



Comparing different models

In this chapter we will make our first expansion of our analysis of the European Dipper data set. We will again concentrate on just the males (MALES.SUR). However, we will now learn how to use SURGE to compare and contrast different models. As noted in considerable detail in Lebreton *et al.* (1992), the analysis of mark-recapture data rests very heavily on model selection. In the process of selecting among different models, we will often be accomplishing 2 separate but equally valuable tasks. First, by selecting the model most appropriate to the data, we can then derive the most meaningful estimates of both survival and recapture rates. Second, model selection is, in fact, hypothesis testing.

In this first example, we will compare just 2 different subsets of models: a model where either survival or recapture (or both) varies with time, or a model where either survival or recapture (or both) are constant with respect to time. In fact, in this family of models, there are 4 different sub-models. They are presented in Fig. 4.1, using the notation suggested in Lebreton *et al.* (1992).

Model	Explanation
ϕ_t, p_t	both survival and recapture time dependent
ϕ, p_t	survival constant, recapture time-dependent
ϕ_t, p	survival time-dependent, recapture constant
ϕ, p	both survival and recapture constant

Fig. 4.1

- In the following, we will go through the steps in fitting each of these 4 models to the data. In fact, the first model (ϕ_t, p_t) is the same model we fit in Chapter 3. Remember, it is also referred to as the Cormack-Jolly-Seber (CJS) model. However, we will go through each of the steps again. Repetition is the most effective learning tool. We will omit some of the more detailed explanations for each step, since they were already covered in Chapter 3.
- However, before we begin, we must emphasize that before you actually use SURGE to compare different models, you need to first confirm that your “starting model” (generally, the most parameterized model - this will be covered in considerable detail later) adequately fits the data. In other words, you MUST conduct a goodness-of-fit (GOF) test for your “starting model”. GOF testing is discussed briefly in the Appendix. For much of this guide we’re going to assume that the “starting model” does fit the data.

MODEL ϕ_t, p_t

- Start SURGE, and answer the various prompts from the “data input screen”. First, enter the name of the output file. For this example, give the output file the name MODELS4.LST (4 models, Chapter 4).
- Next, give SURGE a title appropriate for your output. For this first analysis, we’re analyzing the fully time-dependent CJS model, so we might choose to use “Phi(t),p(t)” for our title. The “Phi” and “p” sections refer to survival and recapture, respectively. Inside the brackets, the letter “t” indicates time-dependence.
- Again, for this example, we’re only going to analyze 1 data set (MALES.SUR). So, we enter the number 1, and the name of the file (Fig. 4.2). These are exactly the same steps as were covered in Chapter 3, pp. 3-1 to 3-2. If you are having any difficulties, re-read those pages.



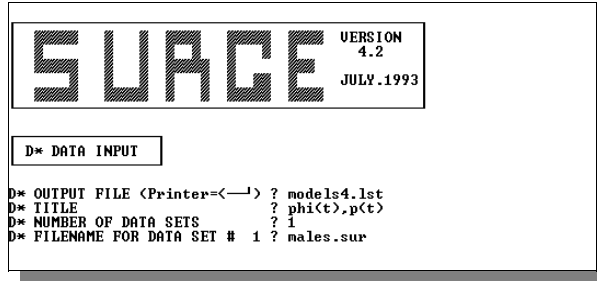


Fig. 4.2

- Again, once you've completed the data input screen, you will immediately be presented with the first of the two "model specification" menus (Fig. 4.3).

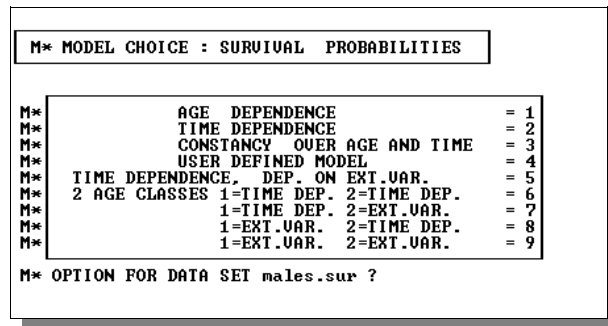


Fig. 4.3

- Now, recall from Chapter 3 that at this stage we are going to tell SURGE which model we want to fit to the data. Again, SURGE gives us several "built-in" options we can select from the menu presented in Fig. 4.3. Later on, we'll see how we can use SURGE to fit any model we design, using a "user-defined model" - choice 4 from the menu. However, for the moment, we're going to fit a "time-dependent" model. Remember, for this first analysis, we are fitting model ϕ, p_t .
- Now, we could simply look back at Chapter 3 to see what we did there,

but it is a good idea to start thinking about each step (rather than simply memorizing a sequence of keyboard commands), and looking closely at what SURGE presents to you on-screen (Fig. 4.3). In this menu, we see several choices for different "built-in" models we can fit. If you read the model descriptions, you should be able to quickly identify which choice you need to make. Obviously, choice 1 is for a age-structured model - in this case, you would select choice 1 if you wanted to model the data with different age classes (i.e., where survival varies among individuals of different ages - Chapter 7)). Clearly, since we want to fit model ϕ, p_t , this would not be the correct choice.

- The next selection, choice 2, is for a "time dependent" model. While this is in fact the correct choice, let's briefly examine the other choices in the menu. Of the remaining 7 choices (3-9), the most commonly used are 3 (constant), 4 (user-defined), and 6 (2 age-classes with time dependence in both age classes). These latter 2 choices will be examined in more detail in later chapters. For the moment, we enter the number 2, and hit the <enter> key.
- It is important to remember that SURGE treats survival and recapture separately, and sequentially. Thus, now that we've told SURGE that we want to fit a time-dependent model, you will immediately be presented with the choice menu for modeling recaptures. As you may remember from Chapter 3, the physical appearance of this menu (Fig. 4.4) is identical to the one you just worked with for modeling survival (Fig. 4.3) - only the title at the top differs.

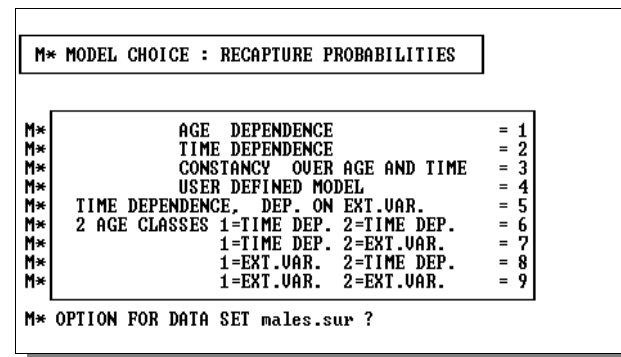


Fig. 4.4

- Again, in this first model, we are fitting the fully time-dependent CJS

model - time dependency in recapture as well as survival. Thus, we again select choice 2 from the menu. This is followed again by the question asking how many time-specific parameters you want in the recapture model. Again, for this model we want full time-dependence (the default), so we simply hit the <enter> key.

- Once we have completed our model choices for survival and recapture, we are asked if we want to add any constraints (Fig. 4.5). We also have the option at this stage to either display the models we've just selected, using SURGE's way of representing the parameter structure (e.g., p. 3-7), or to jump back and re-select the models we want to fit. For the moment, we want to bypass this option (i.e, we don't want to add any constraints, and we will skip looking at the model structure). Since the default is 0, we simply hit the <enter> key.

```

C* CONSTRAINING PARAMETERS
C* There are already 0 constraints
C* HOW MANY MORE CONSTRAINTS? <display model=-1; back to M=-2> :
  
```

Fig. 4.5

- SURGE then asks you if you want to fix any parameters (Fig. 4.6). In this analysis, we don't want to fix any parameters, so we bypass this option by simply hitting the <enter> key (the default is to fix 0 parameters).

```

F* FIXING PARAMETERS
F* HOW MANY PARAMETERS DO YOU WANT TO FIX?: _
  
```

Fig. 4.6

- SURGE will then present you the initial parameter estimates for both survival and recapture, and ask you if you want to change these initial values. The default is 0, so we simply hit the <enter> key to bypass this option.

- SURGE then asks you if you want to change abandon the logit scale transformation (the default) to “no scale”. In most cases, we will want to use the default logit scale, so we simply hit the <enter> key again, accepting the default.
- Now, we are ready to run the analysis. The “analysis screen” (Fig. 4.7) allows you to either proceed with the calculations, or go back to any of the intermediate steps (see pp. 3-5 to 3-6 for more detail). At this point, we simply want to go ahead and run the analysis. We could either enter 0 (for no display, or “-X”, to display the results every “X” iterations (where X=1 or X>4), or X to display detailed results every “X” iterations.. We'll enter -5, to have the results printed onscreen every 5 iterations.

```

O* OUTPUT OF ITERATIVE CALCULATIONS
O* DISPLAY RESULTS EVERY X ITERAT.<0=no display>
  <back to M =-2; back to C =-3; back to I =-4> : -5_
  
```

Fig. 4.7

- We also answer “no” to the query about the variance-covariance matrix. Once the program has completed, you will be presented with the final screen, at which point you can choose to do another analysis, or quit SURGE. Since we want to do more analyses (i.e., we want to fit the other 3 models listed in Fig. 4.1), we do NOT want to quit SURGE. Since we want to fit another, different model to the same data, we enter choice 2, to go back to the model choice menus (Fig. 4.8).

```

D* DATA INPUT 1
M* MODEL CHOICE 2
C* CONSTRAINING PARAMETERS 3
I* INITIAL VALUES AND LOGISTIC TRANSF. 4
QUIT SURGE 5
OPTION # ?
  
```

Fig. 4.8

MODEL ϕ, p_t

- After hitting the <enter> key, you will be asked for a new title. SURGE assumes that if you want to do a new analysis, you'll want a new title (rather obvious). We are now going to fit model ϕ, p_t , so we enter the title "Phi(),p(t)".



Why are we now testing model ϕ, p_t , or any of the other models in Fig. 4.1? As mentioned at the beginning of this chapter, model selection has 2 important purposes: (1) finding the most "parsimonious model" allows us to derive the most robust estimates of both survival and recapture rates; (2) model selection tests hypotheses. For example, as we will see later in this chapter and especially in Chapter 5, comparison of the "fit" of model ϕ, p_t with model ϕ, p_t is a formal test of whether or not survival varies with time, or is constant over time. For the moment, we're concentrating on the "mechanics" of building different models.

- As soon as we enter the new title, we are presented with the model selection menu (Fig. 4.3). However, we now need to consider which choice is appropriate from this menu. Since we are fitting model ϕ, p_t we want to fit a "constant" survival model. The first menu relates to the survival rates, so we select choice 3 from the menu.
- We are then presented with the same menu, but now applied to the recapture rates. In this case, we want to fit a time-dependent model, so we select choice 2 from the menu.
- Once we have made these 2 choices, we're again presented with the "constraints menu" (Fig. 4.9). Again, this menu really gives us the options of doing 3 different things: we can either apply a constraint to the model, or we can either look at the model structure (for example, to confirm that are in fact using the correct model in our analysis), or to go back to the model selection menus. Previously, we simply skipped this step. However, since we changed the model, let's use this opportunity to look at the parameter structure of the model we've just chosen. To do this, we enter -1, and hit the <enter> key.

```
C* CONSTRAINING PARAMETERS
C* There are already 0 constraints
C* HOW MANY MORE CONSTRAINTS? <display model=-1; back to M=-2> : -1_
```

Fig. 4.9

- SURGE will now present you with a schematic representation of the parameter structure of the models you've selected, first for survival, and then for recapture. You've already seen an example of the representation SURGE uses in Chapter 3 (p. 3-7). SURGE uses a triangular matrix to indicate the parameter structure of the model you've chosen. For model ϕ, p_t the parameter structure for survival (which is constant) is shown in Fig. 4.10a:

```
PARAMETER STRUCTURE FOR DATA SET males.sur:
  1  1  1  1  1  1
    1  1  1  1  1
      1  1  1  1
        1  1  1
          1  1
            1
strike <— to continue_
```

Fig. 4.10a

- For recapture rates, which are time-dependent, the parameter structure is shown in Fig. 4.10b.

```
PARAMETER STRUCTURE FOR DATA SET males.sur:
  2  3  4  5  6  7
    3  4  5  6  7
      4  5  6  7
        5  6  7
          6  7
            7
C* HOW MANY MORE CONSTRAINTS? <display model=-1; back to M=-2> : _
```

Fig. 4.10b

- In Chapter 3, we briefly introduced the logic SURGE uses in structuring these matrices, and what the numbers actually represent. Since understanding how this scheme works is essential to using some of the more sophisticated features of SURGE, we will now examine these matrices in further detail.

Model structure matrices and parameter indexing in SURGE

- Much of this discussion refers to the text presented in Lebreton *et al.* (1992), particularly pages 74-76, and in Table 7 in that paper. It will be useful to have these available for reference.
- 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 is important to understanding the structure of the parameter matrices SURGE uses.
- Consider our first model, the CJS model with full time-dependence in both survival and recapture rates. Let’s assume there are no age effects (say, for example, all individuals are marked as adults). In Chapter 3, we represented the parameter structure of this model as shown below in Fig. 4.11:

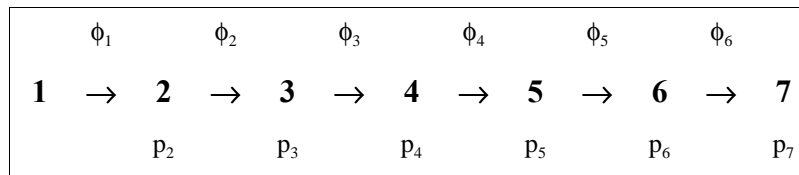


Fig. 4.11

- 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 in Fig. 4-12, and in Table 7 in Lebreton *et al.* (1992). Consider especially Table 7A (where survival is assumed to vary with time, but not cohort) and 7C (where survival varies with cohort but not time).

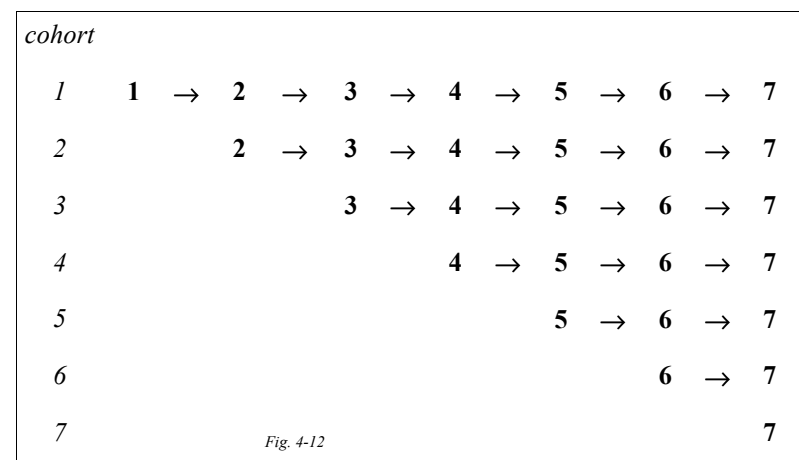


Fig. 4-12

- The occasions are numbered from left to right, starting with occasion 1. Survival rate 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 sample captured are unmarked, and are newly marked and released at occasion 2. These animals comprise the second cohort.
- 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 in Fig. 4.12), but over a given interval (i.e., within a column in Fig. 4.12), survival is the same over all cohorts (i.e., over all rows). Here is the structure for this model (Fig 4.13).

cohort	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6
1	1 → 2	2 → 3	3 → 4	4 → 5	5 → 6	6 → 7
2		2 → 3	3 → 4	4 → 5	5 → 6	6 → 7
3			3 → 4	4 → 5	5 → 6	6 → 7
4				4 → 5	5 → 6	6 → 7
5					5 → 6	6 → 7
6						6 → 7

Fig. 4-13

- 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 SURGE does to generate the parameter or model structure matrix is to reproduce the structure and dimensions of Fig. 4.13, 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, Fig. 4.13 is represented by a triangular matrix of the numbers 1 to 6 (for the 6 survival rates). This is shown in Fig. 4.14.

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

Fig. 4.14

- Again, Fig. 4.14 represents the way that SURGE “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 by 7 as shown in Fig. 4.12 and 4.13. This is because Fig. 4.14 shows the model structure for the survival rates - there are 7 capture occasions, but only 6 recapture intervals.
- This representation will occur in the output file, and on-screen if you choose the appropriate option on the “constraints menu”. Perhaps most importantly, though, this format is the way SURGE keeps track of model structure and parameter indexing. It is **essential** that you understand the relationships presented in Fig. 4.12 through 4.14, since we will make considerable use of this format when we learn how to create a “user-defined model”. In fact, as you will see, in creating a



user-defined model, all we will be doing is creating this triangular matrix of numbers indexing the various parameters of our model. Thus, it is *important for you to take the time now to understand the connections we ve made*. A few more examples will help make them clearer.

- Let's consider the case where survival is constant over time. again, we'll assume that there is no variation in survival among cohorts (i.e., when an animal is marked and released does not significantly influence its survival). Again, for the moment, we'll ignore recaptures, and consider just survival.
- A model with constant survival over intervals and no differences among cohorts is shown in Fig. 4.15.

cohort	ϕ_1	ϕ_1	ϕ_1	ϕ_1	ϕ_1	ϕ_1	ϕ_1
1	1 → 2	2 → 3	3 → 4	4 → 5	5 → 6	6 → 7	
2		ϕ_1	ϕ_1	ϕ_1	ϕ_1	ϕ_1	
3			ϕ_1	ϕ_1	ϕ_1	ϕ_1	
4				ϕ_1	ϕ_1	ϕ_1	
5					ϕ_1	ϕ_1	
6						ϕ_1	

Fig. 4.15

- Within each cohort (row), and among intervals, there is one common survival rate, ϕ_1 . As with the preceding example, what SURGE does to

generate the parameter or model structure matrix is to reproduce the structure and dimensions of Fig. 4.15, replacing the ϕ_1 values with the numerical value of the subscript i . Thus, Fig. 4.15 is represented by a triangular matrix 1's. This is shown in Fig. 4.16 (below).

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

Fig. 4.16

- 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 number (indices) change in value from left to right, then survival 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. Consider what the structure would look like where survival was constant over time, but varied among cohorts. This model is shown in Fig. 4.17 (below).

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

Fig. 4.17

- Now, let's reconsider the model we were working with for the male European dipper data. The model we just fit was ϕ, p_t . In other words, constant survival over time, and time varying recapture rates.
- From the preceding examples, you should now understand that the structure for the survival model is

1	1	1	1	1	1
	1	1	1	1	1
		1	1	1	1
			1	1	1
				1	1
survival					1

and for recapture rate is

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

- However, if SURGE used the models as written above, there would be a problem, since the index 1 occurs in both the survival model and the recapture model. Thus, SURGE simply starts "counting" the indices for the recapture model starting with the first digit after the highest digit for

the survival model. Thus, our model ϕ, p_t would have the following parameter structure for survival and recapture respectively:

1	1	1	1	1	1
	1	1	1	1	1
		1	1	1	1
			1	1	1
				1	1
survival					1
2	3	4	5	6	7
	3	4	5	6	7
		4	5	6	7
			5	6	7
				6	7
recapture					7

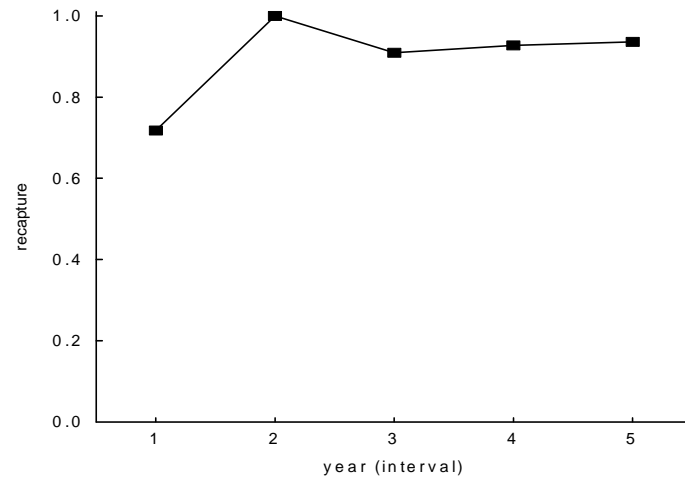
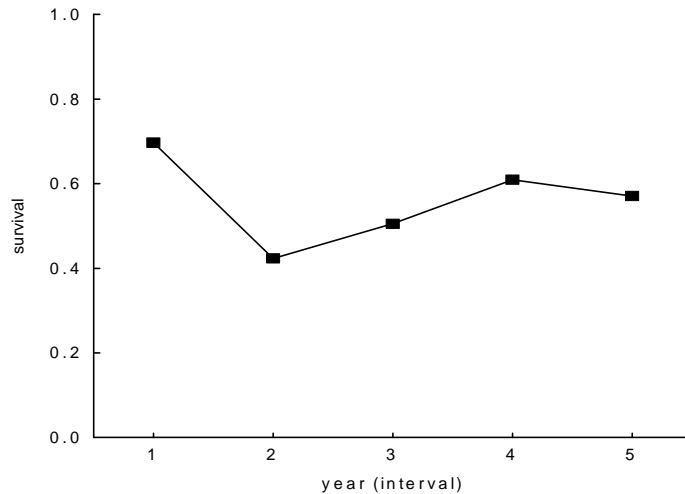
- Recall that these correspond to what SURGE prints on-screen when you ask it to display the models from the "constraint menu" (Fig. 4.10a and 4.10b).
- It is worth going over this again, and studying Table 7 in Lebreton *et al.* (1992); we will examine model structure again in more detail later on, but it is always worth reviewing important concepts frequently.
- Now, let us continue with our analysis. Again, for this model, we do not want to add any constraints, and do not want to fix any of the parameters. SURGE will then present you with its starting value estimates for both survival and recapture. We do not want to change the starting values for either. We also do not want to change the default logit transformation. Once we have responded to these queries, SURGE will again present us with the final screen, which will let us either proceed with the calculations, or go back to any of the intermediate steps. We'll enter "-5" to have the summary results printed onscreen every 5 iterations. We'll also answer "no" to the query about printing the variance-covariance matrix.
- The "final screen" (i.e., Fig. 4.8) allows us to either do another analysis, or quit. We still have 2 more models to run (from Fig. 4.1 - models ϕ, p - time variation in survival, but constant recapture rate, and ϕ, p - where both survival and recapture are constant over time.



- At this point, you should have an intuitive sense for what we do next. Since we want to “run another model”, we would pick choice 2 from the “final menu”, which leads back to the model choice menus (after first prompting you for a new title). You would pick the appropriate selections from the “model choice menus” for both survival and recapture, and proceed as for the preceding 2 models. Then, once you have run all four models, you would pick choice 5 from the “final menu” to exit program SURGE. You’re done! Well, at least you’re done with SURGE. Now all that remains is to analyze the output, which has been sequentially appended to the file you told SURGE to use right at the beginning (in this example, we used the file MODELS4.LST).
- Once you’ve exited SURGE you’ll be staring at the ubiquitous and (to non-DOS or UNIX users) annoyingly uninformative DOS prompt. In you’re using WinSURGE, the Window will close and you’ll be staring at some version of the Windows desktop (which, depending on who you ask, may be little more informative than the DOS or UNIX prompts). Unless you told SURGE to do otherwise, the output file MODELS4.LST will be in the current directory, so all you’ll have to do is browse the output (or print it).
- One convenient feature of SURGE is that the output is nicely formatted to fit into 80 columns. This is convenient in two ways. First, even the most outdated monitor can handle 80 columns, so you’ll be able to see the entire width of the file onscreen if you browse it. There are a large number of programs (which will remain anonymous for the moment) which use >80 column formatting, which makes browsing their results onscreen cumbersome at best. Further, the 80-column formatting of the SURGE output file makes it possible to use one of the many shareware/freeware utilities for printing ASCII text in landscape orientation on most standard HP-compatible laserprinters.
- Let’s look at the output from our analyses. The file MODELS4.LST will contain the results from each of our 4 models (Fig. 4.1), appended in sequence. As noted in Chapter 3, SURGE uses a very consistent output format - in essence, each part of the output file looks the same for each of the 4 models. Only the title, and the actual numbers, differ. As such, you’ll be able to navigate your way around the output with only a bit of practice. The structure of the output file was presented in Chapter 3, and we will dispense with most of the details here. For the moment,

we need only remember that each section of the output file starts with the title. The title is followed by the matrix representation of the survival and recapture models, respectively. These matrices are then followed by the model deviance. If there was a preceding model, then the difference in deviance between the current model and the one immediately preceding it in the output is also printed. This is simply a convenience - it is only saving you the task of calculating the difference in model deviance between 2 successive models. Then, if you are using WINSURGE, you are presented with an estimate of the number of parameters, and the model AIC. These values are followed by parameter estimates themselves, along with estimates of the 95% confidence interval and the standard errors.

- Before we tabulate 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. Often, a good 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). 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 analogous to polynomial regression (and other linear models) - model fit increases with more parameters, but at the cost of reduced precision of the parameter estimates. As we will discuss, the “best” model (for our purposes) is one which balances “fit” and “precision”. However, it is because it fits the best that 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 $\phi_t p_t$ - the CJS model. The parameter estimates (without the standard errors) for ϕ and p are plotted on the following page.
- Note that in these figures we do not include all 6 estimates that SURGE prints 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 SURGE has done is estimate the square-root of the combined probability $\phi_6 p_7$ (which Lebreton *et al.* (1992) refer to as β_7). As discussed in detail in Lebreton *et al.* (1992), 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 the β_7 term is not comparable to either survival or recapture rates separately, it is excluded from our plots.

- Of course, if you're using WinSURGE, the output listing already 'told' you that parameters 6 and 12 are not separately identifiable (i.e., cannot be estimated separately). However, as we've mentioned before, we do not favour slavish reliance on the ability of the software (be it SURGE or any other application) to determine the number of estimable parameters - you first need to develop an understanding of how it is done from first principals.
- From the plot of the survival estimates, while there is no clear pattern, there appears to be a marked reduction in survival in the second and third time intervals. In terms of recaptures, other than the first year, the recapture rate seems to be fairly consistent, and high. Again, we use these plots as aids only - however, as we will see later on, plots without standard errors can be misleading.
- Formally, we determine the "best model" using the statistical formalism described in detail in Lebreton *et al.* (1992). Basically, the goal is to find the most parsimonious model - the model which is consistent with the data, while using the fewest numbers of parameters.
- In this example, we tested 4 models. You should note that in this case, we arbitrarily decided to test these models. They were not chosen with any biological hypothesis in mind. This is important - generally, model selection should be approached in the context of biological hypothesis testing. For any given data set, there are a VERY large number of possible models. Deciding which models to test should reflect biological purposes. This will become more obvious as we further explore the Dipper data set. For the moment, though, we'll simply illustrate the "mechanics" of the model testing procedure.
- In the following table, we list the model, the model deviance, the number of parameters estimated in the model, and the *Akaike Information Criterion* ($AIC = \text{model deviance} + (2 \times \text{number of parameters})$).

model	deviance	# parameters	AIC
ϕ_t, p_t	313.080	11	335.080
ϕ_t, p_t	316.117	7	330.117
ϕ_t, p	315.494	7	329.494
ϕ, p	318.494	2	322.494

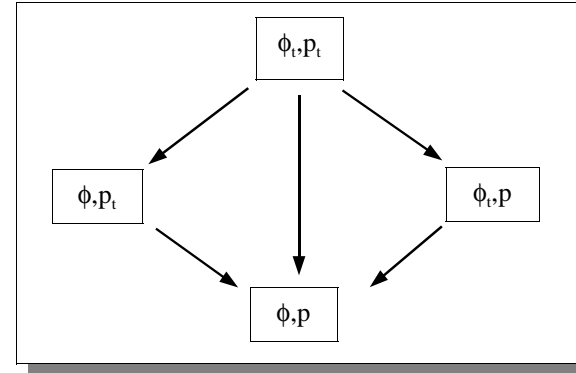


Fig. 4.18a

- Before we go into the specific details of this table, consider a few basic points. First, note that the most parameterized model (ϕ_t, p_t) has the lowest model deviance. This must be true - the more parameters in the model, the greater the “model flexibility”, and the better the model will be able to fit the data. This is analogous to a linear regression model with more parameters having a higher R^2 than a model with fewer parameters. However, the “cost” of more parameters is reduced precision of the parameter estimates. This is key to an intuitive understanding of the AIC - it “penalizes” the better fit of the more parameterized model(s) for the reduced precision of the estimates themselves. Thus, we note that even though model ϕ_t, p_t has the lowest deviance, it also has the greatest number of parameters and the highest AIC value. In contrast, the model deviance for model ϕ, p is the greatest (fits the least well), but because it uses only 2 parameters, it in fact has the lowest AIC of the 4 models. Thus, because it achieves the best compromise between fit and parsimony, the model which assumes that both survival and recapture rates are constant over time is a “better” model than any of the the other three models.
- However, while we can use AIC as an omnibus approach to model selection, it does not easily allow formal testing of the statistical significance of the differences among models (although this is likely to change in the near future). Fortunately, this can be done for nested models using the *likelihood ratio test* (LRT).
- The first step is to determine which models are nested. Consider Fig. 4.18a.

- In Fig. 4.18a, nested models are connected by the arrows. The direction of the arrows leads from a given model to the model nested within it.



Why aren't models ϕ_t, p_t and ϕ_t, p nested? The easiest way to resolve which models 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 I eliminated one or more of the factors from model A? If so, then model B is “nested” within model A. Consider our test models. If we start with model ϕ_t, p_t (model A), we want to know if model ϕ_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 can see easily that if we eliminate “time” from capture in model A, that model A is transformed into model B. Thus, we can say that model B (ϕ_t, p) is nested within model A (ϕ_t, p_t), as indicated in Fig. 4.18a. However, now compare models ϕ_t, p_t and ϕ, 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.

- Any two nested models can be compared statistically using a likelihood ratio test. Recall from Lebreton *et al.* (1992) that, 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.
- We re-draw Fig. 4.18a, showing the difference in deviance among nested models, and the difference in the number of parameters (Fig. 4.18b). The significance of this difference is estimated from any standard χ^2 table. 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 significant to variation in the data - in other words, we are formally **testing the significance of a particular variable in the model.**

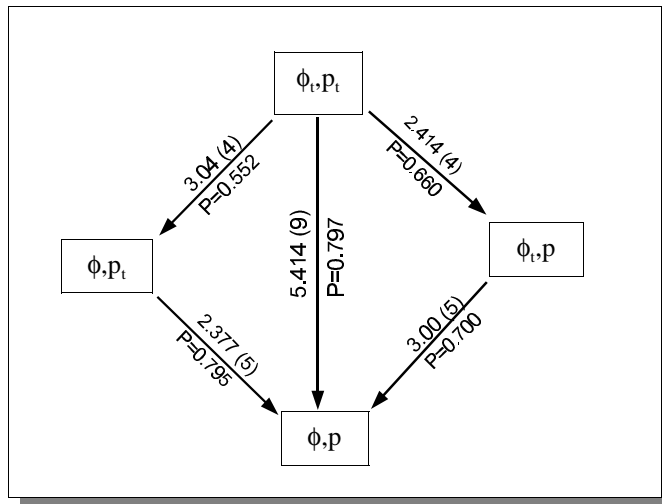
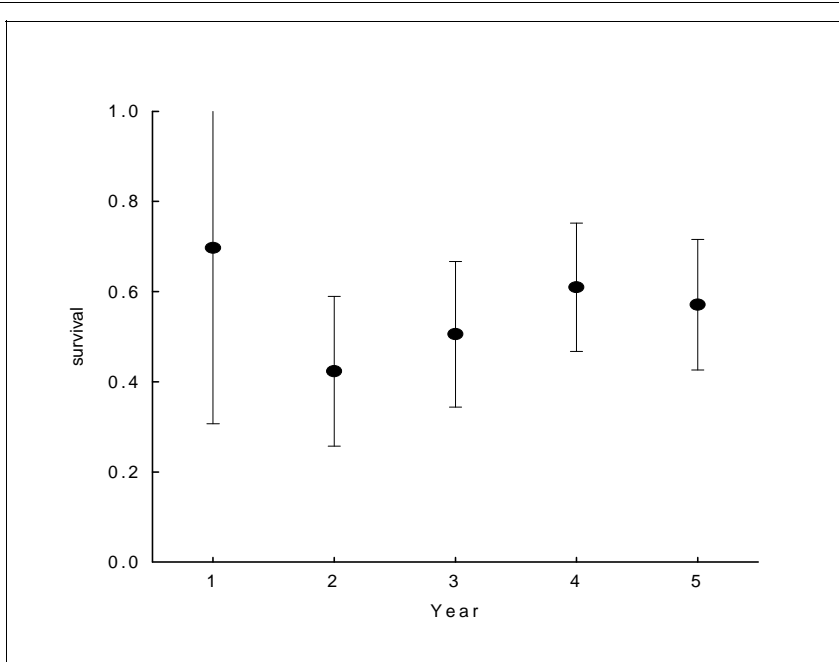


Fig. 4.18b

- As we can see from Fig. 4.18b, there is no significant difference in model fit (difference in deviance) between the most parameterized model (the CJS model ϕ_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).

- However, before we go any further, how else can we interpret these results? More specifically, what hypotheses have we just tested? Consider the test of the CJS model ϕ_t, p_t versus ϕ, p . In this comparison we are formally testing the following “biological 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 (ϕ_t, p_t) to one where it doesn't (ϕ, p). Since the fit of these two models is not significantly different ($\chi^2=3.05, df=4, P=0.552$ - Fig. 4.18b), we can state that “there is no significant annual variation in survival”. This is a formal, statistical result, that has a “meaningful” biological interpretation.
- From Fig. 4.18b, and from the AIC values tabulated on p. 4-11, we see that the most parsimonious model overall is model ϕ, p - i.e., the model where both survival and recapture rate are constant over years.
- How do we interpret the apparent variation in our CJS estimates plotted on p. 4-10? In those plots, we saw that recapture rate was more or less “flat”, or constant (apart from the first value). This appears to be consistent with the conclusions from our model selection results. However, the plot of the survival estimates does not appear at all to fit with our conclusions - the plot seems to “suggest” significant annual variation in survival. In fact, we noted earlier that the estimates of survival varied from 0.7 to 0.42. Why do the results of our “statistics” appear to be so different from our “visual intuition”, based on this plot.
- The reason for the difference is that the plot on p. 4-10 did not include the 95% confidence limits. What would we have concluded if we had plotted them?
- As you can see from the figure on the following page, the 95% CI is very wide, such that there is substantial overlap in the range between the lower and upper bounds for all 5 estimates. Therefore, it is not surprising that we were unable to detect significant annual variation in survival - there was insufficient precision in the estimates to give us sufficient power. Thus, while it can generally be useful to visually examine the estimates from the most parameterized model, your “interpretation” may not be valid unless you also take into account the



relative precision of your estimates. The conclusion we might tentatively make in this case is that, although there appears to be some evidence for variation in survival over years, we were unable to detect it.

- Using the same logic, we have also formally tested the hypothesis that there is significant annual variation in recapture rates - our results indicate no evidence of significant annual variation in recapture rates.
- So, in some ways we're finished with this example analysis of the male Dipper data, at least in terms of the 4 models we've tested. However, before we proceed any further you might have two lingering questions (at least!).
 1. how did we derive the number of parameters for each model, listed in the table on p. 4-11? Knowing the number of parameters is very important since it is necessary for both the likelihood ratio tests (where appropriate), and calculation of the AIC.

2. how do we decide which 2 models to compare to test a specific hypothesis about one or more factors? For example, is the comparison of model ϕ_t, p_t versus ϕ, p_t the most appropriate one to test our hypothesis of significant annual variation in survival?

- Let's consider each of these questions in turn. They are both important.

➡ Counting Parameters

- Perhaps the most frustrating aspect of using the old DOS-based version of SURGE (in its present form) expressed by new users is the fact that it requires you to do several things manually (i.e., SURGE doesn't do them for you). Some of these "manual tasks" are simple - for example, while it may be a nuisance having to manually calculate the difference in deviance between 2 models, it is not overly challenging!
- However, in some cases, counting parameters is more difficult, for two reasons. First, the structure of the model itself determines the number of parameters which may be potentially estimated. This number includes individually identifiable parameters, and those which are not individually identifiable (the compound β values noted in Lebreton *et al.* 1992). Second, there are times when either the sparseness of the data, or other "structural problems" makes it impossible to actually estimate a particular parameter, even if it is theoretically possible to estimate it independently. While modifying SURGE to handle the first problem is fairly straightforward, the second problem is more difficult. If a parameter is truly non-identifiable because of "problems" with the data, then you should not include it in the total for that model. The "trick" is determining when this problem occurs. Such "non-identifiability" is often (but not always) characterized by estimates of 0 or 1, and standard errors that seem extremely small or extremely large. However, presence of such estimates does not always mean there is a problem - see for example the estimate of the recapture rate on the second occasion for the Dipper example in Chapter 3 The estimate for p_2 was 1.0000, with a SE of $\ll 0.00001$. Does this estimate (i) indicate a problem with the data, or (ii) is the recapture rate at this occasion was actually 1.0, and the extremely small standard error simply a function of the back-transformation from the logit scale

(estimates of the SE of estimates near 0 or 1 are known to be biased when using a back-transform from a logit scale - see Lebreton *et al.* 1992)? While there is perhaps no perfectly unambiguous way to make this determination, there are a couple of possible approaches you can use with SURGE to help make this determination (briefly mentioned in Chapter 9).

- The newer WinSURGE program includes assistance in parameter counting - it provide counts of the number of estimated parameters, the number of these parameters which are estimated on the boundary (see above), and the number of structurally non-identifiable parameters (i.e., which parameters are not separately identifiable because of the structure of the model). However, we note that while this is a very useful addition, it is not a “perfect solution” to this problem (see Chapter 9). In some cases, you may still have to do it by hand.
- While you might regard this as a nuisance, it is, in fact, a “useful” one, especially for the beginner, since the very fact that you have to figure out the number of parameters on your own will force you to understand the structure of your model. The more that the software does for you, the greater the tendency to simply treat it all as a “black box” - generating “results” without a thorough understanding of the underlying analysis. Of course, we’re the first to admit that if you have a ‘good data set’, and have a good ‘parameter counting program’ (like WinSURGE), there is no reason not to use this feature of the program.
- 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 rate”. This fraction, also known as the “return rate”, is still widely found in the literature. Unfortunately, naive use of return rate poses real problems, since, in fact, it does not necessarily estimate survival rate 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 rate, 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 rate.
- 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 rate is 50/100 or 0.5. However, does this really mean that “survival” is 50%? Not necessarily. What it means is that $100 * \phi_1 p_2 = 50$, or $\phi_1 p_2 = 0.5$. 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 rates 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? Under this sampling scheme, at least 4 capture histories are possible for individuals marked on the first occasion:

<i>capture history</i>	<i>probability statement</i>
111	$\phi_1 p_2 \phi_2 p_3$
101	$\phi_1 (1-p_2) \phi_2 p_3$
110	$\phi_1 p_2 (1-\phi_2 p_3)$
100	$1-\phi_1 p_2-\phi_1(1-p_2)\phi_2 p_3$

- 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 second 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 rate, since recapture rate 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 rates ϕ_1 and p_2 respectively. Specifically,

$$\frac{N_{111}}{N_{101}} = \frac{\hat{p}_2}{1 - \hat{p}_2}$$

- Of course, SURGE 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 rate 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 rate 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 . Thus, we can re-write our table, and the probability statements, as:

<i>capture history</i>	<i>probability statement</i>
111	$\phi_1 p_2 \beta_3$
101	$\phi_1 (1-p_2) \beta_3$
110	$\phi_1 p_2 (1-\beta_3)$
100	$1-\phi_1 p_2-\phi_1(1-p_2)\beta_3$

- 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. Clearly, 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 Dipper example, there are 7 occasions, so our table of saturated capture histories, and substituting $\beta_7 = \phi_6 p_7$, the associated probability statements, would look like :

<i>Capture history</i>	<i>probability statement</i>
1111111	$\phi_1 p_2 \phi_2 p_3 \phi_3 p_4 \phi_4 p_5 \phi_5 p_6 \beta_7$
0111111	$\phi_2 p_3 \phi_3 p_4 \phi_4 p_5 \phi_5 p_6 \beta_7$
0011111	$\phi_3 p_4 \phi_4 p_5 \phi_5 p_6 \beta_7$
0001111	$\phi_4 p_5 \phi_5 p_6 \beta_7$
0000111	$\phi_5 p_6 \beta_7$
0000011	β_7

<i>Capture history</i>	<i>probability statement</i>
1111111	$\phi_1 p \phi_2 p \phi_3 p \phi_4 p \phi_5 p \phi_6 p$
0111111	$\phi_2 p \phi_3 p \phi_4 p \phi_5 p \phi_6 p$
0011111	$\phi_3 p \phi_4 p \phi_5 p \phi_6 p$
0001111	$\phi_4 p \phi_5 p \phi_6 p$
0000111	$\phi_5 p \phi_6 p$
0000011	$\phi_6 p$



- Now, all you need to do is count how many unique parameters there are. A parameter is unique if 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 rates (ϕ_1 to ϕ_5), 5 recapture rates (p_2 to p_6), and one “beta” term, β_7 , the product of $\phi_6 p_7$.
- Now, a fair question at this point 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: ϕ_1, p and ϕ, p_1 . From our table on p. 4-11, we see that both models have 7 parameters. Let’s confirm this. Again, we use the “saturated capture histories approach”. Now, though we must remember that we do not have full time-dependence for both parameters. Start with the model ϕ_1, p .

- Now, in this case, we do not have a terminal beta term. The terminal product is $\phi_6 p$. Are both parts separately estimable? Yes. Since the constant recapture rate occurs at each occasion, we can use the information from preceding occasions to estimate the value of p . And, if we know the recapture rate p , then we can estimate any of the survival rates, including ϕ_6 . Specifically, ϕ_6 is obtained as

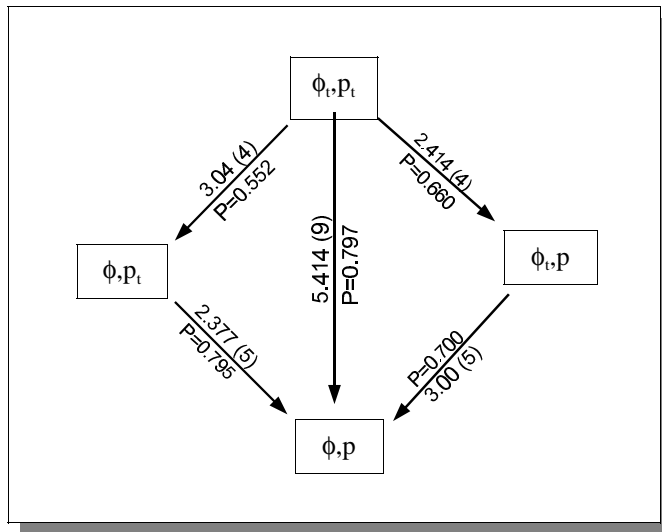
$$\hat{f}_6 = \frac{\hat{b}_7}{\hat{p}}$$

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 rate (p). For the model ϕ, p_1 , we have the same situation (7 estimable parameters) but in reverse: only 1 survival rate (ϕ), and 6 recapture rates (p_2 to p_7).

- Finally, for our model ϕ, p (constant survival and recapture), there are only two estimable parameters. By now, you should be able to prove this to yourself. Try it!

➔ **Where to test for significance of a factor?**

- Recall the schematic diagram we used to illustrate the relationships amongst the 4 models we've used in this example:



- Recall that models joined with arrows are nested (and thus can be compared using the LRT), while models not connected by an arrow are not nested, and must be compared using AIC (with the model having the lower AIC in a given comparison being the preferred model).
- Recall that the LRT provides us with a formal statistical test of the “significance” of one or more factors. For example, when we contrast model ϕ_{t,p_t} with model $\phi_{t,p}$, we are formally testing the hypothesis that there is no significant annual (i.e., time) variation in survival - if the difference between the fit of these 2 models is not significant, then we conclude that there is no evidence for significant annual variation in survival, since the model without time variation in survival fits equally well statistically.
- However, if you look at the diagram carefully, you should notice that

there are 2 different model tests we could use to test for annual variation in survival: ϕ_{t,p_t} versus $\phi_{t,p}$ (as just discussed), or model $\phi_{t,p}$ versus $\phi_{t,p}$. In both cases, we are testing for significant annual variation in survival. Are the two tests completely analogous? If we look closely at the diagram, you will see that the answer is no. In the case of ϕ_{t,p_t} versus $\phi_{t,p}$ the value of the test statistic is 3.04 ($P=0.552, df=4$), while for model $\phi_{t,p}$ versus $\phi_{t,p}$, the test statistic is 3.00 ($P=0.700, df=5$). Although the differences in the test statistic are not dramatically different, because of the differences in the number of parameters involved in the tests, the associated P-values are quite different. Thus, the obvious question is: which of the two tests is the “right one” to use?

- For now, we suggest that the best approach (i.e., the approach most likely to yield the “correct” test) 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”.
- For example, the “more parameterized model” in the two comparisons noted above is ϕ_{t,p_t} and $\phi_{t,p}$ respectively. In each case, the model was tested using LRT against the model nested within that did not contain time variation in survival. However, of these two “starting models”, model $\phi_{t,p}$ is clearly less parameterized than ϕ_{t,p_t} . Further, model $\phi_{t,p}$ is preferred to model ϕ_{t,p_t} ($P=0.552$). Thus, in this case, the more powerful test is likely to be the comparison of $\phi_{t,p}$ versus $\phi_{t,p}$, rather than ϕ_{t,p_t} versus $\phi_{t,p}$.
- Based on our analysis so far, we would conclude that, among the male Dippers in this sample, there is no evidence of significant annual variation in either survival rate or recapture rate. If this was the main hypothesis you wanted to test, then you might decide to end your analysis at this stage.





Why didn't we consider model ϕ_t, p_t versus model ϕ, p as one of our tests of the "significance" of time? The main reason is that in this comparison we are not only testing for time-dependence in survival (ϕ_t versus ϕ), but also for time-dependence in recapture rate (p_t versus p). Because we are in fact testing both parameters simultaneously, we are not (strictly speaking) testing for time-dependence in survival alone. A valid test of significance for a given parameter (ϕ or p) must hold the structure of the other parameter constant. In some senses, the comparison of model ϕ_t, p_t versus model ϕ, p could be considered as an omnibus test of an "overall time-effect" in both survival and recapture, analogous to TEST 1 in Program RELEASE (see Lebreton *et al.* 1992 - p. 82), which is an omnibus test of "overall group differences".

LRT or AIC

- One issue which comes up frequently is whether or not to always use AIC as the model selection criterion, or to use it only as a "tool" with which to (i) identify the most parsimonious model containing the factor(s) of interest you want to test using LRT, and (ii) to compare models which aren't nested. Remember - LRT is valid only when models are nested (see the 'question box' on p. 4-12).
- However, there may be at least 2 reasons to consider using AIC as a general model selection tool, in ALL cases - even when LRT are possible. First, recent theoretical work by Ken Burnham and colleagues (Burnham, Anderson & White (1995) *J Appl Stat* 22: 611-624) suggests that AIC (and derivatives) is as good as LRT under conditions where the data generally meet standard mark-recapture assumptions, and may perform better when these assumptions are violated. In particular, they showed that heterogeneity in capture rates had a negligible effect on model selection using AIC.
- Second, but perhaps as important, AIC provides a criterion which is not affected by the problem of the number of performed tests. As is familiar to most analysts, as the number of tests increases, the

probability of a "significant result" by random chance alone increases proportionately. Controlling for this is often accomplished by adjusting the critical probability of each individual test (α) for the number of tests (k), using either a Bonferroni ($\alpha = \alpha/k$) or Dunn-Šidák ($1 - (1 - \alpha)^{1/k}$) adjustment. However, no such adjustment is needed when using AIC.

- So, should you use AIC instead of LRT? For the moment, we believe that the "jury is still out" on using AIC exclusively, and recommend the basic approaches of using both AIC and LRT described earlier in this chapter. This philosophy will be used in subsequent chapters.

That's it for Chapter 4! In this chapter, we looked at the basic mechanics of using SURGE to test several models, and reviewed some of the logic of model selection. We've looked at the problem of staggered entry of marked individuals into the population (and how this leads logically to the triangular parameter structures). We've also considered the mechanics and application of two statistical "tools": the LRT and the AIC. We've also introduced the problems and procedures of counting parameters.

In the next chapter, we'll expand our understanding of model selection, focussing specifically on comparison of two groups. As you'll see, it is only a simple extension of what we've already learned. The really tricky stuff comes later!

