CHAPTER 13

Time-symmetric open models: recruitment, survival, and population growth rate

So far, we have concentrated on estimation related to the general question 'what is the probability of leaving the population?'. Clearly, death marks permanent departure from the population. Absence from the population can be permanent (like death), or temporary (a subject we'll discuss more fully in a later chapter where we introduce something known as the 'robust design'). However, if we're interested in modeling the dynamics of a population, then we're likely to be as interested in the probability of entry into the population as we are the probability of exit from the population.

So, where to begin? We'll start with the fundamental model of population dynamics. Usually, the assumption (based on even a casual glance at a typical textbook on the subject) is that population dynamics models are based entirely on high-level mathematics. However, while it isn't difficult to find examples of such models, the canonical 'fundamental model' governing population dynamics is quite simple:

population dynamics has to do with the change in abundance over space and/or time (ΔN)

$$\Delta N = \text{'additions' - 'subtractions'}$$

That's it, really. The rest is just 'details' (of course, the details can get messy, and that is what often leads to the higher math referred to above). But the basic idea is simple: when the net number of additions is greater than the net number of subtractions, then clearly the population will grow ($\Delta N > 0$). When the reverse is true, the population will decline. So, population dynamics involves the study and estimation of the relative contributions to the 'additions' and 'subtractions' from the population.

13.1. Population growth: realized vs. projected

Usually, the net growth of a population is expressed as the ratio of successive population abundances: N_{t+1}/N_t. This ratio is usually referred to as λ, leading (frequently) to a whole bunch of confusion – λ as the realized (i.e., observed) ratio of successive population sizes over time, or λ as the *projected* growth rate of a population under specified model conditions? As you may recall, the *projected* growth of a structured population is given as the dominant eigenvalue from a non-negative projection matrix (the 'Leslie' matrix for models structured on age, and the 'Lefkovitch' matrix for situations where some other classification factor – often size or developmental state – is a better (i.e., more informative) demographic

© Cooch & White (2025)

category than age). The word 'projected' is key here – it is the growth rate of the population that would eventually be expected if (and only if) the conditions under which the model are valid are time invariant (i.e., not stochastic). We differentiate between *projected* λ and *realized* λ. Projected λ and realized λ will be equivalent if and only if the growth of the population has achieved stationary, ergodic conditions (the familiar stable-age or stable-age structure at equilibrium). Under such conditions, the population (and each age-class in the population) will be growing at rate λ, such that $\lambda = N_{t+1}/N_t$. So, we suggest qualifying the use of 'λ' with a prefix – either '*projected* λ', or '*realized* λ', and noting (if appropriate) when they are equivalent, and when they are different:

projected λ: the growth rate of the population expected under time invariance (where λ is commonly derived as the dominant eigenvalue from a projection matrix model)

realized λ: the observed growth rate of the population between successive samples (time steps): $\lambda_i = N_{i+1}/N_i$

While *projected* λ is often of considerable interest (especially for prospective management studies), in a retrospective study, where we're interested in assessing the pattern of variation in growth that *has* occurred, it is *realized* λ that is of interest (although there are a variety of analytical methods out there for retrospective analysis that somewhat blur this convenient distinction: for example, the life table response experiment (LTRE) developed by Caswell (2001) is a hybrid of a retrospective technique using prospective perturbation analysis of deviation in the projected λ under a variety of experimental conditions). For our purposes though, we'll keep to the simple distinction we've just described – we want to explore changes in *realized* growth, λ.

13.2. Estimating realized λ

Since $\lambda = N_{t+1}/N_t$, then it seems reasonable that as a first step, we want to derive estimates of abundance for our population at successive time steps, and simply derive the ratio to yield our estimate of λ. Simple in concept, but annoyingly difficult in practice. Why? In large part, because the estimation of abundance in an open population is often very imprecise. Such estimates often suffer rather profoundly from violation of any of a number of assumptions, and are often not worth the effort (abundance estimation in open populations was covered in Chapter 12 – estimation in closed populations is covered in Chapter 15).

So we're stuck, right? Not exactly. The 'solution' comes in several steps. We start with the recognition that our basic purpose in characterizing the change in abundance between years rests on assessing the relative number of 'additions' and 'subtractions'. This basic idea was introduced in Chapter 12.

With a bit of thought, you should realize that an individual which 'dies', or 'permanently emigrates', is clearly a 'subtraction' from the population. As such, we can reasonably state that the number of individuals in the population next year is going to be a function, at least in part, of the number of individuals in the population this year that survive and return to the population next year.

However, we also know that there may be 'additions' to the population, either in the form of permanent immigration, or births (*in situ* recruits). Let the number of individuals this year be N_t. Let φ_i be the probability of surviving and returning to the population ($\varphi_i = S_i F_i$, where S is the true survival probability, and F is the fidelity probability – see chapter 9). Let B_i be the number of new individuals that enter the population between (i) and (i+1). In other words, B_i is the number of individuals in the population at (i+1) that were **not** there at (i).

Thus, we can write:

$$N_{i+1} = N_i \varphi_i + B_i.$$

Next, some simple algebra. First, recall we define λ as the ratio of successive population sizes – $\lambda_i = N_{i+1}/N_i$. Thus, substituting this into the previous expression, and after a bit of re-arranging, we get:

$$\lambda_i = \frac{B_i}{N_i} + \varphi_i.$$

Now, λ and φ are familiar (and explicitly defined above).

What about B_i/N_i? This is the per capita rate of additions to the population (often referred to somewhat 'sloppily' as the recruitment rate, which has a very specific demographic meaning that is often ignored – for purposes of consistency with some of the literature, we'll ignore it too). It is the number of individuals entering the population between (i) and (i+1) (i.e., B_i, the numerator) per individual already in the population at time (i) (i.e., N_i, the denominator). Let's call this ratio 'recruitment rate', f_i.

Thus, we write

$$\begin{aligned}\lambda_i &= \frac{B_i}{N_i} + \varphi_i \\ &= f_i + \varphi_i.\end{aligned}$$

OK, so far so good. But perhaps right about now you're asking yourself 'how does this help?'. We can estimate φ_i fairly well (as discussed in the first several chapters of this guide), but what about recruitment, f_i? After all, f_i is B_i/N_i, both of which are difficult to estimate with any precision in an open population.

13.2.1. Reversing encounter histories: φ and γ

Now for the **big** 'trick', which is so intuitively obvious once we describe it we should probably pause long enough for you to slap yourself in the forehead. Back in 1965, George Jolly, and later Ken Pollock in 1974, noted that the encounter histories carried a lot more information than we often realize.*

Amongst other things, they noted that

> '*if estimating the transitions among encounter occasions going forward in time yields an estimate of the probability of remaining alive and in the population, φ, where $(1 - \varphi)$ is the probability of* leaving *the population, then if you simply reverse the encounter histories, the transition parameter being estimated is the probability of entering the population*'.

Why is this important? It is important because that's precisely what we're after. Recruitment is the process of *entering* the population. So, if we had a parameter that allowed us to estimate the probability of entering a population (i.e., recruiting), then we're clearly on the right track.

In fact, this is precisely what Roger Pradel described in his 1996 paper (Pradel 1996). Re-discovering (and extending) the earlier work of Jolly and Pollock, Pradel explicitly noted the duality between analyzing the encounter history going forward, and going backward in time. He introduced a parameter γ_i (which he referred to as the *seniority* parameter), which he defined as 'the probability that if an individual is alive and in the population at time i that it was also alive and in the population at time i-1'. And, as we will see, this is very much the 'key idea'.

---

* In fact, Jackson (1936, 1939), prompted (it seems) by suggestions from R. A. Fisher, applied methods which strongly predict the approaches later developed more formally by Jolly and Pollock – see Nichols (2016).

Let's pause to highlight the distinctions between φ_i (going *forward* in time) and γ_i (estimated from the reverse encounter history, going *backward* in time):

forward in time	φ_i	probability that if alive and in the population at time *i* (e.g., this year), the you will be alive and in the population at time *i*+1 (e.g., next year)
backward in time	γ_i	probability that if alive and in the population at time *i* (e.g., this year), that you *were* also alive and in the population at time *i*-1 (e.g., last year)

begin sidebar

Understanding 'backwards' encounter histories

Consider the following encounter history: '`101110`'. What this history is telling us, going forward in time, is that the individual was initially encountered and marked at occasion 1, not seen at occasion 2, but then seen on the next 3 occasions (3, 4, and 5), then not seen on occasion 6. The probability expression corresponding to this history would be:

$$\varphi_1(1-p_2)\varphi_2p_3\varphi_3p_4\varphi_4p_5(1-\varphi_5p_6).$$

Now, if we reverse the encounter history ('`101110`' → '`011101`'), then we see that, conditional on being alive and in the population at occasion 5, that the individual was also in the population at occasion 4, and at occasion 3. Given that it was also there at occasion 1, but not encountered at occasion 2 (with probability $1-p_2$), then the probability expression (in terms of γ_i and p_i) corresponding to '`011101`' is:

$$\gamma_5p_4\gamma_4p_3\gamma_3(1-p_2)\gamma_2p_1.$$

end sidebar

13.2.2. Putting it together: deriving λ

Still with us? We hope so, because the 'seniority' parameter γ_i features prominently in what comes next. Remember, our interest in γ_i comes from it having something to do with recruitment.

Now, for the next big step – pay close attention here. Remember from our previous discussion that B_i is the number of individuals entering the population between time (*i*) and (*i*+1). If N_{i+1} is the number of individuals in the population at time (*i*+1), then B_i/N_{i+1} is the proportion (or probability) that an individual in the population at $i+1$ is one that entered the population between (*i*) and ($i+1$). So, if (B_i/N_{i+1}) is the probability that an individual entered the population, then $1-(B_i/N_{i+1})$ is the probability that it was already in the population.

Does this sound familiar? It should. Think back – what is γ_i? It is the probability that if you're in the population at time (*i*) that you were also there at time (*i*-1), which is precisely the same thing! In other words,

$$\gamma_{i+1} = 1 - \frac{B_i}{N_{i+1}}.$$

Thus, since

$$N_{i+1} = N_i\varphi_i + B_i,$$

then we can write

$$\begin{aligned}\gamma_{i+1} &= 1 - \frac{B_i}{N_{i+1}} \\ &= 1 - \frac{\left(N_{i+1} - N_i\varphi_i\right)}{N_{i+1}} \\ &= \frac{N_i\varphi_i}{N_{i+1}} \\ &= \frac{\varphi_i}{\lambda_i}.\end{aligned}$$

Or, re-arranging slightly,

$$\lambda_i = \frac{\varphi_i}{\gamma_{i+1}}.$$

In other words, all we need are estimates of φ_i and γ_{i+1}, and we can derive estimates of realized λ_i, all without ever measuring population abundance (N)! So, we have a technique where, using simple mark-recapture data, we derive an explicit estimate of population growth, with considerable precision (as it turns out), without the need to estimate abundance.

`begin sidebar`

An alternative derivation

We can also derive this expression for λ using a slightly different approach. Let the size of the population at risk of capture (encounter) on occasions i and i+1 be N_i and N_{i+1}, respectively. A subset of these two sets (populations) of animals is the subset of all animals that are alive in the population at both times. At time i+1, the size of this set can be given as $N_i\varphi_i$. At time i, the size of that set can be given as $\gamma_{i+1}N_{i+1}$. Since both relationships give the size of the same set (population size alive and in the population at both times), then we can write

$$N_i\varphi_i = \gamma_{i+1}N_{i+1}$$

Since $\lambda = N_{i+1}/N_i$, then it follows that $\lambda_i = \varphi_i/\gamma_{i+1}$, which is exactly the same relationship we derived above.

`end sidebar`

We also note that since $\lambda_i = \varphi_i + f_i$, then we can clearly also derive the following 2 expressions:

$$\gamma_{i+1} = \frac{\varphi_i}{\varphi_i + f_i}\left(= \frac{\varphi_i}{\lambda_i}\right) \qquad f_i = \varphi_i\left(\frac{1 - \gamma_{i+1}}{\gamma_{i+1}}\right)$$

This is the essence of what we refer to as *time-symmetric models* in **MARK**. They rest on the duality (i.e., the time-symmetry) between estimating φ_i going forward in time, and γ_i going backward in time. Moreover, because φ and γ can both be estimated in a general likelihood expression, not only can we estimate λ and f directly, but we can also derive estimates of the variance in both. Perhaps more importantly, we can address hypotheses about variation in one or more of these parameters using the standard linear models approach we've used throughout – constraining various parameters to be linear functions of one or more covariates.

13.3. Projected λ versus realized λ: are they equivalent?

While the preceding result might appear to be 'the best of all worlds', there are important assumptions, caveats, and conditions under which the time-symmetric models, and the parameters you can estimate with them, are 'meaningful'. First, and most importantly, the parameter λ estimated using time-symmetric models[†] is a measure of the realized rate of change of the age class from which the encounter histories were derived. It is **not** necessarily a measure of the growth rate of the population!

This is so important, we'll repeat it again, with some emphasis:

> **The λ estimated from time-symmetric models is the realized growth rate of the *age class* from which the encounter histories were generated, *which is not necessarily (or even generally) equivalent to the growth rate of the population...***

The λ estimated from the time-symmetric models *may* be a good measure if there is a single age class which contributes most of the variation in the growth of the population as a whole (or if you make somewhat heroic assumptions that the age structure of the population is stationary – in which case, growth of any individual age class is equivalent to the growth of the population as a whole).

Second, the estimate of λ is only biologically meaningful if the study (sample) area stays constant. If you expand your study area, then both f and λ make little biological sense, because the population to which inferences are being made is also expanding or contracting. Further, even if the study area remains constant, some individuals can be missed in the first years of the study (when observers are learning their 'field craft'), and estimates of f, and thus λ, from early years may frequently be biased high. These methods also assume that all animals in the study area have some non-zero probability of being captured. Finally, significant trap response can lead to substantial bias.

But, these caveats notwithstanding, the time-symmetric models are potentially powerful tools for exploring population dynamics. We can look at variation in growth trajectory (λ) without the problems associated with abundance estimation.

13.4. Time-symmetric models in MARK

The main 'class' of time-symmetric models in **MARK** are commonly referred to as 'Pradel models', after the seminal paper by Roger Pradel (1996). To demonstrate the use of Pradel models in **MARK**, we'll make use of the capsid data set (Muir 1957), **`pradel.inp`** (this file is included when you install **MARK** – look in the `MARK/examples` subdirectory on your computer, and drag the input file onto the desktop).

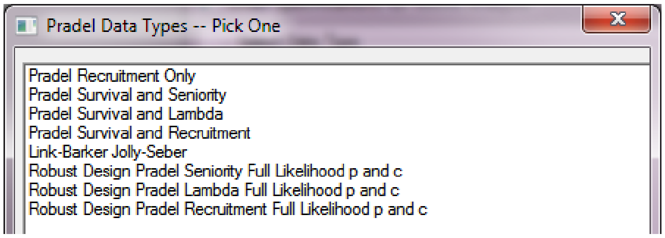
Go ahead and start **MARK**, and begin a new project, reading in the capsid data file. There are 13 occasions, and 1 group.

Once you've specified the input file, and entered the appropriate number of occasions and attribute groups, your next step is to specify which of the Pradel models you want to fit. When you click the '**`Pradel models including robust design`**' data type, you'll be presented with a pop-up window

---

[†] It is important to note that in his 1996 paper, Pradel did not actually use the parameter λ, but instead used ρ to indicate the change in abundance between successive years. While using ρ for *realized* growth rate instead of λ eliminates confusion with use of λ as *projected* growth rate, **MARK** adopts the use of λ, since it is more commonly associated with measures of population growth.

(shown below) listing 8 data types (the first 5 primary models apply to open populations, and following 3 for models within a 'robust design' framework – see Chapter 16 for a full description of the robust design). For now, we'll focus on the (first) 4 'open population' Pradel models.

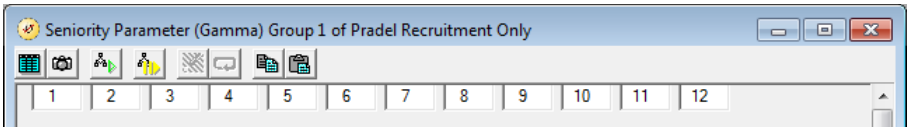


Pradel Recruitment Only
Pradel Survival and Seniority
Pradel Survival and Lambda
Pradel Survival and Recruitment
Link-Barker Jolly-Seber
Robust Design Pradel Seniority Full Likelihood p and c
Robust Design Pradel Lambda Full Likelihood p and c
Robust Design Pradel Recruitment Full Likelihood p and c

At this point, you need to decide which analysis you want to do. You could focus on estimation of the γ_i values alone (i.e., estimate the seniority probabilities). Estimation of γ can be useful in analysis of age-specific variation in 'recruitment to breeding state' (i.e., the transition from pre-breeder to breeder – e.g., Cooch *et al*. 1999).

The next 3 models (and their robust design equivalents) consist of different pairs of parameters (for example, you could estimate survival and realized λ by selecting the '`Survival and Lambda`' data type). You cannot estimate all 3 of the primary parameters simultaneously (i.e., you can't estimate φ, γ, f, and λ in one data type), since they are effectively a linear function of each other (such that estimating any two of them can provide estimates of the remaining parameters).[‡]

Let's run through all 4, starting with the '`Pradel recruitment only`' data type, providing estimates of seniority γ and encounter probability p. Once you click the '`OK`' button, you'll be presented with the PIM for the seniority parameter γ:



1	2	3	4	5	6	7	8	9	10	11	12

Why only 1 row in the PIM? Why not the 'triangular' PIM we've seen for standard recapture models? The answer is because the Pradel models don't allow for 'age effects'. As noted by Franklin (2001), the reason for this is that the likelihood for estimating γ is conditioned on the entire encounter history, not just the portion following first capture as is the case when estimating φ under the CJS model.

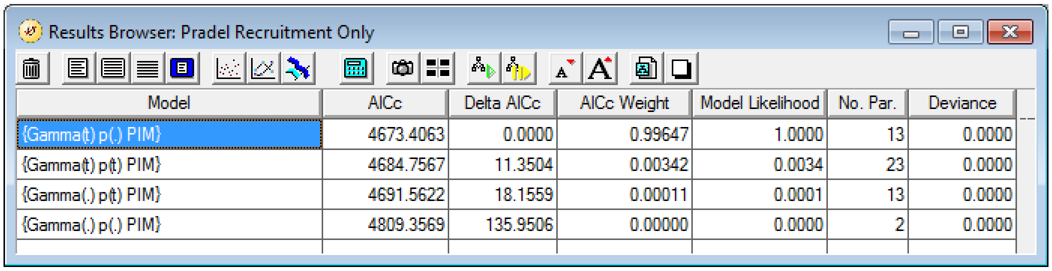
For example, **MARK** conditions on the full encounter history '`001101`' to estimate f and λ, whereas it conditions on only the '`1101`' portion to estimate φ and p. Therefore, age cannot be included because age cannot be estimated back to the initial zeros of the encounter history. Thus, you have no more than one row in the PIM (since each row corresponds to a different release cohort, and cohort and age are collinear). If age-specific estimates are desired, groups of animals can be created based on age.

For our first analysis using the '`Pradel recruitment only`' data type, we'll fit 4 simple models to the capsid data: $\{\gamma_t\, p_t\}$, $\{\gamma_t\, p.\}$, $\{\gamma.\, p_t\}$, and $\{\gamma.\, p.\}$. The models are constructed using PIMs, and the default sin link.

[‡] This is a fairly common situation in parameter estimation. For example, if you have some general model where (say), $A = B + C$, you can often estimate each of the 3 parameters individually (i.e., A, or B, or C), or pairs of parameters (A, B), or (A, C), or (B,C), but not all 3 parameters at the same time.

Thus at least one of the model parameters is (i) not included in the model likelihood, which means that (ii) the missing parameters not in the likelihood will be 'derived by algebra', and (iii) since the derived parameter(s) is not in the likelihood, it isn't in the DM, and thus you can't apply linear constraints to it.

The results browser (below) shows the results of fitting these models to the capsid data.



Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{Gamma(t) p(.) PIM}	4673.4063	0.0000	0.99647	1.0000	13	0.0000
{Gamma(t) p(t) PIM}	4684.7567	11.3504	0.00342	0.0034	23	0.0000
{Gamma(.) p(t) PIM}	4691.5622	18.1559	0.00011	0.0001	13	0.0000
{Gamma(.) p(.) PIM}	4809.3569	135.9506	0.00000	0.0000	2	0.0000

Based on these 4 models, and with a default $\hat{c}$ of 1.0, it appears as though a model with time-variation in γ and a constant encounter probability p is overwhelming supported by the data.

However, since we've mentioned $\hat{c}$, what about it? If you try to run a bootstrap, or median $\hat{c}$, **MARK** will quickly tell you that a neither of these GOF tests are available for the Pradel recruitment only model (and indeed, they aren't available for **any** of the Pradel models). At first glance, it might seem that you could simply flip the encounter histories back around to the 'normal' forward direction (as if you were going to do a CJS analysis), and simply use the CJS bootstrap GOF test.

However, this is inappropriate in general, since the likelihood is based on the full encounter history. Still, it 'may' be reasonable if your only interest is in estimating the γ parameters. For the moment, we can cautiously suggest that if all you're interested in is γ, and are using the Pradel 'seniority only' model, then since this model is identical to taking your encounter histories, flipping them, and running them through the CJS model, then the $\hat{c}$ from the CJS model may be appropriate. But – don't quote us (this is still a work in progress). For the other open population Pradel models ('`Survival and Seniority`', '`Survival and Lambda`', and '`Survival and Recruitment`'), this is likely to be incorrect.

These difficulties notwithstanding, it is likely these other models (being, those models not focussed exclusively on the seniority parameter) that will hold the greatest interest for you, since these models provide information on parameters related to the growth of a population. We will focus in particular on the '`Survival and Lambda`' and '`Survival and Recruitment`' models.

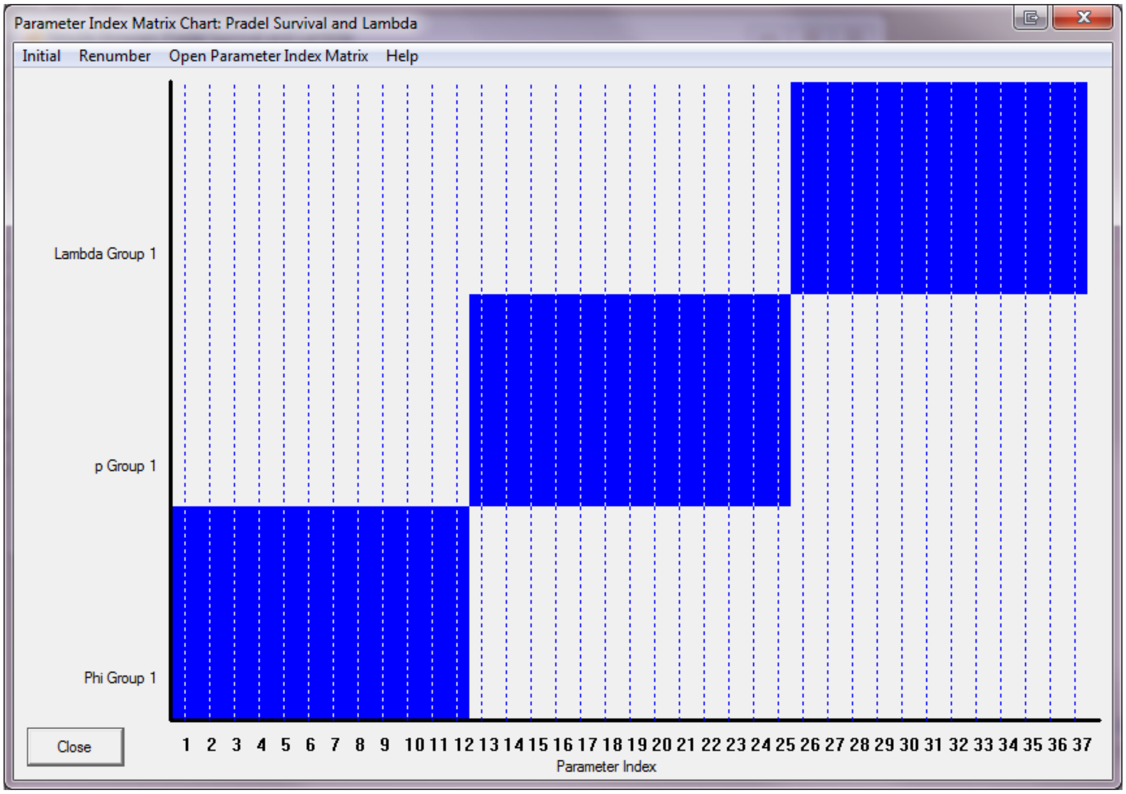
Let's restart our analysis of the capsid data. But, before we do – a summary of the study and sampling design, followed by an important technical aside. The capsid is a small bug. The capsid data were collected by shaking trees in an orchard in some fashion, and collecting the individuals that fell down out of the trees onto sheets arranged below. Given the organism, and the sampling protocol, it is more than likely that a fair number of the capsids were 'lost on capture' (meaning, stepped on, probably).

This is more than a historical sidenote – losses on capture are important considerations when using time-symmetric models, especially when you're working with models where recruitment is a parameter. Unfortunately, Pradel models as parameterized in **MARK** do not handle the losses on capture correctly, since the likelihood(s) being optimized do not incorporate the 'losses on capture parameter' as described in Pradel's (1996) paper.

A workable solution is to use the Link-Barker parameterization (which is one of the data types available in the list you're presented when starting the analysis). Link and Barker (2005) reparameterized the recruitment likelihood of Pradel (1996), ending up at a starting point of the likelihood used in the **POPAN** data type (the **POPAN** and Link-Barker models were introduced in detail in Chapter 12). In this re-parameterized likelihood, the 'probability of entry' parameters in **POPAN** are translated to a recruitment (births and immigration) parameter (f), strictly analogous to the recruitment parameter in Pradel models.

More to the point, the Link-Barker as implemented in **MARK** correctly handles 'losses on capture'. If there are no losses on capture, the 'Link-Barker' likelihood and the 'Pradel' likelihood as parameterized in **MARK** are equivalent, and parameter estimates under the two models will not differ greatly. However, if there are significant losses on capture, the Link-Barker model is preferred. For now, we'll assume no losses on capture, and use Pradel models, but as an exercise, you might redo the following analysis, but instead using the Link-Barker parameterization.

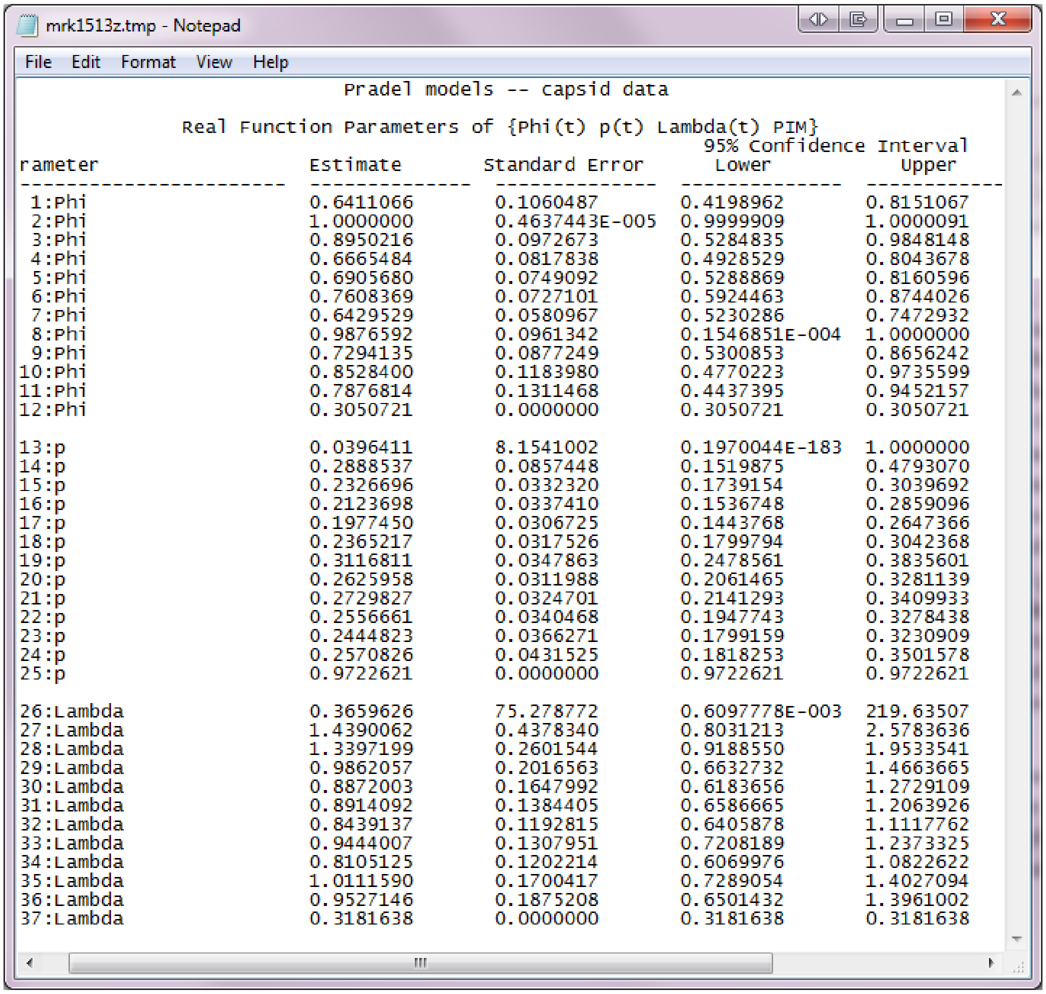
So, we re-visit the capsid data, but this time we will select the '`Survival and Lambda`' model. After we click the '`OK`' button, we are immediately presented with the open PIM for the apparent survival parameter, φ. If you open up the PIM chart (below), you'll see that there are 3 structural parameters involved in this model type: φ, p and λ.



If you look at each PIM separately (i.e., open up the PIM corresponding to each parameter), you'll see that each of them consists of 1 row (the reason for this has been discussed earlier). So, again, the modeling is relatively straightforward – you can apply constraints quite simply to any one or more of the parameters – all three parameters are in the likelihood, and this, you can 'model' any of them (via the DM, for example).

So, at this point, this looks like perhaps the simplest thing we've done so far with **MARK**. However, there are several issues to consider. First, several parameters are confounded under the fully time-dependent model. For example, in model $\{\varphi_t\, p_t\, \lambda_t\}$ the first and last λ are inestimable because φ_1 is confounded with p_1 and φ_{k-1} is confounded with p_{k-1} (for k encounter occasions).

The reconstituted estimates from the fully time-dependent model $\{\varphi_t\, p_t\, \lambda_t\}$ are shown at the top of the next page. We see that the first and last estimates of λ are 'problematic' (very large or impossibly small SE for both estimates). We also have the usual issues of inestimability of terminal survival and encounter parameters.



```
                            Pradel models -- capsid data

              Real Function Parameters of {Phi(t) p(t) Lambda(t) PIM}
                                                           95% Confidence Interval
rameter                  Estimate       Standard Error      Lower           Upper
-----------------------  --------------  --------------  --------------  ------------
 1:Phi                   0.6411066       0.1060487       0.4198962       0.8151067
 2:Phi                   1.0000000       0.4637443E-005  0.9999909       1.0000091
 3:Phi                   0.8950216       0.0972673       0.5284835       0.9848148
 4:Phi                   0.6665484       0.0817838       0.4928529       0.8043678
 5:Phi                   0.6905680       0.0749092       0.5288869       0.8160596
 6:Phi                   0.7608369       0.0727101       0.5924463       0.8744026
 7:Phi                   0.6429529       0.0580967       0.5230286       0.7472932
 8:Phi                   0.9876592       0.0961342       0.1546851E-004  1.0000000
 9:Phi                   0.7294135       0.0877249       0.5300853       0.8656242
10:Phi                   0.8528400       0.1183980       0.4770223       0.9735599
11:Phi                   0.7876814       0.1311468       0.4437395       0.9452157
12:Phi                   0.3050721       0.0000000       0.3050721       0.3050721

13:p                     0.0396411       8.1541002       0.1970044E-183  1.0000000
14:p                     0.2888537       0.0857448       0.1519875       0.4793070
15:p                     0.2326696       0.0332320       0.1739154       0.3039692
16:p                     0.2123698       0.0337410       0.1536748       0.2859096
17:p                     0.1977450       0.0306725       0.1443768       0.2647366
18:p                     0.2365217       0.0317526       0.1799794       0.3042368
19:p                     0.3116811       0.0347863       0.2478561       0.3835601
20:p                     0.2625958       0.0311988       0.2061465       0.3281139
21:p                     0.2729827       0.0324701       0.2141293       0.3409933
22:p                     0.2556661       0.0340468       0.1947743       0.3278438
23:p                     0.2444823       0.0366271       0.1799159       0.3230909
24:p                     0.2570826       0.0431525       0.1818253       0.3501578
25:p                     0.9722621       0.0000000       0.9722621       0.9722621

26:Lambda                0.3659626       75.278772       0.6097778E-003  219.63507
27:Lambda                1.4390062       0.4378340       0.8031213       2.5783636
28:Lambda                1.3397199       0.2601544       0.9188550       1.9533541
29:Lambda                0.9862057       0.2016563       0.6632732       1.4663665
30:Lambda                0.8872003       0.1647992       0.6183656       1.2729109
31:Lambda                0.8914092       0.1384405       0.6586665       1.2063926
32:Lambda                0.8439137       0.1192815       0.6405878       1.1117762
33:Lambda                0.9444007       0.1307951       0.7208189       1.2373325
34:Lambda                0.8105125       0.1202214       0.6069976       1.0822622
35:Lambda                1.0111590       0.1700417       0.7289054       1.4027094
36:Lambda                0.9527146       0.1875208       0.6501432       1.3961002
37:Lambda                0.3181638       0.0000000       0.3181638       0.3181638
```

However, when constraints are placed on either φ or p, some of the variation in these parameters is taken up by λ, γ, or f, which often 'solves the estimability problem'. For example, if you constrained p to be constant over time (e.g., model $\{\varphi_t\, p_{\cdot}\, \lambda_t\}$), you would see that all of the estimates for λ would appear reasonable.

13.4.1. Linear constraints and time-symmetric models

Unfortunately, the issue of applying constraints to models where λ is a structural parameter is somewhat more complex than the preceding example suggests. For example, Franklin (2001) suggests that since λ, γ and f are (often) the parameters of biological interest in the Pradel models, that it is often best to model φ and p as completely time-dependent, and apply the constraints to λ, γ, or f.

While this might seem reasonable, there's a catch. If you do specify constraints on λ, γ, or f, you need to be a bit careful when interpreting the 'meaning' behind constraining these parameters. Since $\lambda_i = \varphi_i + f_i$, if a model is fit with time invariant (i.e., constant) λ, or where λ is constrained to follow a linear trend, but with time varying f, then this implies a direct inverse relationship between survival and recruitment (e.g., if λ is held constant, then if φ_i goes up, then f_i must go down). While this may be true in a general sense, it is doubtful that the link between the two operates on small time scales typically used in mark-recapture studies. As noted by Franklin (2001), models where φ is time invariant while λ is allowed to vary over time are (probably) reasonable, as variations in recruitment are the extra source of 'variation' in λ.

More complex models involving covariates have the same difficulty. Population-level covariates (e.g., weather) are interpretable, but it is potentially difficult to interpret individual-based covariates as operating on population growth. Individual covariates can be used to model φ and p in the Pradel models, since these are covariates that relate to individual parameters, whereas λ is a population level parameter. The root of the problem is that while individual covariates could apply to survival probabilities, the recruitment parameter is not tied to any individual – it is a population-based, average recruitment per individual in the population. What is needed is a generalization of the Jolly-Seber model (Chapter 12) where new entrants to a population are tied to existing members of the population, for example, if newborns were identified with their parents.[§]

So, as long as you take some care, then you'll be OK. There are always challenges in modeling parameters that are linear functions of each other – be advised – think carefully. The Pradel models have great potential, for a number of applications. If you're careful, then you can apply them to a fairly broad set of problems. We introduce once such 'application' in the following section.

begin sidebar

Time-symmetric models and link functions

Several types of parameters have forced link functions, i.e., the link function is changed to the default value unless the user specifies '**`Parm-Specific`**' link functions. Specifically, the λ and f (recruitment) parameters of the Pradel and Link-Barker data types are set to a *log* link function, even if the user selects the **`sin`**, **`logit`**, **`loglog`**, or **`cloglog`** link functions. Likewise, the population estimates ($\hat{N}$) in the (*i*) Jolly-Seber and POPAN and (*ii*) closed captures data types (discussed in Chapter 12 and Chapter 15, respectively) are also set to a log link function when the user selects the sin, logit, loglog, or cloglog link functions for the model. The reason for these changes from the user-specified link function for the model is that link functions that constrain these parameters to the $[0, 1]$ interval will not work because the real parameters λ, f and N should not be in the $[0, 1]$ interval.

Note: **MARK** will force the log link for λ, regardless of what other link function you select – but, there will be no indication of this in the output. So, for example, if you are evaluating the β_i values for a particular model, the β_i values are estimated on the log scale, so (i) if reconstituting by hand, you need to use the back-transform for the log link, and (ii) assessing if $\lambda > 1$ is equivalent to asking if $\beta_i > 0$.

In addition, there is at least one other situation involving Pradel models where you may need to pay particular attention to the link function. There is a logical necessity that $\varphi_i \le \lambda_i$ for time-symmetric models where λ occurs as a structural parameter, i.e., is a parameter included in the likelihood. Consider for example, the situation where $f = 0$ (i.e., if there was no recruitment). In the absence of recruitment, the population would decline over time – the rate of the decline in a given year could not be less than φ_i, and thus $\varphi_i \le \lambda_i$. You can enforce this constraint using the *cumulative logit* link function (introduced in Chapter 6, section 6.8.1). As you might recall, the cumulative logit link (CLogit) function is useful for constraining a set of parameters to monotonically increase.

Suppose for some parameter θ that you desire the relationship of $\hat{\theta}_1 \le \hat{\theta}_2 \le \hat{\theta}_3$, but do not want to enforce the strict linear relationship on the logit scale, such that

$$\text{logit}\big[\theta_2\big] - \text{logit}\big[\theta_1\big] = \text{logit}\big[\theta_3\big] - \text{logit}\big[\theta_2\big].$$

To use the CLogit link, you have to specify a separate CLogit link for each set of parameters that are to be constrained. In addition, you also have to specify the order of the parameters for the set.

[§] An alternative approach based on random effects is discussed in Appendix D (section D.4.4).

For this example, the link function for each of the 3 parameters would be:

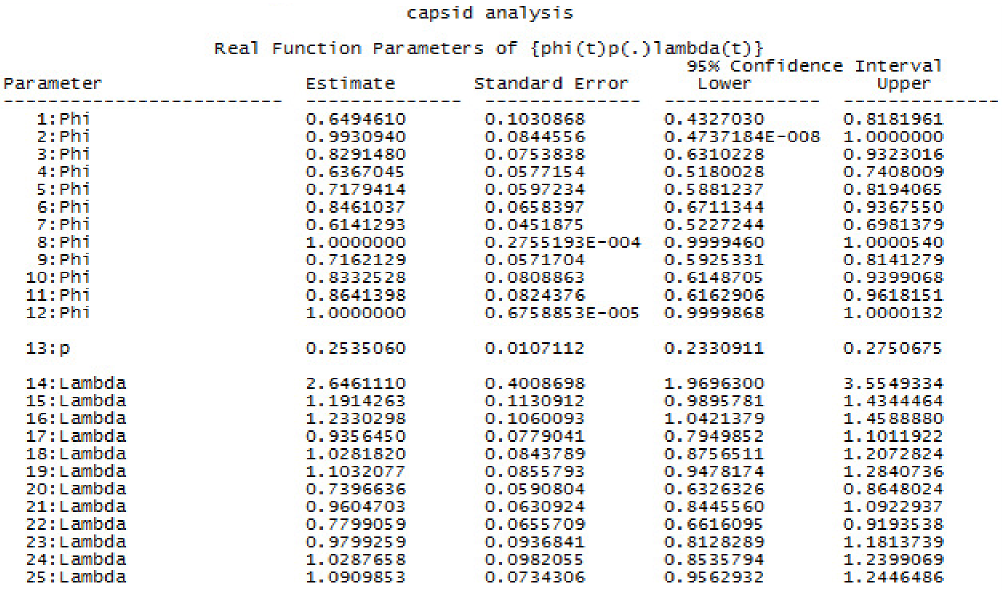
θ_1: `CLogit(1,1)`

θ_2: `CLogit(1,2)`

θ_3: `CLogit(1,3)`

Consider the situation where you want to constrain $\varphi_i \leq \lambda_i$. We'll demonstrate the steps using the capsid data set introduced earlier (contained in **`pradel.inp`**). Re-start your analysis of these data, and fit model $\{\varphi_t\, p_{.}\, \lambda_t\}$ to the data (i.e., the '**`Pradel Survival and Lambda`**' data type; time-dependence in φ and λ, but a constant 'dot' model for encounter probability p). We'll start using the default sin link for all parameters (remembering that in fact **MARK** will use the log link for λ).

The parameter estimates are shown below:



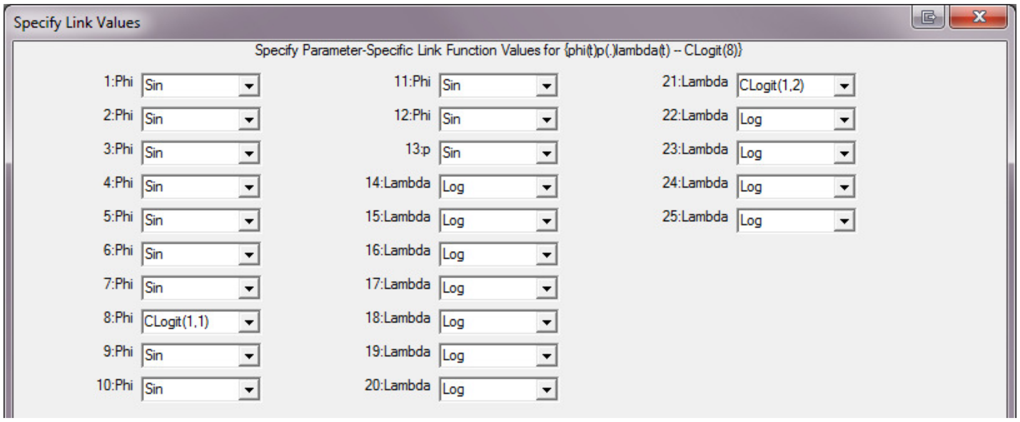
```
                                capsid analysis

                    Real Function Parameters of {phi(t)p(.)lambda(t)}
                                                             95% Confidence Interval
Parameter                    Estimate       Standard Error      Lower           Upper
-------------------------   --------------  --------------  --------------  --------------
   1:Phi                     0.6494610       0.1030868       0.4327030       0.8181961
   2:Phi                     0.9930940       0.0844556       0.4737184E-008  1.0000000
   3:Phi                     0.8291480       0.0753838       0.6310228       0.9323016
   4:Phi                     0.6367045       0.0577154       0.5180028       0.7408009
   5:Phi                     0.7179414       0.0597234       0.5881237       0.8194065
   6:Phi                     0.8461037       0.0658397       0.6711344       0.9367550
   7:Phi                     0.6141293       0.0451875       0.5227244       0.6981379
   8:Phi                     1.0000000       0.2755193E-004  0.9999460       1.0000540
   9:Phi                     0.7162129       0.0571704       0.5925331       0.8141279
  10:Phi                     0.8332528       0.0808863       0.6148705       0.9399068
  11:Phi                     0.8641398       0.0824376       0.6162906       0.9618151
  12:Phi                     1.0000000       0.6758853E-005  0.9999868       1.0000132

  13:p                       0.2535060       0.0107112       0.2330911       0.2750675

  14:Lambda                  2.6461110       0.4008698       1.9696300       3.5549334
  15:Lambda                  1.1914263       0.1130912       0.9895781       1.4344464
  16:Lambda                  1.2330298       0.1060093       1.0421379       1.4588880
  17:Lambda                  0.9356450       0.0779041       0.7949852       1.1011922
  18:Lambda                  1.0281820       0.0843789       0.8756511       1.2072824
  19:Lambda                  1.1032077       0.0855793       0.9478174       1.2840736
  20:Lambda                  0.7396636       0.0590804       0.6326326       0.8648024
  21:Lambda                  0.9604703       0.0630924       0.8445560       1.0922937
  22:Lambda                  0.7799059       0.0655709       0.6616095       0.9193538
  23:Lambda                  0.9799259       0.0936841       0.8128289       1.1813739
  24:Lambda                  1.0287658       0.0982055       0.8535794       1.2399069
  25:Lambda                  1.0909853       0.0734306       0.9562932       1.2446486
```

While $\hat{\varphi}_i \leq \hat{\lambda}_i$ for most parameters, this is clearly not the case for $\hat{\varphi}_8 = 1.0000 > \hat{\lambda}_8 = 0.9605$. Since this particular pair of parameters violates the logical necessity that $\hat{\varphi}_i \leq \hat{\lambda}_i$, we could simply re-run the model applying the CLogit link to these parameters only. To do this, we'll re-run model $\{\varphi_t\, p_{.}\, \lambda_t\}$, except that this time we'll select the '**`Parm-Specific`**' link function option. Once you click the '**`OK to Run`**' button, you'll be presented with a popup window allowing you to specify the link function for a given parameter. Here, we're going to apply the CLogit link between φ_8 and λ_8.

The CLogit link is specified by manually entering the word '`CLogit`' into the appropriate spot in this window. The Clogit function has 2 arguments: the first is the particular set (or pair) of parameters you want to apply the link to, and the second identifies the member of the pair. For example, `CLogit(1,1)` and `CLogit(1,2)` refer to the first pair of parameters, corresponding to φ_1 and λ_1, respectively. `CLogit(2,1)` and `CLogit(2,2)` refer to the second pair of parameters, corresponding to φ_2 and λ_2, respectively. And so on.

The completed link function specification window for our capsid analysis is shown at the top of the next page.

Note: if you are applying the CLogit link to only a subset of the φ and λ parameter pairs, you must remember to specify the log link function for the λ parameters you are not constraining using the CLogit. For the capsid analysis, we specify the log link for parameters 14 → 20, and 22 → 25 (top of the next page). If we run the model with the CLogit link function applied to φ_8 and λ_8, we'll see that $\hat{\varphi}_8 = 0.9998 \leq \hat{\lambda}_8 = 0.9998$.

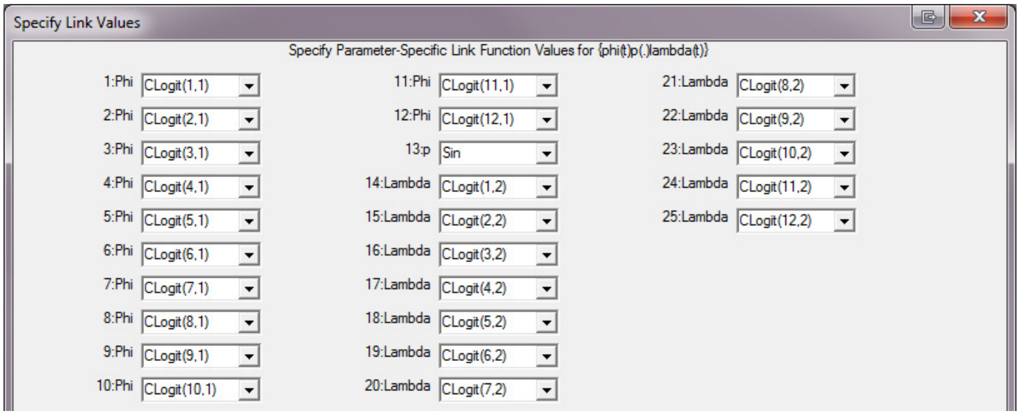


Specify Parameter-Specific Link Function Values for {phi(t)p(.)lambda(t) -- CLogit(8)}

1:Phi Sin
2:Phi Sin
3:Phi Sin
4:Phi Sin
5:Phi Sin
6:Phi Sin
7:Phi Sin
8:Phi CLogit(1,1)
9:Phi Sin
10:Phi Sin
11:Phi Sin
12:Phi Sin
13:p Sin
14:Lambda Log
15:Lambda Log
16:Lambda Log
17:Lambda Log
18:Lambda Log
19:Lambda Log
20:Lambda Log
21:Lambda CLogit(1,2)
22:Lambda Log
23:Lambda Log
24:Lambda Log
25:Lambda Log

However, applying the CLogit only to those parameters which are identified (based on a first analysis) as violating the logical constraint that $\hat{\varphi}_i \leq \hat{\lambda}_i$ might appear rather *post hoc*. What about *a priori* applying a CLogit link to *all* of the φ and λ parameters? We'll see in a moment why this is **not** a good idea. For now, we'll plunge ahead as if it were.

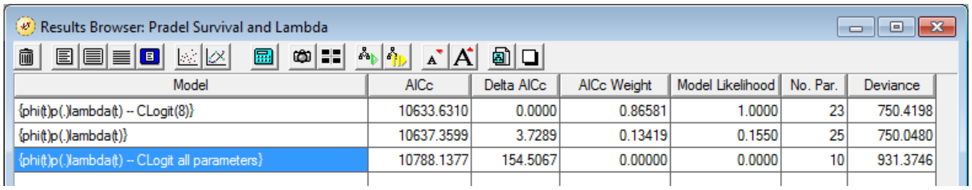
Here is the completed link specification window if we apply the CLogit link function to all successive pairs of φ and λ parameters:



Specify Parameter-Specific Link Function Values for {phi(t)p(.)lambda(t)}

1:Phi CLogit(1,1)
2:Phi CLogit(2,1)
3:Phi CLogit(3,1)
4:Phi CLogit(4,1)
5:Phi CLogit(5,1)
6:Phi CLogit(6,1)
7:Phi CLogit(7,1)
8:Phi CLogit(8,1)
9:Phi CLogit(9,1)
10:Phi CLogit(10,1)
11:Phi CLogit(11,1)
12:Phi CLogit(12,1)
13:p Sin
14:Lambda CLogit(1,2)
15:Lambda CLogit(2,2)
16:Lambda CLogit(3,2)
17:Lambda CLogit(4,2)
18:Lambda CLogit(5,2)
19:Lambda CLogit(6,2)
20:Lambda CLogit(7,2)
21:Lambda CLogit(8,2)
22:Lambda CLogit(9,2)
23:Lambda CLogit(10,2)
24:Lambda CLogit(11,2)
25:Lambda CLogit(12,2)

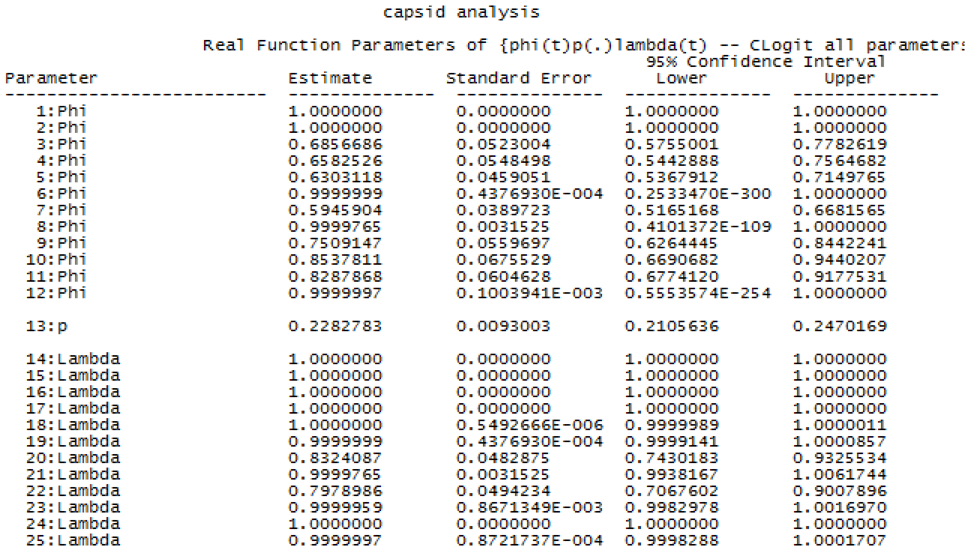
Pay attention to how successive pairs of parameters are indexed – the first argument in `CLogit(n,p)` is n = 'which pair?', while p = 'which parameter in the pair?'. Click the '**OK**' button, and add the results to the browser.



Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance
{phi(t)p(.)lambda(t) -- CLogit(8)}	10633.6310	0.0000	0.86581	1.0000	23	750.4198
{phi(t)p(.)lambda(t)}	10637.3599	3.7289	0.13419	0.1550	25	750.0480
{phi(t)p(.)lambda(t) -- CLogit all parameters}	10788.1377	154.5067	0.00000	0.0000	10	931.3746

Hmmm. Something has definitely 'gone wrong'. The first two models in the browser are the model with and without the CLogit link constraint applied to $[\varphi_8, \lambda_8]$, respectively. However, when we apply the CLogit link to *all* of the φ and λ parameters, we see that not only is the model deviance quite different (931.37 versus $\approx$ 750), but the number of estimated parameters is **much** lower (10 versus 23 & 25).

If we look at the estimates (below),



```
                                     capsid analysis

                   Real Function Parameters of {phi(t)p(.)lambda(t) -- CLogit all parameter
                                                                  95% Confidence Interval
Parameter                  Estimate       Standard Error      Lower           Upper
-------------------------  --------------  --------------  --------------  --------------
   1:Phi                   1.0000000       0.0000000       1.0000000       1.0000000
   2:Phi                   1.0000000       0.0000000       1.0000000       1.0000000
   3:Phi                   0.6856686       0.0523004       0.5755001       0.7782619
   4:Phi                   0.6582526       0.0548498       0.5442888       0.7564682
   5:Phi                   0.6303118       0.0459051       0.5367912       0.7149765
   6:Phi                   0.9999999       0.4376930E-004  0.2533470E-300  1.0000000
   7:Phi                   0.5945904       0.0389723       0.5165168       0.6681565
   8:Phi                   0.9999765       0.0031525       0.4101372E-109  1.0000000
   9:Phi                   0.7509147       0.0559697       0.6264445       0.8442241
  10:Phi                   0.8537811       0.0675529       0.6690682       0.9440207
  11:Phi                   0.8287868       0.0604628       0.6774120       0.9177531
  12:Phi                   0.9999997       0.1003941E-003  0.5553574E-254  1.0000000

  13:p                     0.2282783       0.0093003       0.2105636       0.2470169

  14:Lambda                1.0000000       0.0000000       1.0000000       1.0000000
  15:Lambda                1.0000000       0.0000000       1.0000000       1.0000000
  16:Lambda                1.0000000       0.0000000       1.0000000       1.0000000
  17:Lambda                1.0000000       0.0000000       1.0000000       1.0000000
  18:Lambda                1.0000000       0.5492666E-006  0.9999989       1.0000011
  19:Lambda                0.9999999       0.4376930E-004  0.9999141       1.0000857
  20:Lambda                0.8324087       0.0482875       0.7430183       0.9325534
  21:Lambda                0.9999765       0.0031525       0.9938167       1.0061744
  22:Lambda                0.7978986       0.0494234       0.7067602       0.9007896
  23:Lambda                0.9999959       0.8671349E-003  0.9982978       1.0016970
  24:Lambda                1.0000000       0.0000000       1.0000000       1.0000000
  25:Lambda                0.9999997       0.8721737E-004  0.9998288       1.0001707
```

we see that the 'problem' is that many of the estimates of λ are estimated at the boundary, even though λ was previously estimated as > 1 for many of the estimates (see listing of estimates on p. 12).

What has happened? Well, the answer is somewhat explicit in the name of the link function: Clogit. The cumulative logit function is still a *logit* link, meaning, it constrains estimates to be bounded $[0, 1]$. While this isn't a problem for some parameters, it is clearly a problem for parameters such as λ, which are not upper-bounded at 1. So, you could/should apply the CLogit link function to enforce the constraint that $\varphi_i \leq \lambda_i$ only for those pairs of parameters where $\lambda_i \leq 1$.

end sidebar

13.5. Extensions using the φ and f parametrization...

Suppose you are interested in the relative degree to which recruitment or survival influences the growth of a population. Formally, you want to determine the degree to which λ is 'sensitive' to variation in one or more elements of the demography of the population. Such a sensitivity analysis is usually expressed in terms of the rate of change of λ given a certain change in one of the matrix elements (a_{ij}). Done on a log scale, this 'sensitivity' analysis becomes an 'elasticity' analysis (the log scale expresses relative proportional contributions to λ). Now, in the typical 'sensitivity' or 'elasticity' analysis, the point is to determine what *would* happen to growth if a specified change is made in one or more vital rates. So, a *prospective* analysis. In the prospective context, sensitivity and elasticity are together referred to as 'perturbation' techniques, since the goal is to see how much projected growth would change *in the future* as the system is 'perturbed' by changing one of the vital rates which contributes to population growth. There is a very large literature on the application of prospective perturbation techniques in conservation and population management (Caswell 2001 being the canonical reference).

However, in the *retrospective* context, the story is a little bit different. In this situation, we have an estimated time series of $\hat{\lambda}_i$, which we might estimate from our mark-encounter data. We want to know what the relative contributions of $\hat{\varphi}_i$ and $\hat{f}_i$ have been to the observed variation in $\hat{\lambda}_i$. In other words, what *has* driven the estimated pattern of variation in population growth over time.

Jim Nichols and colleagues addressed this very question (Nichols *et al*. 2000). The approach they developed is very intuitive, and the result is rather elegant. The basic idea is that since γ_{i+1} is the probability that an individual in the population at time i+1 (and thus contributing to N_{i+1}) was also there at time i (and thus included in N_i), and since $N_{i+1} = N_i\varphi_i + B_i$, then

$$N_i\varphi_i = N_{i+1}\gamma_{i+1} \qquad \text{and} \qquad B_i = \left(1-\gamma_{i+1}\right)N_{i+1}.$$

Since $\lambda_i = N_{i+1}/N_i$, then

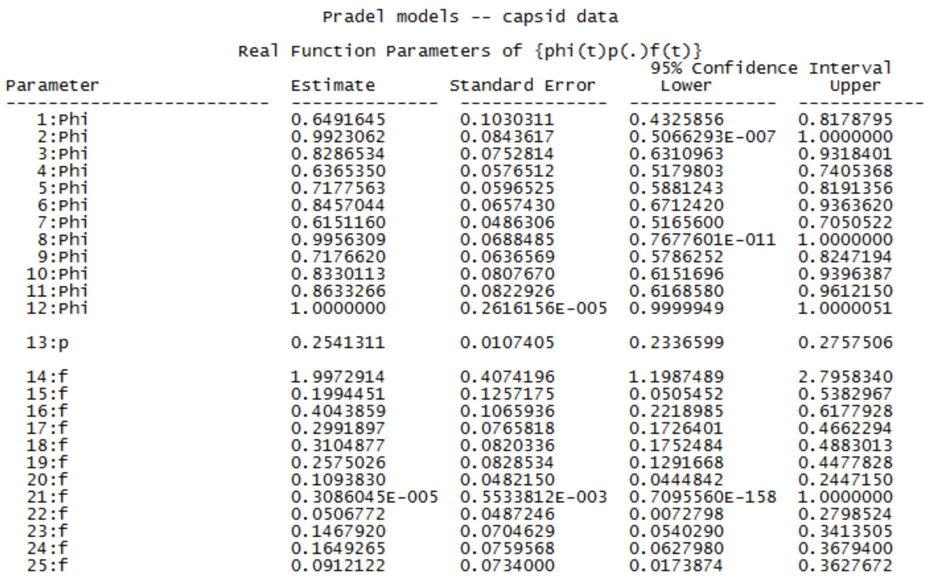
$$E(\lambda_i) = \frac{\left[\gamma_{i+1}N_{i+1} + \left(1-\gamma_{i+1}\right)N_{i+1}\right]}{E(N_i)}.$$

Since the abundance term, N_{i+1}, is the same for both product terms in the numerator, then the γ_{i+1} in the first term in the numerator is interpretable as the contribution of survivors from time i to time i+1, while the $(1-\gamma_{i+1})$ in the second term of the numerator is the contribution of new recruits into the population. So, the two terms give the proportional contributions of survivors and new recruits to λ, which is conceptually analogous to the elasticities of both terms. The details (and several very clever extensions to multi-state and robust design models) are discussed at length in the Nichols paper.

However, this approach involves 'doing algebra' with estimates of γ only. A more direct, and perhaps more flexible approach is to re-parameterize the likelihood in terms of survival and recruitment directly. This is the basis of the '**`Survival and Recruitment`**' Pradel model implemented in **MARK**.

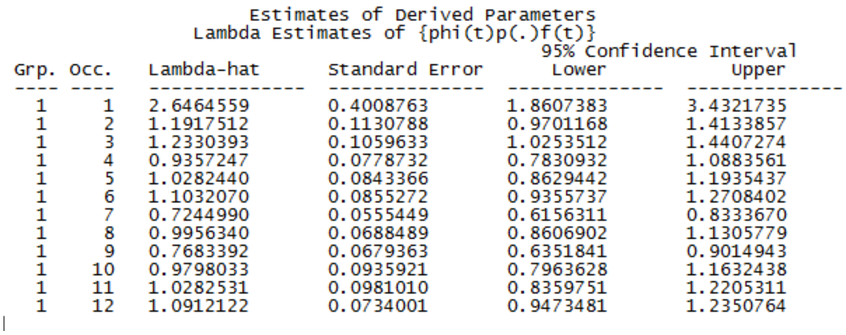
Re-start the analysis of the capsid data set, this time selecting the '**`Survival and Recruitment`**' data type. If you look at the PIM chart, you'll see that there are only 3 structural parameters: φ, p and f. For this model type, the realized growth λ is estimated as a *derived parameter* (meaning, it is derived 'by algebra', outside the likelihood). This has one immediate implication, which refers back to our earlier discussion of applying constraints to Pradel models. Because λ is a derived parameter (not in the likelihood), you cannot put a constraint on λ. Alternatively, you might apply constraints to the underlying demographic processes which contribute to λ (i.e., φ and f).

Let's proceed by fitting a single Pradel model to the capsid data: model $\{\varphi_t\ p_{\cdot}\ f_t\}$. We'll use constant encounter probabilities over time to eliminate some of the confounding problems inherent in fully time-dependent models. Here are the reconstituted parameter estimates from this model



```
                               Pradel models -- capsid data

                       Real Function Parameters of {phi(t)p(.)f(t)}
                                                               95% Confidence Interval
Parameter                   Estimate       Standard Error       Lower           Upper
--------------------------  --------------  --------------  --------------  ------------
   1:Phi                    0.6491645       0.1030311        0.4325856       0.8178795
   2:Phi                    0.9923062       0.0843617        0.5066293E-007  1.0000000
   3:Phi                    0.8286534       0.0752814        0.6310963       0.9318401
   4:Phi                    0.6365350       0.0576512        0.5179803       0.7405368
   5:Phi                    0.7177563       0.0596525        0.5881243       0.8191356
   6:Phi                    0.8457044       0.0657430        0.6712420       0.9363620
   7:Phi                    0.6151160       0.0486306        0.5165600       0.7050522
   8:Phi                    0.9956309       0.0688485        0.7677601E-011  1.0000000
   9:Phi                    0.7176620       0.0636569        0.5786252       0.8247194
  10:Phi                    0.8330113       0.0807670        0.6151696       0.9396387
  11:Phi                    0.8633266       0.0822926        0.6168580       0.9612150
  12:Phi                    1.0000000       0.2616156E-005   0.9999949       1.0000051

  13:p                      0.2541311       0.0107405        0.2336599       0.2757506

  14:f                      1.9972914       0.4074196        1.1987489       2.7958340
  15:f                      0.1994451       0.1257175        0.0505452       0.5382967
  16:f                      0.4043859       0.1065936        0.2218985       0.6177928
  17:f                      0.2991897       0.0765818        0.1726401       0.4662294
  18:f                      0.3104877       0.0820336        0.1752484       0.4883013
  19:f                      0.2575026       0.0828534        0.1291668       0.4477828
  20:f                      0.1093830       0.0482150        0.0444842       0.2447150
  21:f                      0.3086045E-005  0.5533812E-003   0.7095560E-158  1.0000000
  22:f                      0.0506772       0.0487246        0.0072798       0.2798524
  23:f                      0.1467920       0.0704629        0.0540290       0.3413505
  24:f                      0.1649265       0.0759568        0.0627980       0.3679400
  25:f                      0.0912122       0.0734000        0.0173874       0.3627672
```

and the reconstituted estimates of λ (on the real scale):



```
                         Estimates of Derived Parameters
                     Lambda Estimates of {phi(t)p(.)f(t)}
                                                     95% Confidence Interval
 Grp. Occ.    Lambda-hat        Standard Error        Lower            Upper
 ---- ----   --------------    --------------    --------------   --------------
   1     1   2.6464559         0.4008763         1.8607383        3.4321735
   1     2   1.1917512         0.1130788         0.9701168        1.4133857
   1     3   1.2330393         0.1059633         1.0253512        1.4407274
   1     4   0.9357247         0.0778732         0.7830932        1.0883561
   1     5   1.0282440         0.0843366         0.8629442        1.1935437
   1     6   1.1032070         0.0855272         0.9355737        1.2708402
   1     7   0.7244990         0.0555449         0.6156311        0.8333670
   1     8   0.9956340         0.0688489         0.8606902        1.1305779
   1     9   0.7683392         0.0679363         0.6351841        0.9014943
   1    10   0.9798033         0.0935921         0.7963628        1.1632438
   1    11   1.0282531         0.0981010         0.8359751        1.2205311
   1    12   1.0912122         0.0734001         0.9473481        1.2350764
```

To reinforce the relationship between survival, recruitment, and λ, let's compare (i) the derived estimate of λ provided by **MARK** (above), and (ii) the value of the sum of the estimates of survival and recruitment.

Since

$$\lambda_i = \varphi_i + \frac{B_i}{N_i} = \varphi_i + f_i,$$

then (i) and (ii) should be identical, as they are shown to be (to within rounding error) in the following table for a sub-sample of intervals:

interval	*derived* $\hat{\lambda}$	$\hat{\varphi}$	$\hat{f}$	$\hat{\varphi} + \hat{f}$
2	1.1918	0.9923	0.1994	1.1918
3	1.2330	0.8287	0.4044	1.2330
4	0.9357	0.6365	0.2992	0.9357
5	1.0282	0.7178	0.3105	1.0282
6	1.1032	0.8457	0.2575	1.1032
7	0.7245	0.6151	0.1094	0.7245

We could express the proportional contribution of (say) survival φ to realized growth λ simply as $\varphi/(\varphi + f)$. The variance of this proportion could be estimated using the Delta method (Appendix B). ¶

How do these calculated proportions compare to those based on interpreting γ_{i+1} as the proportion of λ due to survival of individuals from $N_i \rightarrow N_{i+1}$, and $(1 - \gamma_{i+1})$ as the proportion due to new recruits (*sensu* Nichols *et al*. 2000)?

At the top of the next page, we re-tabulate the first couple of estimates of $\hat{\varphi}_i$ and $\hat{f}_i$, using the '`Survival and Recruitment`' data type (above), and also include estimates of $\hat{\gamma}_{i+1}$ using the '`Survival and Seniority`' data type, using model $\{\varphi_t\, p_.\, \gamma_t\}$.

¶ If we let $\theta = \varphi/(\varphi + f)$, then

$$\widehat{\text{var}}(\hat{\theta}) \approx \left[\left(\frac{\hat{f}}{(\hat{\varphi}+\hat{f})^2}\right) \quad \left(-\frac{\hat{\varphi}}{(\hat{\varphi}+\hat{f})^2}\right)\right] \begin{bmatrix} \widehat{\text{var}}(\hat{\varphi}) & \widehat{\text{cov}}(\hat{\varphi},\hat{f}) \\ \widehat{\text{cov}}(\hat{f},\hat{\varphi}) & \widehat{\text{var}}(\hat{f}) \end{bmatrix} \begin{bmatrix} \dfrac{\hat{f}}{(\hat{\varphi}+\hat{f})^2} \\ -\dfrac{\hat{\varphi}}{(\hat{\varphi}+\hat{f})^2} \end{bmatrix}.$$

where the variance and covariance of $\hat{\varphi}$ and $\hat{f}$ can be output from **MARK** (see Appendix B).

interval	*derived* $\hat{\lambda}$	$\hat{\varphi}$	$\hat{f}$	$\hat{\varphi}/\hat{\lambda}$	$\hat{f}/\hat{\lambda}$	$\hat{\gamma}_{i+1}$	$(1-\hat{\gamma}_{i+1})$
2	1.1918	0.9923	0.1994	0.8326	0.1673	0.8327	0.1673
3	1.2330	0.8287	0.4044	0.6721	0.3279	0.6720	0.3280
4	0.9357	0.6365	0.2992	0.6802	0.3198	0.6803	0.3196
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

We see clearly that the proportional contribution of survival φ or recruitment f to realized growth λ, calculated as φ/λ or f/λ, is entirely equivalent to using γ_{i+1} and $(1-\gamma_{i+1})$, respectively.

By considering the proportional contribution of survival and recruitment to λ, we can interpret these parameters as non-asymptotic analogs of sensitivity and elasticity. Thus, for example, we might consider how much population growth might decrease if we reduced survival by some factor δ. We see clearly from above that it would be reduced by $\delta\varphi$, which is equivalent to $\delta\gamma_{i+1}$.

One final point: all we've done up until now is talk about *net* 'additions' and 'subtractions' to the population. We haven't partitioned these any further. For example, we haven't partitioned additions into '*in situ* recruits' and 'immigrants'. We may, in fact, not be satisfied with simply using a 'summary accounting' like 'total recruits' or 'total subtractions' – we may want to know how many of each are due to underlying, lower-level processes (like births, immigration, deaths, or emigration). However, to do that, we'd need to consider different approaches: the Jolly-Seber (and related) models (introduced in Chapter 12), and the robust design (which we introduce in Chapter 16). But, if partitioning λ into summary contributions of total recruits and total losses is what you're after, then the Pradel models may be of some use. For a comprehensive review of time-symmetric open models and their application to a number of interesting biological questions, see Nichols (2016).

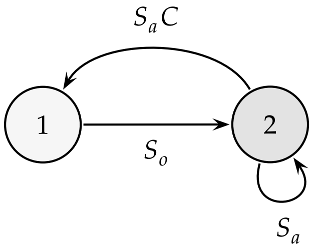
`begin sidebar`

λ = 'survival' + 'recruitment': be careful!

Another example of a potential pitfall. In the preceding, we made use of the fact that realized λ can be estimated as the sum of survival and per capita recruitment, both estimated over a given interval.

However, you need to be careful in how you are 'handling' recruitment. Consider, for example, a population with 2 age classes: babies, and adults. Assume that adults start breeding at age 2 years (i.e., they don't breed as yearlings), and on average produce C babies. Babies survive with probability S_o, and thus become adults, which survive with probability S_a.

Assuming the population is censused after breeding, the population can be described by the life-cycle graph shown below:



In the graph, node 1 refers to babies (offspring), and node 2 refers to adults (age 1 yr and older). The self-loop on node 2 indicates that survival does not vary with age among adults. The fertility arc (connecting node 2 back to node 1) represents the expected contribution of each individual in node 2 at time (i) to the baby age class (node 1) at time (i+1).

From the life cycle graph (preceding page), we can derive the corresponding projection matrix

$$\mathbf{A} = \begin{bmatrix} 0 & S_a C \\ S_o & S_a \end{bmatrix}.$$

Assume that $C = 0.42$, $S_a = 0.9$, and $S_o = 0.5$. Thus, the projection matrix **A** is

$$\mathbf{A} = \begin{bmatrix} 0 & S_a C \\ S_o & S_a \end{bmatrix} = \begin{bmatrix} 0.000 & 0.378 \\ 0.500 & 0.900 \end{bmatrix},$$

from which we can determine that projected $\lambda = 1.0757$. [The use of life cycle diagrams, projection matrices, and various metrics extracted from such matrices, is discussed in most modern texts on population biology – Caswell (2001) is the standard reference].

OK, but what if you had used a different approach, based on the logic underlying the derivation of the Pradel models? In other words, $\lambda_i = \varphi_i + f_i$. Assume we know that $(\varphi_i =) S_a = 0.90$. That would appear to be half of our equation for λ. What about recruitment, f_i? For the Pradel models, we're interested in recruitment to the adult age class – the number of individuals entering the adult population between (i) and (i+1) for each individual adult at (i).

If you stare at the life-cycle graph on the preceding page, it might seem obvious that recruitment is simply the number of babies who become adults. True – but how many babies are there? Recall that we're estimating growth rate λ without having estimates of abundance. Well, as a first stab at the answer, you might think that the number of babies surviving to adulthood is a function of how many babies are produced by current adults (which is $S_a.C.S_o$; because this is a post-breeding census, you pre-multiply by S_a since a current adult has to survive in order to produce babies next year – this is indicated by the product $S_a.C$ on the fertility arc connecting node 2 back to node 1 in the life-cycle diagram). So, you might try to derive λ as $\lambda = S_a + S_a.C.S_o = 0.9 + 0.189 = 1.089$.

Unfortunately, this value ($\lambda = 1.089$) is not the same as the 'true' value of projected growth rate derived from the projection matrix ($\lambda = 1.0757$). Why the difference? The difference is due to the fact that recruitment between (i) and (i+1) is a function of how many babies there were at time (i). The product $S_a.C.S_o$ gives the projected recruitment between (i+1) and (i+2)! Why? Look carefully – the product $S_a.C.S_o$ covers two time intervals: one for current adults (S_a), and one for babies produced next year by those adults (S_o).

So, how would you solve this problem? Fairly easily – you simply need to remember that for an exponentially growing population, ΔN for any age class over (t) time intervals is simply $N\lambda^t$. Similarly, since the projection of an exponentially growing population is time-symmetric, you could also project backwards, and say that the size of the population (t) time units in past is simply $N\lambda^{-t}$.

Which is important...because?? It's important because you want to know how many of babies at time (i) will recruit (become adults) between (i) and (i+1). Since the product $S_a.C.S_o$ in fact gives recruitment 2 time steps ahead, what you need to do is 'back-step' this product by 1 time step, which (as noted) is given simply by λ^{-1}. For our numerical example, where λ (from the matrix) is given as 1.0757, then $(1.0757)^{-1} = 0.92963$. So, we correct (or 'back-step') our recruitment term as $(0.92963).S_a.C.S_o = 0.1757$. Thus, $\lambda = 0.9 + 0.1757 = 1.0757$, which is exactly the same value we got from the projection matrix.

OK – admittedly a somewhat 'artificial' problem, but remember: although the basic logic underlying the temporal symmetry approach to estimating λ is relatively simple, you do need to pay close attention to what is going on.

end sidebar

13.6. 'average' realized growth rate

Following estimation of the time-specific realized growth rates, λ_i, there is natural interest in the average growth rate over the course of the study. You might recall from section 6.14 in Chapter 6 that estimating the 'average' for a parameter can be somewhat more complicated than you might expect.

This is especially true in the present situation, where we are interested in estimating $\bar{\hat{\lambda}}$. Here, the complication is that we’re calculating the mean of the ratio of successive population sizes, where the population sizes at each time step are outcomes of an underlying, geometric stochastic generating process. As such, the most appropriate ‘average’ to report is the *geometric* mean of the individual $\hat{\lambda}_i$, not the more familiar *arithmetic* mean.

You might recall that the geometric mean of a set of n numbers $\{x_1, x_2, \ldots, x_n\}$ is given as

$$y = \sqrt[n]{\prod_{i=1}^{n} x_i}.$$

An important result is that unless the set $\{x_1, x_2, \ldots, x_n\}$ are all the same number (i.e., $\{x_1 = x_2 = \cdots = x_n\}$), then the geometric average is always less than the arithmetic average. [*Note*: if the geometric mean, in general, is new to you, it is worth consulting the following `-sidebar-` before proceeding much further.]

`begin sidebar`

arithmetic mean, geometric mean, and population growth...

The first ‘statistical’ calculation you usually learn how to do is the computation of an ‘average’. This is how the teacher comes up with grades (in many cases) so it’s a basic bit of math most students have an inherent interest in.

What you typically learned to calculate is what is known as the *arithmetic average*. For example, given three random values x_1, x_2, and x_3 then arithmetic mean y_{AM} is

$$y_{AM} = \frac{x_1 + x_2 + x_3}{3}.$$

However, one of the most important averages in ecology is something known as the *geometric mean*, or *geometric average*, which given the same three random values x_1, x_2, and x_3 is

$$y_{GM} = \sqrt[3]{x_1 x_2 x_3}.$$

In other words, the geometric mean of three numbers is the cube-root of their product.

If you have n numbers, the geometric mean is

$$\begin{aligned} y_{GM} &= \sqrt[n]{\prod_{i=1}^{n} x_i} \\ &= e^{\left(\frac{1}{n}\sum\left(\log(x)\right)\right)}. \end{aligned}$$

For example, let $x_1 = 2$, $x_2 = 2.5$, and $x_3 = 4$. Given these values, the arithmetic mean is $(2 + 2.5 + 4)/3 = 2.833$, whereas the geometric mean is

$$\begin{aligned} y_{GM} &= \sqrt[3]{2 \times 2.5 \times 4} \\ &= \sqrt[3]{20} \\ &= 2.714. \end{aligned}$$

Consider the simplest possible case of two numbers: x_1 and x_2. Unless x_1 and x_2 are the same number, then *the geometric average is always less than the arithmetic average*.

Here is a proof by induction of this inequality of the two means. Assume we have 2 values: x_1 and x_2. Assume that $x_1 \neq x_2$, and thus, $x_1 - x_2 \neq 0$. The proposition is that

$$\frac{x_1 + x_2}{2} \geq \sqrt{x_1 x_2}.$$

Since $x_1, x_2 > 0$, then we can write

$$\begin{aligned} \left(\sqrt{x_1} - \sqrt{x_2}\right)^2 &\geq 0 \\ x_1 - 2\sqrt{x_1 x_2} + x_2 &\geq 0. \end{aligned}$$

Factoring the $\sqrt{x_1 x_2}$ term yields

$$\frac{x_1 + x_2}{2} \geq \sqrt{x_1 x_2},$$

where the LHS is the arithmetic mean of x_1 and x_2, and the RHS is the geometric mean of the same two values. The LHS (arithmetic mean) is greater than the RHS (geometric mean). Q.E.D

end sidebar

Now, why do we care about the distinction between an arithmetic and geometric mean? We care because the geometric mean is the appropriate average for stochastic population growth. Consider two successive years of population growth, with λ_1 being the growth factor for the first year, and λ_2 being the growth factor for the second year. Then, the population size after the first year, starting at size N_0, is

$$N_1 = \lambda_1 N_0 \qquad \leftarrow \text{after 1 year}$$

while after 2 years, the population size is given as

$$\begin{aligned} N_2 &= \lambda_2 N_1 \\ &= \lambda_2 \lambda_1 N_0 \quad \leftarrow \text{after 2 years} \end{aligned}$$

Thus, the λ_i values for each year i are simply multiplied together when projecting of geometric growth through time.

What can we use this for? Well, let's step back a moment, and recall from some population biology class you might have taken that under *time-invariance*, such that λ is constant over time, we can write

$$N_t = N_0 \lambda^t.$$

Let $t = 3$. Thus

$$\begin{aligned} N_3 &= N_0 \lambda^3 \\ &= N_0 \cdot \left(\lambda \cdot \lambda \cdot \lambda\right). \end{aligned}$$

Clearly, in this case the growth rate over each year is the same, and is given by λ. So, the expected change in size over $t = 3$ time steps is simply the average annual growth rate, raised to the 3^{rd} power. And, thus in reverse, the average growth rate in a given year would simply be the 3^{rd} root of the product of the individual λ values. For example, if $N_0 = 10$, and $N_3 = 11.57625$, then since $N_t = N_0 \lambda^t$ under time invariance, then $\lambda^3 = (11.57625/10) = 1.157625$, and thus the average growth rate (again, assuming time invariance) is simply $\sqrt[3]{1.157625} = 1.05$.

Now, consider the more typical case where λ varies from year to year. If λ_1 is the geometric growth factor for the first year, λ_2 is the geometric growth factor for the second year, and λ_3 is the geometric growth factor for the third year then

$$N_3 = \lambda_3\lambda_2\lambda_1 N_0.$$

So, what is the *average* growth rate over the 3 years? Consider the following simple numerical example: let $\lambda_1 = 1.05$, $\lambda_2 = 1.01$, and $\lambda_3 = 0.98$. The *arithmetic* mean of these three growth rates is 1.013. Is this the appropriate growth rate to project population size in 3 years?

Let’s see what happens. Let N_0=10. Thus, after three years, we project the population size in 3 years will be $N_3 = (10 \cdot 1.05 \cdot 1.01 \cdot 0.98) = 10.393$. But, if instead we had used the *arithmetic* average in the projection equation, we would project the population size after three time steps to be $(10 \cdot 1.013 \cdot 1.013 \cdot 1.013) = 10.405$, which is higher than it should be $(10.405 > 10.393)$.

But, what if instead we use the *geometric* mean? The geometric mean of our annual growth rates is

$$\begin{aligned} &\sqrt[3]{(1.05 \times 1.01 \times 0.98)} \\ &= \sqrt[3]{1.03929} = 1.01293. \end{aligned}$$

So, the projected size of the population after three time steps, using the *geometric* mean growth rate, would be $(10 \times 1.01293 \times 1.01293 \times 1.01293) = 10.393$, which is **exactly** what it should be, based on our earlier projection using the individual λ_i values. The constant-environment equivalent of the fluctuating environment is an environment with a constant λ that is the geometric average of the λ’s in the fluctuating environment. We refer to this mean stochastic growth rate as λ_s. And, as noted earlier, the geometric mean is always smaller than the arithmetic mean, and so, we expect that the *stochastic growth rate will be lower than the deterministic growth rate*.

OK, back to Pradel models, and calculating the most appropriate ‘average’ growth rate over the time series of estimated growth rates, $\hat{\lambda}_i$. From the preceding, we see that using a simple arithmetic mean calculated over the set of $\hat{\lambda}_i$ would be incorrect – it would overestimate the stochastic growth rate, which is more appropriately estimated using the geometric mean.

How can we estimate the geometric mean growth rate using **MARK**? In fact, you have two options. You can either (i) derive estimates of the geometric mean over the set of ML estimates of $\hat{\lambda}_i$ by hand (a fairly straightforward exercise using a spreadsheet of the estimates), or, (ii) you can use *derived estimates* of λ on the log scale, which **MARK** generates automatically.

Why is this second approach useful? Recall again from some earlier population biology class that $\lambda = e^r$, where r is the instantaneous (intrinsic) rate of growth, whereas λ is the ratio of population sizes at 2 discrete points in time.

We can write the stochastic growth rate λ_s over T time steps as

$$\begin{aligned} \lambda_s &= \left(\lambda_1 \times \lambda_2 \times \lambda_3 \cdots \times \lambda_T\right)^{1/T} \\ &= \left(e^{r_1} \times e^{r_2} \times e^{r_3} \cdots \times e^{r_T}\right)^{1/T}, \end{aligned}$$

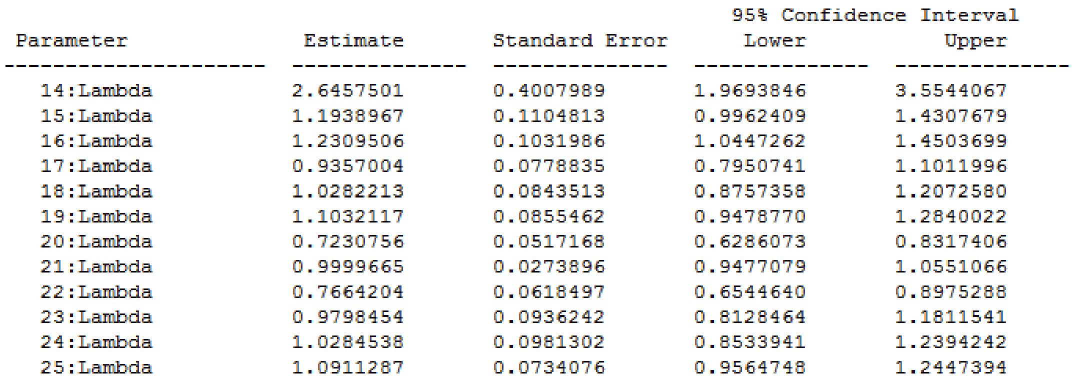
which, taking logs, can be rewritten as

$$\ln \lambda_s = \frac{r_1 + r_2 + r_3 \cdots + r_T}{T}.$$

So, we can calculate $\ln(\lambda_s)$ simply as the arithmetic mean $\bar{r}$.

Let's explore application of these algebraic relationships using the capsid data set we introduced earlier in this chapter. Using the '**`Pradel Survival and Lambda`**' data type, we'll fit model $\{\varphi_t\, p_{\cdot}\, \lambda_t\}$ to the data, using the CLogit link applied to parameters φ_8 and λ_8 to enforce the logical constraint that $\varphi_i \leq \lambda_i$ (this was discussed in the `-sidebar-` starting on p. 11 of this chapter).

The estimates of realized growth rate, $\hat{\lambda}_i$ for the 12 intervals for the capsid data set are shown below. For our subsequent calculations, we'll exclude the estimate over the first interval, $\hat{\lambda}_1 = 2.6457$, as being 'suspicious' (see earlier discussion on the potential for bias and confounding for initial estimates from Pradel models).



```
                                                              95% Confidence Interval
 Parameter                   Estimate       Standard Error      Lower           Upper
-------------------------  --------------  --------------  --------------  --------------
   14:Lambda                2.6457501       0.4007989       1.9693846       3.5544067
   15:Lambda                1.1938967       0.1104813       0.9962409       1.4307679
   16:Lambda                1.2309506       0.1031986       1.0447262       1.4503699
   17:Lambda                0.9357004       0.0778835       0.7950741       1.1011996
   18:Lambda                1.0282213       0.0843513       0.8757358       1.2072580
   19:Lambda                1.1032117       0.0855462       0.9478770       1.2840022
   20:Lambda                0.7230756       0.0517168       0.6286073       0.8317406
   21:Lambda                0.9999665       0.0273896       0.9477079       1.0551066
   22:Lambda                0.7664204       0.0618497       0.6544640       0.8975288
   23:Lambda                0.9798454       0.0936242       0.8128464       1.1811541
   24:Lambda                1.0284538       0.0981302       0.8533941       1.2394242
   25:Lambda                1.0911287       0.0734076       0.9564748       1.2447394
```

Now, how do we calculate the best estimate of 'average' growth rate? From the preceding discussion, we know that the simple arithmetic mean of the set of estimates $\hat{\lambda}_2 \rightarrow \hat{\lambda}_{12}$ (1.0074) would be incorrect as a measure of this average growth, since it would overestimate the average expected stochastic growth of the population over any given interval.

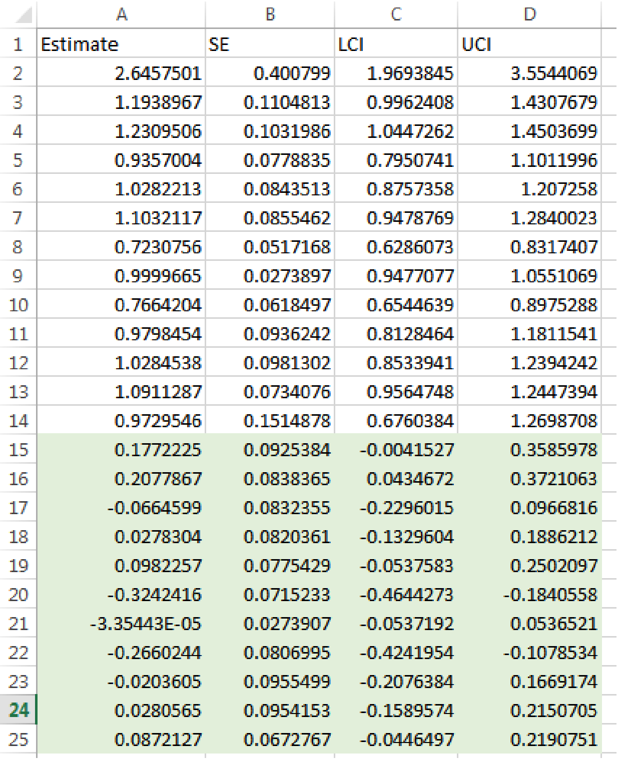
Instead, we should focus on the *geometric* mean of the estimates. We could simply take our set of estimates $\hat{\lambda}_i$, export them to a spreadsheet, and calculate the geometric mean 'by hand'. Alternatively, and perhaps more conveniently, we could export the derived estimates of $\ln(\hat{\lambda}_i)$ from **MARK**, and take the arithmetic mean of these derived estimates, recalling that $\ln(\lambda_i) = r_i$, and that $\ln(\lambda_s)$ is simply the arithmetic mean $\bar{r}$. Recall that **MARK** generates estimates of $\ln(\hat{\lambda}_i)$ as *derived* parameters for all of the Pradel data types, which makes this approach very straightforward.

Simply select '**`Output | Specific Model Output | Parameter Estimates | Derived Estimates | Copy to Excel`**'. The exported estimates are shown at the top of the next page. The first 12 rows are the estimates of $\hat{\lambda}_i$, while the next 12 rows are the same estimates, but reported on the log scale, $\ln(\hat{\lambda}_i)$. For example, $\ln(\hat{\lambda}_2) = 0.1772225 = \ln(1.1938967)$, where $\hat{\lambda}_2 = 1.1938967$.

We focus here on the set of estimates $\ln(\hat{\lambda}_2) \longrightarrow \ln(\hat{\lambda}_{12})$, shown on the following page with light green shading. The *arithmetic* average of this set is $\bar{r} = \ln(\lambda_s) = -0.004616859$. Thus, our estimate of the stochastic, geometric growth rate on the real scale is $\hat{\lambda}_s = e^{(-0.004616859)} = 0.995393783$, which as expected is less than the arithmetic mean of 1.0074. Not only that, the geometric mean suggests a slow decline in population size over the long-term, whereas the arithmetic mean suggests a slow increase.

At this point, you may (and should) also be wondering about the variance of our estimated stochastic growth rate. We could simply take the variance calculated on the log scale, and using the Delta method (see Appendix B), back-transform our estimated variance to the usual real scale.

But, recall from section 6.14 in Chapter 6 that such an approach is generally biased, since it fails to take into account the conditional sampling variances of our time-specific estimates of growth rate. The preferred approach is to use a 'random effects, variance components approach', which could be

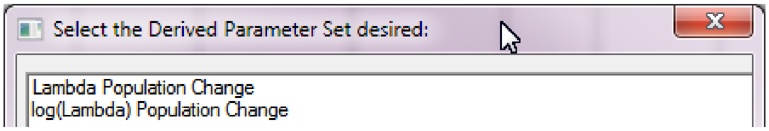


	A	B	C	D
1	Estimate	SE	LCI	UCI
2	2.6457501	0.400799	1.9693845	3.5544069
3	1.1938967	0.1104813	0.9962408	1.4307679
4	1.2309506	0.1031986	1.0447262	1.4503699
5	0.9357004	0.0778835	0.7950741	1.1011996
6	1.0282213	0.0843513	0.8757358	1.207258
7	1.1032117	0.0855462	0.9478769	1.2840023
8	0.7230756	0.0517168	0.6286073	0.8317407
9	0.9999665	0.0273897	0.9477077	1.0551069
10	0.7664204	0.0618497	0.6544639	0.8975288
11	0.9798454	0.0936242	0.8128464	1.1811541
12	1.0284538	0.0981302	0.8533941	1.2394242
13	1.0911287	0.0734076	0.9564748	1.2447394
14	0.9729546	0.1514878	0.6760384	1.2698708
15	0.1772225	0.0925384	-0.0041527	0.3585978
16	0.2077867	0.0838365	0.0434672	0.3721063
17	-0.0664599	0.0832355	-0.2296015	0.0966816
18	0.0278304	0.0820361	-0.1329604	0.1886212
19	0.0982257	0.0775429	-0.0537583	0.2502097
20	-0.3242416	0.0715233	-0.4644273	-0.1840558
21	-3.35443E-05	0.0273907	-0.0537192	0.0536521
22	-0.2660244	0.0806995	-0.4241954	-0.1078534
23	-0.0203605	0.0955499	-0.2076384	0.1669174
24	0.0280565	0.0954153	-0.1589574	0.2150705
25	0.0872127	0.0672767	-0.0446497	0.2190751

implemented using either a ‘moments-based’ approach (Appendix D), or using MCMC (Appendix E).

For (relative) simplicity, we will briefly demonstrate the steps for the using the ‘moments-based’ method. We will fit a random effects ‘**`intercept only`**’ (i.e., mean) model to the capsid data. The ‘**`intercept only`**’ model assumes that the parameters in the model (say, λ_i) are drawn from some underlying distribution specified by mean μ and variance σ^2 (see Appendix D for complete details – our intent here is to demonstrate the mechanics only).

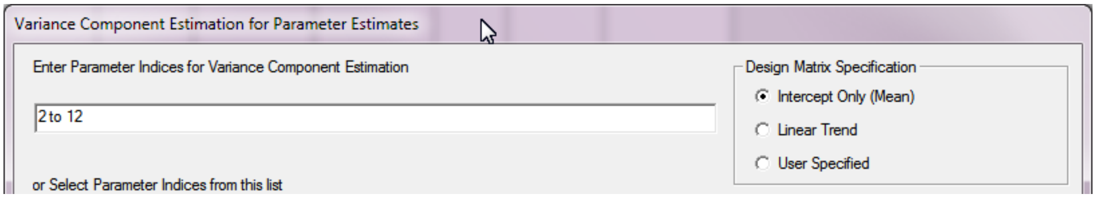
First, retrieve the model $\{\varphi_t\, p_{.}\, \lambda_t\}$ in the browser. Then, select ‘**`Output | Specific Model Output | Variance Components | Derived Parameter Estimates`**’. This will spawn another window (shown below) asking you whether you want to use $\hat{\lambda}_i$ (‘**`Lambda Population Change`**’, or $\ln(\hat{\lambda}_i)$ (‘**`log(Lambda) Population Change`**’) in the calculations.



Although earlier in this section we suggested that working with $\ln(\lambda)$ is more ‘convenient’ in the context of doing subsequent calculations in a spreadsheet, convenience isn’t really the issue here. Here, it is important to think carefully about what the variance components analysis yields for either λ or $\ln(\lambda)$, and based on that, deciding which is the more appropriate.

To that end, we’ll compare both approaches – starting with the variance decomposition for λ. First, we select ‘**`Lambda Population Change`**’.

Then, in the variance component specification window, we specify parameters '**`2 to 12`**', and the '**`Intercept Only (Mean)`**' model:



Variance Component Estimation for Parameter Estimates

Enter Parameter Indices for Variance Component Estimation: 2to 12

Design Matrix Specification

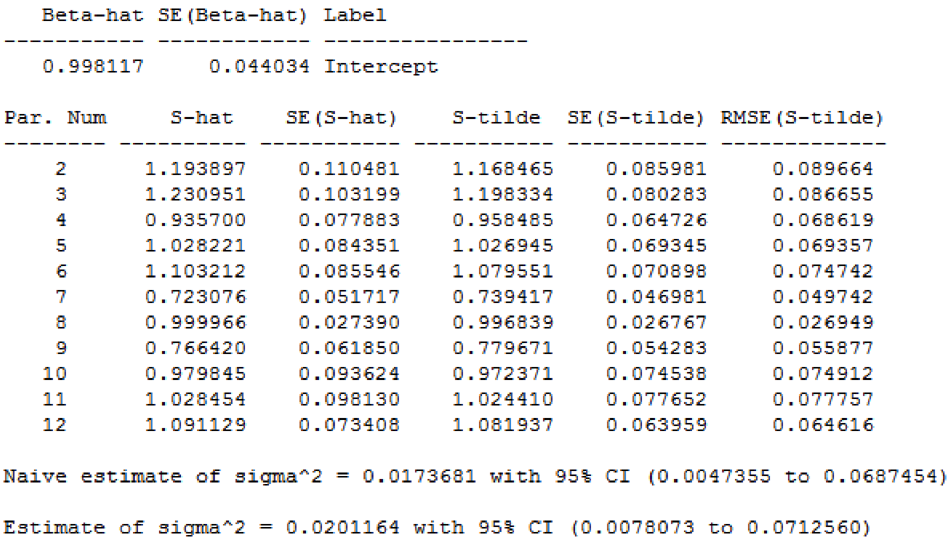
● Intercept Only (Mean)

○ Linear Trend

○ User Specified

or Select Parameter Indices from this list

Here is the output of the variance decomposition of $\hat{\lambda}_2 \rightarrow \hat{\lambda}_{12}$, using the default mean (intercept-only) model:



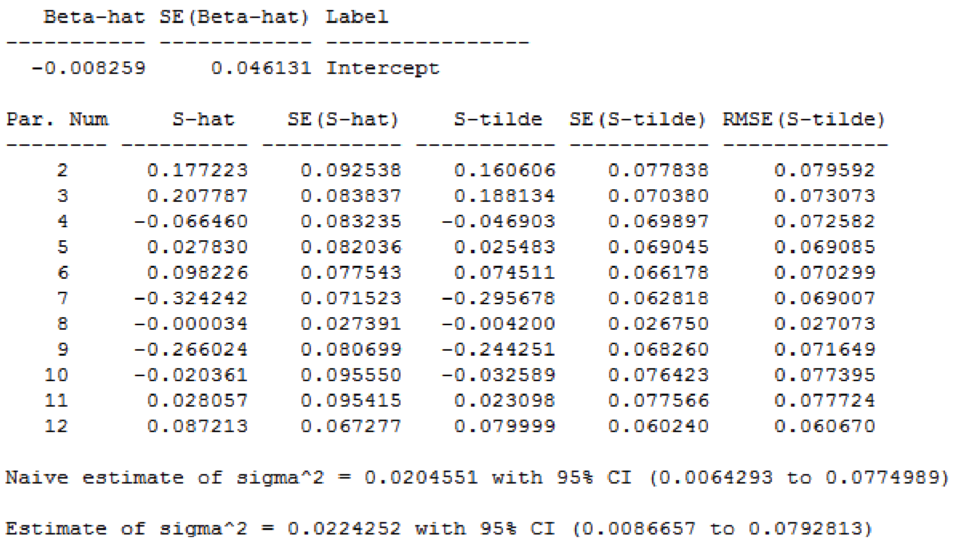
```
   Beta-hat  SE(Beta-hat)  Label
 ----------- ------------ ----------------
   0.998117     0.044034  Intercept

Par. Num       S-hat     SE(S-hat)     S-tilde   SE(S-tilde) RMSE(S-tilde)
-------- ----------- ----------- ----------- ----------- -------------
    2      1.193897    0.110481    1.168465    0.085981     0.089664
    3      1.230951    0.103199    1.198334    0.080283     0.086655
    4      0.935700    0.077883    0.958485    0.064726     0.068619
    5      1.028221    0.084351    1.026945    0.069345     0.069357
    6      1.103212    0.085546    1.079551    0.070898     0.074742
    7      0.723076    0.051717    0.739417    0.046981     0.049742
    8      0.999966    0.027390    0.996839    0.026767     0.026949
    9      0.766420    0.061850    0.779671    0.054283     0.055877
   10      0.979845    0.093624    0.972371    0.074538     0.074912
   11      1.028454    0.098130    1.024410    0.077652     0.077757
   12      1.091129    0.073408    1.081937    0.063959     0.064616

Naive estimate of sigma^2 = 0.0173681 with 95% CI (0.0047355 to 0.0687454)

Estimate of sigma^2 = 0.0201164 with 95% CI (0.0078073 to 0.0712560)
```

The value $\hat{\beta} = 0.998117$ reported near the top of the output is, in fact, quite close to the *arithmetic* mean of the shrinkage estimates $\tilde{S}_i$ (labeled '**`S-tilde`**' in the output): $\bar{\tilde{S}}_i = 1.0024$ (the reason for the slight difference between the two is discussed in Appendix D). In fact, if we were interested in the *arithmetic* mean of the $\hat{\lambda}_i$, then $\hat{\beta} = 0.998117$ would be our best, most robust estimate.

But, to quantify average stochastic growth rate, we want the *geometric* mean, not the *arithmetic* mean. Clearly, we can't 'get there from here' using a variance decomposition of the $\hat{\lambda}_i$ estimates. We need to re-do our variance components analysis, this time using the derived $\ln(\hat{\lambda}_i)$ estimates.

The output from this re-analysis, using $\ln(\hat{\lambda}_i)$, is shown at the top of the next page. The value $\hat{\beta} = -0.008259$ reported near the top of the output is again quite close to the *arithmetic* mean of the shrinkage estimates, $\ln(\tilde{\lambda}_i)$ (again, labeled '**`S-tilde`**' in the output), -0.00653. The arithmetic mean of the ML estimates of $\ln(\hat{\lambda}_i)$ (labeled '**`S-hat`**') is identical to what we reported earlier (-0.00462).

If we had a basis for accepting that a random effects model was a more parsimonious model for these data than was the original fixed effects model, $\{\varphi_t\, p_.\, \lambda_t\}$, then our best estimate of the stochastic growth rate on the log scale would be -0.008259, with an estimated process variance on the log scale of 0.02243. If we back-transform from the log scale to the real scale, our estimate of stochastic growth rate is $\exp(-0.008259) = 0.9918$, which is somewhat smaller than the estimated arithmetic mean of the $\hat{\lambda}_i$ estimates shown above. This is perhaps expected (since for a random sample, the $GM < AM$), but



```
   Beta-hat  SE(Beta-hat)  Label
-----------  ------------  ----------------
  -0.008259      0.046131  Intercept

Par. Num      S-hat     SE(S-hat)     S-tilde   SE(S-tilde)  RMSE(S-tilde)
--------  ----------  -----------  -----------  -----------  -------------
     2      0.177223     0.092538     0.160606     0.077838       0.079592
     3      0.207787     0.083837     0.188134     0.070380       0.073073
     4     -0.066460     0.083235    -0.046903     0.069897       0.072582
     5      0.027830     0.082036     0.025483     0.069045       0.069085
     6      0.098226     0.077543     0.074511     0.066178       0.070299
     7     -0.324242     0.071523    -0.295678     0.062818       0.069007
     8     -0.000034     0.027391    -0.004200     0.026750       0.027073
     9     -0.266024     0.080699    -0.244251     0.068260       0.071649
    10     -0.020361     0.095550    -0.032589     0.076423       0.077395
    11      0.028057     0.095415     0.023098     0.077566       0.077724
    12      0.087213     0.067277     0.079999     0.060240       0.060670

Naive estimate of sigma^2 = 0.0204551 with 95% CI (0.0064293 to 0.0774989)

Estimate of sigma^2 = 0.0224252 with 95% CI (0.0086657 to 0.0792813)
```

we note that this may not always be the case when comparing β estimates for the intercept (i.e., means) using the variance components approach.

Using a Delta method approximation, our estimate for the back-transformed process variance is $\approx (\exp(-0.008259)^2 \times 0.02243) = 0.0228$.[‖] To derive a 95% CI for our estimated of stochastic growth, we first calculate the CI on the log scale using the estimated '`SE(Beta-hat)`'$= 0.046131$, as $-0.008259 \pm (1.96 \times 0.045181) \rightarrow [-0.098676, 0.082158]$, which when back-transformed from the log scale, is $[0.9060, 1.0856]$

Note: beyond the ability to estimate mean growth rate directly, the log transformation of the λ_i estimates results in a distribution which is generally nearer to normal than the generally log-normal distribution of the untransformed λ_i values. This may provide some improvement in performance of the 'method of moments' approach to estimation of the mean and process variance of the random distribution.

It is worth mentioning, however, that despite these advantages, the 'method of moments' approach to variance components analysis is sensitive to the length of the time-series, and is generally thought to be robust only when number of samples is ≥ 15 (see Appendix D). For the capsid example (above), we have only 12 estimates of λ_i, 11 if we exclude the potentially biased estimate over the first interval. In practice, we would be somewhat cautious in evaluation of the estimated mean stochastic growth rate from data with a relatively small number of years in the sample.

13.7. Time-symmetric models and Jolly-Seber estimation

We began this chapter by noting that population dynamics in the broad sense is determined by the net balance of 'additions' and 'subtractions'. The time-symmetric models, which we've introduced in this chapter, are one of several approaches available in **MARK** for partitioning one or more components of the dynamics of a population.

However, there are some important differences between time-symmetric models and the classical approaches generally referred to as *Jolly-Seber* models, introduced earlier in Chapter 12. In fact, the time-symmetric models considered in this chapter, in particular, the 'Pradel' models, are in effect a special

---

[‖] Because the transformation here (log) is non-linear, potentially strongly so depending on the range of the data, the first-order Delta approximation may not be particularly accurate – see section B.3.1 in Appendix B.

(conditional) case of the more general Jolly-Seber model (as discussed in Chapter 12). One important difference is that the 'Pradel' models, and the Cormack-Jolly-Seber (CJS) models we've considered in detail for analysis of live encounter data, condition on events since marking, and do not explicitly try to model events prior to the first encounter.

Another major difference is that neither the 'Pradel' or 'Link-Barker', or CJS models, specifically estimate abundance. In many cases, however, estimating abundance in open populations is important. For that, you need to consider Jolly-Seber and related models (Chapter 12).

13.8. Summary

That's the end of our very quick stroll through the time-symmetric 'Pradel' and 'Link-Barker' models in **MARK**. We've seen how a simple 'flip' of the encounter history can yield all sorts of interesting information on the processes underlying the dynamics of our population. We can estimate population growth, without the 'messy' job of estimating abundance. Moreover, we can partition variation in population growth due to relative contributions of recruitment and mortality. Pretty neat stuff. But, we wouldn't want to presume that estimating abundance, and all of the other elements which contribute to the dynamics of a population, are not important – we'll deal with this in the next chapter.

13.9. References

Caswell, H. (2001) *Matrix Population Models: Construction, Analysis, and Interpretation. 2nd Edition*. Sinauer Associates.

Cooch, E. G., Lank, D. B., Rockwell, R. F. and Cooke, F. (1999) Body size and recruitment in snow geese. *Bird Study*, **46**, 112-119.

Franklin, A. B. (2001) Exploring ecological relationships in survival and estimating rates of population change using program **MARK**. pp. 350-356 *in* R. Field, R. J. Warren, H. K. Okarma & P. R. Sievert (editors). *Wildlife, Land, and People: Priorities for the 21st Century*. The Wildlife Society, Bethesda, MD.

Link, W. A., and Barker, R. J. (2005) Modeling association among demographic parameters in analysis of open population capture-recapture data. *Biometrics*, **61**, 46-54.

Muir, R. C. (1957) On the application of the capture-recapture method to an orchard population of *Blepharidopterus angulatus*. *Report of East Mulling Research Station for 1957*, pp. 140–147.

Nichols, J. D., Hines, J. E., Lebreton, J.-D., and Pradel, R. (2000) Estimation of contributions to population growth: a reverse-time capture-recapture approach. *Ecology*, **81**, 3362-3376.

Nichols, J. D. (2016) And the first one now will later be last: time-reversal in Cormack-Jolly-Seber models. *Statistical Science*, **31**, 175-190.

Pradel, R. (1996) Utilization of capture-mark-recapture for the study of recruitment and population growth rate. *Biometrics*, **52**, 703-709.