censored and is not part of the estimation during the interval).

The second position (`D`) in the pair is '`0`' if the animal survived to the end of the interval. It is a '`1`' if it died sometime during the interval. As the fate of every animal is assumed known at every occasion, the sampling probabilities (p) and reporting probabilities (r) are 1. The following examples will help clarify the coding:

encounter history	*probability*	*interpretation*
`10 10 10 10`	$S_1S_2S_3S_4$	tagged at occasion 1 and survived until the end of the study
`10 10 11 00`	$S_1S_2(1-S_3)$	tagged at occasion 1, known alive during the second interval, and died during the third interval
`10 11 00 00`	$S_1(1-S_2)$	tagged at occasion 1 and died during the second interval
`11 00 00 00`	$(1-S_1)$	tagged at occasion 1 and died during the first interval
`10 00 00 10`	S_1S_4	tagged at occasion 1, *censored* for interval 2 and 3 (not detected, or removed for some reason), and re-inserted into the study at occasion 4
`00 00 10 11`	$S_3(1-S_4)$	tagged at occasion 3, died during the 4th interval
`10 00 00 00`	S_1	tagged at occasion 1, known alive at the end of the first interval, but not released at occasion 2 and therefore *censored* after the first interval

Estimation of survival probabilities is based on a release (1) at the start of an interval and survival to the end of the interval (0), mortality probabilities are based on a release (1) and death (1) during the interval; if the animal then was censored, it does not provide information about S_i or $1 - S_i$).

Some 'rules' for encounter history coding for known-fate analysis:

a. The two-digit pairs each pertain to an interval (the period of time between occasions).

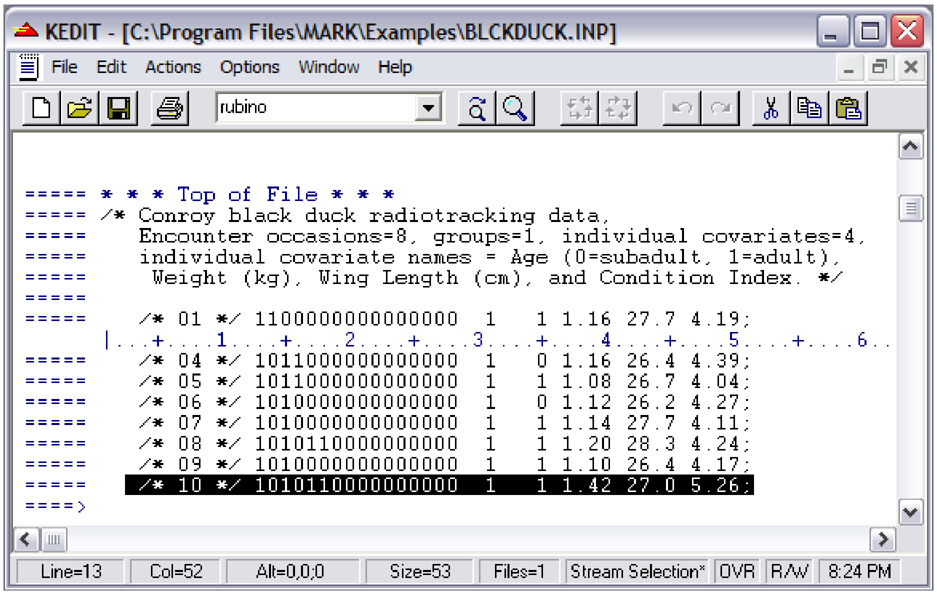
b. There are only 3 possible entries for each interval:
   - `10` = an animal survived the interval, given it was alive at the start of the interval
   - `11` = an animal died during the interval, given it was alive at the start of the interval
   - `00` = an animal was censored for this interval

c. In order to know the fate of an animal during an interval, one must have encountered it **both** at the beginning **and** the end of the interval.

16.4. Worked example: black duck survival

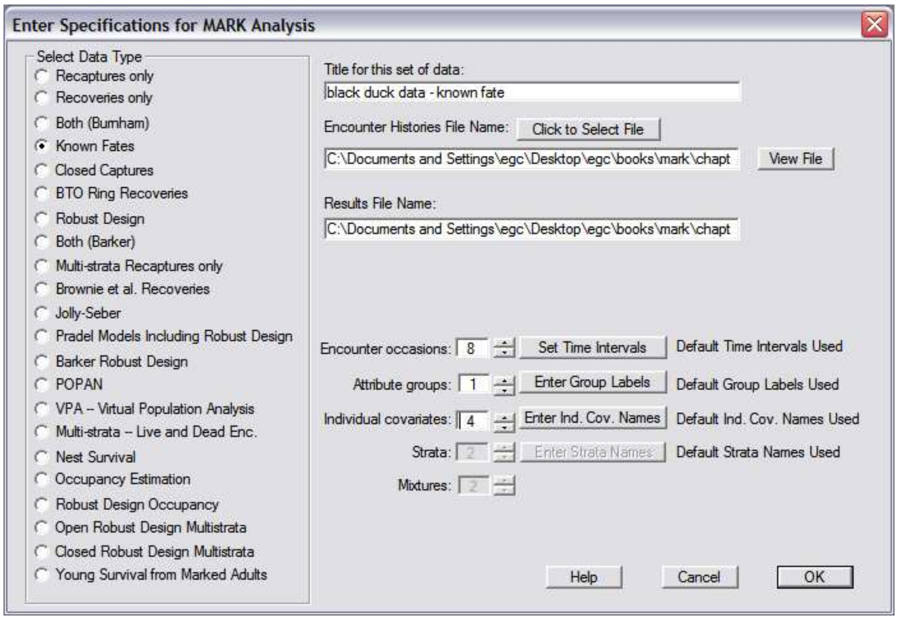
Here, we consider the black duck radio-tracking data from Conroy *et al.* (1989). These data are contained in the `BLCKDUCK.INP` file contained in the **MARK** examples subdirectory that is created when you install **MARK** on your computer. The data consists of 50 individual encounter histories, 8 encounter occasions, 1 group, and 4 individual covariates: `age` (0 = subadult, 1 = adult), `weight` (kg), `wing` (length, in cm) and `condition`. In this study, it was suspected that variation in body size, condition (or both) might significantly influence survival, and that the relationship between survival and these covariates might differ between adult and subadult ducks.

Here is what a portion of the `BLCKDUCK.INP` file looks like, showing the encounter histories and covariate values for the first 10 individuals:

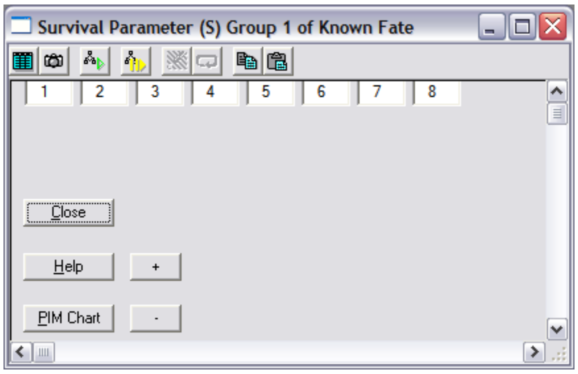


For example, the 10th individual in the data set has the encounter history '`1010110000000000`', meaning: marked and released alive at the start of the first interval, was detected alive at the start of the second interval, and then died during the third interval. The individual was radio-marked as an adult, and weighed 1.42 kilograms, had a wing length of 27.0 cm, and a condition index of 5.26. Ah – but look carefully – notice that in this `.INP` file, age is not coded as a classification variable (as is typically done for 'groupings' of individuals – see Chapter 2), but instead as a dichotomous covariate. Coding dichotomous groups as simple linear covariates is perfectly acceptable – sometimes it is more straightforward to implement – the only 'cost' (for large data sets) might be efficiency (the numerical estimation can sometimes be slower using this approach). However, the advantage of coding age as an individual covariate is that if age turns out to be not important, then you are not required to manipulate PIMs for 2 age groups.

Obviously, this has some implications for how we specify this data set in **MARK**. Start a new project in **MARK**. Select '**`known fates`**' as the data type. Enter 8 encounter occasions. Now, the 'trick' is to remember that even though there are two age groups in the data, we're coding age using an individual covariate – as such, there is still only 1 attribute group, not two. So, leave attribute groups at the default of 1. For individual covariates, we need to 'tell' **MARK** that the input file has 4 covariates which we'll label as `age`, `weight`, `wing`, and `cond` (for condition), respectively.



Once we've specified the data type, we'll proceed to fit a series of models to the data. Let's consider models S_t, S_{age}, $S_.$, S_{weight}, S_{wing}, and S_{cond}. Clearly, the most parameterized of these models is model S_t, so we'll start with that. Here is the PIM for survival:

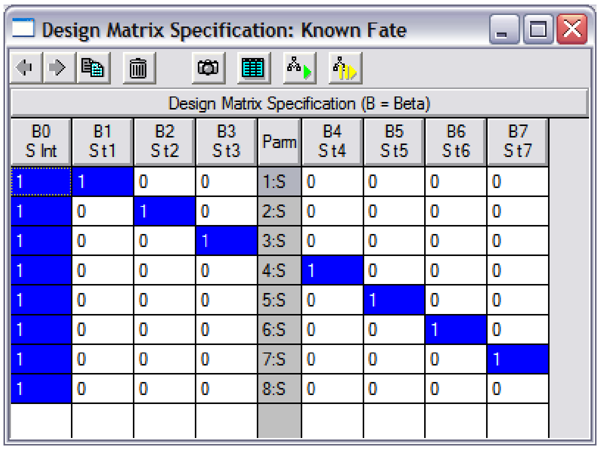


Not only is this particular PIM rather 'boring' (only a single row), in fact, there are no other PIMs for this analysis! Why? Easy – for known-fate data, we assume that all individuals are detected at each occasion, conditional on being alive and in the sample (i.e., we assume detection probability equals 1). Thus, the only parameter to model is the survival parameter (this should make sense – look back at the table on page 4 of this chapter – notice that the probability expressions corresponding to the different encounter histories are functions only of S_i – no encounter probability is included).

Why only a single row? Again, the assumption in the 'known-fate' analysis is that all individuals are released on the same occasion – presumably, at the start of the study (we'll consider staggered entry designs later). So, a single row, since all individuals in the analysis are in the same release cohort. Of course, this means that you aren't able to separate 'time' effects from 'age' effects in the analysis – at least, using the '**known fates**' data type in **MARK**. Remember, the age factor in this analysis is acting as a *classification* variable – and does not indicate the effects of aging (getting older over the course of the study) on survival. If you're marked individuals are all adults, then this may not be a particular problem. But, if your marked sample are subadults or young individuals, or perhaps a heterogeneous mixture of adults which might contain some proportion of transient individuals (see Chapter 7), you might have a problem. We'll deal with this later on in the chapter. For now, we'll proceed with the analysis, assuming all the assumptions of the classic known-fates analysis have been met.

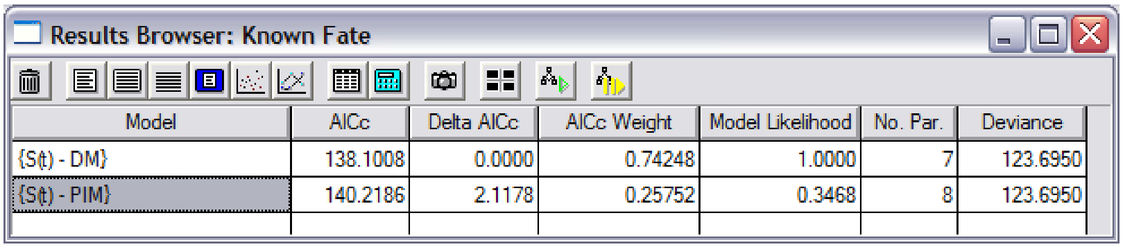
Given the preceding discussion, it should be clear that for a known-fate data, the PIMs (and as a result, the model-fitting in general) is very straightforward. The default PIM (shown on the previous page) corresponds to model S_t. We go ahead, fit the model, and add the results to the browser. Recall that the default link function when using the PIM approach to model fitting is the sin link.

But, also recall that ultimately, we want to use the complete flexibility of the design matrix for fitting models in **MARK**. So, let's 're-build' our starting model S_t, using the design matrix. In this case, since the model we're fitting corresponds to the fully time-dependent model, we can generate the design matrix simply by selecting '**Design | Full**', which yields the following design matrix:



B0 S Int	B1 S t1	B2 S t2	B3 S t3	Parm	B4 S t4	B5 S t5	B6 S t6	B7 S t7
1	1	0	0	1:S	0	0	0	0
1	0	1	0	2:S	0	0	0	0
1	0	0	1	3:S	0	0	0	0
1	0	0	0	4:S	1	0	0	0
1	0	0	0	5:S	0	1	0	0
1	0	0	0	6:S	0	0	1	0
1	0	0	0	7:S	0	0	0	1
1	0	0	0	8:S	0	0	0	0

Go ahead and fit this model, and add the results to the browser – label the model '`S(t) - DM`', to indicate it is the S_t model, constructed using a design matrix (DM) approach. Once you've added this model to the results browser (shown at the top of the next page), you'll notice that the two models (which are structurally identical) report different numbers of estimated parameters – 7 estimated parameters for the model fit using the DM (and the logit link), and 8 estimated parameters for the model fit using the PIM approach (and the sin link).

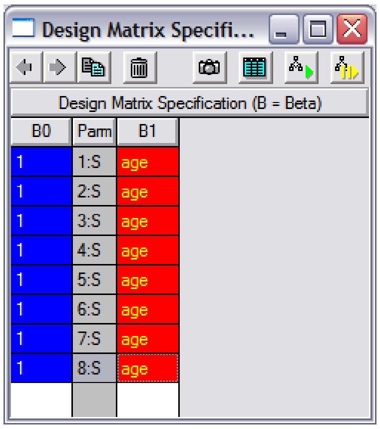


Results Browser: Known Fate

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{S(t) - DM}	138.1008	0.0000	0.74248	1.0000	7	123.6950
{S(t) - PIM}	140.2186	2.1178	0.25752	0.3468	8	123.6950

In fact, what we see here is fairly common for known-fate studies – in many such studies, the sampling interval is often relatively short, such that the survival probabilities over each interval are often relatively close to 1.0. We discussed previously (Chapter 6) how the different link functions behave when parameter values are near the $[0, 1]$ boundary. In the present example, examination of the reconstituted parameter values on the probability scale are in many cases close to the boundary – the two models differ in the estimate of survival for the last interval – the sin link estimates survival for the final interval at 1.00, whereas the logit link estimates the survival as 1.00, but fails to properly count this parameter as being estimated. We know that the number of estimated parameters for this analysis is 8 – so, we manually adjust the number of parameters for the model fit using the design matrix from 7 to 8 (when we do so, we see that the AIC_c and related statistics for the two models are now identical). We then delete the model fit using the PIM, since it is redundant to the model fit using the DM.

The next model in our candidate model set is model S_{age}. Recall that for this analysis, age is entered as a linear covariate in the `.INP` file, where age = 0 for subadults, and age = 1 for adults. Recall from Chapter 11 that individual covariates are introduced directly into the design matrix. So, for model S_{age}, the design matrix will look like



With this design matrix, we can interpret β_2 as the difference in survival between subadults and adults, i.e., what should be added on a logit scale to the subadult survival estimate to obtain the adults survival estimate (interpretation of the β_i terms in the linear model is discussed at length in Chapter 6). We run this model, and add the results to the browser. We do much the same thing for each of the remaining models in the model set – each time, making simple modifications to the design matrix. The results browser showing the results from all of the models in our candidate model set is shown at the top of the next page. Interpretation and processing of the results follows the usual process outlined in earlier chapters, so we will not elaborate further here.

Results Browser: Known Fate

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{S(.)}	136.1581	0.0000	0.32786	1.0000	1	134.1440
{S(condition)}	136.5835	0.4254	0.26504	0.8084	2	132.5408
{S(t) - DM}	138.1008	1.9427	0.12412	0.3786	7	123.6950
{S(age)}	138.1515	1.9934	0.12101	0.3691	2	134.1088
{S(wing)}	138.1862	2.0281	0.11893	0.3628	2	134.1435
{S(t) - PIM}	140.2186	4.0605	0.04305	0.1313	8	123.6950

16.5. Pollock's staggered entry design

The usual application of the Kaplan-Meier method assumes that all animals are released at occasion 1 and they are followed during the study until they die or are censored. Often new animals are released at each occasion (say, weekly); we say this entry is 'staggered' (Pollock *et al*. 1989). Assume, as before, that animals are fitted with radios and that these do not affect the animal's survival probability. This staggered entry fits easily into the K-M framework by merely redefining the n_i to include the number of new animals released at occasion i. Therefore, conceptually, the addition of new animals into the marked population causes no difficulties in data analysis.

But, you might be wondering how you handled staggered entry designs in **MARK** – after all, how do you handle more than one cohort, if the survival PIM has only one row? If you think that the survival of the newly added animals is identical to the survival of animals previously in the sample, then you can just include the new animals in the encounter histories file with pairs of '`00`' `LD` codes prior to when the animal was captured and first put into the sample.

But what if you think that the newly added animals have different survival. Obviously, you need more rows. How? As it turns out, there is a straightforward and fairly intuitive way to tweak the known-fate data type (in this case, allowing it to handle staggered entry designs) – you simply add in additional groups for each release occasion (each release cohort), thus allowing cohort-specific survival probabilities. For this to work, you need to fix the survival probabilities for these later cohorts prior to their release to 1, because there is no data available to estimate these survival rates. With multiple groups representing different cohorts, analyses equivalent to the upper-triangular PIMs of the CJS and dead recovery data types can be developed.

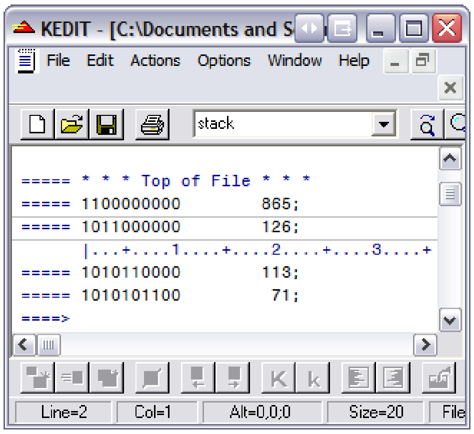
16.5.1. Staggered entry – worked example

To demonstrate the idea, we'll consider a somewhat complex example, involving individuals radio-marked as young – the complexity lies in how you handle the age-structure. Within a cohort, age and time are collinear, but with multiple release cohorts, it is possible to separate age and time effects (see Chapter 7). We simulated a dataset (`staggered.inp`) where individuals were radio-marked as young and followed for 5 sampling intervals – assume each interval is (say) a month long. We assume that all individuals alive and in the sample were detected, and that all fatalities were recorded (detected). We let survival in the interval following marking be 0.4, while survival in subsequent intervals (with a given cohort) was 0.8. For convenience, we'll refer to the two age classes as 'newborn' and 'mature', respectively.

If this were a typical 'age' analysis (see Chapter 7), this would correspond to the following PIM structure:

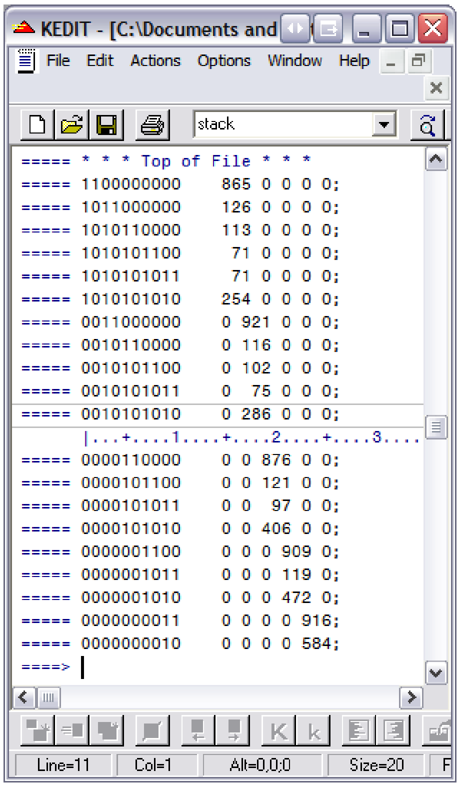
1	2	2	2	2
	1	2	2	2
		1	2	2
			1	2
				1

But, here we are considering a known-fate data, with staggered entry. To begin, let's first have a look at the `.INP` file – the first few lines are shown below:

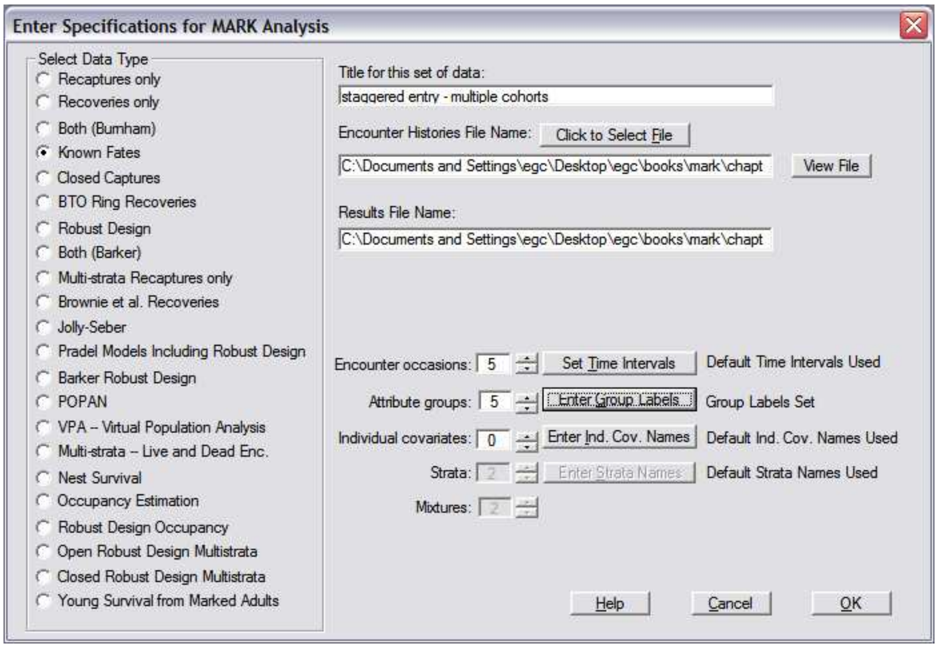


Now, at first glance, the structure of this file might seem perfectly reasonable. There are 5 occasions, in `LDLD` format. We see, for example, there were 865 individuals marked and released in the first cohort which died during the first interval (as newborns), 126 which were marked and released in the first cohort which died during the second interval (as mature individuals), and so on. But, what about the second cohort, and the third cohort, and so on?

How do we handle these 'additional cohorts'? As mentioned, we accomplish this in the known-fate data in **MARK** by specifying multiple groups – one group for each additional release cohort (in this case, 5 groups). However, while it is easy enough to specify 5 groups in the data specification window in **MARK**, we first need to modify the `.INP` file to indicate multiple groups. Recall from earlier chapters (especially Chapter 2), that each grouping requires a *frequency* column. So, 5 groups mean 5 frequency columns – not just the single frequency column we start with. The fully modified `staggered.inp` file is shown at the top of the next page (we'll let you make the modifications yourself). Notice that there are now 5 frequency columns – the first frequency column corresponds to number of individuals marked and released in cohort 1, the second frequency column corresponds to the number of individuals marked and released in cohort 2, and so on. Pay particular attention to the structure of these frequency columns.



Now that we've modified the `.INP` file (above), we can go ahead and run the analysis in **MARK**. We select the known-fate data type, and specify 5 groups (which we'll label as `C1`, `C2`,. . . ,`C5`, for cohort 1, cohort 2, and so on, respectively, to cohort 5):



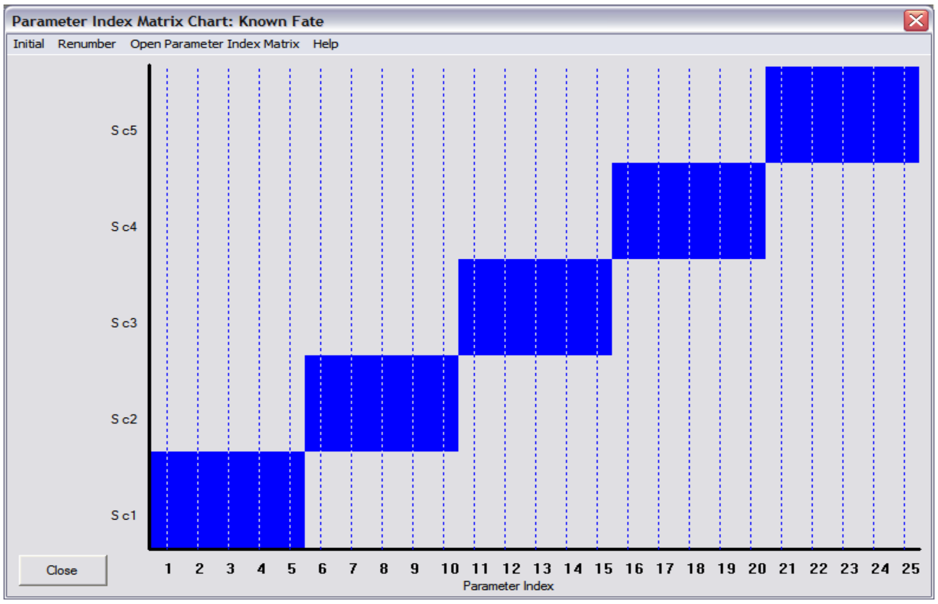
OK, so far, so good. Now for the only real complication – how to structure the PIMs for each cohort, and which parameters to fix to 1, in order for the analysis to make sense. Let's consider the following 2 models for our model set: $S_{a_2 \times cohort}$, and S_{a_2}. The first model indicates 2 age classes (newborn, and mature), with differences among cohorts. This corresponds to the following PIM structure:

1	6	6	6	6
	2	7	7	7
		3	8	8
			4	9
				5

The second model has differences in survival among the two age classes, but no differences among cohorts. This corresponds to the following PIM structure:

1	2	2	2	2
	1	2	2	2
		1	2	2
			1	2
				1

OK, so these are the 2 models we want to fit in **MARK**. The challenge is figuring out how to build them, and which parameters to fix. Clearly, the first model `S(a2 - cohort)` is the most general (since it has the most parameters), so we'll start there. Here is the default PIM chart for these data:



We see from the following PIM structure for this model that the first cohort consists of 2 age classes, as does the second, third, and so on. So, we might choose to simply right-click on the various 'blue-boxes' in the PIM chart, and select '**age**' – specifying 2 age-classes. Now, while you could, with some care, get this to work, there is an alternative approach which, while appearing to be more complex (and initially perhaps less intuitive), is in fact much easier.

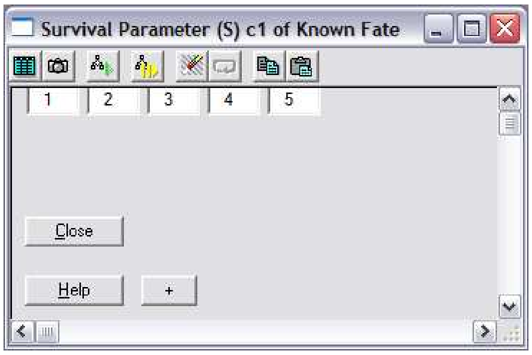
The key is in remembering that in the known-fates staggered entry analysis, we treat each cohort as if it were a separate group, fixing any '`00`' cells preceding the initial encounter in a cohort to 1.0. Again, keep in mind that each row (cohort) represents (analytically) a separate group. And, as noted, we want to fix the estimate for any of the preceding '`00`' cells to 1.0. Where do these cells occur? We've added them to the PIM in the following:

1	6	6	6	6
00	2	7	7	7
00	00	3	8	8
00	00	00	4	9
00	00	00	00	5

Now for the big step – if all of the '`00`' cells are ultimately to be fixed to 1.0, then we clearly would need only one parameter to code for them. So, let's rewrite the PIM, using the parameter 1 for the '`00`' cells, and then increasing the value of all of the other parameters by 1:

2	7	7	7	7
1	3	8	8	8
1	1	4	9	9
1	1	1	5	10
1	1	1	1	6

OK, now what? Well, each cohort is a group. So, we open up the PIM chart for each of the 5 groups (cohorts) in our example analysis – each of them has the same structure: a single line – here is the starting PIM for cohort 1:

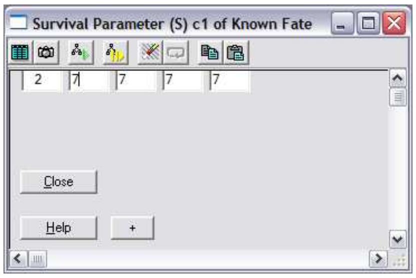


So, remembering that we want the overall PIM structure (over all cohorts) to look like

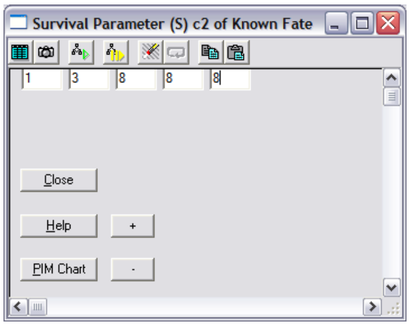
2	7	7	7	7
1	3	8	8	8
1	1	4	9	9
1	1	1	5	10
1	1	1	1	6

then it should be clear how to modify the PIM for cohort 1 - it needs to be modified to correspond to the first row of the overall PIM structure.

In other words, for cohort 1

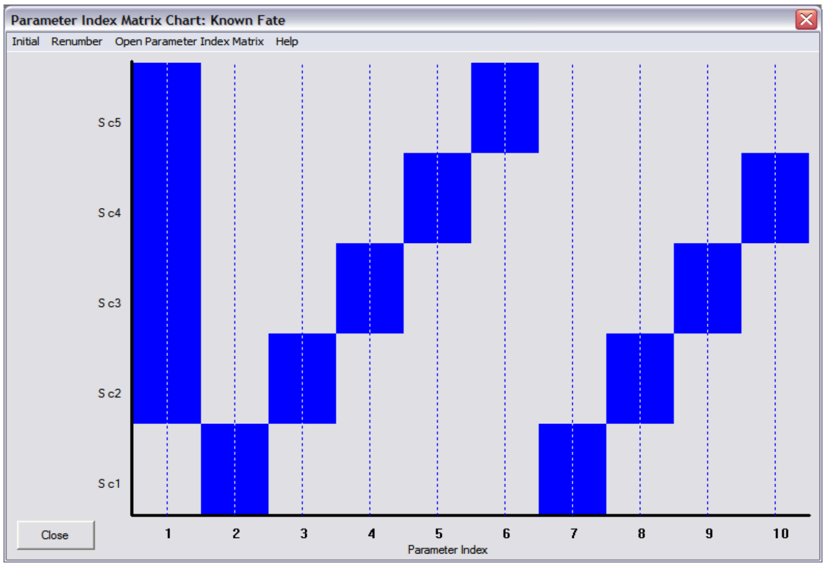


and for cohort 2,



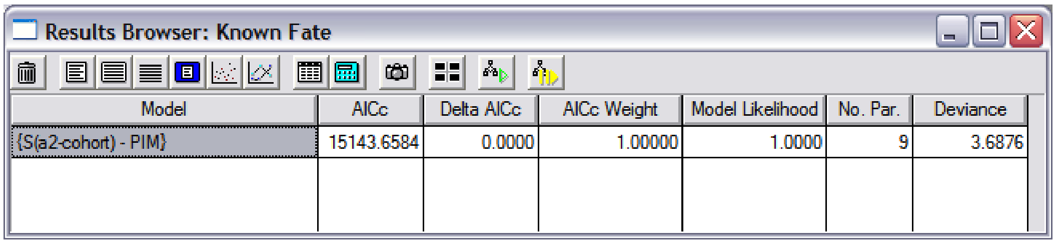
and so on – each PIM modified to match the corresponding row (representing a specific cohort) in the overall PIM.

Before we run our analysis, it's worth looking at the PIM chart for the model we've just created:



Note that the new parameter 1 occurs only in groups (cohorts) 2 to 6. The 'staircase' pattern for parameters 2 to 6, and 7 to 10 shows that we're allowing survival to vary among release cohorts as a function of age: in the first period following marking (*newborns*, parameters 2 to 6), and subsequent intervals (*mature*, 7 to 10). Note that in cohort 5, there are no 'mature' individuals.

Now, all that is left to do is to run the model, and add the results to the browser. All you need to do is remember that parameter 1 is fixed to 1.0. Go ahead and run the model, after first fixing the appropriate parameter to 1.0 – add the results to the browser – call the model '`S(a2 - cohort) - PIM`' (we add the word PIM to indicate the model was built by modifying the PIMs).


Results Browser: Known Fate

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{S(a2-cohort) - PIM}	15143.6584	0.0000	1.00000	1.0000	9	3.6876

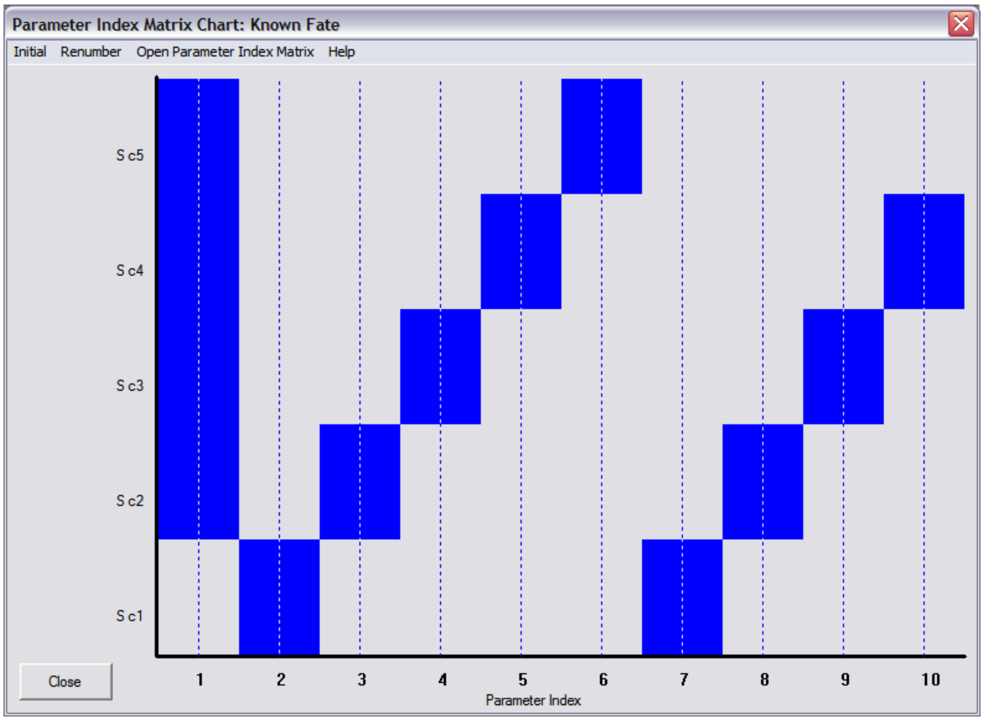
OK, what about the second model – model `S(a2)` (no cohort effect)? Well, if you reached this point in the book (i.e., have worked through the preceding chapters), you might realize that this model corresponds to

1	2	2	2	2
	1	2	2	2
		1	2	2
			1	2
				1

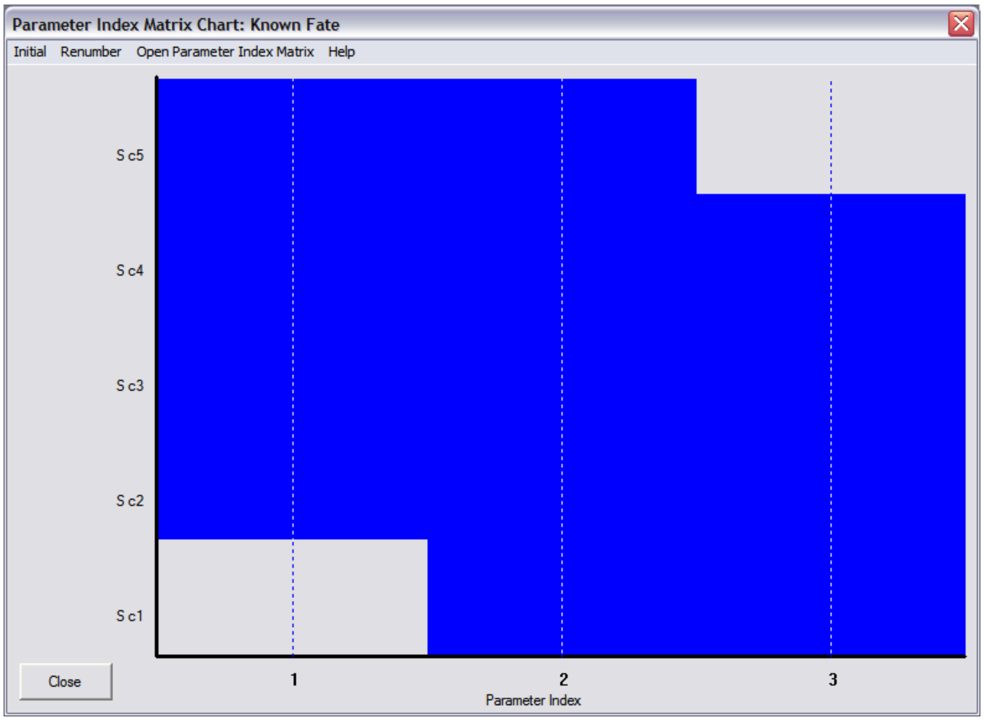
Again, if we add a parameter 1 to indicate the '`00`' cells preceding the first encounter within each cohort, and subsequently increment the parameter indexing for all other parameters by 1, we get

2	3	3	3	3
1	2	3	3	3
1	1	2	3	3
1	1	1	2	3
1	1	1	1	2

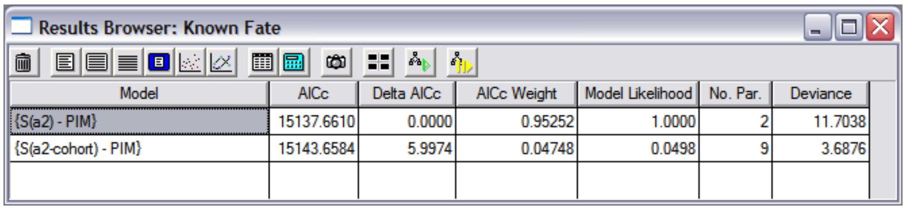
We can build this model conveniently by simply modifying the PIM chart for the preceding model `S(a2 - cohort)`. Recall that the PIM chart for that model was (see top of the next page)



So, to build model `S(a2)`, all we need to do is 'remove' the cohort variation for parameters 2 to 6, and 7 to 10 – this is shown in the modified PIM chart, below:



Now, run this model, first fixing parameter 1 to 1.0, label it '`S(a2) - PIM`', and add the results to the browser:



Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{S(a2) - PIM}	15137.6610	0.0000	0.95252	1.0000	2	11.7038
{S(a2-cohort) - PIM}	15143.6584	5.9974	0.04748	0.0498	9	3.6876

As expected, model `S(a2)` (the true, underlying model which we used to generate the data) gets virtually all of the AIC weight, relative to the other model. And, the reconstituted parameter estimates are very close to the true underlying values.

Now, while 'fiddling' with the PIM chart (and the underlying PIMs) is convenient for these simple models, we know from earlier chapters that there are structural limits to the types of models we can construct this way. Most obviously, we can't use the PIM approach to build models with additive effect. Ultimately, it's to our advantage to build models using the design matrix (DM), since all reduced parameter models can be constructed simply by manipulating the structure of the DM for the most general model. Let's build the DM for model '`S(a2 - cohort)`', which is the most general model of the two models in our candidate model set).

First, we start by writing out the conceptual structure of the linear model corresponding to this model:

$$S = \text{cohort} + \text{age} + \text{cohort.age}$$

The first term is fairly straightforward – we have 5 cohorts, so we need $(5 - 1) = 4$ columns to code for cohort. What about age? Well, look again at the PIM for this model:

2	7	7	7	7
1	3	8	8	8
1	1	4	9	9
1	1	1	5	10
1	1	1	1	6

Remembering that parameter 1 is fixed at 1.0, and is thus a constant. We can ignore it for the moment (although we do need to account for it in the DM). Pay close attention to the parameters along and above the diagonal. These represent each of the two age classes in our model – the vary among rows within an age class, but are constant among columns within a row, specifying cohort variation for a given age class, but no time variation (recall from Chapter 7 that a fully age-, time- and cohort-dependent model is generally not identifiable, since the terms are collinear). So, we have 2 age classes, meaning we need $(2 - 1) = 1$ column to code for `age`. What about `cohort`? Well, 5 cohorts, so $(5 - 1) = 4$ columns to code for `cohort`. Again, hopefully familiar territory. If not, go back and re-read Chapter 6.

But, what about the interaction terms (`age.cohort`) – do we need $(4 \times 1) = 4$ columns? If you think back to some of the models we constructed in Chapter 7 (age and cohort models), especially those models involving individuals marked as young only you might see how we have to handle interaction terms for this model. Recall from Chapter 7 that the interaction columns in the DM reflected 'plausible' interactions – if a specific interaction of (say) age and time wasn't possible, then there was no column in the DM for that interaction. For example, for an analysis of individuals marked as young, there can be no interaction of age (young or adult) with time in the first interval, since if the sample are all marked as young, then there are no marked adults in the first interval to form the interaction (i.e., there can be no plausible interaction of age and cohort in the first interval, since only one of the two age classes is present in the first interval).

OK, so what does this have to do with our known-fate data? The key word is 'plausible' – we build interactions only for interactions that are plausible, given the structure of the analysis. In this case, there are only 2 true age classes (newborn, and mature). All of the other 'age' classes are 'logical' – we've 'created' them to handle the preceding '`00`' terms in the PIM. They are not true 'age' classes, since there are no marked animals in those classes. As such, there are no interactions between cohort and any of these logical '`00`' age classes – we need only consider the interactions of the two true 'biological' age classes (newborn, and mature), with cohort. But, how many columns? Look closely again at the PIM:

2	7	7	7	7
1	3	8	8	8
1	1	4	9	9
1	1	1	5	10
1	1	1	1	6

Pay particular attention to the fact that the 'newborn' age class shows up in all 5 cohorts, while the 'mature' age class shows up only in the first 4 cohorts (and not in the fifth). So, not all age.cohort interactions are 'plausible'. Which ones are 'plausible'? Well, both age classes are represented in the first 4 cohorts, but both age classes are represented only over intervals 2 to 4. Thus, we only need include cohorts 2, 3 and 4, in the interaction terms. See the pattern? If not, try again. It's very similar to problems we considered in Chapter 7.

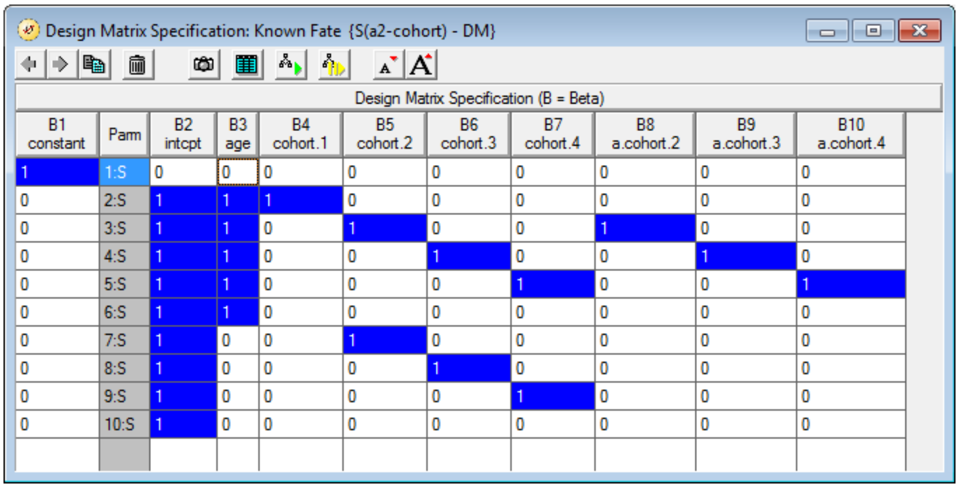
OK, penultimate step – what about parameter 1? Well, as noted earlier, since it's fixed to 1.0, then it's simply a constant across cohorts, and thus, enters into the linear model as a single parameter.

Now, finally, we're ready to write out the linear model corresponding to `S(a2 - cohort)`.

$$
\begin{aligned}
\hat{S} = \;& \beta_1(\texttt{constant}) \\
&+ \beta_2(\texttt{intercept}) \\
&+ \beta_3(\texttt{age}) \\
&+ \beta_4(\texttt{c}_1) + \beta_5(\texttt{c}_2) + \beta_6(\texttt{c}_3) + \beta_7(\texttt{c}_4) \\
&+ \beta_8(\texttt{age.c}_2) + \beta_9(\texttt{age.c}_3) + \beta_{10}(\texttt{age.c}_4)
\end{aligned}
$$

Is this correct? It has the same number of terms (10), as there are parameters in the PIM chart, so it would seem to be correct.

The next step, then, is to actually build the DM. We start by having **MARK** present us with a 10-column '**`reduced`**' DM as the starting point. The completed DM for this model is shown at the top of the next page. Column 1 (labeled `B1`) contains a single '1' - this represents parameter 1, which is a constant – fixed to 1.0 for all cohorts. The next column (labeled `B2`) represents the intercept for the 'age and cohort' part of the model. Column `B3` codes for `age` – `1` for newborn individuals, and `0` for mature individuals (note the different number of rows for each age class – this is key – 5 rows for newborns, and 4 rows for mature individuals). Columns `B4` to `B7` code for `cohort`. Note how the first row for newborn individuals for cohort 1 is coded, and note that this row does not show up for mature individuals – since, in cohort 1, there are no mature individuals! Finally, the interaction terms – columns `B8` to `B10`, for those '`age.cohort`' combinations that represent 'plausible' interactions.

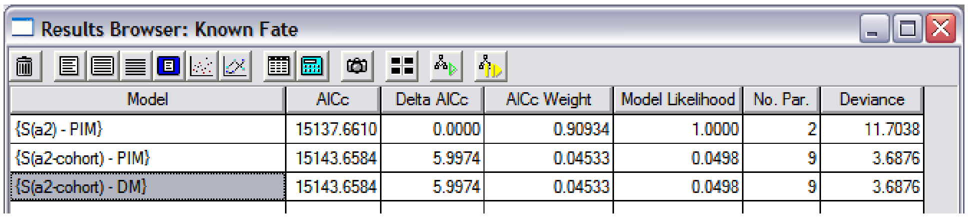


Design Matrix Specification: Known Fate {S(a2-cohort) - DM}

Design Matrix Specification (B = Beta)

B1 constant	Parm	B2 intcpt	B3 age	B4 cohort.1	B5 cohort.2	B6 cohort.3	B7 cohort.4	B8 a.cohort.2	B9 a.cohort.3	B10 a.cohort.4
1	1:S	0	0	0	0	0	0	0	0	0
0	2:S	1	1	1	0	0	0	0	0	0
0	3:S	1	1	0	1	0	0	1	0	0
0	4:S	1	1	0	0	1	0	0	1	0
0	5:S	1	1	0	0	0	1	0	0	1
0	6:S	1	1	0	0	0	0	0	0	0
0	7:S	1	0	0	1	0	0	0	0	0
0	8:S	1	0	0	0	1	0	0	0	0
0	9:S	1	0	0	0	0	1	0	0	0
0	10:S	1	0	0	0	0	0	0	0	0

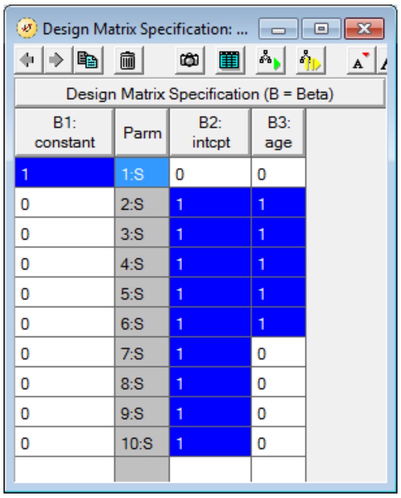
Go ahead and run this DM-based model (label it `S(a2-cohort - DM)`), and confirm that the results exactly match those for the model you constructed using the PIM chart, as shown below:



Results Browser: Known Fate

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{S(a2) - PIM}	15137.6610	0.0000	0.90934	1.0000	2	11.7038
{S(a2-cohort) - PIM}	15143.6584	5.9974	0.04533	0.0498	9	3.6876
{S(a2-cohort) - DM}	15143.6584	5.9974	0.04533	0.0498	9	3.6876

Now that you have the DM for the general model, try constructing model `S(a2)` – the second model. We already did this a few pages back using the PIM approach, but we can generate the same model easily using the DM approach by simply deleting (i) the columns of the DM coding for `cohort`, and (ii) the (`age.cohort`) interaction columns:

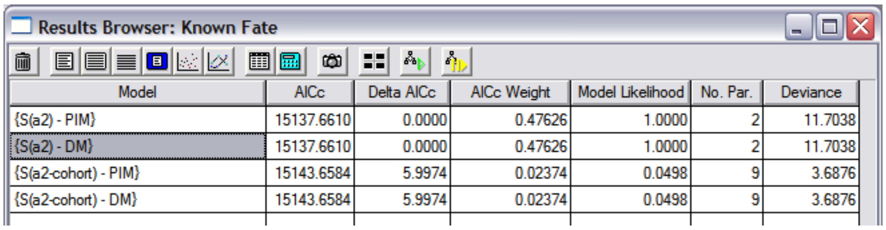


Design Matrix Specification: ...

Design Matrix Specification (B = Beta)

B1: constant	Parm	B2: intcpt	B3: age
1	1:S	0	0
0	2:S	1	1
0	3:S	1	1
0	4:S	1	1
0	5:S	1	1
0	6:S	1	1
0	7:S	1	0
0	8:S	1	0
0	9:S	1	0
0	10:S	1	0

If you run this model, again you'll see the results exactly match those for model `S(a2)` built using the PIM approach:



Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{S(a2) - PIM}	15137.6610	0.0000	0.47626	1.0000	2	11.7038
{S(a2) - DM}	15137.6610	0.0000	0.47626	1.0000	2	11.7038
{S(a2-cohort) - PIM}	15143.6584	5.9974	0.02374	0.0498	9	3.6876
{S(a2-cohort) - DM}	15143.6584	5.9974	0.02374	0.0498	9	3.6876

We'll leave building an additional model `S(a2+cohort)` (i.e., a model with additive effects between age and cohort) to you as an exercise (hint: simply delete the interaction columns from the design matrix for model `S(a2-cohort)`).

So, we see that by treating different release cohorts as 'groups', we can use the known fate data type in **MARK** to handle staggered entry designs. Are there any other design types we can handle using known fate data? In fact, there are, but they involve using a different approach, based on treating known fate data in a live-encounter, dead-recovery context.

16.6. Known fate and joint live-dead encounter models

As noted earlier, the encounter history format for known-fate data is structurally similar to the classic `LDLD` format used for Burnham's live encounter-dead recovery analysis (Chapter 9). Recall that in that case, it is possible to observe an individual alive at the start of a particular interval (`L`), and dead at some point during the interval (`D`).

With a little thought, you might think that you could apply the live encounter-dead recovery model structure directly to known-fate data, if you simply fix the 'detection parameters' (r and p), and the 'fidelity parameter ' (F) to 1 (remember, for a known-fate data, we assume we know the fate of all individuals). With a little more thought, however, you might realize there is a complication – the live encounter-dead recovery model does not correctly handle the censoring of '`00`' `LD` pairs in a known-fate data. In the live encounter-dead recovery data type, the '`00`' is handled as an animal that was not detected as either alive or dead on this occasion. In a known-fate data, the '`00`' indicates that the animal was censored from the study. The distinction is made clearer in the following table, where we contrast the probability expressions, and interpretations, of the encounter history '`100010`' under the known-fate, and live-dead encounter models, respectively.

model	*probability*	*interpretation*
known fate	S_1S_3	tagged at occasion 1, censored for interval 2 (not detected, or removed for some reason), and re-inserted into the study at occasion 3.
live-dead	$S_1F_1S_2(1-p_2)S_3p_3$ $+S_1F_1S_2(1-p_2)(1-S_3)(1-r_3)$	(i) tagged at occasion 1, stays in sample, survives to occasion 2 but not encountered, survives to occasion 3, where it is encountered alive, not shot; (ii) tagged at occasion 1, stays in sample, survives to occasion 2 but not encountered, survives to occasion 3, where it is encountered alive, shot, but not recovered.

Clearly, the probability expressions differ considerably between the two model types. And, as such, you can't simply apply the live-dead encounter model to known-fate data without somehow accounting for the difference in how the '00' values in the encounter history are handled. Specifically, how can you 'tell' the live-dead model that a '00' means 'censored' and not either 'dead and missed', or 'live and missed'?

One way to handle this is to break up the encounter history and use a '-1 coding' – in other words, take the '10 00 10' encounter history and make it into 2 encounter histories as:

```
10 00 00   -1;
00 00 10    1;
```

Now, the live-dead model correctly handles the pair of encounter histories to allow the animal to be in the sample for the first interval, and then be removed from the sample. The animal is then re-injected back into the sample for interval 3. If all the r and p parameters are fixed to 1, and you also fix F to 1, then you will get the identical estimates of survival from the live-dead and known fate approaches.

To see that the preceding statement is true, first examine the probability of the first encounter history: $S_1 + (1 - S_1)(1 - r_1)$, which reduces to just S_1 because $r_1 = 1$. The probability of the second encounter history is $S_3 + (1 - S_3)(1 - r_3)$, which again reduces to just S_3. So, the product of these 2 encounter histories is identical to the probability of the original encounter history under the known fate model.

To make this 'trick' of splitting known fate encounter histories to allow censoring, let's consider a bit more complex example. Take the encounter history '10 10 00 10 11'. The known fate probability is just $S_1S_2S_4(1 - S_5)$. The split encounter history for live-dead coding looks like:

```
10 10 00 00 00  -1;
00 00 00 10 11   1;
```

The probability expression corresponding to the first piece is just $S_1F_1p_2(S_2+(1-S_2)(1-r_2))$, which reduces to just S_1S_2 because all F, p, and r parameters are fixed to 1. The second probability is $S_4F_4p_5(1-S_5)r_5$, which reduces to $S_4(1-S_5)$. The preceding might seem like a lot of work just to 'trick' the Burnham live-dead model into being able to handle known-fate data. Clearly, for 'typical' known-fate data, use of the known-fate data type in **MARK** is decidedly more straightforward (and, not surprisingly, why it's there in the first place). However, there are some situations where using the live-dead model is particularly helpful – we consider two such applications in the following.

16.6.1. Live-dead and known fate models (1) 'radio impact'

One of the most pressing questions with known fate data is 'What is the impact of the radio on the animal's survival?' A useful solution to this question can be obtained by marking some animals with non-intrusive tags. For example, one sample of ducks can be radio-marked, whereas a second can be banded with leg bands. Now, the data must be analyzed with a different model that incorporates the live detection probability p and the dead detection probability r.

The way to do this is to use the live-dead model, and specify 2 groups. The first group would consist of the radio-marked sample, where all the p, r, and F parameters are fixed to 1. The second group would consist of the leg-banded sample, where all the parameters are estimated. The power of this design comes into play when we compare a model with survival estimated separately for each group against the equivalent model but with survival estimated in common across both groups. The comparison of these 2 models provides a powerful test of the effects of the radios on survival. For a well-designed study, we might consider using a likelihood-ratio test between these 2 models to test the null hypothesis of no radio effect directly. Alternatively, we could use the Akaike weights to assign probabilities to which hypothesis we believe is most likely the truth.

16.6.2. Live-dead and known fate models: (2) 'temporary emigration'

The live-dead data type can also be used to estimate the fidelity (F) to a study area for known fate data. The approach is to code the `LD` pair as '`00`' for animals that leave the study area. That is, animals that leave the study area are not censored as if the radio failed, but rather included in the sample with `00` for periods when they are off the study area. Then, given that $p = 1$ and $r = 1$, F is estimated. So, consider what the probability would be for the encounter history '`10 10 10 00 00`' when $p = 1$ and $r = 1$ so that these terms are left out of the expression: $S_1F_1S_2F_2S_3(1-F_3)$. With F estimated, the only way to account for trailing `00` values is to have the animal emigrate. Remember that the Burnham joint live-dead data type assumes permanent emigration.

What if you want to model temporary emigration? The solution in this case is to use the Barker joint live-dead data type (see Chapter 9), where the parameter F' is the probability that an animal not available for capture (i.e., off the study area) returns to the study area. So consider the probability of the encounter history '`10 10 00 00 10`' with $p = 1$ and $r = 1$, along with no probability of sightings in between capture occasions (i.e., $R = 0$ and $R' = 0$): $S_1F_1S_2(1-F_2)S_3(1-F_3')S_4F_4'S_5$. The point here is that the Barker joint live-dead data type can also be used to estimate the temporary emigration probability from known fate data, and hence can also be used to assess the effects of radios on animals against a sample marked in a different fashion.

16.7. Censoring

Censoring appears 'innocent' but it is often not. If a substantial proportion of the animals do not have exactly known fates, it might be better to consider models that allow the sampling parameters to be < 1. In practice, one almost inevitably will lose track of some animals. Reasons for uncertainty about an animal's fate include radio transmitters that fail (this may or may not be independent of mortality) or animals that leave the study area. In such cases, the encounter histories must be coded correctly to allow these animals to be censored. Censoring often require some judgment.

When an animal is not detected at the end of an interval (i.e., immediately before occasion j) or at the beginning of the next interval (i.e., immediately after occasion $j+1$), then its fate is unknown and must be entered as a '`00`' in the encounter history matrix. Generally, this results in 2 pairs with a '`00`' history; this is caused by the fact that interval j is a `00` because the ending fate was not known and the fact that the beginning fate for the next interval ($j+1$) was not known. Censored intervals almost always occur in runs of two or more (e.g., '`00 00`' or '`00 00 00`'). See the example above where the history was '`10 00 00 11`'.

In this example, the animal was censored but re-encountered at the beginning of interval 4 (alive) and it died during that interval. It might seem intuitive to infer that the animal was alive and, thus, fill in the 2 censored intervals with '`10 10`' – this is incorrect and results in bias. The reason for this bias is because a dead animal is less likely to be encountered at a later occasion than if it lives. So, you have a biased sampling process – animals are mostly encountered because they are alive, and hence estimates of survival become too high if the '`00`' values are replaced with '`10`'.

Censoring is assumed to be independent of the fate of the animal; this is an important assumption. If, for example, radio failure is due to mortality, bias will result in estimators of $\hat{S}_i$. Of course, censoring reduces sample size, so there is a trade-off here. If many animals must be censored, then the possible dependence of fates and censoring must be a concern. In such cases, you probably should be analyzing the data with the live encounters-dead recovery data type, and explicitly estimate the p and r parameters.

16.8. Goodness of fit and known fate models

Consider a model where all the parameters are specific to both the time-interval, as well as the cohort (i.e., year marked and released). This is a fully-saturated model where there are as many unknown parameters as there are cells. Note, the saturated model always fits the data perfectly (by definition and design). The concept of a saturated model is necessary in computing model deviance. As discussed earlier in Chapter 5, the deviance of model j in the candidate model set is defined as

$$\text{Deviance} = -2\ln\big(\mathcal{L}_j(\theta)\big) - \big[-2\ln\big(\mathcal{L}_{\text{saturated}}(\theta)\big)\big]$$

Typically, for most data types, the saturated model contains many uninteresting parameters – its use is primarily heuristic, in allowing use to estimate the deviance of some less general model, relative to the saturated model.

Now, if sample size is large (i.e., there are no cells with small expectations), then the deviance is asymptotically χ^2 with df equal to (the number of cells in the saturated model) - (the number of estimable parameters in model j). OK, fine, this is the basis of the likelihood ratio test discussed earlier in Chapter 5. What does this have to do with GOF testing for known-fate data?

Well, the problem with known-fate data is this – for known-fate models where all individuals enter at the same time (or even with staggered entry data), the saturated model where each cohort has its own survival estimate for each occasion is a sensible model, and as such, there is no way to estimate the deviance of the saturated model from itself. Because the saturated model fits the data perfectly, there is no GOF test for classical known-fate data. In reality, this is the same with all models in **MARK** – we just assume (i.e., make an assumption) that some reduction of the saturated model to a biologically reasonable model is okay, and use this reduction to assess GOF.

To help you understand this point, consider a simple radio-tracking study where 100 radios are put on a single age/sex class for one occasion. The saturated model is the simple survival estimate based on the binomial distribution. There is only one data point, hence one degree of freedom, and that df is used to make the estimate of survival. Thus, it is fairly obvious that there is no GOF test available – to obtain a GOF test, we would have to assume a reasonable biological model that is reduced from the saturated model. This selection can be pretty arbitrary (obviously).

16.9. Known-fate models and derived parameters

Typically you are doing a known fate analysis to be able to estimate survival over an interval, say 1 year. However, you also want to know something about how survival changes within the year, or maybe because of censoring and radio failure problems, you want to include animals in the analysis that only appeared for a period of time within the year period. For example, you are doing a bear study where you have staggered entries and some radio failures or collars that dropped off that you have kept track of on a monthly interval. However, you are interested in estimating annual survival. How do you get an estimate of annual survival from 12 monthly estimates?

MARK provides derived parameter estimates that are the product of all the estimates for the intervals in the PIMs. So, suppose you have a 3-year study, where you want 3 annual estimates of survival, but you have 36 months of data. The clever way of setting up your analysis is to define 3 groups for the known fate model, each with 12 occasions (months), with the 3 groups corresponding to the 3 years of interest. Then, when you examine the derived parameter estimates, you will find 3 estimates, representing the 3 years. Variances and covariances of the derived parameters are computed with the Delta method (Appendix B).

Derived parameter estimates can be used in model averaging and variance components analyses, so you further have all of the power of these methods available for your analysis of annual survival rates.

Part of the 'art' of how to set up the known fate data type is whether attribute variables should be incorporated as groups or individual covariates. Derived parameter estimates are a function of the individual covariates used to compute them, so whether age in the black duck example is treated as a group or an individual covariate won't make a difference in the estimates. However, if age is handled as a group variable, the derived estimates are clear. To get derived estimates when age is an individual covariate means that you must specify individual covariate values to obtain the correct estimates.

16.10. Known-fate analyses and 'nest success models'

Suppose you want/need to estimate the survival of radio-tracked animals when the animals are not monitored in discrete intervals, as generally required by the known fate data type. Consider that such data are no different than a set of (say) nests, where all the nests are not visited on the same day. As such, you could apply a 'nest success model' to the data – in such a model, the daily survival rate is estimated for each day of the study based on the sample of animals available on that day, and the exact day of death is not required (just as the exact day that a nest was destroyed is not known). We call these kinds of data '*ragged telemetry data*' because the sampling scheme is ragged, but useful estimates can still be obtained. Nest success analysis is the subject of our next chapter.

16.11. Summary

Known-fate models are a very important model type – most commonly applied in situations where individuals are marked with radios (i.e., radio telemetry studies). The presence of a radio makes is feasible (under usual circumstances) to determine the 'fate' of the individual: is it alive, or dead? Present, or absent? And so on. Although the assumption that detection and reporting probabilities are both 1.0 simplifies aspects of the modeling considerably, a number of complex, elegant approaches to handling known-fate data are possible – especially when known-fate data are combined with data from other sources.