

APPENDIX E

Markov Chain Monte Carlo (MCMC) estimation in MARK ...

Markov Chain Monte Carlo (more conveniently, MCMC) is a parameter estimation procedure that is frequently (but not exclusively) associated with Bayesian inference, that has been implemented in **MARK** for 2 primary purposes:

1. to provide the capability to more flexibly model and estimate the mean and variance (i.e., variance decomposition) of both univariate and multivariate *hyperdistributions* (i.e., the joint distribution of 2 sets of parameters)
2. to provide the capability to derive more intuitive credible intervals for the estimated parameters.

In this appendix, we discuss the basic theory and mechanics of using MCMC in **MARK**, for a variety of problems which would be difficult (at best) to implement in any other way. We defer a review of the theory and mechanics underlying MCMC to an Addendum at the end of the appendix. If you have no background at all in MCMC, or Bayesian inference, you are encouraged to have a look at the Addendum before proceeding too far in this appendix.* While it is possible to proceed in applying MCMC in **MARK** without a fair understanding of ‘how it works’, this is counter to our view that you are always better off if you actually know a bit about ‘what **MARK** is doing’. Those of you with stronger backgrounds might want to flip through the Addendum at some stage, if only to give you some insights as to the details of how things are implemented in **MARK**.

It is assumed that the reader already has a basic knowledge of some standard encounter-mark-reencounter models as described in detail in this book (e.g., dead recovery and live recapture models – referred to here generically as capture-recapture). We also assume familiarity with the variance components and random effects models presented in Appendix D – we strongly suggest you work through Appendix D in full before continuing here, if you have not done so already.

We introduce the subject of – and some of the motivation for – this appendix by example. In the following we consider two relatively common scenarios (out of a much larger set of possibilities) where a ‘different analytical approach’ (i.e., MCMC) might be helpful at least, or essential (in the case of the second example).

* *Note:* we do not mean to imply that this appendix, or the addendum to same, are in any way a substitute for formal study of MCMC in general, and Bayesian inference in particular. What we present here is only intended as a minimum sufficient introduction to get you started. Take a class, or study one of the many very good books on the subject.

scenario 1 – parameters as random samples

Consider a Cormack-Jolly-Seber (CJS) time-specific model $\{S_t, p_t\}$ wherein survival (S) and capture probabilities (p) are allowed to be time varying for $(k + 2)$ capture occasions, equally spaced in time. If $k \geq 20$ we are adding many survival parameters into our model as if they were unrelated; however, more parsimonious models are often needed.

Consider a reduced parameter model – at the extreme, we have the model $\{S, p_t\}$ wherein $S_1 = S_2 = \dots = S_k = S$. However, this model may not fit well even if the general (time-dependent) CJS model fits well and there is no evidence of any explainable structural time variation, such as a linear time trend, in this set of survival rates, or variation as a function of an environmental covariate. Instead, there may be unstructured time variation in the S_i that is not easily modeled by any simple smooth parametric form, yet which cannot be wisely ignored. In this case it is both realistic and desirable to conceptualize the actual unknown S_i as varying, over these equal-length time intervals, about a conceptual population mean $E(S) = \mu$, with some population variation, σ^2 (Fig. E.1).

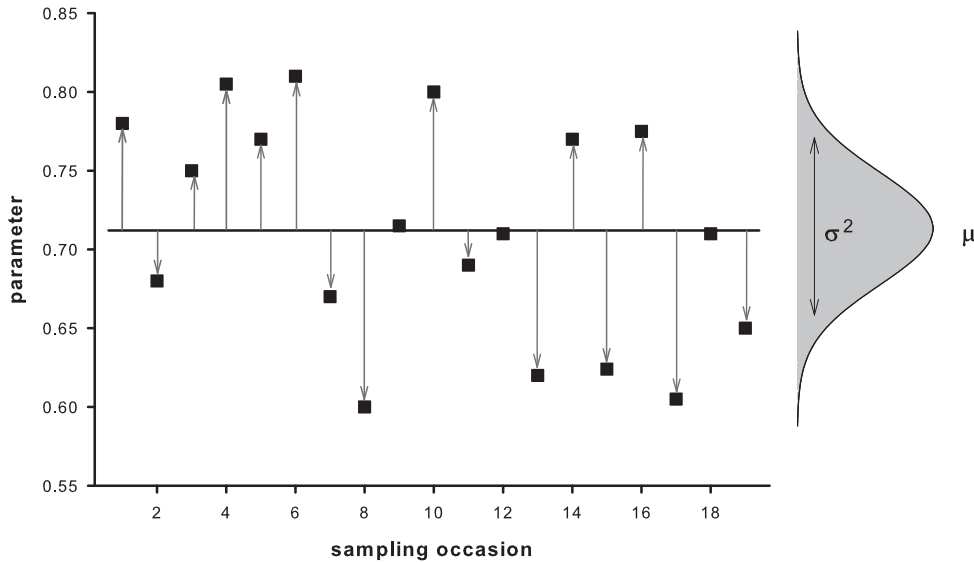


Figure E.1: Schematic representation of variation in occasion-specific parameters, θ_i , as if the parameters were drawn randomly from some underlying distribution with mean μ and variance σ^2 .

Here, by population, we will mean a conceptual statistical distribution of survival probabilities, such that the S_i may be considered as a sample from this distribution. Hence, we proceed as if S_i are a *random* sample from a distribution with mean μ and variance σ^2 . The parameter σ^2 is now the conventional measure of the unstructured variation in the S_i , and we can usefully summarize $S_1 \dots S_k$ by two parameters: μ and σ^2 . The complication is that we do not know the S_i ; we have only estimates \hat{S}_i , subject to non-ignorable sampling variances and covariances, from a capture-recapture model wherein we traditionally consider the S_i as fixed, unrelated parameters. We would like to estimate μ and σ^2 , and adjust our estimates to account for the different contributions to the overall variation in our estimates due to sampling, and the environment.

In Appendix D, we considered estimation of these 2 parameters using a random effects model based on a ‘methods of moments’ approach. We will see in this appendix how we can not only estimate μ and

σ^2 using MCMC, but how MCMC will allow much greater flexibility for more complex problems than the ‘method of moments’ approach.

scenario 2 – covariation between 2 structural parameters

In Chapter 8, we introduced ‘dead recovery’ models – so named because the ‘encounter data’ consist of recoveries of dead marked individuals. One dead recovery parametrization (Brownie) is commonly used for analysis of recovery data where the mortality event is influenced by harvest. For harvested species, an individual marked and released alive can experience one of 3 fates (Fig. E.2): (1) it can survive the year with some probability (S), (2) it can be ‘harvested’ (i.e., some ‘action’ leading to permanent removal) with some probability (K), or (3) it can ‘die’ from ‘natural’ causes with probability ($1 - S - K$) (i.e., it might actually die from some reason other than harvest, or permanently emigrate the sampling area, at which point it appears dead). Conditional on being harvested, (i) the individual may be retrieved (probability c), and (ii) reported (i.e., the individual identification number of the harvested individual is submitted to some monitoring agency), with probability (λ). The product ($Kc\lambda$) is referred to as the ‘recovery rate’, f .

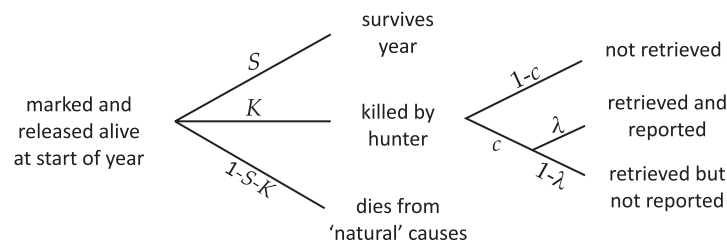


Figure E.2: Probability of different fates for marked individuals subject to harvest (Brownie parameterization).

Note that f is related to the ‘survival’ process, since an individual which is shot, retrieved and reported, does not survive. So, there is some anticipated structure relating S and f . Traditionally, the relationships between survival and harvest are broadly dichotomized as reflecting either *additive* or *compensatory* mortality (Fig. E.3).

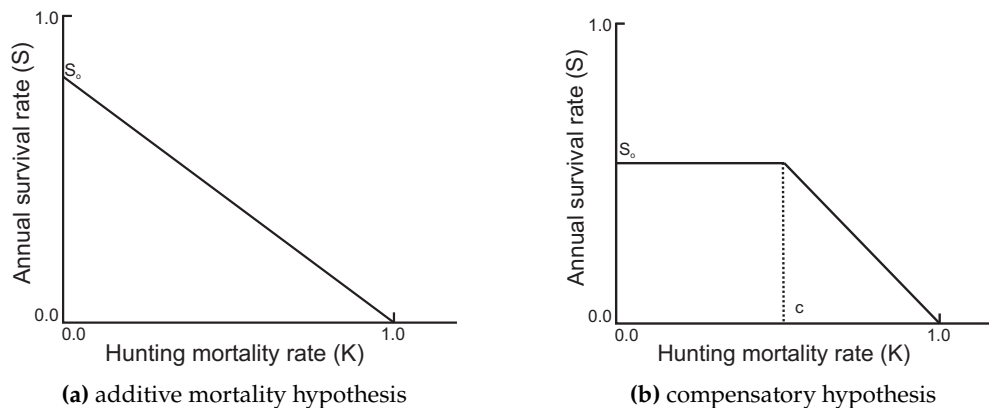


Figure E.3: Patterns of variation in survival, S , and mortality rate due to harvest, K .

In the case of additive mortality (Fig. E.3a), the sources of mortality are, just that, 'additive' – they add together, so that both natural and harvest mortality are combined in some additive way. In contrast, with compensatory mortality (Fig. E.3b), we are, in effect, assuming that we harvest only those animals which were likely to have died from natural causes in the first place - harvest does not increase the overall mortality rate (at least over a certain range). Moreover, if survival is a constant over some range, even when mortality due to harvest is increasing, then this implies that over that range, natural mortality is decreasing!

This is the key difference between compensatory and additive mortality – in compensatory mortality, natural mortality varies as a function of how much mortality there is due to harvest, while in additive mortality, the 2 simply add together. Mechanistically, compensatory mortality is generally believed to reflect the process(es) of density-dependence, while additive mortality reflects density-independence. This dichotomy between additive and compensatory mortality was first articulated by Anderson & Burnham in 1976.

If we assume that $c\lambda$ is a constant, then any variation in f must be proportional to variation in K . And thus, the structural relationship between S and f potentially yields important insights in differentiating between the additive and compensatory hypotheses, which has important implications for harvest management (since K , and thus f are under management control, at least to some extent; see the Williams, Nichols & Conroy (2002) book for an exhaustive treatment of the subject). If you look at figures (E.3a) and (E.3b) for a moment, it should be fairly clear that negative process correlation between harvest and survival rates is consistent with at least partially additive harvest effect on survival, whereas process correlations > 0 are consistent with a hypothesis that harvest mortality is fully compensated by other sources of mortality (Anderson & Burnham 1976).

In theory, then, all we need to do is look at the correlation of estimates of S and f . Easy enough in principle, but recall that estimates of these two parameters are generally not independent – there is significant sampling covariance within and between parameters. Thus, we can't simply take the estimates and 'do statistics on statistics' (i.e., calculate the bivariate correlation ρ between S and f). Fortunately, MCMC provides a solution, because MCMC approaches produce parameter estimates that are not influenced by sampling covariance. So, we can use the MCMC capabilities in **MARK** to derive a robust estimate of S and f , and the correlation between them.*

The ability to model the covariance between structural parameters in a statistically valid way using MCMC might also be of some interest for many other data types. For example, a negative correlation between recruitment f and apparent survival φ in time-symmetric models (Chapter 13) might be consistent with some hypotheses concerning cost of reproduction. Another example might be the relationship between abundance and the probability of temporary emigration in robust design models (Chapter 16), where a negative correlation between N and γ'' (the probability of temporarily emigrating) might be consistent with some hypotheses concerning density-dependence of individuals temporarily leaving the sampled population (which might be of interest if, say, the sampled population represents only breeding individuals, and temporary emigration is equivalent to non-breeding). Still another example might involve looking for interesting relationships between apparent survival S and state transition probabilities ψ , in multi-state models (Chapter 10).

In this appendix, we will consider application of MCMC to estimation of variance components, and modeling the relationship amongst structural parameters, using program **MARK**. We will outline the 'mechanics' of applying MCMC using program **MARK**, by means of a series of 'worked examples'.

* For example, this approach has been applied to greater sage-grouse recovery data – Sedinger, J. S., G. C. White, S. Espinosa, E. R. Parte & C. E. Braun. (2010) Assessing compensatory versus additive harvest mortality: an example using greater sage-grouse. *Journal of Wildlife Management*, **74**, 326-332. For an experimental approach, see Sandercock, B. K., E. B. Nilsen, H. Brøseth & H. C. J. Pedersen. (2011) Is hunting mortality additive or compensatory to natural mortality? Effects of experimental harvest on the survival and cause-specific mortality of willow ptarmigan. *Journal of Animal Ecology*, **80**, 244-58.

E.1. Variance components analysis revisited – MCMC approach

In Appendix D, we introduced the concept of ‘process variation’ among a set of parameters, and the mechanics of estimating process variation in **MARK**, using a moments-based estimator. Here we introduce an alternative approach, using MCMC.

E.1.1. Example 1 – binomial survival re-visited

We start by re-visiting the binomial survival example introduced in Appendix D. Again, we imagine a scenario where we are conducting a simple ‘known fate’ analysis (Chapter 17). In each of 10 years ($k = 10$), we mark and release $n = 25$ individuals, and determine the number alive, y , after 1 year (since this is a known-fate analysis, we assume there is no error in determining whether an animal is ‘alive’ or ‘not alive’ on the second sampling occasion). Here, though, we’ll assume that the survival probability in each year, S_i , is drawn from $\mathcal{N}(0.5, 0.05)$ (i.e., distributed as an independent normal random variable with mean $\mu = 0.5$ and process variance $\sigma^2 = 0.05^2 = 0.0025$). Conditional on each S_i , we generated y_i (number alive after one year in year i) as an independent binomial random variable $B(n, S_i)$. Thus, our maximum likelihood estimate of survival for each year is $\hat{S}_i = y_i/n$, with a conditional sampling variance of $\widehat{\text{Var}}(\hat{S}_i | S_i) = [\hat{S}_i(1 - \hat{S}_i)]/n$, which given $\mu = 0.5$, and $\sigma^2 = (0.05)^2 = 0.0025$, is approximately 0.01.

Table (E.1) gives the values of S_i , y_i and \hat{S}_i for our ‘example data’. Clearly, for a ‘real analysis’, we would not know the true values for S_i – we would have only \hat{S}_i , and generally only have $\hat{E}_S[\text{Var}(\hat{S}_i | S_i)]$ as $\widehat{\text{Var}}(\hat{S}_i | S_i)$. From Table (E.1) we see that the *empirical* standard deviation of the 10 estimated survival rates (i.e., the \hat{S}_i) is 0.106. However, we should not take $(0.106)^2$ as an estimate of σ^2 because such an estimate includes *both* process and sampling variation. Clearly, we want to subtract the estimated sampling variance from the total variation to get an estimate of the overall process variation.

Table E.1: Single realization from simple binomial survival example, $k = 10$, $E(S) = 0.5$, $\sigma = 0.05$, where $\hat{S}_i = y_i/n$ are $B(25, S_i)$, hence expected $SE(\hat{S}_i|S) \approx 0.1$.

year (i)	S_i	\hat{S}_i	$\widehat{SE}(\hat{S}_i S_i)$
1	0.603	0.640	0.096
2	0.467	0.360	0.096
3	0.553	0.480	0.100
4	0.458	0.440	0.100
5	0.506	0.480	0.100
6	0.498	0.320	0.093
7	0.545	0.600	0.098
8	0.439	0.400	0.098
9	0.488	0.560	0.099
10	0.480	0.560	0.099
mean	0.504	0.484	0.100
SD	0.050	0.106	

In Appendix D, we applied an approach based on a linear ‘method of moments’. The results from the variance components analysis of these data presented in Appendix D are shown below:

```

Beta-hat SE(Beta-hat)
-----
0.482526 0.033946

S-hat SE(S-hat) S-tilde SE(S-tilde) RMSE(S-tilde)
-----
0.640000 0.096000 0.548337 0.048955 0.103917
0.360000 0.096000 0.431320 0.048955 0.086505
0.480000 0.099920 0.481505 0.049351 0.049374
0.440000 0.099277 0.465242 0.049289 0.055377
0.480000 0.099920 0.481505 0.049351 0.049374
0.320000 0.093295 0.412990 0.048656 0.104951
0.600000 0.097980 0.530797 0.049160 0.084887
0.400000 0.097980 0.448615 0.049160 0.069139
0.560000 0.099277 0.514014 0.049289 0.067410
0.560000 0.099277 0.514014 0.049289 0.067410

Naive estimate of sigma^2 = 0.0015950 with 95% CI (-0.0042991 to 0.0277050)
Estimate of sigma^2 = 0.0019503 with 95% CI (-0.0039312 to 0.0280522)
Estimate of sigma = 0.0441616 with 95% CI (0.0000000 to 0.1674878)
Trace of G matrix = 4.7017092

```

Starting from the top – the first line of the output (above) reports a ‘Beta-hat’ of 0.482526. As discussed in Appendix D, this is our most robust estimate of the mean survival probability. This estimate is followed by the estimate of ‘SE(Beta-hat)’ which is our most robust estimate of total variance (i.e., process + sampling variation). In the absence of sampling covariance, it is estimated as the square-root of the sum of estimated process variation, $\hat{\sigma}^2$, and sampling variation, $E[\widehat{\text{Var}}(\hat{S}_i | S_i)]$, divided by k , where k is the number of parameter estimates.

Next, a table of various parameter estimates. The first two columns are the ML estimates of survival \hat{S}_i (‘S-hat’), followed by the standard error for the estimate (‘SE(S-hat)’). Next, the ‘shrinkage’ estimates \tilde{S}_i (‘S-tilde’) and corresponding SE and RMSE.

Finally, the estimates for process variation. First, **MARK** reports the ‘Naive estimate of sigma^2’ = 0.001595. This is followed by the ‘Estimate of sigma^2’ = 0.0019503 (and the ‘Estimate of sigma’ = 0.044162). As discussed in Appendix D, these are the ‘preferred’ estimates for process variance. We concentrate here on using MCMC in **MARK** to derive similar – hopefully, identical – estimates for σ .

We begin by opening up the binomial-example.dbf file (from Appendix D). The browser should contain at least the general model $\{S_i\}$. Retrieve the general model. We’re now going to re-run it, but this time, making use of the MCMC capabilities in **MARK**. Click the ‘Run’ icon in the toolbar, and bring up the ‘Setup Numerical Estimation Run’ window (shown at the top of the next page). Here, notice that we’ve checked the ‘MCMC Estimation’ box, indicating we want to use the MCMC capabilities in **MARK** for the estimation. We’ve also checked the ‘Provide initial parameter estimates’. This has been shown to be a good standard practice for MCMC-based estimation. Finally, we’ve selected the ‘Logit’ as the link function. Because the logit link is a monotonic transformation (as opposed to the sin link – see the discussion of ‘link functions’ presented in Chapter 6), evaluation of the moments of the posteriors is somewhat easier (and less prone to error). Finally, we’ve added the word ‘MCMC’ to the model name, to indicate that the model will be estimated using MCMC (although this isn’t really necessary, since the model results are not added to the browser).

Setup Numerical Estimation Run

Title for Analyses: binomial survival -- known fate -- VC analysis

Model Name: {S(t)} -- general model - MCMC

Fix Parameters: No Parameters Fixed

Link Function:

- ☐ Sin
- ☒ Logit
- ☐ LogLog
- ☐ CLogLog
- ☐ Log
- ☐ Identity
- ☐ Absolute
- ☐ Parm-Specific

Var. Estimation:

- ☐ Hessian
- ☒ 2ndPart

☒ MCMC Estimation

Numerical Estimation Options:

- ☐ List Data
- ☒ Provide initial parameter estimates
- ☐ Use Alt. Opt. Method
- ☐ Profile Likelihood CI
- ☐ Set digits in estimates
- ☐ Set function evaluations
- ☐ Set number of parameters
- ☐ Standardize Individual Covariates
- ☐ Do not standardize design matrix

Real Par. Estimates from Individual Covariates:

- ☐ First Encounter History Covariate Values
- ☒ Mean Individual Covariate Values
- ☐ User-specified Covariate Values

Buttons: Help, Cancel Run, OK to Run

Once you've made these changes, click the '**OK to Run**' button. This will spawn a new window (shown below) which will allow you to specify the '**Markov Chain Monte Carlo Parameters**':

Markov Chain Monte Carlo Parameters

Basics

Random Number Seed: 0

Number of tuning samples: 4000

Number of burn in samples: 1000

Number of samples to store: 10000

Name of binary file to store samples: MCMC.BIN

Name of CSV file to store summary data: MCMC.CSV

Default SD of normal proposal distribution: 0.5

Convergence Diagnostics:

- ☐ Estimate sample sizes using gibbsit procedure
- ☒ Single chain, with no convergence diagnostics
- ☐ More than 1 series, using convergence of Markov chains

Number of chains: 10

Hyperdistribution Specification:

Number of hyperdistributions: 0

☐ Hyperdistribution means modeled with a design matrix of 2 columns

☐ Variance-covariance matrix specified

Prior Distributions for Parameters not Included in Hyperdistributions:

- ☐ No Prior -- Prior Ratio = 1
- ☒ Default Prior: Mean 0.0 Sigma 1.75
- ☐ Specify Priors Individually

Buttons: Help, Cancel, OK

We'll quickly go down the list of those parameters we're going to need to consider when using MCMC for the binomial survival example. First, the '**Random Number Seed**'. It defaults to 0, meaning, the random numbers used to generated the samples will be different for each run. For almost all purposes, this is entirely acceptable, so we leave the seed at the default value of 0 in most cases. The purpose of this entry is to allow you to specify a random number seed if you want to duplicate results from a previous analysis.

Next, 3 boxes indicating the number of ‘samples’ we want to make. The specific details concerning these options are detailed in the Addendum to this appendix, but briefly:

Number of ‘burn in’ samples – the Metropolis-Hastings algorithm used by **MARK** takes random samples from the posterior distribution. Typically, initial samples from the Markov chain are not completely valid because the chain has not stabilized. The ‘burn in’ samples allow you to discard these initial samples.

Number of ‘tuning’ samples – MCMC is based on acceptance/rejection of a ‘proposed’ value, drawn from a ‘proposal distribution’ – typically $\mathcal{N}(0, \sigma)$, where σ is chosen estimated to give a 40-45% acceptance rate. That is, σ is estimated during the ‘tuning’ phase to accept the new proposal 40-45% of the time.

Number of samples to store – After the initial ‘burn in’ period (see above), samples from the posterior distribution are saved to enable computation of summary statistics describing this (posterior) distribution. [Note: thinning the sample is an option within **MARK**, but can also be done after the fact using **SAS** or **R**.]

For smaller problems, these defaults are generally sufficient. For larger data sets, and more complex models, you will typically need to increase the size of the various samples (in particular, the ‘**Number of samples to store**’).

Next, a box to let you specify the name of binary file to store samples (it defaults to **MCMC.BIN**). The samples are saved in a binary file (meaning, you can’t simply open up the file in an ASCII editor). The binary file can be read, however, with a **SAS** code or an **R** code to perform more sophisticated analysis than are available from **MARK**. Example **SAS** and **R** code is provided in the **MARK** helpfile. This box is followed by another which lets you specify the name of a .CSV file to store summary data (means, medians, percentiles...) from the sample data (it defaults to **MCMC.CSV**). This file can be opened in Excel, or equivalent spreadsheet software.

Finally, a box to set the default SD of the normal proposal distribution. The default is 0.5. As noted above, for beta parameters that are not a part of hyperdistributions, the default step size to generate the next value of the parameter is generated as a random normal variable with a SD specified in this edit box. Typically, you want to accept the new parameter value about 45% of the time, so you can adjust this SD to approximately obtain this acceptance rate. (Discussed more fully in the Addendum).

The next 3 sections are particularly important. First, the ‘**Hyperdistribution Specification**’. One of the primary purposes of the MCMC algorithm in **MARK** is to estimate mean and variance of sets of parameters, i.e., estimate the values of μ and σ given a hyperdistribution of a set of beta parameters. Using this edit box, you can specify the number of hyperdistributions that you want to model. If you leave the number of hyperdistributions set to the default of 0, **MARK** will use MCMC to derive estimates of parameters and credibility intervals for those estimates, but will not estimate μ or process variance σ^2 . For our re-analysis of the binomial survival data, we want to specify 1 hyperdistribution (i.e., to estimate μ and σ^2 among the set of 10 survival estimates).

As you will observe, once you set the number of hyperdistributions to a value > 0 , then two options will be made available (‘active’). First, you can choose to model the hyperdistribution means using a design matrix. Checking this check box allows you to model the means of the hyperdistributions with linear models specified in the MCMC hyperdistribution design matrix. For this example, we will not specify a linear model with a design matrix, so we’ll leave the box unchecked.

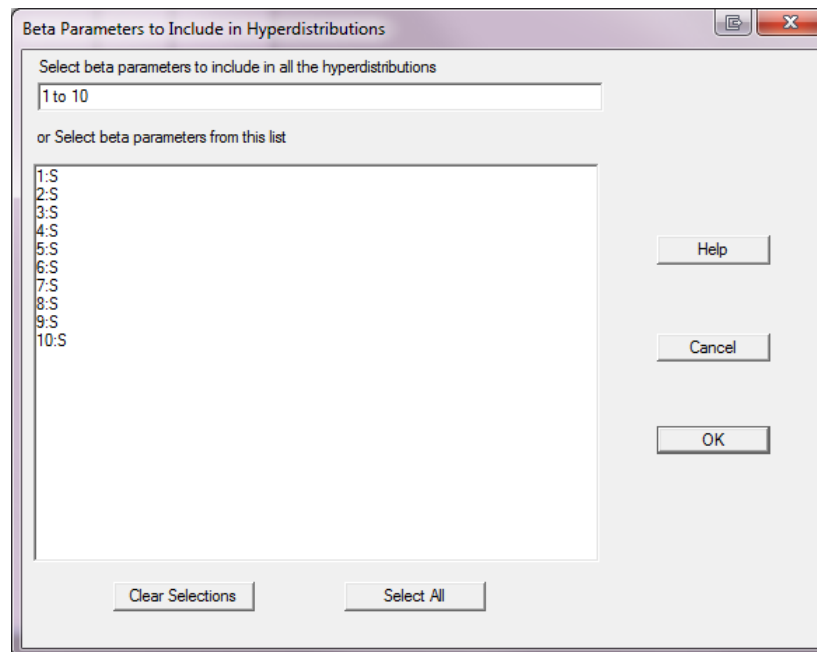
Second, you can specify the structure of the variance-covariance matrix among parameters included in the hyperdistributions. This specification is especially helpful if you are interested in looking at covariance among different parameters in a particular model (e.g., the covariance of S and f in a Brownie

dead recovery model; example scenario 2, on p. E-3). Since that is not our interest here (clearly, since there is only a single parameter type, S_i), we will leave this option unchecked as well.

The next item involves the specification of ‘**Prior Distributions for Parameters not included in Hyperdistributions**’. Although the most likely use of the MCMC estimation procedure is to estimate the mean and variance of hyperdistributions, most models in **MARK** include other nuisance parameters in the models. These parameters also require a prior distribution. Three options are provided. The first is to ignore the prior distribution, and never use it to decide whether a new value in the Markov Chain is accepted or rejected. The second option is to specify a default prior distribution, consisting of a normal distribution with the mean and variance provided. All parameters not included in a hyperdistribution will use this normal prior. The third option is to specify the prior distribution for each parameter individually. However, only normal priors are allowed, so you can only specify a mean and standard deviation appropriate for each of the non-hyperdistribution parameters.

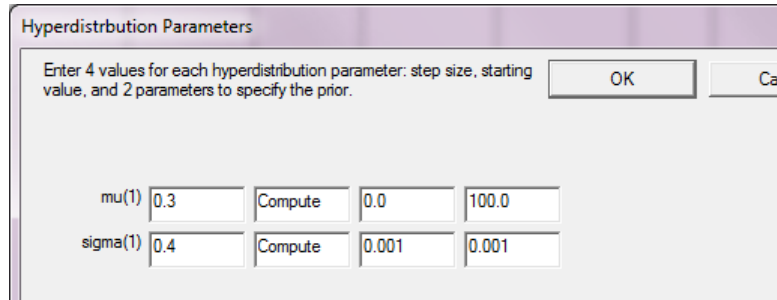
Finally, a section letting you specify various ‘**Convergence Diagnostics**’. This group of controls determines what diagnostic values will be generated. Checking the ‘**GIBBSIT sample size**’ box will produce a table of estimated sample sizes for the burn-in period and the samples to store. Selecting the ‘**multiple Markov chains**’ option will produce a diagnostic statistic (\hat{R}) useful for determining if the Markov chains have adequately sampled the posterior distribution (i.e., if the multiple chains have converged, indicated by $\hat{R} \rightarrow 1.0$). The default is a single chain, which is usually sufficient for simple problems (such as the binomial survival analysis), given enough samples. The **MARK** helpfile should be consulted for further details on these options.

Once you have specified the appropriate values, or want to accept the defaults, click the ‘**OK**’ button to continue. Since we have specified > 0 hyperdistributions to be modeled (1, for this example), additional dialog windows will request information on these hyperdistributions.



For this example, we have 10 survival parameters (S_1, S_2, \dots, S_{10}), so we enter ‘1 to 10’ as the parameters to include in the hyperdistribution. Here, we include all 10 survival parameters. In situations where one or more parameters are confounded, we generally exclude them from the hyperdistributions

(similar to excluding confounded parameters in moment-based RE models, as discussed in Appendix D). Click the 'OK' button. Based on the lists of means and standard deviations selected, a new dialog window appears that is requesting information about how to estimate these hyperparameters.



The dialog box is titled "Hyperdistribution Parameters". It contains a text box with the instruction: "Enter 4 values for each hyperdistribution parameter: step size, starting value, and 2 parameters to specify the prior." To the right of the text box are "OK" and "Cancel" buttons. Below the text box, there are two rows of input fields. The first row is for $\mu(1)$ and the second row is for $\sigma(1)$. Each row has four input boxes: the first for the step size, the second for the starting value (with a "Compute" button between the first and second boxes), and the third and fourth for the prior parameters. The values entered are: for $\mu(1)$, step size 0.3, starting value 0.0, and prior parameters 100.0 and 0.001; for $\sigma(1)$, step size 0.4, starting value 0.001, and prior parameters 0.001 and 0.001.

Parameter	Step size	Starting value	Prior parameter 1	Prior parameter 2
$\mu(1)$	0.3	0.0	100.0	0.001
$\sigma(1)$	0.4	0.001	0.001	0.001

Four inputs are requested for each hyperparameter. The first edit box is to specify the step size to be used to generate new parameter values with which to sample the posterior distribution. As with the default step size, the goal is to tune the estimation so that approximately 45% of the steps are accepted.

The second edit box is to specify an initial value to start the Markov Chain. The default is 'compute', which tells **MARK** to compute an estimate from the initial values of the beta parameters. It is generally recommended practice that you should run the model that you want to use for MCMC estimation as a typical **MARK** analysis, so that you can provide initial estimates to start the MCMC estimation.

The third and fourth edit boxes specify the parameters for the *prior* distribution to be used with this hyperdistribution. For mean parameters, a normally distributed prior is used. The third edit box specifies the mean, and the fourth edit box specifies the standard deviation. The default values for μ are mean = 0 and standard deviation = 100, giving a very flat and uninformative prior. For σ parameters, a γ prior is used to model the σ in a transformation: $1/\sigma^2$ is assumed to be distributed as a γ distribution with parameters α (third edit box) and β (fourth edit box). Again, the defaults of $\alpha = \beta = 0.001$ result in a very flat, uninformative prior. In other words, the defaults specify uninformative 'flat' priors for the two hyperparameters, μ and σ .

Once you click 'OK', you'll be asked to specify 'starting values', which you can retrieve from the general model $\{S_t\}$, which is already in the browser (remember – retrieving from a model in the browser only works if the model you're retrieving was run with the same link function as we're using here – logit). Once you have set the starting values, and clicked 'OK', **MARK** will spawn a command windows ('DOS box'), which will show you the progress of the MCMC sampling (i.e., which chain **MARK** is working on, which iteration). In practice, you will quickly discover that this stage can (and frequently is) be the 'rate-limiting step' for using MCMC in **MARK** (or any other software), especially for complex problems (which in the context of **MARK**, can also mean 'lots of parameters, and lots of data'), where even the default 10,000 samples (plus 5,000 for burn-in and tuning) can take a very long time. There isn't much you can do easily to speed things up – in general, the only guaranteed way to speed things up is to get a faster computer.

For our re-analysis of the binomial data, this is a probably a moot point. **MARK** will probably generate the 15,000 total samples in only a few seconds on even a 'mediocre' computer. Once **MARK** has finished generating the samples, the resulting output is not stored in the results browser, or in the results file. Rather, the output, containing various summary statistics, is placed in an editor window, as well as a .CSV file that can be opened in Excel, or equivalent spreadsheet software. Summaries are provided for each of the beta parameters, hyperdistribution parameters, and real parameters are the mean, standard deviation, median, and mode, plus the percentage of trials when a new value was accepted (labeled

as the proportion of jumps accepted). In addition, the frequency percentiles of 2.5, 5, 10, 20, 80, 90, 95, and 97.5, as well as 80%, 90% and 95% highest posterior density intervals, are listed. These values can be used to create credibility intervals for the parameters (discussed in more detail, below). In addition, as noted, the binary output file is available to use for additional, more sophisticated analysis, with the **SAS** code or **R** code provided in the **MARK** helpfile.

When you scroll through the output in the editor window, you'll eventually come to the summary statistics for the posterior samples. Here they are for the binomial survival example (*note*: MCMC is based on taking random samples from the posterior – the word ‘random’ explicitly indicates that the values you get in your data summaries are unlikely to precisely match those reported here. But, they should at least be relatively close).

Parameter	LOGIT Link Function Mean	Parameters of {S(t)} Standard Dev.	-- general model} Median	Mode	Jumps(%)	Jump Size
1:S	0.0489572	0.2331556	0.0248514	0.0041018	43.2	0.30007
2:S	-0.1428307	0.2230992	-0.1175747	-0.0956812	43.2	0.30007
3:S	-0.0624551	0.2087515	-0.0597061	-0.0899707	43.3	0.30007
4:S	-0.0902734	0.2092891	-0.0784611	-0.0755440	56.8	0.18009
5:S	-0.0607908	0.2057351	-0.0564780	-0.0477804	43.4	0.30007
6:S	-0.1696255	0.2364368	-0.1384957	-0.1123492	61.2	0.15190
7:S	0.0179133	0.2191342	0.0030244	0.0069413	47.5	0.25311
8:S	-0.1142002	0.2135478	-0.0998745	-0.1057864	35.0	0.42175
9:S	-0.0079053	0.2107560	-0.0150001	-0.0059292	52.2	0.21350
10:S	-0.0071009	0.2103838	-0.0148864	0.0098952	43.0	0.30007
11:mu(1)	-0.0588212	0.1489109	-0.0568440	-0.0905101	51.2	0.07688
12:sigma(1)	0.1744596	0.1435179	0.1314652	0.0482029	50.5	1.31665
13:-2log Likelihood	346.11226	2.7115905	346.27226	346.46076		
-2log Likelihood for means of beta estimates = 343.19938						
Penalty function 2*p_D for DIC = 5.8257631						
DIC = 349.02514						
WAIC = 349.06356						

At the top of the output (above), it is important to note that what you see here are ‘logit link function parameters’. In other words, you’re presented with the mean (and SD), median and mode of the distribution of the samples for each parameter (S_i , and the hyperparameters μ and σ) on the logit scale. Meaning, a back-transformation of some sort will be needed to generate parameter estimates on the real probability scale. More on this in a moment.

What information do we glean from the output? We see that the mean, median, and mode are different for most parameters, occasionally markedly so. This difference indicates in part that the posterior is potentially not ‘nice and symmetrical’ (if the distribution was perfectly symmetrical, then the mean, median and mode would be identical). There is clearly some uncertainty on choosing a particular value (mean, median, mode) to generate back-transformed estimates on the real probability scale. In general, using the mean seems to work well, but if in doubt, consult your local Bayesian.

The right-hand column gives the proportion of jumps for each parameter. Ideally, we’d like this percentage to be in the 40-50% range (the proportion of jumps is an index to how efficiently we’re sampling the posterior). Some theory has been proposed that 40-50% is ‘optimal’ for exploring the posterior (optimal being a function of time to convergence, and the ability to explore all regions of the posterior distribution), which **MARK** adopts, in a fashion. As noted earlier, you can ‘tweak’ the jump size by changing the step size **MARK** uses to generate new parameter values with which to sample the posterior. In addition to the summary statistics for each of the parameters, **MARK** also reports the estimated $-2 \ln \mathcal{L}$ for the model, including the $-2 \ln \mathcal{L}$ based on the means of the beta estimates (where near equality of the two values indicates that using the mean is probably acceptable for this example).

Finally, **MARK** reports several measures of ‘fit’ of the model to the data – the DIC (Deviance Information Criterion; Spiegelhalter *et al.* 2002, Celeux *et al.* 2006) and the WAIC (Watanabe-Akaike Information Criterion; WAIC; Watanabe 2013) for the model, either of which in theory could be used for model selection, model averaging, and other purposes. The DIC is a hierarchical modeling

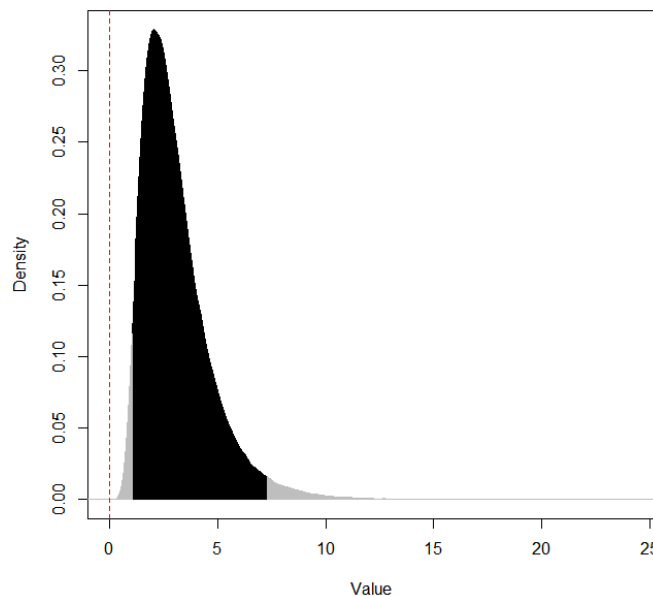
generalization of the AIC (Akaike information criterion) and BIC (Bayesian information criterion). The DIC is calculated as the sum of the model deviance (the difference in $-2 \ln \mathcal{L}$ of the current model and the saturated model), and the effective number of parameters. The WAIC substitutes the logarithm of the posterior predictive density for the deviance and uses a different calculation for the penalty.

Both the DIC and WAIC have the same ‘fit plus parameter penalty’ structure as do the AIC and BIC. And, like the AIC and BIC, both are asymptotic approximations as the sample size becomes large. Generally, both are only valid when the posterior distribution is approximately multivariate normal. Because of current challenges on how best to handle model selection and averaging in a Bayesian context, both the DIC and WAIC are printed in the output for ‘informational purposes’ only. Use at your own risk (see Hooten & Cooch 2019, for a discussion of both criterion in model selection).

Immediately below the summary statistics of the parameters on the logit scale are the percentile tabulations for each parameter – useful in generating (say) a 95% credibility interval, on the logit scale. Two different types of percentiles are generated: the first based on simple frequency percentiles for the posterior distribution (i.e., 2.5%, 5%, ..., 95%, 97.5%), and a second based on the highest posterior density (HPD). For unimodal, (more-or-less) symmetric posterior distributions, the frequency percentile- and HPD-based credible intervals won’t differ much.

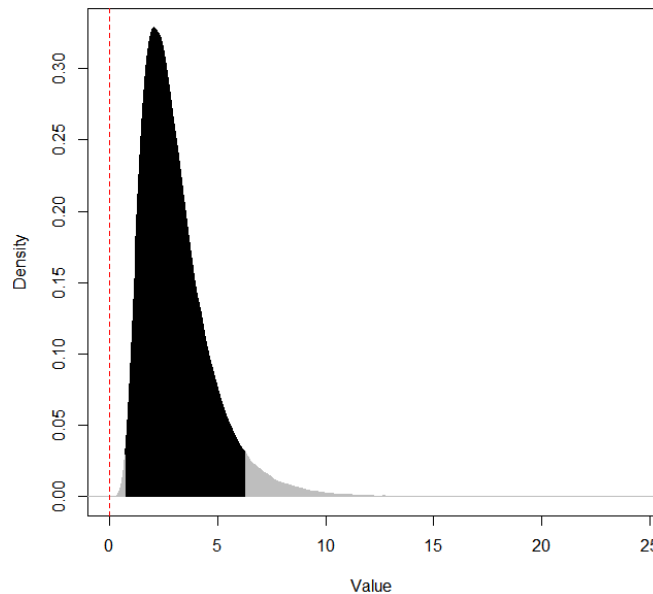
However, the HPD is potentially the more useful basis for specifying a credible interval when the posterior distribution is skewed, or not symmetric. In brief, the HPD is an interval that spans a specified percentage of the distribution (say, 95%), such that every point inside the interval has a higher plausibility than any point outside the interval. For a simple, univariate parameter θ with a unimodal posterior probability density, calculation of a simple HPD is relatively straightforward: (i) sort the posterior values for θ in ascending order, (ii) for $j = 1, 2, \dots, N - [(1 - \alpha)N]$, when N posterior samples, compute $N(1 - \alpha)\%$ credible intervals (e.g., for a posterior consisting of $N = 1,000$ samples, for an $\alpha = 0.95$ (i.e., for a 95% HPD), you construct $1,000 \times (1 - 0.95) = 50$ intervals), where the credible interval is $[\theta_j, \theta_{(j+[(1-\alpha)N])}]$. Finally, (iii) the HPD interval is the one with the smallest interval width among all credible intervals generated in step (ii).

Let’s take a closer look at how these intervals differ. Consider a large sample of log-normal random deviates, with mean and standard deviation on the log scale of 1.0 and 0.5, respectively. The following is a density plot of the sample data – clearly, this distribution is not symmetrical:



The left- and right-hand grey tails are the 2.5% percentiles, constructed such that the area of both tails is the same, with the dark (black) area representing 95% of the probability mass. The 95% frequency percentiles shown in the figure are [1.0672, 7.2661]. But, consider the probability of $\theta = 1$. Based on the density plot, $\theta = 1$ is far more likely to occur than $\theta = 6$, and yet $\theta = 6$ is included in the frequency percentile-based credibility interval, whereas $\theta = 1$ is not. Clearly, this does not inspire much confidence (pun partially intended...) in our 95% credibility interval for θ based on simple frequency percentiles.

Contrast this with a credibility based on highest posterior density (HPD), indicated in the following density plot. Here, the grey tails represent those values of θ where our belief in the credibility of that values is $< 5\%$, whereas the dark, black area represents the values of θ for which the sum of our belief in those points (i.e., the area under the curve) is 95%.



The 95% HPD interval shown in the preceding plot, [0.7027, 6.2893], is clearly quite different than the one based on frequency quantiles (above). The important distinction here is that the probability at either the left-hand or right-hand ‘border’ between the grey and black regions is identical. In other words, if you drew a horizontal line parallel to the x -axis, it would intersect the probability function at the same point at either ‘border’.

Which credibility interval should we report? Well, to some degree, it might seem reasonable to use the HPD generally, because it is often more defensible than the quantile-based interval when the posterior distribution is asymmetrical or (worse) multi-modal. When the posterior is symmetrical and unimodal, then the HPD and quantile-based intervals are effectively identical in practice.

The biggest potential problem for the HPD is its lack of invariance under transformation. And, neither the frequency percentile-based probability interval or the HPD interval take the prior distribution into account. Yet another, computationally more challenging approach which does take the prior into account is based on the ‘lowest posterior loss’ (LPL) functions. [At present, calculation of LPL credibility intervals has not been implemented in **MARK**.] See Bernardo (2005) for the rather messy details.*

* Bernardo, J. M. (2005). Intrinsic Credible Regions: An Objective Bayesian Approach to Interval Estimation. *Sociedad de Estadística e Investigación Operativa*, 14, 317-384.

Immediately following the frequency percentile- and HPD-based credible intervals in the **MARK** output are the estimates and moments for the individual structural parameters (but note – not the hyperparameters), back-transformed to real probability scale.

Here are the reconstituted estimates for the binomial survival example:

Real Function Parameters of {S(t) -- general model - MCMC}				
Parameter	Mean	Standard Dev.	Median	Mode
1:S	0.5122326	0.0587721	0.5048525	0.4923201
2:S	0.4628398	0.0549111	0.4677703	0.4718443
3:S	0.4838063	0.0497401	0.4840120	0.4831943
4:S	0.4765415	0.0516942	0.4783840	0.4868165
5:S	0.4838067	0.0501470	0.4840088	0.4876349
6:S	0.4551692	0.0574450	0.4621490	0.4729862
7:S	0.5059414	0.0552734	0.5007306	0.4977442
8:S	0.4692801	0.0517626	0.4724324	0.4739214
9:S	0.4984989	0.0523712	0.4953406	0.4938752
10:S	0.4981472	0.0523627	0.4949541	0.4852393

Now, it is critical to remember that for this analysis, we are estimating a model where we've specified a hyperdistribution for the individual S_i , with hyperparameters μ and σ . In other words, this is a mean (intercept only) random effects model. Thus, the reported estimates for S_i (above) are, in fact, shrinkage estimates, \tilde{S}_i , where the estimates are 'shrunk' towards the common mean of the hyperdistribution. Shrinkage estimates are introduced and discussed in detail in Appendix D.

Also, note that what is reported for each parameter is the mean of the posterior, back-transformed from the logit to the real probability scale. For example, from the preceding table of real function parameters, we see that the estimated mean for parameter S_1 on the logit scale was $\tilde{S}_{1,\text{logit}} = 0.0506165$.

If we back-transform from this value on the logit scale to the real probability scale, we get

$$\tilde{S}_1 = \frac{e^{0.0506165}}{1 + e^{0.0506165}} = 0.51265,$$

which is close to what **MARK** reports for $\tilde{S}_1 = 0.51223$ above. Why the slight difference? The answer is because **MARK** does not simply back-transform the estimate for $\tilde{S}_{1,\text{logit}} = 0.0506165$ (as we have done here) – rather, **MARK** takes the entire posterior sample on the logit scale, and back-transforms it to the real probability scale, and then takes the mean of this back-transformed distribution.

How do these back-transformed estimates of the individual \tilde{S}_i compare to estimates derived using maximum likelihood and the 'method of moments' approach (from p. E-6)? In the following table (E.2), we compare the first 5 estimates from both methods (rounded to 3 significant digits). As we can see, the results from both approaches match pretty well, which they should, given we used a non-informative prior for both hyperparameters (see the following – sidebar –).

Table E.2: Comparison of shrinkage estimates from the ML 'methods of moments' (Appendix D) with shrinkage estimates from MCMC analysis of the binomial survival data.

method	i				
	1	2	3	4	5
\tilde{S}_i (ML)	0.548	0.431	0.482	0.465	0.482
\tilde{S}_i (MCMC)	0.512	0.463	0.483	0.477	0.484

begin sidebar

ML estimates and MCMC

In the preceding example, we fit a model where we specified a hyperdistribution for the S_i , with hyperparameters μ and σ . The estimates of the S_i values were, therefore, shrinkage estimates, \hat{S}_i . How would we use MCMC to generate what are equivalent to ordinary ML estimates? Simple enough – we simply re-run the MCMC analysis without specifying the hyperdistribution on the individual S_i , with a non-informative prior on the beta parameters.

However, we need to think a bit about how **MARK** handles priors for parameters which are not included in a hyperdistribution – meaning, in this case, all the parameters. These parameters also require a prior distribution. As noted earlier, three options are provided in **MARK**. The first is to ignore the prior distribution, and never use it to decide whether a new value in the Markov Chain is accepted or rejected. The second option (which is selected by default) is to specify a default prior distribution, consisting of a normal distribution with the mean and variance provided. All parameters not included in a hyperdistribution will use this normal prior. The third option is to specify the prior distribution for each parameter individually. However, only normal priors are allowed, so you can only specify a mean and standard deviation appropriate for each of the non-hyperdistribution parameters.

Care must be taken in what prior is used for parameters not included in a hyperdistribution. A mean of zero is appropriate for most parameters. However, a very large standard deviation will result in a back-transformed logit value with a ‘U-shaped’ distribution, i.e., the real parameter value is much more likely to take on the values close to zero or close to 1. A standard deviation of 1.5 results in a back-transformed distribution that is about 95% of the probability between [0.05, 0.95], so is actually results in a pretty reasonable prior distribution on the real scale. Further, some consideration must be given to how the prior distributions of a function of the beta parameters will interact. For example, the intercept and slope of a trend model might not be appropriately specified as $N(0, 1.5)$ priors, depending on how the beta parameters are expected to change over the range of the data.

The other consideration about the prior is that if it is non-informative, then the mode of the posterior distribution, as generated by MCMC, should be identical with the MLE. This can be easily demonstrated by algebra. Consider estimation of a simple binomial (say, a coin toss experiment). If the probability of tossing a head is θ , then (from Chapter 1) the probability distribution is given by the binomial distribution, i.e.,

$$f(x | \theta) = \frac{n!}{x!(n-x)!} \theta^x (1-\theta)^{n-x}.$$

As discussed briefly in Chapter 1, it is clear that the likelihood given this expression is maximized at $\theta = x/n$.

In a Bayesian context, we generally imagine placing a prior distribution on the parameter θ . Since the parameter θ lies within the interval [0, 1], then an appropriate prior is also bounded on the interval [0, 1]. In the absence of prior information (or belief) we might consider a non-informative (‘flat’) prior which makes all possible values of θ equally likely. For convenience, the Bayesian often uses a general prior distribution which makes it convenient to modify the prior to reflect a range of prior beliefs.

One common approach is to take the beta distribution (which is conjugate for the binomial) such that $P(\theta) \propto \theta^{\alpha-1} (1-\theta)^{\beta-1}$. The shape variables α and β are the prior parameters, and can be chosen to reflect differing prior beliefs. If $\alpha = \beta = 1$, then the distribution is flat (uniform) on the interval [0, 1] (Fig. E.4). Note also that whenever $\alpha = \beta$, the prior distribution is symmetric around $\theta = 0.5$ (it is only symmetric and flat if $\alpha = \beta = 1$).

If we adopt a beta prior on θ , then the posterior is given as

$$\pi(\theta | x) \propto f(x | \theta) p(\theta) \propto \theta^{x+\alpha-1} (1-\theta)^{n-x+\beta-1}.$$

Thus, we see that the posterior is itself in the form of a beta distribution, with parameters $(x + \alpha)$ and $(n + \beta - x)$.

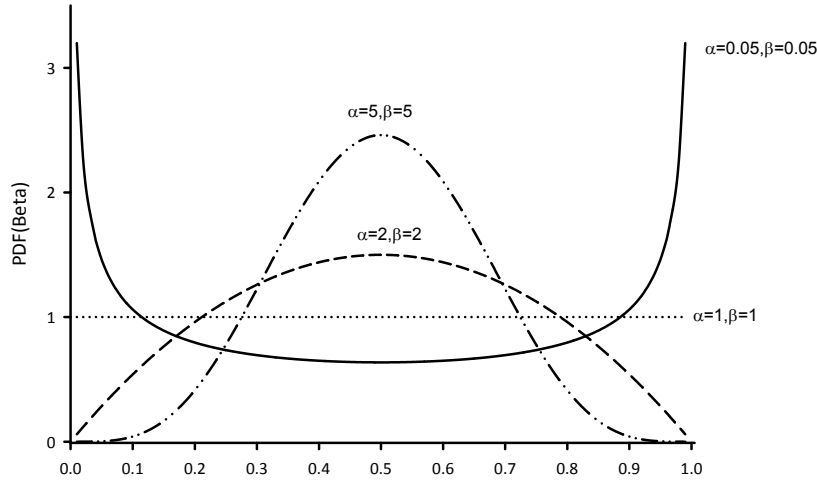


Figure E.4: Shape of the beta probability distribution for different values of the shape parameters α and β . The distribution is symmetric for $\alpha = \beta$, and is uniform for $\alpha = \beta = 1$.

By the method of moments (see first few pages of Appendix B), we can show that this posterior distribution has a mean of

$$\frac{x + \alpha}{n + \alpha + \beta},$$

and a mode of

$$\frac{x + \alpha - 1}{n + \alpha + \beta - 2}.$$

Now, both the expected mean and mode of the posterior appear quite different than the MLE of (x/n) . But, look closely at the expressions for the expected mean and mode of the posterior. You should see that, in fact, the MLE is embedded in the two expressions.

Why is this important? It is important because this suggests that the Bayesian estimate for θ will differ from the ML estimate for θ as a function of the values of α and β used in the prior. Given that, what happens if we use a flat, non-informative prior, $\alpha = \beta = 1$? If we do, then we see that the expected mean of the posterior is

$$\frac{x + \alpha}{n + \alpha + \beta} = \frac{x + 1}{n + 1 + 1} = \frac{x + 1}{n + 2},$$

while for the mode,

$$\frac{x + \alpha - 1}{n + \alpha + \beta - 2} = \frac{x + 1 - 1}{n + 1 + 1 - 2} = \frac{x}{n}.$$

In other words, for a ‘flat’, non-informative prior, the mode of the posterior and the MLE are *identical*. Also, note that as the sample size increases (i.e., as x and n both increase), then the posterior mean will converge on the mode and the MLE. (For completeness, we also note that as the sample size gets larger, the variance of the posterior gets smaller – just as with ML estimation).

So, if we use a non-informative prior, the parameter estimates from **MARK** should match the MLE pretty closely. The following table (E.3, top of the next page) compares the MLE and MCMC estimates for the first 5 parameters. As expected, the estimates are close between the two methods.

Table E.3: Comparison of maximum likelihood (ML) estimates to estimates from MCMC analysis of the binomial survival data.

method	<i>i</i>				
	1	2	3	4	5
\tilde{S}_i (ML)	0.640	0.360	0.480	0.440	0.480
\tilde{S}_i (MCMC)	0.635	0.367	0.474	0.450	0.477

end sidebar

Finally, at the bottom of the output, you'll see some 'snippets' of **SAS** and **R** code, respectively:

```

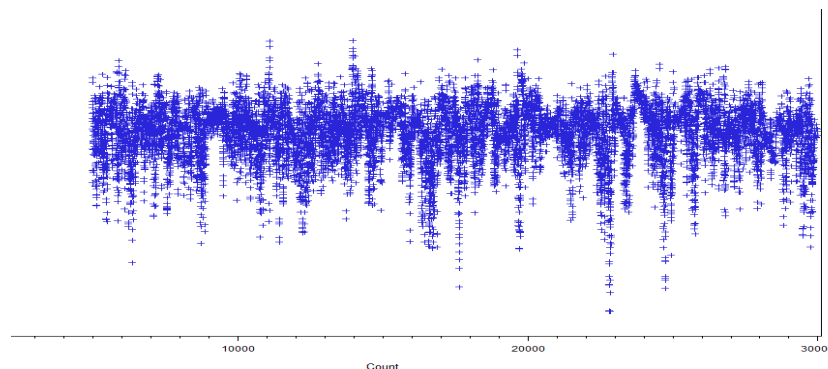
SAS Macro Variable Values:
%let ncovs=10; * Number of beta estimates;
%let nmeans=1; * Number of mean estimates;
%let ndesigns=0; * Number of design matrix estimates;
%let nsigmas=1; * Number of sigma estimates;
%let nrhos=0; * Number of rho estimates;
%let nlogit=10; * Number of real estimates;
%let MCMCfile='MCMC.BIN'; * Name and path to the MCMC.BIN file;

R Variable Values:
ncovs <- 10; # Number of beta estimates
nmeans <- 1; # Number of mean estimates
ndesigns <- 0; # Number of design matrix estimates
nsigmas <- 1; # Number of sigma estimates
nrhos <- 0; # Number of rho estimates
nlogit <- 10; # Number of real estimates
filename <- "MCMC.BIN"; # Name and path to the MCMC.BIN file

```

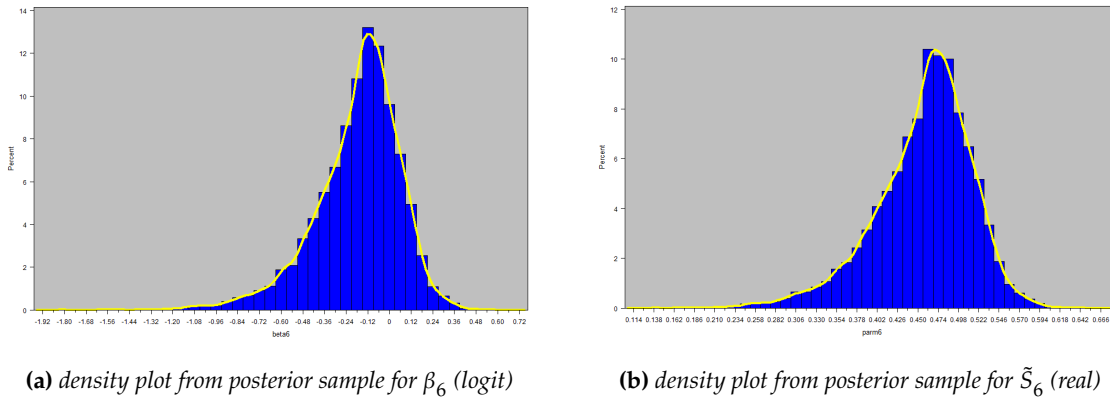
These bits of code contain the values for different variables you need to 'copy and paste' into the **SAS** or **R** code to further process the sample data. This code can be found in the **MARK** helpfile.

One particularly helpful stage of further analysis of the sample data is the visualization of the density and trace of the posterior samples. For example, here (Fig. E.5) is the time-series (trace) for a single chain for β_6 (based on 25,000 samples), after dropping the tuning and 'burn-in' samples:

**Figure E.5:** Time-series 'trace' of posterior samples for β_6 , on the logit scale. Trace based on 25,000 samples.

We expect that the trace should look like what has been described as a ‘fuzzy lawn’ – roughly an equal number of samples above and below some imaginary horizontal line drawn through the central mass of the samples. More specifically, we hope that the sampler doesn’t seem to get ‘stuck’ at any point (i.e., that it is able to explore the entire domain of the posterior distribution). Based on this somewhat subjective criterion, the trace for β_6 (Fig. E.5) would probably be considered as ‘satisfactory’. [As an exercise, try changing the step size parameters, and looking at see what this does to the trace. As noted, we try to set the step size to balance efficiency – moving around the posterior in as few samples as possible – with the ability to actually ‘reach’ all parts of the posterior.]

In the following figure (E.6), we compare the sample density for β_6 (on the logit scale) and the back-transformed parameter \tilde{S}_6 (on the real probability scale).



(a) density plot from posterior sample for β_6 (logit)

(b) density plot from posterior sample for \tilde{S}_6 (real)

Figure E.6: Density plots for MCMC samples for β_6 (on the logit scale), and the back-transformed value S_6 (on the real probability scale).

We are primarily interested in establishing that the density plots are unimodal. If the plots have > 1 mode, then there is a chance the sampler was ‘stuck’ in a local minimum, and has biased our parameter estimates (this is discussed in some detail with respect to multi-state models (Chapter 10), where such multi-modal posterior distributions are not uncommon). As discussed earlier, we can also consider attributes of the ‘width’ of the density plot (wide, narrow) to establish estimated parameter precision.

These are only 2 of a couple of graphical representations of the MCMC data. In addition, there are a very large (and seemingly ever-growing) number of diagnostic tools (some graphical, some analytical) to allow you to more fully explore the MCMC sample data. In this Appendix, we will be working primarily with examples where the posterior distribution has ‘good properties’ (although we do include one example where the results are somewhat ‘problematic’, reflecting some of the fairly common challenges you might face in practice). Working with MCMC (in **MARK**, or otherwise) inevitably means you’ll spend considerable time evaluating and working with these tools.

E.1.2. estimating the hyperparameters μ and σ

OK, so we see how we can use MCMC to derive estimates for the annual survival parameters, either ML estimates, \hat{S}_i , or shrinkage estimates, \tilde{S}_i . The most obvious reason we might choose to do so is because the MCMC approach generates the percentiles we can use to specify a 95% credibility interval. Not only is this arguably more ‘defensible’ in the conceptual sense (since the credibility interval is perhaps more meaningful, and certainly less confusing, than the more familiar ‘frequentist’ 95% confidence interval, since the latter is based on expectations from a theoretical number of replicates of the data), but the

credible interval may actually perform better (or at least as well) than the profile likelihood-based CI's for parameters estimated near the $[0, 1]$ boundaries.

But our primary interest here is the estimation of the hyperparameters for the random effects model. Recall from p. E-6 that, on the real probability scale, $\hat{\mu}_{\text{MOM}} = 0.482527$, and $\hat{\sigma}_{\text{MOM}} = 0.0441609$ (where the 'MOM' subscript indicates that the estimates were derived using the 'method of moments' approach to variance components analysis discussed in Appendix D). From the results of our MCMC analysis (p. E-11), we see that $\hat{\mu}_{\text{MCMC}} = -0.0604595$, and $\hat{\sigma}_{\text{MCMC}} = 0.1792769$, on the logit scale (here, the 'MCMC' subscript indicates that the estimates were derived using MCMC).

What happens if we back-transform our estimates from the MCMC analysis from the logit scale to the real probability scale? We'll start with μ – our best estimate of the overall mean of the parameter:

$$\hat{\mu} = \frac{e^{-0.0604595}}{1 + e^{-0.0604595}} = 0.48489,$$

which is close to the value estimated using the 'method of moments' approach ($\hat{\mu}_{\text{MOM}} = 0.482527$). So, it appears straightforward to derive our estimate of the overall mean μ , by taking the estimate from the MCMC analysis on the logit scale, and back-transforming to the real probability scale.

What about the estimate of process variance, σ ? What happens if we simply back-transform $\hat{\sigma}_{\text{MCMC}} = 0.1792769$ from the logit to the real probability scale?

$$\hat{\sigma} = \frac{e^{0.1792769}}{1 + e^{0.1792769}} = 0.5447,$$

which is clearly not close (not even remotely) to the estimate from the 'method of moments' approach ($\hat{\sigma}_{\text{MOM}} = 0.044161$), which we accept to be 'correct' for these data (see Appendix D).

There are several issues to consider here. First, perhaps we shouldn't be trying to back-transform at all, and should evaluate our estimates directly on the logit scale on which they are estimated. Often the logit scale is the biologically relevant scale at which to work. The logit scale is more likely to provide a linear scale to model the effects of environmental covariates, e.g., precipitation or temperature. However, this sidesteps the objective of comparing the estimates of σ from MCMC with those generated by the 'method of moments' random effects approach. While we might be satisfied and generally safe using estimates of process variance from the moments-based approach, that particular method doesn't apply well to complex models, and we would like to be able to use the more flexible MCMC approach in such cases. Moreover, one of the uses of process variance is in analysis stochastic projection models (as described in the introduction to Appendix D), which are projected using transition probabilities on the real $[0, 1]$ probability scale, not the logit scale.

A second possibility might be that we're not properly accounting for the effects of the transformation on the estimated variance on the transformed scale. If you've worked through Appendix B, it might seem reasonable to consider using the Delta method (Appendix B) to estimate the variance μ after transformation from the logit to the real probability scale.

From Appendix B, we write the transformation function $f(\mu)$ as

$$f(\mu) = \frac{e^{\mu}}{1 + e^{\mu}}.$$

Thus, to first order,

$$\widehat{\text{Var}} \approx (f'(\mu))^2 \sigma_{\mu}^2$$

$$\begin{aligned}
&= \left(\frac{\partial f(\mu)}{\partial \mu} \right)^2 \widehat{\text{Var}}(\hat{\mu}^2) \\
&= \left(\frac{e^{\hat{\mu}}}{1 + e^{\hat{\mu}}} - \frac{(e^{\hat{\mu}})^2}{(1 + e^{\hat{\mu}})^2} \right)^2 \widehat{\text{Var}}(\hat{\mu}).
\end{aligned}$$

From p. 11, we see that $\hat{\mu}_{\text{MCMC}} = -0.06046$. Now, what should we use for the estimate of the variance of μ ? There are 2 values in the output on p. E-11 that you might consider for the variance term in the Delta approximation. First, we see that the variance for the posterior for the parameter μ is estimated as $0.145678^2 = 0.0212218$. We also have the estimate of σ as a parameter itself, $\hat{\sigma} = 0.17928$, such that $\widehat{\text{Var}}(\hat{\sigma}) = 0.17928^2 = 0.03214$. Let's try the variance of μ (0.0212218) first.

$$\begin{aligned}
\widehat{\text{Var}} &\approx \left(\frac{e^{-0.06046}}{1 + e^{-0.06046}} - \frac{(e^{-0.06046})^2}{(1 + e^{-0.06046})^2} \right)^2 (0.0212218) \\
&= 0.0013240.
\end{aligned}$$

Thus, $\widehat{\text{Var}}(\hat{\mu})$ on the logit scale would be approximated as $\sqrt{0.0013240} = 0.03214$. While this is considerably closer to the estimate from the 'method of moments' approach (0.04416) than our naive back-transformed estimate of 0.5447, there is still a fair discrepancy (almost 30% difference) between the estimates.*

Now, let's repeat the calculation, but this time, using the estimate of $\widehat{\text{Var}}(\hat{\sigma}) = 0.17928^2 = 0.03214$ for the variance term.

$$\begin{aligned}
\widehat{\text{Var}} &\approx \left(\frac{e^{-0.06046}}{1 + e^{-0.06046}} - \frac{(e^{-0.06046})^2}{(1 + e^{-0.06046})^2} \right)^2 (0.03214) \\
&= 0.0020051,
\end{aligned}$$

and, thus, our estimate of $\hat{\sigma}$ on the real probability scale would be approximated as $\sqrt{0.0020051} = 0.0448$. We are rather pleased to observe that this value is close to the estimate from the 'method of moments' analysis (0.0442), differing only in the fourth decimal place.

However, before we become overly satisfied, it's important to understand 'why' this seems to have worked. The key is in remembering that μ and σ are being estimated as parameters, and while there is uncertainty in the estimate of each, such that each has its own sampling variance (based on the distribution of MCMC samples for each parameter), the estimate of $\hat{\sigma}$ (as the mean of the posterior for the σ parameter) is in fact the best estimate of the random variation of the annual S_i around the mean, μ , which is itself estimated as the mean of the posterior for the μ parameter. So, the correct value to use for the variance in the Delta approximation is the square of the estimate of σ itself.

Does this approximation always work? As noted in Appendix B, the Delta method assumes that the transformation function is effectively linear in the region where most of the data reside. For some parameters, especially those near the 0 or 1 boundaries, this may not necessarily be the case. Further,

* In fact, this estimate of 0.03214 is approximately equal to the estimate for σ you would obtain if (i) you took the entire posterior sample, back-transformed it from the logit scale to the real probability scale, and then (ii) estimated σ from this back-transformed distribution. If you think hard about what is meant by the estimate of σ you used in the Delta transformation to generate the value of 0.03214, you should be able to see the reason for this.

in the present example, we used a non-informative ‘flat prior’. If instead we had used an informative prior – in particular, a prior that was asymmetrical over the $[0, 1]$ interval – then it is quite likely that the Delta method may not work particularly well.

begin sidebar

estimating σ by simulation

As noted in Appendix B, where we introduce the Delta method approximation to estimating the variance of functions of one or more parameters (as we just applied, above), another approach to estimating the variance is to use numerical simulation or bootstrapping. Such numerical methods are less convenient in many instances than the Delta method, but are generally less susceptible to violation of some of the necessary assumption required for the Delta method to perform well.

Here, we introduce a simple approach here for estimating the process variance σ^2 , based on MCMC sample data. We will demonstrate the method using the binomial survival example. Recall that the ‘method of moments’ approach generated estimates of $\hat{\mu} = 0.48253$ and $\hat{\sigma} = 0.04416$. Recall also that our MCMC estimates of μ and σ on the logit scale (based on the mean of the posterior distribution for both parameters) were $\hat{\mu}_{\text{MCMC}} = -0.06046$ and $\hat{\sigma}_{\text{MCMC}} = 0.17928$.

What we’re going to do is simulate data on the logit scale, given estimates of μ and σ , back-transform the simulated data from the logit scale to the real probability scale, and then estimate σ from these back-transformed simulated data. The only challenge is deciding how to simulate data on the logit scale. As noted earlier, **MARK** samples from the posterior based on the assumption of a logit normal proposal distribution. Thus, our approach will be to

- simulate a logit normal sample based on $(\hat{\mu}_{\text{MCMC}} + \mathcal{N}(0, \hat{\sigma}_{\text{MCMC}}))$
- back-transform the simulated logit normal data to the real probability scale
- estimate μ and σ from the back-transformed simulated data

Here is a short snippet of **R** code that implements this sequence of steps, using parameter estimates from the binomial survival analysis. To facilitate comparison, we also include the estimates of μ and σ from the ‘methods of moments’ approach (above):

```
# enter parameters from MCMC
mu=-0.0604595; sigma_mean=0.1792769;

# draw 100000 random samples from logit normal distribution
lmsamp <- rnorm(100000,mean=mu,sd=sigma_mean)

# back-transform simulated data from logit -> real
backtrans <- exp(lmsamp)/(1+exp(lmsamp));
```

All that remains is looking at the estimates of μ and σ from the back-transformed data:

```
# now estimate and assess mean and sigma estimate from back-transformed data
mean(backtrans);
0.4846203

sqrt(var(backtrans));
0.04444851
```

These values estimated from the back-transformed data simulated using estimates of μ and σ from the MCMC analysis are clearly close to the estimates of the mean (0.482527) and σ (0.0441609) from the ‘method of moments’ random effects model.

end sidebar

What about credibility intervals for our estimates for $\hat{\mu}$ and $\hat{\sigma}$? Returning to the **MARK** output for the analysis of the binomial survival data, we see that the 95% frequency percentile-based interval reported on the logit scale for $\hat{\mu}$ and $\hat{\sigma}$ were $[-0.3477, 0.2828]$ and $[0.0280, 0.5470]$, respectively. The 95% HPD for $\hat{\mu}$ and $\hat{\sigma}$ were $[-0.3601, 0.2580]$ and $[0.0194, 0.4537]$, respectively.

Which credibility interval should we report? As discussed earlier, we might generally use the HPD, because it is often more defensible than the quantile-based interval when the posterior distribution is asymmetrical or (worse) multi-modal. For the binomial survival data, there is evidence suggesting that the posterior distribution for the mean μ is likely symmetrical, since the reported mean, median, and mode are all quite close: -0.0605 , -0.0617 , and -0.0695 , respectively. In contrast, the mean, median and mode reported for the variance σ are quite different (0.1793 , 0.1251 , and 0.0411 , respectively), suggesting the posterior distribution for σ is likely asymmetrical.

Our intuition is supported by visual examination of the probability density plots for both parameters. We see that the plot for the mean μ (Fig. E.7a) is indeed quite symmetrical, while for the variance σ , the posterior probability distribution (Fig. E.7b) is quite asymmetrical (strong right-skew) as expected – reporting the HPD might be more appropriate for this parameter.

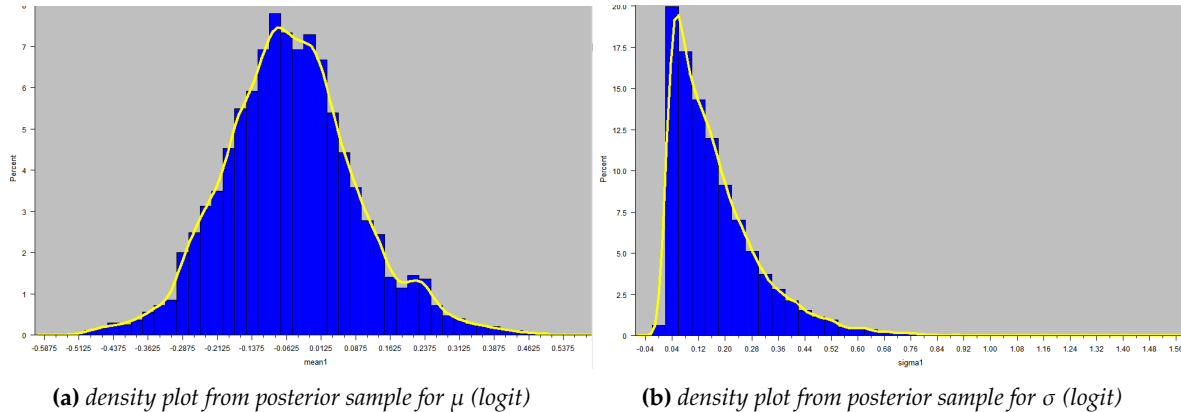


Figure E.7: Density plots for MCMC samples for μ and σ (on the logit scale).

hyperparameter estimates and the design matrix...

There is a subtle, but very important point that you must be aware of when considering estimates of the hyperparameters. What the MCMC routine(s) in **MARK** estimate is the mean (μ) and standard deviation (σ) of the β parameters, *not* the real parameters. In contrast, when estimating μ and σ using the ‘method of moments’ approach, we are generally using the real parameter estimates.

For example, consider the following simulated CJS data set, **MCMC_DM.inp**: 11 sampling occasions, 500 individuals released per occasion, with a true generating model of $\{\varphi_t p\}$. The true mean and standard deviation of the parameters values used in the simulation were $\mu = 0.671$ and $\sigma = 0.0681$.

We’ll start by fitting 3 ‘different versions of the true generating model, $\{\varphi_t p\}$ ’ that are, in fact, identical, in terms of the reconstituted real parameter estimates, but which differ from each other in ‘structure’. First, we’ll fit a model using a simple time-dependent PIM for φ , and a constant PIM for p .

Then, we’ll fit a model with an identity design matrix, using the same underlying PM structure.

You should recall from Chapter 6 that the PIM-only model assumes an identity design matrix, so, these two models are entirely equivalent. You should also recall that when using an identity design matrix, each interval has its own β parameter (this becomes important in a moment).

B1: t1	B2: t2	B3: t3	B4: t4	B5: t5	Parm	B6: t6	B7: t7	B8: t8	B9: t9	B10: t10	B11: p
1	0	0	0	0	1:Phi	0	0	0	0	0	0
0	1	0	0	0	2:Phi	0	0	0	0	0	0
0	0	1	0	0	3:Phi	0	0	0	0	0	0
0	0	0	1	0	4:Phi	0	0	0	0	0	0
0	0	0	0	1	5:Phi	0	0	0	0	0	0
0	0	0	0	0	6:Phi	1	0	0	0	0	0
0	0	0	0	0	7:Phi	0	1	0	0	0	0
0	0	0	0	0	8:Phi	0	0	1	0	0	0
0	0	0	0	0	9:Phi	0	0	0	1	0	0
0	0	0	0	0	10:Phi	0	0	0	0	1	0
0	0	0	0	0	11:p	0	0	0	0	0	1

Finally, we'll fit a model using an 'offset-coded' design matrix, where the final interval is used as the reference (intercept):

B1: int	B2: t1	B3: t2	B4: t3	B5: t4	Parm	B6: t5	B7: t6	B8: t7	B9: t8	B10: t9	B11: p
1	1	0	0	0	1:Phi	0	0	0	0	0	0
1	0	1	0	0	2:Phi	0	0	0	0	0	0
1	0	0	1	0	3:Phi	0	0	0	0	0	0
1	0	0	0	1	4:Phi	0	0	0	0	0	0
1	0	0	0	0	5:Phi	1	0	0	0	0	0
1	0	0	0	0	6:Phi	0	1	0	0	0	0
1	0	0	0	0	7:Phi	0	0	1	0	0	0
1	0	0	0	0	8:Phi	0	0	0	1	0	0
1	0	0	0	0	9:Phi	0	0	0	0	1	0
1	0	0	0	0	10:Phi	0	0	0	0	0	1
0	0	0	0	0	11:p	0	0	0	0	0	1

If you run these 3 models, you'll see (below) that (as expected), they all have identical fits to the data:

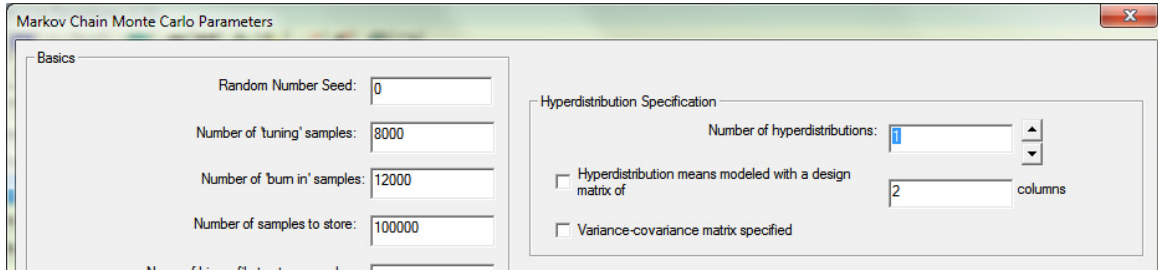
Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance	-2Log(L)
{phi(t)p(.) - logit - PIM}	19047.7167	0.0000	0.33333	1.0000	11	742.7386	19025.6890
{phi(t)p(.) - DM-identity}	19047.7167	0.0000	0.33333	1.0000	11	742.7386	19025.6890
{phi(t)p(.) - DM-offset}	19047.7167	0.0000	0.33333	1.0000	11	742.7386	19025.6890

Next, we'll use a 'method of moments' random effects approach to estimate μ and σ for the 10 survival parameters ($\varphi_1 \rightarrow \varphi_{10}$), on the real probability scale. As expected, the fit of the random effects models to each of the 3 model structures is identical (as shown in the browser below). And, as expected, each random effects model yields the same estimate of $\hat{\mu} = 0.664275$, and $\hat{\sigma} = 0.0569992$.

Model	AICc	Delta AICc	AICc Weight	Model Likelihood	No. Par.	Deviance	-2Log(L)
{phi(t)p(.) - logit - PIM: Random Effects fit}	19046.9113	0.0000	0.19978	1.0000	10.49349	742.9485	19025.8990
{phi(t)p(.) - DM-identity: Random Effects fit}	19046.9113	0.0000	0.19978	1.0000	10.49349	742.9485	19025.8990
{phi(t)p(.) - DM-offset: Random Effects fit}	19046.9113	0.0000	0.19978	1.0000	10.49349	742.9485	19025.8990
{phi(t)p(.) - logit - PIM}	19047.7167	0.8054	0.13355	0.6685	11	742.7386	19025.6890
{phi(t)p(.) - DM-identity}	19047.7167	0.8054	0.13355	0.6685	11	742.7386	19025.6890
{phi(t)p(.) - DM-offset}	19047.7167	0.8054	0.13355	0.6685	11	742.7386	19025.6890

Next, we'll use MCMC to estimate μ and σ as parameters of hyperdistributions we create. But – and this is critical – remember that **MARK** creates hyperdistributions for the β parameters, not the real parameters (whereas in the preceding step, we used the ‘method of moments’ random effects models to estimate μ and σ for the *real* parameter estimates).

For each of the 3 model forms (PIM, identity DM, offset DM), we applied MCMC, using 8,000 tuning samples, 12,000 burn-in samples, and 100,000 samples to generate the posterior from which we make inference.



In the following (Table E.4), we compare the estimates of μ and σ for each of the 3 model forms, derived using either the ‘method of moments’ random effects model (MOM), or the MCMC approach:

Table E.4: Comparison of estimates of μ and σ using ‘method of moments’ random effects (MOM), using real parameter estimates $\hat{\varphi}_1 \rightarrow \hat{\varphi}_{10}$, and MCMC back-transformed from the mode of the posterior for $\hat{\beta}_1 \rightarrow \hat{\beta}_{10}$ on the logit scale \rightarrow real probability scale.

model	MOM		MCMC	
	$\hat{\mu}$	$\hat{\sigma}$	$\hat{\mu}$	$\hat{\sigma}$
PIM	0.6643	0.0570	0.6628	0.0545
DM (identity)	0.6643	0.0570	0.6660	0.0552
DM (offset)	0.6643	0.0570	0.5028	0.0888

As we see, the MCMC estimates from the PIM-based and identity DM-based models are near identical, both with each other, and with the MOM estimates. On the other hand, the MCMC estimates for the offset DM-based model are clearly ‘off’, in a big way.

Why? The answer is simple, but important. In the offset DM, the individual β parameters do not correspond to individual estimates of φ on the real probability scale, which would be the case using the identity DM (which is assumed using the PIM-only approach). In the offset DM, the β parameters form the logistic function from which each individual φ is derived: β_1 is the intercept, and $\beta_2 \rightarrow \beta_{10}$ represent the effect of a particular time step as a deviation from this intercept. This is discussed in considerable detail in Chapter 6.

So, for the offset DM model, estimates of μ and σ are derived on the distribution of β estimates from the logistic model, and not some underlying distribution of parameter values. We can confirm this by applying the MOM approach to the β parameters $\beta_1 \rightarrow \beta_{10}$ from the DM-offset model. Doing so yields estimates of $\hat{\mu} = 0.008450$ and $\hat{\sigma} = 0.3925032$, respectively, which are quite close to the posterior modes from the MCMC analysis (and, when back-transformed, yield estimates of $\hat{\mu} = 0.5021$ and $\hat{\sigma} = 0.0981$ on the real probability scale, which are very close to those shown for this model in Table E.4).

Thus, if you're using MCMC to derive estimates of μ and σ (and, as we'll demonstrate shortly, there are cases where doing so offers more flexibility than the 'method of moments' random effects approach), you need to understand that your choice of DM (explicit, or otherwise) may have a big impact on your estimates – because you are estimating from the distribution of β parameter estimates, and not the estimates of φ on the real probability scale. In general, estimates of μ and σ will only be easily interpretable if you use an identity structure for the design matrix.

E.1.3. Example 2 – California mallard survival re-visited

Here, we will re-visit an analysis of a long-term dead recovery data set based on late summer banding of adult male mallards (*Anas platyrhynchos*), banded in California every year from 1955 to 1996 ($k = 41$) (Franklin *et al.*, 2002). This example was introduced in Appendix D (section D.4.2), where we focussed on the fitting of various random effects models to the mallard recovery data, including a model where survival, S , varied around some unknown mean μ , with unknown process variance σ^2 . We referred to this model as $\{S_{\mu, \sigma^2} f_t\}$. If you still have the analysis files for these data that you generated in Appendix D (**california-male-mallard.dbf** and **california-male-mallard.fpt**), this model might still be in your results browser. If not, you can either go back to section D.4.2 in Appendix D and re-create the model, or simply follow our summary of the results for fitting this model to the data (shown below):

```
California mallard data (males) -- random effects c-hat = 1.1952000

Beta-hat SE(Beta-hat) Label
-----
0.629440 0.015657 Intercept

Par. Num S-hat SE(S-hat) S-tilde SE(S-tilde) RMSE(S-tilde)
-----
1 0.423020 0.046257 0.447682 0.040257 0.047211
2 0.689787 0.087061 0.666101 0.060185 0.064678
3 0.666269 0.105860 0.647494 0.064989 0.067647
4 0.526082 0.079841 0.553727 0.055754 0.062231
5 0.685033 0.123080 0.638352 0.068445 0.082849
6 0.481012 0.095575 0.523707 0.058176 0.072161
7 0.586389 0.102964 0.600340 0.062774 0.064306
8 0.751006 0.136897 0.684031 0.071333 0.097847
9 0.529065 0.083367 0.562724 0.056508 0.065774
<<snipped to save space>>
32 0.610498 0.056343 0.621005 0.045188 0.046393
33 0.667301 0.063895 0.666712 0.049460 0.049464
34 0.685156 0.063149 0.699888 0.048406 0.050598
35 0.935051 0.081332 0.862595 0.058412 0.093069
36 0.577746 0.051772 0.611583 0.041696 0.053699
37 0.738200 0.075641 0.732753 0.054859 0.055129
38 0.846765 0.108590 0.793773 0.066163 0.084768
39 0.535913 0.064761 0.525179 0.047116 0.048323
40 0.675469 0.067890 0.663663 0.052836 0.054139
41 0.583104 0.085443 0.593429 0.061669 0.062527

Naive estimate of sigma^2 = 0.0079384 with 95% CI (0.0031332 to 0.0173404)

Estimate of sigma^2 = 0.0081075 with 95% CI (0.0041907 to 0.0166408)

Estimate of sigma = 0.0900415 with 95% CI (0.0647353 to 0.1289993)
```

Recall that in Appendix D, we used time-structure for the recovery parameter f . We do so because any constraint applied to f will impart (or 'transfer') more of the variation in the data to the survival parameter S , such that the estimated process variance $\hat{\sigma}^2$ will be 'inflated', relative to the true process

variance. In general, you want to estimate variance components using a fully time-dependent model, for all parameters, even if such a model is not the most parsimonious given the data. This is also true if using the MCMC approach we're introducing here.

We see from the results (above) of fitting model $\{S_{\mu, \sigma^2} f_t\}$ to the data, that the estimated mean survival on the real probability scale was $\hat{\mu} = 0.630244$, and process variance was $\hat{\sigma}^2 = 0.0895355^2 = 0.0080166$. Now, let's see if we can replicate this analysis, using the MCMC capabilities in **MARK**. First, retrieve model $\{S_t f_t\}$ in the browser to make it the 'active' model. We are going to re-run this model, using MCMC. Make sure that you check the logit link function. If you built the model using the logit link in the first instance, you can and should check the box telling **MARK** you will provide initial values from that model. If not, take a moment and re-run the model first, using standard ML estimation and the logit link. This is a large (41 years of data), complex dataset, and experience indicates you should probably use the simulated annealing optimization routine (so check the 'alternate optimization' box). Be advised that convergence can take some time.

Once finished, we're ready to start our MCMC analysis. Re-run the model, checking the '**logit**' link function, '**Provide initial parameter estimates**', and (of course) the '**MCMC Estimation**' check-boxes.

Next, you'll specify the '**Monte Carlo simulation parameters**'. We'll use the standard defaults for the number of '**tuning samples**' (4,000) and '**burn-in**' (1,000). However, because of the size and complexity of the data set, and the number of parameters being carried in the model, we'll increase the number of MCMC '**samples to store**' from the default of 10,000 to 25,000. Finally, we'll set the '**Number of hyperdistributions**' to 1 (specified by the parameters μ and σ). We'll use default settings for everything else. Once you're ready, click the '**OK**' button.

Next, you'll be presented with a popup window asking you to select which parameters you want to include in the single hyperdistribution. We're interested in estimating μ and σ for the survival parameter, S . Since all S_i and f_i parameters are identifiable in a Brownie model (Chapter 8), we enter '1 to 41' (corresponding to survival parameters $\{S_1, S_2, \dots, S_{41}\}$).

Next, we're asked to enter 4 values specifying the step size, starting value, and the two parameters governing the '**Hyperdistribution parameters**'. We'll accept the defaults (i.e., we'll use default starting values, step size, and the default noninformative 'flat' priors for both μ and σ).

Finally, we're asked to provide '**Initial parameter estimates**' – we'll retrieve starting estimates from model $\{S_t f_t\}$ fit with the logit link (by selecting it from the list of candidate models already in the browser). One the starting values have been retrieved, click the '**OK**' button, and the MCMC sampler will start. Remember – we're doing 30,000 total iterations of the sampler, which for a large and complex data set, will take some time (15-20 minutes on a typical desktop computer).

Once the MCMC samples are finished, **MARK** will dump the summary statistics to the editor. Since our main focus here is on estimation of μ and σ , we'll focus on that part of the output:

```

78:f          -2.6935079      0.0764739      -2.6928311
79:f          -2.8313228      0.0794477      -2.8312507
80:f          -2.9239283      0.0906455      -2.9222336
81:f          -2.7381426      0.0708654      -2.7372413
82:f          -2.5601021      0.0746314      -2.5597806
83:f          -2.4440585      0.0970018      -2.4444575
84:mu(1)      0.5678008      0.0641315      0.5667889
85:sigma(1)   0.3883343      0.0694615      0.3821488
86:-2log Likelihood    65283.015      12.890003      65282.456
      -2log Likelihood for means of beta estimates =    65215.190
      DIC =      595.14177

```

We see that the estimate (on the logit scale) of $\hat{\mu}_{\text{MCMC}} = 0.5678008$, and $\hat{\sigma}_{\text{MCMC}} = 0.3883343$. The

back-transform of μ from the logit to the real probability scale yields

$$\hat{\mu} = \frac{e^{0.5678008}}{1 + e^{0.5678008}} = 0.638256,$$

which is close to the estimate $\hat{\mu} = 0.630244$ from the ‘methods of moments’ approach (above), as expected.

What about $\hat{\sigma}$? Recall from the binomial survival analysis we covered earlier that we can’t simply take a back-transform of $\hat{\sigma} = 0.3883343$ from the logit scale to the real probability scale. Instead, we can try either the Delta method, or a numerical simulation approach (as covered in the preceding -sidebar-). For purposes of comparison, we’ll use both here. Recall that to first order,

$$\widehat{\text{Var}} \approx \left(\frac{e^{\hat{\mu}}}{1 + e^{\hat{\mu}}} - \frac{(e^{\hat{\mu}})^2}{(1 + e^{\hat{\mu}})^2} \right)^2 \widehat{\text{Var}}(\hat{\mu}^2).$$

From above, we see that $\hat{\mu}_{\text{MCMC}} = 0.5678008$. From the analysis of the binomial survival data, we know that we use the estimate of σ as a parameter, $\hat{\sigma}_{\text{MCMC}} = 0.3883343$, such that $\widehat{\text{Var}}(\hat{\sigma}) = 0.3883343^2 = 0.150805$.

Thus,

$$\begin{aligned} \widehat{\text{Var}} &\approx \left(\frac{e^{0.5678008}}{1 + e^{0.5678008}} - \frac{(e^{0.5678008})^2}{(1 + e^{0.5678008})^2} \right)^2 (0.150805) \\ &= 0.0080391, \end{aligned}$$

and, thus, our estimate of $\hat{\sigma}$ on the real probability scale would be approximated as $\sqrt{0.0080391} = 0.089661$, which is close to the estimate from the ‘methods of moments’ analysis (0.089536).

Using the numerical simulation approach introduced earlier, the estimates of μ and σ are

Estimation of mu and sigma on back-transformed scale transformed logit

This is estimated S (mu): 0.633954, compared to moments estimate of 0.630244

This is estimated sigma: 0.0876645, compared to moments estimate of 0.089536

Again, close to the values estimated using the ‘methods of moments’ approach.

However, in our preceding two examples, mean survival was near the middle of the interval (0.4-0.6). Why is this important? As noted by White *et al.* (2009), the back-transformation of an estimate of σ to the value of σ on the real scale depends on the mean of the distribution. So, an estimate of $\sigma = 0.1$ with a mean of 0 on the logit scale results in a back-transformed real variable with a mean of 0.5 and $\sigma = 0.0025$ with n large. But, an estimate of $\sigma = 0.1$ but with a mean of 4 on the logit scale back-transforms to a real variable with a mean 0.982 and $\sigma = 0.0018$. White *et al.* noted that because of this relationship, ‘...interpretation of the estimates of the process variance on the logit scale must consider the mean as well’. Note that on the logit scale, $|\mu| \geq 3$ when back-transformed is approaching either the 0 or 1 boundary on the real probability interval over $[0, 1]$. The logit scale on the interval $[\mu = -3, \mu = 3]$ is linear, or nearly so.

So, as a final test before moving on, we'll consider a simulated live encounter mark-recapture (CJS) data set, where true apparent survival alternates from 0.8 to 0.9 in successive years (i.e., $\varphi_1 = 0.8, \varphi_2 = 0.9, \varphi_3 = 0.8 \dots$), with encounter probability constant at $p = 0.5$. Thus, the true mean survival $\mu = 0.85$, and the true process variance $\sigma^2 \rightarrow 0.0025$. The data set consists of 17 sampling occasions, so 16 intervals for survival, and 16 occasions for recapture. Given that apparent survival alternates between 0.8 and 0.9 in successive years, this means 8 intervals over which survival is 0.8, and 8 intervals over which survival is 0.9. True $\mu = 0.85$, and true $\sigma^2 = 0.00267$. For the simulation, we released 500 newly marked individuals per occasion. To simplify the modeling, and to let us use all 16 estimates of φ , we used model $\{\varphi_i p\}$ as our starting, fixed effects model. Normally, we estimate process variance using a fully time-dependent model (see Appendix D for a discussion of this point), but fixing p constant eliminates confounded parameters, and doing so is unlikely to bias results of one method of estimating σ versus the other.

Using these simulated data (contained in **sigmasim.inp**), we estimated the process variance, using both the 'methods of moments' and MCMC approaches. Using the 'methods of moments' approach, $\hat{\mu}_{\text{MOM}} = 0.8517$, which is pretty close to the true mean of 0.85. Estimated $\hat{\sigma}_{\text{MOM}} = 0.0586$ (such that $\hat{\sigma}^2 = 0.00344$), which is not too far off the true process variance of $\sigma^2 = 0.00267$. The discrepancy undoubtedly reflects, at least in part, constraining the encounter probability p to be constant over time. Of more importance here, though, is the MCMC estimate, and if our back-transformation of these estimates from the logit scale to the real scale gives us values close to those from the 'method of moments' approach.

The MCMC analysis (using default values for number of samples), generated estimates of $\hat{\mu}_{\text{MCMC}} = 1.8227$ and $\hat{\sigma}_{\text{MCMC}} = 0.5188$. Using the Delta method, the back-transformed estimate for the mean on the real scale is $\hat{\mu} = 0.8609$, which perhaps not surprisingly is quite close to the true mean of 0.85. For σ , the back-transformed estimate of $\hat{\sigma} = 0.0621$ (such that $\hat{\sigma}^2 = 0.00386$). So, even when true mean survival is relatively close to the boundary, estimates of σ from either the 'method of moments' approach (0.0586) or the MCMC approach (0.0621) are quite close. Whether this holds as μ gets much closer to the boundary needs further study. In such cases, a different link function might be needed (as was the case for some problems considered in Appendix D). In fact, for this example, using the identity link for the MCMC analysis, the reported values for μ (0.8516) and σ (0.0621) are identical to those generated using the back-transformation from parameters estimated on the logit scale.

For now, we will defer further discussion of this issue, and proceed under the assumption that, in practice, back-transformed estimates of σ from the logit scale to the real probability scale are robust and unbiased.

E.1.4. Example 3 – environmental covariates re-visited

In the preceding two examples, we explored the mechanics of using MCMC in **MARK** to derive an estimate of process variance, σ^2 . If that is all we were interested in, we might choose not to bother with MCMC, when the 'method of moments' approach has been shown to be extremely robust, and is much faster computationally than MCMC.

However, as mentioned in the preamble to this Appendix, there are situations where the 'method of moments' approach is insufficiently flexible to allow us to estimate process variance for certain types of models. We demonstrate this by re-considering an example presented in Appendix D, where apparent survival is believed to vary as a function of a binary environmental covariate (water level). Again, we imagine we have some live encounter (CJS) data collected on a fish population studied in a river that is subject to differences in water level. We hypothesize that annual fish survival is influenced by variation in water level. For this study, we simulated $k = 21$ occasions of mark-recapture data (contained in **level-covar.inp**). Over each of the 20 intervals between occasions, water flow was characterized as either 'average' (A) or 'low' (L) (more specific covariate information was not available). Here is the time

series of flow covariates: {AAAAALLAALALALLLAL}.

We began our analysis of these data in Appendix D considering 3 fixed effect models for apparent survival, φ : $\{\varphi_t p_t\}$, $\{\varphi \cdot p_t\}$ and $\{\varphi_{level} p_t\}$. The results from fitting these 3 models to the data are shown below:

Model	AICc	Delta AICc	AICc Weight	No. Par.	Deviance
{phi(t)p(t)}	16551.9575	0.0000	0.99993	39	1870.9737
{phi(level)p(t)}	16570.9746	19.0171	0.00007	22	1924.2826
{phi(.)p(t)}	16581.0875	29.1300	0.00000	21	1936.4079

There was strong evidence for variation over time in apparent survival, but no support for an effect of water level. If you look at the estimates from model $\{\varphi_{level}\}$ for average ($\hat{\varphi}_{avg} = 0.709$, SE = 0.0106) and low ($\hat{\varphi}_{low} = 0.650$, SE = 0.0100), the lack of any support for this model may not be surprising. At least, based on considering water level as a fixed effect.

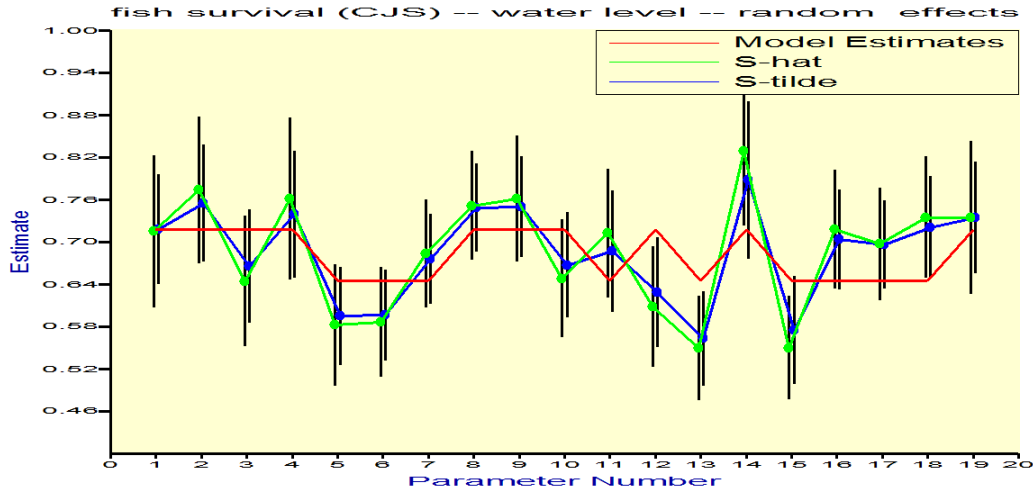
Next, we considered 2 different random effects models – one a simple intercept only (mean) model, which seemed to be consistent with the strong support for the simple time variation model $\{\varphi_t\}$, and a model where survival was thought to vary randomly around a (water) level-specific mean. In other words, we hypothesized $\mu_{low} \neq \mu_{avg}$. In Appendix D, we also assumed that $\sigma_{low}^2 = \sigma_{high}^2$. As noted, this is not directly testable using the moments-based variance components approach in **MARK**.

But, in fact, estimating a separate μ and σ for each water level is possible, using an MCMC approach. (Recall that we included parameters $\varphi_1 \rightarrow \varphi_{19}$ in the random effect – φ_{20} was not included because it was confounded with p_{21} .) The results browser containing the 3 fixed effects models, and the 2 random effects models, is shown below:

Model	AICc	Delta AICc	AICc Weight	No. Par.	Deviance
{phi(t)p(t): water level model – Random Effects Trace G=15.2401619}	16546.4694	0.0000	0.58836	35.24016	1873.0840
{phi(t)p(t): intercept only model – Random Effects Trace G=15.8100799}	16547.3766	0.9072	0.37380	35.81008	1872.8391
{phi(t)p(t)}	16551.9575	5.4881	0.03784	39	1870.9737
{phi(level)p(t)}	16570.9746	24.5052	0.00000	22	1924.2826
{phi(.)p(t)}	16581.0875	34.6181	0.00000	21	1936.4079

What is especially noteworthy here is that the random effects model with water level-specific means, $\{\varphi_{\mu_{level}\sigma_{level}^2}\}$, was the most parsimonious model in the model set, despite having no support whatsoever when considered in a fixed effects design. The near equivalence of the AIC_c weights between this model and the simpler ‘intercept only’ random effects model suggested that we couldn’t differentiate between the two, but whereas our initial conclusion strongly rejected the hypothesis that there was an influence of water level on apparent survival, our random effects modeling would suggest that perhaps we shouldn’t be quite so sure.

A plot showing the ML and shrinkage estimates, and (importantly here) the underlying model (the red line), for model $\{\varphi_{\mu_{level}\sigma_{level}^2}\}$, is shown at the top of the next page. The red line clearly indicates that there were 2 separate means being modeled, for the low and average water flow years, respectively. The estimated process variance is $\hat{\sigma}^2 = 0.00313$, and the estimate of the estimate for $\beta_1 = 0.0717$ in the linear model indicates that survival is higher in ‘average’ water level years (since we used ‘low’ level years as the reference level in our design matrix, above). What is also important here, is that the shrinkage estimates are clearly not constrained to fall exactly ‘on the red line’ – they represent shrunk estimates of apparent survival as if each estimate was drawn randomly from a sample with a water level-specific mean.



Now, let's look at the estimates for μ and σ from the most parsimonious model, $\{\varphi_{\mu_{\text{level}} \sigma_{\text{level}}^2}\}$. The model was structured to estimate μ separately for each water level. But, recall because of how we structured the design matrix, what is estimated for μ is the linear model:

$$\text{logit}(\varphi) = \beta_1 + \beta_2(\text{level}).$$

The estimates for the linear model coefficients were $\hat{\beta}_1 = 0.645584$, and $\hat{\beta}_2 = 0.071666$ (note that these coefficients are reported in the real scale, not the logit scale). Since we used a dummy coding of '1' for an 'average' water level year', then

$$\hat{\varphi}_{\text{avg}} = 0.645584 + 0.071666(1) = 0.717250,$$

and

$$\hat{\varphi}_{\text{low}} = 0.645584 + 0.071666(0) = 0.645584.$$

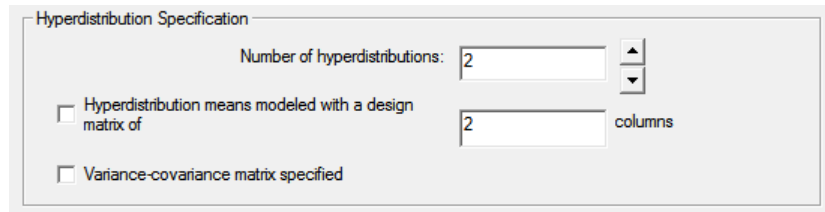
MARK also reports $\hat{\sigma} = 0.0559457$, but this is calculated over both water levels together (such that we might predict that 0.0559457 is the average of the process variances for each water level; we deal with this below). What we want, though, is an estimate of process variance for each water level separately. Is this possible using the 'method of moments' approach? The answer is 'no'. Both $\hat{\beta}_1$ and $\hat{\beta}_2$ are reported with an estimated SE, but these SE are estimated from the sum of process and sampling variation (i.e., total variance).

Thus, even though we could, with a bit of algebra, use these SE to come up with estimates of the variance for both water levels, the calculated variance would be total variance for each water level, and not process variance, which is what we're really after.

Our solution is to use the MCMC capabilities in **MARK**, to directly estimate process variance for both water levels separately. We simply specify 2 different hyperdistributions (one for average water level, and one for low water level), and estimate μ and σ for each hyperdistribution separately. Start by retrieving the fully time-dependent model $\{\varphi_t p_t\}$. We will re-run this model, using MCMC. We will use a logit link, and will use estimates from the fixed effects model $\{\varphi_t p_t\}$ as initial estimates.

Once you click the 'OK' button, you're presented with the window which lets you set the MCMC parameters. We'll keep the default values for the number of samples, the specification of priors and will use a single chain. The only change we need to make to the defaults in this window is the number of hyperdistributions – now we want 2: one for average water level, and one for low water level.

The first thing we do is change the value for ‘Number of hyperdistributions’ from 0 to 2:



Hyperdistribution Specification

Number of hyperdistributions: 2

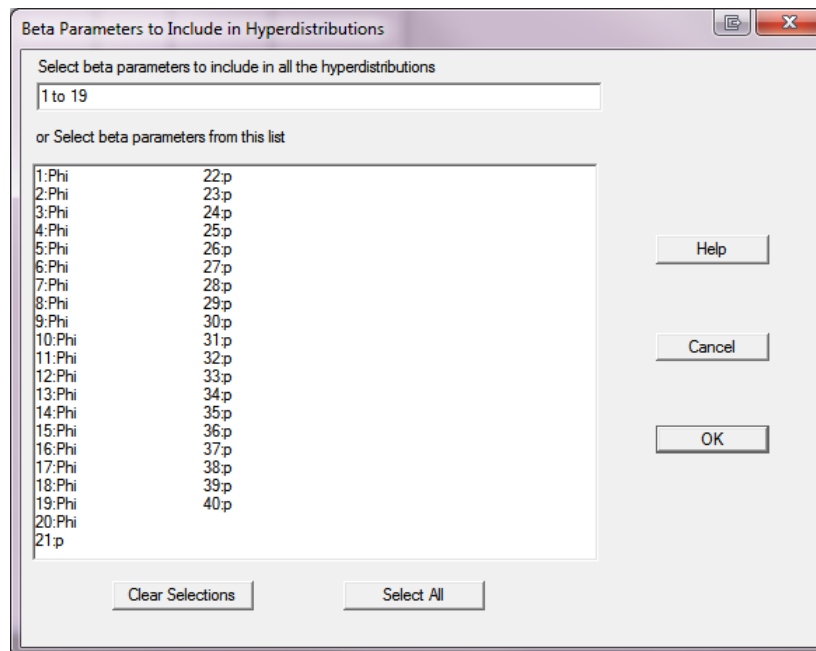
☐ Hyperdistribution means modeled with a design matrix of 2 columns

☐ Variance-covariance matrix specified

What we need to do next is specify which parameters are included in which hyperdistribution. With a bit of thought, you should realize that we want to include the parameters corresponding to average water years in one of the hyperdistributions, and the parameters corresponding to low water years in the other. **MARK** gives you a couple of ways to do this – in practice, you’ll decide which one you find easier. Since this is the first time we’ve dealt with > 1 hyperdistribution, we’ll demonstrate both approaches.

First, in the ‘**Hyperdistribution Specification**’ box, you’ll notice that the second item in the box is a checkbox to allow you model the ‘**hyperdistribution means with a design matrix**’. This is turned greyed out by default. The fact that it doesn’t immediately ‘turn on’ when you set the number of hyperdistributions > 1 suggests there is another, default approach to specifying the hyperdistributions.

Indeed, there is. The default approach has a couple of steps. First, once you’ve set the number of hyperdistributions to 2 (above), click the ‘**OK**’ button. This will bring up a window (shown below) where you list all the parameters which will be included in a hyperdistribution.



Beta Parameters to Include in Hyperdistributions

Select beta parameters to include in all the hyperdistributions

1 to 19

or Select beta parameters from this list

1:Phi	22:p
2:Phi	23:p
3:Phi	24:p
4:Phi	25:p
5:Phi	26:p
6:Phi	27:p
7:Phi	28:p
8:Phi	29:p
9:Phi	30:p
10:Phi	31:p
11:Phi	32:p
12:Phi	33:p
13:Phi	34:p
14:Phi	35:p
15:Phi	36:p
16:Phi	37:p
17:Phi	38:p
18:Phi	39:p
19:Phi	40:p
20:Phi	
21:p	

Help

Cancel

OK

Clear Selections

Select All

You’ve seen this window before, but in the previous example, we had only a single hyperdistribution. Here, we’re specifying 2 hyperdistributions. However, this is not the point where you assign a given parameter to a given hyperdistribution – it is merely where you tell **MARK** how many, and which parameters will, ultimately, be included in one hyperdistribution or another. For our purposes, we will enter ‘1 to 19’, as shown on the previous page.

Once we click the ‘OK’ button, **MARK** will present you with a window (shown at the top of the next page) where you ‘Specify the mean of the hyperdistribution for each parameter’.

Specify Mean of the Hyperdistribution for each Parameter

Specify Parameter-Specific Mean Value for each Hyper

1:Phi	mu(1)	11:Phi	mu(2)
2:Phi	mu(1)	12:Phi	mu(2)
3:Phi	mu(1)	13:Phi	mu(2)
4:Phi	mu(1)	14:Phi	mu(2)
5:Phi	mu(1)	15:Phi	mu(2)
6:Phi	mu(1)	16:Phi	mu(2)
7:Phi	mu(1)	17:Phi	mu(2)
8:Phi	mu(1)	18:Phi	mu(2)
9:Phi	mu(1)	19:Phi	mu(2)
10:Phi	mu(2)		

OK Cancel Default Reset All Paste

By default (as shown below), **MARK** will assume that if you specified two hyperdistributions, that half of the parameters will be included in the first hyperdistribution (i.e., μ_1), and the other half will be included in the second hyperdistribution (i.e., μ_2).

Specify Mean of the Hyperdistribution for each Parameter

Specify Parameter-Specific Mean Value for each Hyper

1:Phi	mu(1)	11:Phi	mu(2)
2:Phi	mu(1)	12:Phi	mu(2)
3:Phi	mu(1)	13:Phi	mu(2)
4:Phi	mu(1)	14:Phi	mu(2)
5:Phi	mu(1)	15:Phi	mu(2)
6:Phi	mu(1)	16:Phi	mu(2)
7:Phi	mu(1)	17:Phi	mu(2)
8:Phi	mu(1)	18:Phi	mu(2)
9:Phi	mu(1)	19:Phi	mu(2)
10:Phi	mu(2)		

But, this default might not be what you want. Moreover, if the total number of parameters over both hyperdistributions is odd (as it is in the present example, where we are including 19 parameters in the two hyperdistributions), you can’t divide them evenly between the two hyperdistributions. Or, as in the present case, where each type of year (average water level, low water level) in our study is associated with a different hyperdistribution.

Since this isn’t what **MARK** defaults to, what you need to do at this point is go through the list of parameters – all 19 of them – and for each one in turn, decide which hyperdistribution they belong to, and select (from the drop-down menu) or manually enter the appropriate choice: mu(1) or mu(2). Now, at this point, you need to remember which year (interval, parameter) corresponds to which water level.

Recall that the time series of water level covariates was: {AAAAALLAAALALALLLAL}, where ‘A’ is average, and ‘L’ is low. So, first 4 years had average water levels, followed by 3 years of low water levels, and so on. In the figure shown below, we’ve used $\mu(1)$ for average water level years, and $\mu(2)$ for low water level years. Note we only use the first 19 years, to eliminate confounding between the final φ and p estimates.

Specify Mean of the Hyperdistribution for each Parameter

Specify Parameter-Specific Mean Value for each Hyper

1:Phi	mu(1)
2:Phi	mu(1)
3:Phi	mu(1)
4:Phi	mu(1)
5:Phi	mu(2)
6:Phi	mu(2)
7:Phi	mu(2)
8:Phi	mu(1)
9:Phi	mu(1)
10:Phi	mu(1)
11:Phi	mu(2)
12:Phi	mu(1)
13:Phi	mu(2)
14:Phi	mu(1)
15:Phi	mu(2)
16:Phi	mu(2)
17:Phi	mu(2)
18:Phi	mu(2)
19:Phi	mu(1)

Once you click the ‘OK’ button, MARK will present the exact same sort of window, except here you specify the σ hyperparameter for each hyperdistribution. We follow the exact same process as above, assigning $\sigma(1)$ to average water years, and $\sigma(2)$ to low water years.

Specify Sigma of the Hyperdistribution for each Parameter

Specify Parameter-Specific Sigma Value for each Hyper

1:Phi	sigma(1)
2:Phi	sigma(1)
3:Phi	sigma(1)
4:Phi	sigma(1)
5:Phi	sigma(2)
6:Phi	sigma(2)
7:Phi	sigma(2)
8:Phi	sigma(1)
9:Phi	sigma(1)
10:Phi	sigma(1)
11:Phi	sigma(2)
12:Phi	sigma(1)
13:Phi	sigma(2)
14:Phi	sigma(1)
15:Phi	sigma(2)
16:Phi	sigma(2)
17:Phi	sigma(2)
18:Phi	sigma(2)
19:Phi	sigma(1)

Once you click the ‘OK’ button, you will be presented with the familiar window for specifying starting values, step size for the sampler, and the parameters governing the shape of the prior, for each of the 4 hyperparameters. For this example, we’ll accept the defaults.

Finally, you’re asked to enter initial parameter estimates, which you can retrieve from the appropriate model from the browser. Click ‘OK’, the MCMC sampler will starting running.

Here are the estimates (on the logit scale) for $\mu(1)$ and $\sigma(1)$ (corresponding to $\hat{\mu}_{avg}$ and $\hat{\sigma}_{avg}$, respectively), and $\mu(2)$ and $\sigma(2)$ (corresponding to $\hat{\mu}_{low}$ and $\hat{\sigma}_{low}$), where ‘avg’ and ‘low’ are the

two different levels of the water flow covariate:

```

39:p                0.0051108      0.1313146      0.0031311      -0.0178870
40:p                0.2805627      1.1353757      -0.1102470      -0.6830918
41:mu (1)           0.9493033      0.0930781      0.9468457      0.9192309
42:mu (2)           0.6155881      0.1199335      0.6143002      0.6284445
43:sigma (1)        0.2029072      0.1188329      0.1862616      0.1784712
44:sigma (2)        0.3101054      0.1218244      0.2879500      0.2621693
45:-2log Likelihood 16514.862      9.0845322      16514.222      16513.685
      -2log Likelihood for means of beta estimates = 16504.298
DIC = 1922.8737

```

Now, before we back-transform the MCMC estimates from the logit scale to the real probability scale, let's first step back and look at the other way we could have specified the parameters in each hyperdistribution. We'll re-run our analysis, except that this time, after specifying 2 hyperdistributions, we'll go ahead and check the box which allows us to model the hyperdistribution means using a design matrix:

Hyperdistribution Specification

Number of hyperdistributions: 2

☒ Hyperdistribution means modeled with a design matrix of 2 columns

☐ Variance-covariance matrix specified

Once we check the box, we're asked to specify the number of columns in the DM. Since our classification factor has 2 levels (average, and low), you might instinctively (at this point in the book, answers to such questions concerning the DM might have reached the level of 'instinct') decide that you need 2 columns: 1 for the intercept, and 1 coding for water level. Now, as we'll see shortly, adopting this fairly standard (and familiar) linear models 'intercept offset' approach may not be the easiest or best approach to generating estimates of μ for each hyperdistribution.

After specifying 2 hyperdistributions, and indicating you want a DM with 2 columns, click the 'OK' button. You'll then be asked to specify the parameters you want to include in the hyperdistributions ('1 to 19', for this example).

Now, when you click the 'OK' button at this stage, MARK will generate a window that represents the familiar design matrix 'spreadsheet' (shown at the top of the next page), with the number of columns equal to what you specified a couple of steps ago (for this example, 2 columns). Notice that both columns are initially all 0's – meaning, you need to manually specify the structure you want for the means of the hyperdistributions.

Earlier, we suggested that most of you are probably imagining modifying this DM template to reflect a familiar 'intercept offset' linear model. In other words,

$$\text{logit}(\varphi) = \beta_1 + \beta_2(\text{water level}),$$

where β_2 would represent the effect of changing water level relative to the reference level. If, for example, we used a '1' to code for the average water level, then β_2 would represent the degree to which apparent survival in years with an average water level deviated from the years with low water level. Alternatively, rather than using an 'intercept offset' approach, you could simply code each water level in its own column – in other words, if say the first column represented average water level, then entering a '1' for each average year. In the second column (which now represents low water level), you'd enter a '1' for those years with low water levels.

Parm	B1:	B2:
1:Phi	0	0
2:Phi	0	0
3:Phi	0	0
4:Phi	0	0
5:Phi	0	0
6:Phi	0	0
7:Phi	0	0
8:Phi	0	0
9:Phi	0	0
10:PH	0	0
11:PH	0	0
12:PH	0	0
13:PH	0	0
14:PH	0	0
15:PH	0	0
16:PH	0	0
17:PH	0	0
18:PH	0	0
19:PH	0	0

Both forms of the DM are shown in Fig. (E.8), below:

Parm	B1:	B2:
1:Phi	1	1
2:Phi	1	1
3:Phi	1	1
4:Phi	1	1
5:Phi	1	0
6:Phi	1	0
7:Phi	1	0
8:Phi	1	1
9:Phi	1	1
10:Phi	1	1
11:Phi	1	0
12:Phi	1	1
13:Phi	1	0
14:Phi	1	1
15:Phi	1	0
16:Phi	1	0
17:Phi	1	0
18:Phi	1	0
19:Phi	1	1

(a) standard 'intercept offset' coding

Parm	B1:	B2:
1:Phi	1	0
2:Phi	1	0
3:Phi	1	0
4:Phi	1	0
5:Phi	0	1
6:Phi	0	1
7:Phi	0	1
8:Phi	1	0
9:Phi	1	0
10:Phi	1	0
11:Phi	0	1
12:Phi	1	0
13:Phi	0	1
14:Phi	1	0
15:Phi	0	1
16:Phi	0	1
17:Phi	0	1
18:Phi	0	1
19:Phi	1	0

(b) single-factor 'identity' coding

Figure E.8: Alternate DM structures for modeling water level effect

In a moment, we'll address the question of which DM to use, and why. For the moment, assume we're using the 'intercept offset coding', shown in Fig. (E.8a). Once you click the 'OK' button, you're

presented with a window that lets you specify the σ for each hyperdistribution. In fact, it is exactly the same window we saw before for σ , and we modify it in exactly the same way:

So, we can use the DM to model the means of the hyperdistributions, but not the variances. This should make sense (since analysis of variance and linear models in general relate to structural relationships among means, and parsimonious estimates of those models, given the variance).

Go ahead and finish things up, and run the MCMC sampler for this model, coded using the ‘intercept offset coding’ DM. Here are results from our analysis:

```

---
40:p                0.2756508      1.1628521      -0.1756704      -0.5326210
41:designbeta(1)    0.6150444      0.1092533      0.6092173      0.5804592
42:designbeta(2)    0.3278746      0.1450860      0.3290370      0.3822768
43:sigma(1)        0.1932216      0.1209980      0.1748662      0.0655464
44:sigma(2)        0.3022333      0.1132706      0.2818723      0.2603783
45:-2log Likelihood 16514.719      8.8534776      16514.144      16513.136
-2log Likelihood for means of beta estimates = 16507.461
DIC = 1919.4241

```

Now, re-do the analysis, but this time, use the alternate approach to coding the DM, shown in Fig. (E.8b). Here are the results from our analysis using that DM:

```

---
39:p                0.0010649      0.1297505      0.0026155      -0.0012348
40:p                0.4672519      1.0422928      0.2065664      -0.1892676
41:designbeta(1)    0.9514356      0.0931333      0.9461958      0.9310929
42:designbeta(2)    0.6114962      0.1139695      0.6117010      0.6072719
43:sigma(1)        0.1918787      0.1178243      0.1740883      0.1396514
44:sigma(2)        0.2999334      0.1094985      0.2840983      0.2723184
45:-2log Likelihood 16515.030      9.0251843      16514.489      16513.997
-2log Likelihood for means of beta estimates = 16495.978
DIC = 1931.5289

```

We quickly notice that the σ values are quite similar – this is not surprising, since the DM does not model σ .

For the estimates of μ , we’ll start with the estimates derived from the model fit using the ‘intercept

offset coding’ (Fig. E.8a). The designbeta(n) estimates correspond to the intercept (β_1) and slope (β_2) parameters in the following linear model:

$$\begin{aligned}\text{logit}(\hat{\phi}) &= \hat{\beta}_1 + \hat{\beta}_2(\text{water level}) \\ &= 0.6150444 + 0.3278746(\text{water level}).\end{aligned}$$

Thus, during an average water year,

$$\begin{aligned}\text{logit}(\hat{\phi}) &= 0.6150444 + 0.3278746(1) \\ &= 0.9429190,\end{aligned}$$

while in a low water year,

$$\begin{aligned}\text{logit}(\hat{\phi}) &= 0.6150444 + 0.3278746(0) \\ &= 0.6150444.\end{aligned}$$

Back-transforming from the logit scale to the probability scale for real parameters,

$$\begin{aligned}\hat{\mu}_{\text{avg}} &= \frac{e^{0.9429190}}{1 + e^{0.9429190}} & \hat{\mu}_{\text{low}} &= \frac{e^{0.6150444}}{1 + e^{0.6150444}} \\ &= 0.719689 & &= 0.649091\end{aligned}$$

which are close to the estimates from the ‘method of moments’ approach we derived earlier ($\hat{\mu}_{\text{avg}} = 0.717250$ and $\hat{\mu}_{\text{low}} = 0.645584$).

Now let’s look at the estimates using the DM shown in Fig. (E.8b). We see that $\hat{\beta}_1 = 0.951436$, while $\hat{\beta}_2 = 0.611496$. What are these values? Simple – they are the estimates for $\text{logit}(\hat{\phi}_{\text{avg}})$ and $\text{logit}(\hat{\phi}_{\text{low}})$, respectively! There is no linear model – they are estimates for each water level (average, low). [If you know why, good. If not, go back and study Chapter 6 again!] You’ll notice that these estimates are similar to the values calculated on the logit scale for each water level, using the linear model (above). For example, 0.719689 vs. 0.717250 for the average water level. They should, in theory, be identical. And, if we’d used ML estimation on the data, they would be. But here we are using MCMC, and the slight differences between the values are because MCMC involves random sampling (and no 2 Markov chains will yield identical results). Both values are close to those derived by manually specifying the mean of the hyperdistribution for a set of parameters (top of p. E-28). In fact, this approach, and the approach using the DM shown in Fig (E.8b) are in fact entirely equivalent (again, the estimates reported differ slightly, due to the random nature of MCMC estimation).

Finally, what about process variation? Recall that a primary motivation for using MCMC for this example was because the MCMC approach allows us to estimate process variance σ separately for each water level. Using the Delta approximation, and our estimates from p. E-28, we find the back-transform of $\hat{\sigma}_{\text{MCMC, avg}} = 0.2029072$ on the logit to the real scale yields $\hat{\sigma}_{\text{avg}} = 0.040819$, while the back-transformation of $\hat{\sigma}_{\text{MCMC, low}} = 0.3101054$ on the logit to the real scale yields $\hat{\sigma}_{\text{low}} = 0.070622$. While we can’t compare these values to estimates from the ‘method of moments’ approach (since that approach doesn’t allow us to estimate σ separately for average and low water levels), we recall (from earlier in this section) that the overall estimated process variance for both water levels together from the ‘method of moments’ analysis was $\hat{\sigma} = 0.0559457$, which is quite close to the average of the estimates of σ we just derived for each water level using MCMC $([0.04082 + 0.07062]/2 = 0.05572)$.

Some closing comments on example 3...

In this example, we clearly didn't need to use a design matrix (DM) to achieve our objective. Our intent was simply to demonstrate some of the mechanics, rather than suggesting this was a good approach for this particular problem. As you know from other chapters in this book, the DM is a powerful tool for modeling parameters. The same applies here, but keep in mind that the DM available in the MCMC part of **MARK** are more limited than the full-blown DM you've used for linear models for various data types (in particular, you are using the DM to model the means of parameters, and the number of means will always be less than the number of occasions).

Also, you may have noticed that we've progressed fairly far into this Appendix without making much reference to 'model selection'. While we did mention earlier that **MARK** generates the DIC and WAIC for a given model, we did not do much more than define it, and suggested that you're welcome to use it, but you're on your own if you do (there is a fair bit of literature out concerning use of the DIC or WAIC for model selection. This isn't because we've suddenly decided that model selection (and related things like model averaging) are any less important. On the contrary, these continue to be vitally important for the way we analyze data to address interesting questions. Unfortunately, model selection in the Bayesian context (where you generally find MCMC being used most frequently – pun completely intended...) is not as straightforward as evaluating models based on nominal AIC weights. A lot of smart folks are thinking pretty hard about this and related problems.

E.1.5. Example 4 – group + time as random effects

Suppose you've done a mark-recapture study with 5 groups (say, different levels of some experimental treatment), and 8 sampling occasions. You are interested in estimating the mean and process variance for some parameter, but unlike our earlier examples, we're not interested in estimating the hyperparameters over time only, but among groups as well. In other words, we want to estimate $\{\mu_{\text{time}}, \sigma_{\text{time}}\}$, and $\{\mu_{\text{group}}, \sigma_{\text{group}}\}$.

It is reasonable to wonder if it is possible to estimate both parameters at the same time (i.e., in the same model)? Unfortunately, the answer is, 'no', for the following reason. Remember that μ and σ are estimated based on the hyperdistribution of the β parameters (on the sin or logit scale), not the real parameters. So, at minimum, to estimate μ and σ for some factor, you need to create a model structure (using the DM), where you have separate parameters for different levels of that factor. You might recall from pp. 22-25, that to estimate μ and σ over a set of parameters, that this generally requires an *identity* DM structure, where each β represents a separate parameter (i.e., different level of a given factor).

In our present example, the two factors are (grp) and time (time). So, with 5 groups, and 8 occasions (7 intervals), you'd need 12 columns in the DM for a given parameter (say, φ): 5 for grp, and 7 for time.

However, with a bit of thought, you'll see there is a problem. We know from Chapter 6 that the linear model for $\varphi = \text{grp} + \text{time}$, with 5 groups, and 8 occasions (7 intervals), would have 11 β parameters, not 12 (1 for the intercept, $(5 - 1) = 4$ for grp, and $(7 - 1) = 6$ for time:

$$\begin{aligned} \text{logit}(\varphi) = & \beta_1 + \beta_2(\text{grp}_1) + \beta_3(\text{grp}_2) + \beta_4(\text{grp}_3) + \beta_5(\text{grp}_4) \\ & + \beta_6(\text{time}_1) + \beta_7(\text{time}_2) + \beta_8(\text{time}_3) + \beta_9(\text{time}_4) + \beta_{10}(\text{time}_5) + \beta_{11}(\text{time}_6). \end{aligned}$$

It is not possible to specify a DM with 'identity' structure for both grp and time in a single model, since such a model would have 12 parameters, 1 more than needed to fully specify the additive model.

As a result, we need to fit 2 separate models., to get estimates of μ and σ for each factor: (i) grp with identity structure, time with the standard offset, and (ii) grp with the standard offset, time with identity structure.

Here is a part of the DM for the (i) grp with identity structure, time with the standard offset model:

B1: g1	B2: g2	B3: g3	B4: g4	B5: g5	B6: t1	B7: t2	B8: t3	B9: t4	B10: t5	B11: t6	Parm	B12: p
1	0	0	0	0	1	0	0	0	0	0	1:Phi	0
1	0	0	0	0	0	1	0	0	0	0	2:Phi	0
1	0	0	0	0	0	0	1	0	0	0	3:Phi	0
1	0	0	0	0	0	0	0	1	0	0	4:Phi	0
1	0	0	0	0	0	0	0	0	1	0	5:Phi	0
1	0	0	0	0	0	0	0	0	0	1	6:Phi	0
1	0	0	0	0	0	0	0	0	0	0	7:Phi	0
0	1	0	0	0	1	0	0	0	0	0	8:Phi	0
0	1	0	0	0	0	1	0	0	0	0	9:Phi	0
0	1	0	0	0	0	0	1	0	0	0	10:Phi	0
0	1	0	0	0	0	0	0	1	0	0	11:Phi	0
0	1	0	0	0	0	0	0	0	1	0	12:Phi	0
0	1	0	0	0	0	0	0	0	0	1	13:Phi	0
0	1	0	0	0	0	0	0	0	0	0	14:Phi	0
0	0	1	0	0	1	0	0	0	0	0	15:Phi	0
0	0	1	0	0	0	1	0	0	0	0	16:Phi	0
0	0	1	0	0	0	0	1	0	0	0	17:Phi	0
0	0	1	0	0	0	0	0	1	0	0	18:Phi	0
0	0	1	0	0	0	0	0	0	1	0	19:Phi	0
0	0	1	0	0	0	0	0	0	0	1	20:Phi	0
0	0	1	0	0	0	0	0	0	0	0	21:Phi	0
0	0	0	1	0	1	0	0	0	0	0	22:Phi	0

and here is part of the DM for the (ii) grp with offset structure, time with identity structure model:

B1: g1	B2: g2	B3: g3	B4: g4	B5: t1	B6: t2	B7: t3	B8: t4	B9: t5	B10: t6	B11: t7	Parm	E
1	0	0	0	1	0	0	0	0	0	0	1:Phi	0
1	0	0	0	0	1	0	0	0	0	0	2:Phi	0
1	0	0	0	0	0	1	0	0	0	0	3:Phi	0
1	0	0	0	0	0	0	1	0	0	0	4:Phi	0
1	0	0	0	0	0	0	0	1	0	0	5:Phi	0
1	0	0	0	0	0	0	0	0	1	0	6:Phi	0
1	0	0	0	0	0	0	0	0	0	1	7:Phi	0
0	1	0	0	1	0	0	0	0	0	0	8:Phi	0
0	1	0	0	0	1	0	0	0	0	0	9:Phi	0
0	1	0	0	0	0	1	0	0	0	0	10:Phi	0
0	1	0	0	0	0	0	1	0	0	0	11:Phi	0
0	1	0	0	0	0	0	0	1	0	0	12:Phi	0
0	1	0	0	0	0	0	0	0	1	0	13:Phi	0
0	1	0	0	0	0	0	0	0	0	1	14:Phi	0
0	0	1	0	1	0	0	0	0	0	0	15:Phi	0
0	0	1	0	0	1	0	0	0	0	0	16:Phi	0
0	0	1	0	0	0	1	0	0	0	0	17:Phi	0
0	0	1	0	0	0	0	1	0	0	0	18:Phi	0
0	0	1	0	0	0	0	0	1	0	0	19:Phi	0
0	0	1	0	0	0	0	0	0	1	0	20:Phi	0
0	0	1	0	0	0	0	0	0	0	1	21:Phi	0
0	0	0	1	1	0	0	0	0	0	0	22:Phi	0

Have a look back at pp. 22-25, to make sure you understand how to construct these 2 DM.

We'll fit these models to some simulated live encounter data contained in `dm_grp.inp` – 5 groups, 8 sampling occasions..* These data we're simulated under a true generating model $\{\varphi_{g+t} p.\}$, with $\{\mu_{\text{time}} = 0.6785, \sigma_{\text{time}} = 0.0488\}$, and $\{\mu_{\text{group}} = 0.6785, \sigma_{\text{group}} = 0.0791\}$. Note that for a true generating model $\{\varphi_{g+t} p.\}$, with no trend over time, then $\mu_{\text{group}} \equiv \mu_{\text{time}}$.

We begin with (i) `grp` with identity structure, `time` with the standard offset. For the MCMC run, we specify a single hyperdistribution, using parameters $\beta_1 \rightarrow \beta_5$:

For this model, each of the parameters $\beta_1 \rightarrow \beta_5$ correspond to a different level of the `grp` factor, allowing us to estimate μ and σ over the set of parameters. Basing inference on the mean of the posterior, $\hat{\mu}_{\text{grp, logit}} = 0.8654531$, the back-transformed estimate is $\hat{\mu}_{\text{grp, real}} = 0.7038$, which isn't too far off from the true parameter value $\mu_{\text{grp}} = 0.6785$.

It is worth noting that the mean of the posterior is very similar to the reported median (0.8629936) and mode (0.8474103). Such is not the case for σ , where the posterior distribution is typically strongly right-skewed, such that the reported mean, median and mode are quite different (0.4585094, 0.3990607 and 0.3216484, respectively). If we compare the largest and smallest values, the back-transformed estimates of σ are $\hat{\sigma}_{\text{grp, real}} = 0.0956$ and $\hat{\sigma}_{\text{grp, real}} = 0.0671$, respectively. The true, canonical value for $\sigma_{\text{grp}} = 0.0791$ is approximately half-way between these two 'extreme' values.

We focus next on (ii) `grp` with offset structure, `time` with identity. For the MCMC run, we again specify a single hyperdistribution, but here we are using parameters $\beta_5 \rightarrow \beta_{11}$:

* We note that 5 groups, and 7 intervals, are insufficient to do a decent job of estimating μ and σ for either factor. As discussed in Appendix D, the general recommendation is ≥ 10 levels. Our purpose here is merely to demonstrate the mechanics.

For this model, each of the parameters $\beta_5 \rightarrow \beta_{11}$ correspond to a different level of the time factor, allowing us to estimate μ and σ over the set of parameters. Basing inference on the mean of the posterior, $\hat{\mu}_{\text{time, logit}} = 0.9963532$, the back-transformed estimate is $\hat{\mu}_{\text{time, real}} = 0.7303$, which again isn't too far off from the true parameter value $\mu_{\text{grp}} = 0.6785$.

As above, the posterior for σ is strongly right-skewed, so we resort to comparing the largest (mean) and smallest (mode) moments calculated from the posterior distribution. The back-transformed estimates of σ are $\hat{\sigma}_{\text{time, real}} = 0.0532$ and $\hat{\sigma}_{\text{time, real}} = 0.0430$, respectively. As was the case above, the true, canonical value for $\sigma_{\text{time}} = 0.0488$ is approximately half-way between these two 'extreme' values.

E.2. Hyperdistributions between structural parameters

In the second motivating example presented at the start of this appendix ('scenario 2'; p. E-3), we considered an example based on analysis of dead recovery data, where we had interest in the correlation between the two structural parameters S (survival) and f (recovery rate). This interest was motivated by some *a priori* expectation that predicted that the sign of any correlation between S and f might indicate support for one of two competing models relating harvest to mortality. As noted, the difficulty in assessing the correlation between these two parameters is that there is covariance both within (over sampling occasions) and between the two parameters. Thus, the parameters are not independent samples, and we can't simply take estimates of \hat{S}_i and \hat{f}_i and estimate the correlation. This would amount to little more than 'doing statistics on statistics', which is almost always a poor approach.

Here we consider 2 worked examples – one based on a dead recovery analysis, and the other on a time-symmetric model approach. Our intent is to focus on the mechanics, rather than provide an exhaustive list of models where this approach could be implemented. But, it may suffice to say that the approach we'll describe here could be used to model covariance among structural parameters in any model where there might be interest. Other than greater flexibility for handling estimation of process variation (which has been our focus up until this point), the ability to consider multivariate hyperdistributions is one of the main motivations for use of MCMC in **MARK**. Not only are the types of questions which can be addressed using this approach numerous and varied, in most cases, using MCMC is the only viable means of considering them.

E.2.1. Example 1 - dead recovery analysis ($\text{corr}\{S, f\}$)

For this example we consider the estimation of the correlation of 2 structural parameters in a Brownie dead recovery model (survival, S , and recovery rate, f). To demonstrate the mechanics of using MCMC in **MARK** to estimate the correlation, we simulated a data set with 11 occasions (thus, 10 estimates of survival, and 11 estimates of recovery rate). We drew our set of underlying survival and recovery probabilities used to simulate the data from a random bivariate normal distribution with $S_i = \mathcal{N}(0.8, 0.025)$, and $f_i = \mathcal{N}(0.1, 0.015)$, with $\rho(S, f) = -0.65$. In other words, we drew a random sample of survival and recovery probabilities from a distribution where the 2 parameters were negatively correlated. Recall (from Chapter 8) that there is one more recovery parameter ($k = 11$) than survival parameter ($k = 10$). Thus, we drew 10 pairs of parameter values for $S_{1 \rightarrow 10}$ and $f_{1 \rightarrow 10}$. For f_{11} , we used $f_{11} = 0.1$, which was the true mean of the distribution from which the f_i were drawn.

The sample of probabilities (i.e., parameter values) we ended up with for each parameter is shown at the top of the next page. The correlation of these true parameters used in our simulation $\rho(S_{1 \rightarrow 10}, f_{1 \rightarrow 10}) = -0.76$ (remember – we took a small sample of parameter values from the underlying bivariate normal distribution with overall $\rho = -0.65$. The value -0.76 is the estimate of ρ from this sample). It is this correlation between $\rho(S_{1 \rightarrow 10}, f_{1 \rightarrow 10})$ that we will try to estimate using MCMC.

survival (S)	recovery (f)
0.804	0.118
0.802	0.099
0.807	0.097
0.790	0.109
0.846	0.085
0.787	0.114
0.818	0.092
0.798	0.107
0.797	0.097
0.793	0.123
–	0.100

For each sampling occasion, we simulated marking and release of 10,000 individuals. We ran a single simulation, and used the encounter histories generated by that simulation for our analysis. These simulated encounter data (in LDLD format) are contained in **brownie-corr.inp**.

Start **MARK**, select '**Dead recoveries | Brownie**' as the data type, set the number of occasions to 11. Click '**OK**', and then go ahead and fit the full time-dependent default model, $\{S_t, f_t\}$, using the logit link. Add the results to the browser.

Now, we're going to re-run this model, using MCMC. Run the model, but this time, check the '**MCMC estimation**' check-box. It is also a good habit to use initial parameter estimates to start the sampler, so check the '**Provide initial parameter estimates**' box as well. Then click '**OK**'.

Next, we need to set various parameters to govern the MCMC sampler. We'll use the default 4,000 tuning and 1,000 burn-in samples, but we'll increase the number of samples to store from 10,000 to 50,000 (you'll see why in a moment).

Next, we need to specify 2 hyperdistributions (one for S , and one for f). And, now, the new step – we need to check the box to allow us to specify the '**Variance-covariance matrix**', as shown in the following:

Hyperdistribution Specification

Number of hyperdistributions: 2

☐ Hyperdistribution means modeled with a design matrix of 2 columns

☒ Variance-covariance matrix specified

After you click '**OK**', we're asked to specify the '**Beta parameters to include in the hyperdistributions**'. Our interest is in the correlation between $S_{1 \rightarrow 10}$ and $f_{1 \rightarrow 10}$, so we enter '1 to 10, 11 to 20'.

In the next step, **MARK** defaults to precisely what we want – the mean ($\mu(1)$ and $\mu(2)$) and variance ($\sigma(1)$ and $\sigma(2)$) are correctly associated with parameters $1 \rightarrow 10$ and $11 \rightarrow 20$, respectively. Remember, there is no S_{11} corresponding to f_{11} , so we use only 10 for each parameter.

Specify Mean of the Hyperdistribution for each Parameter

Specify Parameter-Spec

1:S	<input type="text" value="mu(1)"/>	11:f	<input type="text" value="mu(2)"/>
2:S	<input type="text" value="mu(1)"/>	12:f	<input type="text" value="mu(2)"/>
3:S	<input type="text" value="mu(1)"/>	13:f	<input type="text" value="mu(2)"/>
4:S	<input type="text" value="mu(1)"/>	14:f	<input type="text" value="mu(2)"/>
5:S	<input type="text" value="mu(1)"/>	15:f	<input type="text" value="mu(2)"/>
6:S	<input type="text" value="mu(1)"/>	16:f	<input type="text" value="mu(2)"/>
7:S	<input type="text" value="mu(1)"/>	17:f	<input type="text" value="mu(2)"/>
8:S	<input type="text" value="mu(1)"/>	18:f	<input type="text" value="mu(2)"/>
9:S	<input type="text" value="mu(1)"/>	19:f	<input type="text" value="mu(2)"/>
10:S	<input type="text" value="mu(1)"/>	20:f	<input type="text" value="mu(2)"/>

Specify Sigma of the Hyperdistribution for each Parameter

Specify Parameter-Spec

1:S	<input type="text" value="sigma(1)"/>	11:f	<input type="text" value="sigma(2)"/>
2:S	<input type="text" value="sigma(1)"/>	12:f	<input type="text" value="sigma(2)"/>
3:S	<input type="text" value="sigma(1)"/>	13:f	<input type="text" value="sigma(2)"/>
4:S	<input type="text" value="sigma(1)"/>	14:f	<input type="text" value="sigma(2)"/>
5:S	<input type="text" value="sigma(1)"/>	15:f	<input type="text" value="sigma(2)"/>
6:S	<input type="text" value="sigma(1)"/>	16:f	<input type="text" value="sigma(2)"/>
7:S	<input type="text" value="sigma(1)"/>	17:f	<input type="text" value="sigma(2)"/>
8:S	<input type="text" value="sigma(1)"/>	18:f	<input type="text" value="sigma(2)"/>
9:S	<input type="text" value="sigma(1)"/>	19:f	<input type="text" value="sigma(2)"/>
10:S	<input type="text" value="sigma(1)"/>	20:f	<input type="text" value="sigma(2)"/>

Finally, we're at the point where we want to specify the variance-covariance structure for the 2 hyperdistributions.

HyperDistributions Variance-Covariance Matrix																				
Design Matrix Specification (B = Beta)																				
Param	B1:	B2:	B3:	B4:	B5:	B6:	B7:	B8:	B9:	B10:	B11:	B12:	B13:	B14:	B15:	B16:	B17:	B18:	B19:	B20:
1:S	sigma(1)	0	0	0	0	0	0	0	0	0	rho(1)	0	0	0	0	0	0	0	0	0
2:S		sigma(1)	0	0	0	0	0	0	0	0	0	rho(1)	0	0	0	0	0	0	0	0
3:S			sigma(1)	0	0	0	0	0	0	0	0	0	rho(1)	0	0	0	0	0	0	0
4:S				sigma(1)	0	0	0	0	0	0	0	0	0	rho(1)	0	0	0	0	0	0
5:S					sigma(1)	0	0	0	0	0	0	0	0	0	rho(1)	0	0	0	0	0
6:S						sigma(1)	0	0	0	0	0	0	0	0	0	rho(1)	0	0	0	0
7:S							sigma(1)	0	0	0	0	0	0	0	0	0	rho(1)	0	0	0
8:S								sigma(1)	0	0	0	0	0	0	0	0	0	rho(1)	0	0
9:S									sigma(1)	0	0	0	0	0	0	0	0	0	rho(1)	0
10:S										sigma(1)	0	0	0	0	0	0	0	0	0	rho(1)
11:f											sigma(2)	0	0	0	0	0	0	0	0	0
12:f												sigma(2)	0	0	0	0	0	0	0	0
13:f													sigma(2)	0	0	0	0	0	0	0
14:f														sigma(2)	0	0	0	0	0	0
15:f															sigma(2)	0	0	0	0	0
16:f																sigma(2)	0	0	0	0
17:f																	sigma(2)	0	0	0
18:f																		sigma(2)	0	0
19:f																			sigma(2)	0
20:f																				sigma(2)

What you see pictured above is the variance-covariance matrix, for the 2 hyperdistributions. Along the left-hand side, you'll see a column listing the parameters $S_1 \rightarrow S_{10}$, and $f_1 \rightarrow f_{10}$. There is no horizontal reference bar analogous to the vertical reference bar in the linear models DM separating different types of parameters. Along the diagonal in blue (i.e., immediately above the 'lower-triangle' in black) we see 20 'blue boxes' with white lettering (actually, unless you've changed the default DM colors in **MARK**, what you will see on your computer is red boxes with white lettering – for purposes of increasing contrast here in the book, we've changed from a red color scheme to blue). Starting from the upper-left corner, and moving down the diagonal, the first 10 cells along the diagonal are labeled 'sigma(1)', while the next 10 are labeled 'sigma(2)'. Remember, in a variance-covariance matrix, the variances of the parameters are on the diagonal. That is exactly what we see here. But, because we have 'collected' parameters into hyperdistributions, we see 'sigma(1)' for parameters $1 \rightarrow 10$, rather than 'sigma(1)', 'sigma(2)', ..., 'sigma(10)'.

Now, you should also recall that off the diagonal in a VC matrix are the covariances between particular parameters. These you will need to manually enter. If you leave the above diagonal cells at the default of '0', you would end up simply estimate process variance, 'sigma(1)' and 'sigma(2)'. However, recall what we are trying to do here – we want to estimate the correlation, ρ , between $S_{1 \rightarrow 10}$ and $f_{1 \rightarrow 10}$, which correspond to parameters $1 \rightarrow 10$ and $11 \rightarrow 20$, respectively. So, S_1 is paired with f_1 , S_2 is paired with f_2 , and so on. Meaning, we need to find the point above the diagonal where parameter S_1 (row 1), pairs up with parameter f_1 (row 11 – note that the row numbers match the parameter indexing). So in matrix element [1, 11], we manually type in – carefully – the expression 'rho(1)'. We do the same thing for parameter S_2 (row 2), and corresponding recovery parameter f_2 (row 12) – in matrix element [2, 12] we again enter the expression 'rho(1)'. We do not enter 'rho(2)' (i.e., we don't increment the indexing for 'rho(n)'), because we are estimating one correlation coefficient (i.e., 'rho(1)') between both sets (hyperdistributions) of parameters.

So, off the diagonal, a diagonal vector of 10 elements where you've entered the expression 'rho(1)'. You can do this manually, or it may be more efficient to use the design matrix command '**Copy Value Diagonal**', especially for large matrices where many values of the rho parameter must be entered. You also have the option to paste a VC matrix into the window, but you may want to specify the entire matrix (both above and below the diagonal) and paste it into the window, because trying to construct the values to paste with different numbers of elements per row can be painful. For the values below the diagonal, specify zeros, as these will be ignored.

Note that only zeros or 'rho' values are allowable values in the upper off-diagonal portion of the matrix, and only sigma(n)'s are valid on the diagonal. So, technically, this matrix is a correlation matrix *above* the diagonal, and a standard deviation vector *on* the diagonal.

[begin sidebar](#)

more on the MCMC VC matrix...

As mentioned, the default VC matrix is all zero values, which is the same as if no variance-covariance matrix is specified. On the diagonal of the matrix are the sigma values (which you should not need to change if they were correctly specified in the specification of the hyperdistribution). You only have to specify the off-diagonal elements above the diagonal, as the matrix is symmetric and the lower off-diagonal elements should be blacked out.

In the present example, we've specified a VC matrix to estimate the correlation, rho(1), between 2 hyperdistributions. Another potentially useful matrix to estimate is the *autocorrelation* of the beta parameters, as shown below (we show the first 5 beta parameters only):

sigma(1)	rho(1)	rho(1)**2	rho(1)**3	rho(1)**4
	sigma(1)	rho(1)	rho(1)**2	rho(1)**3
		sigma(1)	rho(1)	rho(1)**2
			sigma(1)	rho(1)
				sigma(1)

This VC matrix models the correlation between *consecutive* parameters. If β_1 and β_2 are correlated as rho(1), and β_2 and β_3 are also correlated as 'rho(1)', then β_1 and β_3 have to be correlated with 'rho(1)**2'. This autocorrelation matrix can be obtained easily by right-clicking the VC matrix window and selecting the '**Autocorrelation matrix**' option for the list of options. Likewise, you can get back to the default matrix of zero off-diagonal elements with the '**No correlation**' menu choice.

[end sidebar](#)

Back to our example - Once you've completed filling out the VC matrix, click the '**OK**' button.

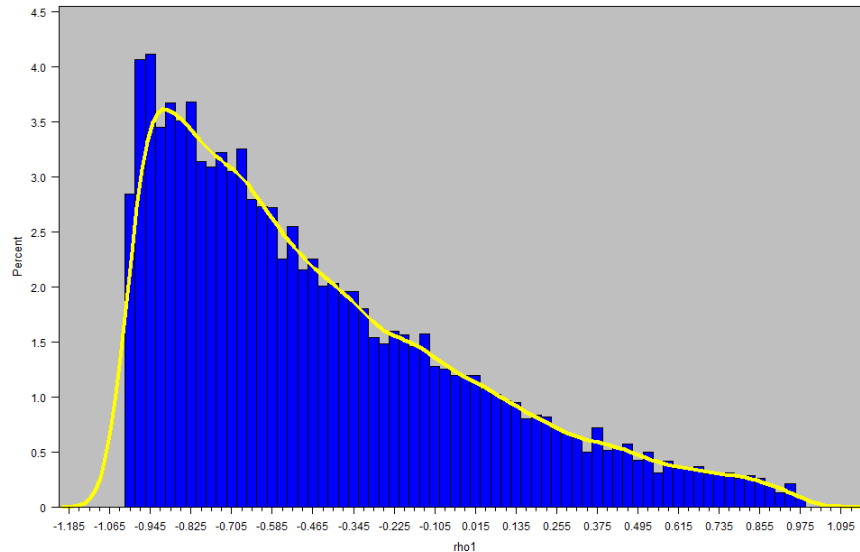
We'll accept the defaults for the priors – but will note that the prior for the correlation ρ is $\mathcal{U}(-1, 1)$. More on this in a moment.

Once you've pulled in the initial parameter estimates from the appropriate model in the browser, go ahead and run the MCMC sampler. Here are the results we generated, based on 50,000 samples:

20:t	-1.9664312	0.0179214	-1.9661824	-1.9641209
21:f	-2.2156381	0.0211403	-2.2158994	-2.2173419
22:mu(1)	1.4241278	0.0263609	1.4247749	1.4275046
23:mu(2)	-2.1617753	0.0506300	-2.1614787	-2.1560266
24:sigma(1)	0.0631657	0.0329045	0.0560224	0.0455499
25:sigma(2)	0.1551130	0.0425683	0.1471698	0.1334634
26:rho(1)	-0.4433035	0.4553181	-0.5606491	-0.9999635
27:-2log Likelihood	268155.78	5.3284515	268155.19	268154.08
-2log Likelihood for means of beta estimates = 268141.63				
DIC =	78.075570			

Note that the estimate of the correlation – based on correlation on the logit scale – is $\hat{\rho} = -0.443304$. We need to pause and make several points here.

1. Earlier, we noted that we were going to increase the number of samples from the default of 10,000 to 50,000. Why? Well, to answer, let's have a look at the frequency histogram (below) based on the distribution of 50,000 samples from the posterior for ρ .



Clearly, this is not a nicely symmetrical distribution – as such, the most appropriate ‘moment’ of this distribution to use for inference about a point estimate for ρ is open to some debate. From the results shown above, we see estimates of -0.443304 , -0.56065 and -0.99996 for the mean, median, and mode, respectively. Clearly, the reported mode isn't plausible.

But, what about the mean and median values? Which one should we choose? In this instance, the point may be moot since the 95% frequency-based credibility interval for $\hat{\rho}$ is $[-0.9917, 0.6739785]$, which clearly bounds 1.0. So does the 95% HPD, $[-1.000, 0.2729]$. This is not particularly encouraging, given large samples of marked and released individuals in the simulation, and a relatively strong true negative correlation between S and f simulated in the data. Some solace, perhaps, given that the estimate is at least in the right direction (negative), although we know this only because we know the underlying structure of the true model we used to generate the data. Keeping all of this in mind, consider that the situation would have been even more uncertain if we'd used the default of 10,000 samples, instead of 50,000 (try it for yourself).

2. How would the situation change if we used *a priori* expectation, and changed the prior on ρ from $\mathcal{U}(-1, 1)$ to $\mathcal{U}(-1, 0)$, or perhaps a different, informative prior (say, $\mathcal{B}(5, 2)$)? You might recall that when we ran our analysis, the last step involved specifying the priors for the univariate and any multivariate hyperparameters.

mu(1)	0.4	Compute	0.0	100.0		
mu(2)	0.4	Compute	0.0	100.0		
sigma(1)	0.4	Compute	0.001	0.001		
sigma(2)	0.4	Compute	0.001	0.001		
rho(1)	0.4	Compute	-1.0	1.0	1.0	1.0

For the univariate hyperparameters μ_i and σ_i , four inputs are requested. The first edit box is to specify the step size to be used to generate new parameter values with which to sample the posterior distribution. As with the default step size, the goal is to tune the estimation so that approximately 45% of the steps are accepted. The second edit box is to specify an initial value to start the Markov Chain. The default is 'compute', which tells **MARK** to compute an estimate from the initial values of the beta parameters. It is generally recommended practice that you should run the model that you want to use for MCMC estimation as a typical **MARK** analysis, so that you can provide initial estimates to start the MCMC estimation.

The third and fourth edit boxes specify the parameters for the *prior* distribution to be used with this hyperdistribution. For mean parameters, a normally distributed prior is used. The third edit box specifies the mean, and the fourth edit box specifies the standard deviation. The default values for μ are mean = 0 and standard deviation = 100, giving a very flat and uninformative prior over the range of the possible values of the mean. For σ parameters, an inverse gamma distribution prior is used with parameters alpha and beta. The mean of a gamma distribution is α times β , and the variance is α times β^2 . Values of $\alpha = 0.001$ and $\beta = 0.001$ result in a reasonably flat prior distribution. In other words, the defaults specify uninformative 'flat' priors for μ and σ .

For the multivariate hyperparameter, ρ , the default prior is a beta distribution over the range specified by the lower bound in the third edit box to the upper bound in the fourth edit box (see figure at bottom of preceding page). The default values for the range of ρ are $[-1, 1]$, but to force the correlation to be positive, you might use $[0, 1]$, or forced to be negative, $[-1, 0]$.

In addition, the fifth and sixth boxes for ρ (above) specify the α and β parameters of the beta distribution (see -sidebar- starting on p. E-15). The default is $\alpha = \beta = 1.0$, which gives a uniform distribution on the range specified (see Fig. E.4 on p. E-16). The shape of the beta prior on ρ over the interval $[-1, 1]$ can be changed by modifying the values of α and β (Fig. E.10).

For our present example, let's see what happens if we use a beta prior for ρ which gives greater weight to negative values (using, say, $\mathcal{B}(2, 5)$). Doing so changes the estimates of the mean and median on the logit scale for ρ to -0.5389 and -0.5863 , respectively. Both are negative. But, what about the credibility intervals? The frequency percentile-based 95% interval is $[-0.9298, 0.1105]$, while the HPD is $[-0.1275, 0.0046]$. We might be somewhat

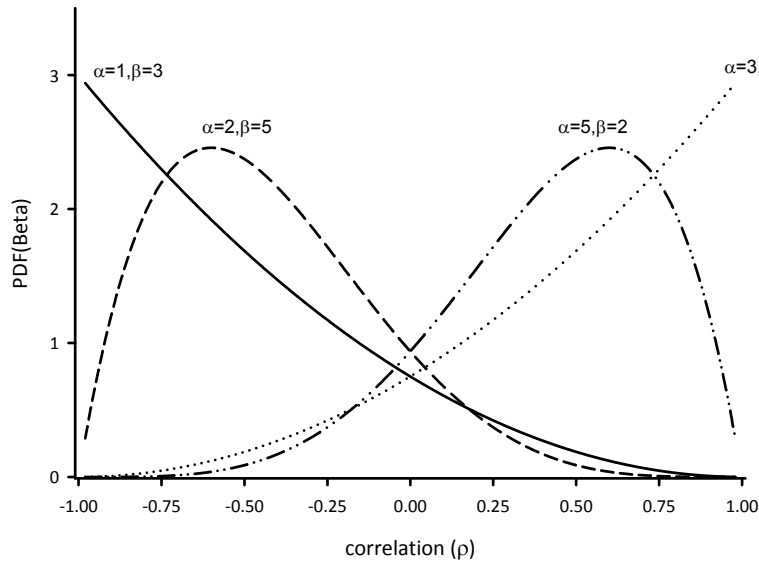


Figure E.10: Shape of the beta probability distribution for unequal values of the shape parameters α and β . The distribution is symmetric around $\rho = 0$ over the interval $[-1, 1]$ for $\alpha = \beta$.

comforted by the fact that either credibility bound is ‘more supportive’ of a negative correlation. Of course, the burden would now fall on us to justify the use of an informative prior, but that is part of the ‘fun’ of ‘being Bayesian’ (according to some).

3. However, while we might be somewhat ‘satisfied’ or ‘content’ with our results, at least those based on the use of the informative beta prior $\mathcal{B}(2, 5)$ (see point 2, above), we must think – hard – about whether or not this estimate of ρ is robust, and whether or not it represents what we think (hope?) it does – the covariation of survival S and recovery rate f . In fact, there might be a real problem here – since $f = Kc\lambda$ (discussed earlier in the appendix), then in fact there is an implicit relationship between survival S and the probability of mortality due to harvest, which is clearly a function of f . In fact, under some assumptions, there is an intrinsic ‘part-whole’ correlation between S and f such that the null hypothesis for our analysis might not be $\rho = 0$, but rather, the null expectation for ρ might be some other non-zero value. As such, a fair and robust interpretation of $\hat{\rho}$ would depend on knowing what this intrinsic correlation might be (for a brief discussion of one approach to estimating the covariance function for S and f , see last example in Appendix B).
4. Finally, the estimate $\hat{\rho}$ is reported on the logit scale. OK – you might think, no problem. Back-transform the estimate from the logit scale to the real probability scale. But, take a moment to think about it. The correlation ρ is estimated on the interval $[-1, 1]$. A correlation of $\rho = 0.0$ (i.e., no correlation) would back-transform to $\exp(0)/(1 + \exp(0)) = 0.5$. If the correlation was perfectly negative, $\rho = -1.0$, then the naive back-transformed value would be $\exp(-1)/(1 + \exp(-1)) = 0.269$. In fact, a parameter bounded on the interval $[-1, 1]$ on the logit scale would back-transform to a parameter bounded on the interval $[0.269, 0.731]$ on the real probability scale, not $[-1, 1]$ or even $[0, 1]$. In fact, the latter would require ρ being estimated on the logit scale on the interval $[-\infty, +\infty]$, while the former is not mathematically possible, since the back-transform from the logit cannot generate a transformed value < 0 .

Fortunately, it doesn't matter here. Imagine you have some estimates of 2 parameters on the real probability scale, say S and f . Suppose they have a true correlation on the real scale of ρ_{real} . It turns out that if you take the data (i.e., the parameter estimates on the real scale), and logit transform them, and then calculate the bivariate correlation on the logit transformed data, you'll find that $\rho_{\text{logit}} \simeq \rho_{\text{real}}$. You can easily prove this for yourself – simulate a number of sets of parameters from a bivariate normal with known correlation, transform the data from the real scale to the logit scale, calculate the correlation of the logit transformed data, and compare with the correlation on the real scale. Given a large enough number of simulations, you'll see that indeed $\rho_{\text{logit}} \simeq \rho_{\text{real}}$.

Why? Simple – because you're estimating a parameter (ρ) describing the linear covariance between two parameters, and the transformation from real to logit is itself effectively linear, over the typical range of data. It is a 'basic result' that the bivariate correlation of X_1 and X_2 is invariant to a *linear* transformation. It is also *largely* invariant to nonlinear transformation, provided the transformation is *monotonic* (in other words, the sin link would not work particularly well). Again, this is easy enough to demonstrate for yourself. In other words, saying that ' $\hat{\rho} = xyz$ on the logit scale' is effectively the same as saying that ' $\hat{\rho} = xyz$ on the familiar $[-1, 1]$ scale'.

An additional way you can confirm this for yourself is to re-run the MCMC analysis, using the identity link. Normally we don't recommend using the identity link function, but it tends to be fairly stable for Brownie dead recovery models. The practical advantage of using the identity link is there is no extra 'thinking' involved in considering whether a parameter is estimated on this scale or that, or how to transform to the real probability scale. Based on 50,000 samples, and using the identity link, the mean and median for the estimated posterior for $\hat{\rho}$ were -0.60791 and -0.80254 , respectively. Since the logit transform is essentially linear here, we would not anticipate that using an identity link would change the estimates much (they don't – both the mean and median are relatively close to what we saw above using the logit link).

5. So, we conclude that our best estimate of the correlation between S and f for our simulated data is probably $\hat{\rho} \approx (-0.5 \leftrightarrow -0.65)$. But, the credibility interval nearly bounds 0 (using an informative beta prior on ρ), so we might not be all that excited by this result. In fact, we should probably be suspicious of trying to 'tell a story' based on only 10 pairs of parameters in the first place. We re-visit the issue of the length of the time series used in specifying the hyperdistributions later.

E.2.2. Example 2 – time-symmetric model

A final example application considers correlation between the process parameters φ (apparent survival) and f (recruitment), in a temporal symmetry model (Chapter 13). We will re-visit the moth (*Gonodontis bidentata*) data analysis introduced in Appendix D. The data (contained in `moth-example.inp`) consist of records for 689 male moths that were captured, marked, and released daily over 17 days in northwest England. These moths were nonmelanic; demographic parameters were estimated as part of a larger study looking at comparative fitness of distinct color morphs.

In Appendix D, we used 'method of moments' random effect models, focussing on estimation of process variance, and possible trend, in realized growth rate λ . Our focus here is on estimating correlation between the process parameters which jointly determine realized population change (since $\lambda = \varphi + f$). We might be interested if they covary positively (such that high survival years are also characterized by high recruitment – the 'a good year for one demographic parameter is also a good year for another' hypothesis), or negatively (such that increased recruitment, say, correlates with reduced

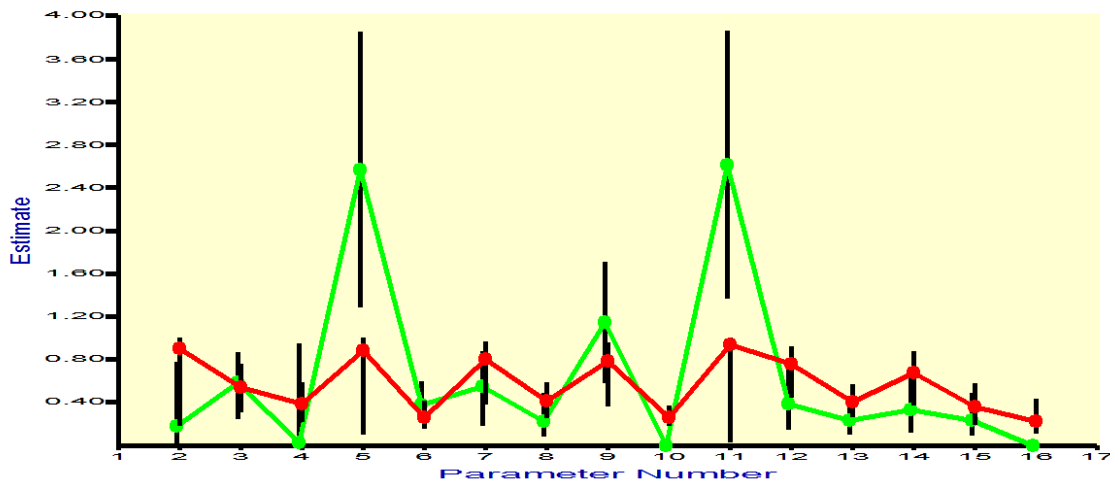
survival. Potentially evidence for the ubiquitous ‘density-dependence’ hypothesis which has featured prominently in population ecology for a long time).

Go ahead and start **MARK**, and import the moth data. Select the ‘**Pradel survival and recruitment**’ data type with 17 sampling occasions. For purposes of convenience, let’s set the starting general model to $\{\varphi_i p, f_i\}$. By fixing encounter probability p to be constant over time, we’re eliminating some of the potential confounding between φ and f in a fully time-dependent model. The key word here is ‘some’. Even with a time-constant p , experience has shown that the first estimates of λ (and as such, φ and f) are likely biased. So, we’ll exclude φ_1 and f_1 , and φ_{16} and f_{16} from our hyperdistributions, and base our MCMC estimation of correlation between φ and f on the remaining 14 estimates.

We’ll fit our general model using ‘**parm-specific**’ link functions. For φ and p , both of which are bounded $[0, 1]$, we’ll use the logit link. For f (recruitment), which cannot be negative (lower bound of 0), but which can be (and frequently is) ≥ 1 , we’ll use the log link. We’ll also use simulated annealing for the optimization (preliminary analysis shows that the estimation of some parameters for these data is somewhat sensitive to starting values used in the optimization).

[Note: using ‘**parm-specific**’ link functions, or the log link for all parameters, yields the same model deviance. However, the parameter count differs depending on which link function approach is used.]

Here is a plot of $\hat{\varphi}_i$ and \hat{f}_i , for occasions $i = 2$ to 17, where the red line represents estimates for apparent survival $\hat{\varphi}_i$, and the green line represents estimates for per capita recruitment, \hat{f}_i :



There would seem to be some evidence – based on the completely non-rigorous criterion of ‘visual assessment’ of positive covariation between the two parameters. We of course prefer a more rigorous approach, and will proceed with formal MCMC estimation of the correlation between φ_i and f_i .

Proceed as in the previous example – let’s set the number of samples to 50,000. For specifying the hyperdistributions, we want to use parameters 2 to 16 for one (φ_i), and 19 to 33 for the other (f_i). Remember, we are dropping the first φ and f estimates from the hyperdistributions.

Once you have set up `mu(1)` and `mu(2)`, and `sigma(1)` and `sigma(2)`, you’ll be presented with the template for the variance-covariance matrix. Here, because we have 30 total parameters, the VC matrix is getting a bit dense (shown at the top of the next page) – almost rivaling some of the DM seen for the more complicated robust design models introduced in Chapter 16.

Design Matrix Specification (B = Beta)																														
Param	B1:	B2:	B3:	B4:	B5:	B6:	B7:	B8:	B9:	B10:	B11:	B12:	B13:	B14:	B15:	B16:	B17:	B18:	B19:	B20:	B21:	B22:	B23:	B24:	B25:	B26:	B27:	B28:	B29:	B30:
2.Ph	sigma(0	0	0	0	0	0	0	0	0	0	0	0	0	0	rho(1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.Ph		sigma(0	0	0	0	0	0	0	0	0	0	0	0	0	0	rho(1)	0	0	0	0	0	0	0	0	0	0	0	0	0
4.Ph			sigma(0	0	0	0	0	0	0	0	0	0	0	0	0	0	rho(1)	0	0	0	0	0	0	0	0	0	0	0	0
5.Ph				sigma(0	0	0	0	0	0	0	0	0	0	0	0	0	0	rho(1)	0	0	0	0	0	0	0	0	0	0	0
6.Ph					sigma(0	0	0	0	0	0	0	0	0	0	0	0	0	0	rho(1)	0	0	0	0	0	0	0	0	0	0
7.Ph						sigma(0	0	0	0	0	0	0	0	0	0	0	0	0	0	rho(1)	0	0	0	0	0	0	0	0	0
8.Ph							sigma(0	0	0	0	0	0	0	0	0	0	0	0	0	0	rho(1)	0	0	0	0	0	0	0	0
9.Ph								sigma(0	0	0	0	0	0	0	0	0	0	0	0	0	0	rho(1)	0	0	0	0	0	0	0
10.Ph									sigma(0	0	0	0	0	0	0	0	0	0	0	0	0	0	rho(1)	0	0	0	0	0	0
11.Ph										sigma(0	0	0	0	0	0	0	0	0	0	0	0	0	0	rho(1)	0	0	0	0	0
12.Ph											sigma(0	0	0	0	0	0	0	0	0	0	0	0	0	0	rho(1)	0	0	0	0
13.Ph												sigma(0	0	0	0	0	0	0	0	0	0	0	0	0	0	rho(1)	0	0	0
14.Ph													sigma(0	0	0	0	0	0	0	0	0	0	0	0	0	0	rho(1)	0	0
15.Ph														sigma(0	0	0	0	0	0	0	0	0	0	0	0	0	0	rho(1)	0
16.Ph															sigma(0	0	0	0	0	0	0	0	0	0	0	0	0	0	rho(1)
19.f																sigma(0	0	0	0	0	0	0	0	0	0	0	0	0	0
20.f																	sigma(0	0	0	0	0	0	0	0	0	0	0	0	0
21.f																		sigma(0	0	0	0	0	0	0	0	0	0	0	0
22.f																			sigma(0	0	0	0	0	0	0	0	0	0	0
23.f																				sigma(0	0	0	0	0	0	0	0	0	0
24.f																					sigma(0	0	0	0	0	0	0	0	0
25.f																						sigma(0	0	0	0	0	0	0	0
26.f																							sigma(0	0	0	0	0	0	0
27.f																								sigma(0	0	0	0	0	0
28.f																									sigma(0	0	0	0	0
29.f																										sigma(0	0	0	0
30.f																											sigma(0	0	0
31.f																												sigma(0	0
32.f																													sigma(0
33.f																														sigma(

However, if you understood what we did in the preceding example, then this one should not be much more difficult, despite the scale of the VC matrix. You simply need to remember to ‘match’ up pairs of φ and f parameters. In this case, φ_2 with f_2 , φ_3 with f_3 , and so forth. The parameter φ_2 is in row 1, while parameter f_2 is in row 16. This, we click in the cell in row 1, column 16, and enter ‘rho(1)’. We then simply copy this down the diagonal (using the right-click menu design menu option), as shown. With some practice, this tends to be relatively easy to accomplish.

Proceed through the next steps, retrieving estimates from the appropriate model in the browser as initial values, and then start the sampler. Here are the results from our analysis, based on 50,000 samples.

```

32:f -1.8793636 0.539814 -1.8139282 -1.653294 /
33:f -3.6299603 0.9897197 -3.4812543 -3.2572262
34:mu(1) 0.3479257 0.3453461 0.3160946 0.2813022
35:mu(2) -1.2506498 0.4283580 -1.2128292 -1.1596338
36:sigma(1) 1.0592197 0.3652815 0.9969333 0.8981278
37:sigma(2) 1.4064406 0.4035213 1.3377597 1.2411260
38:rho(1) 0.7905274 0.1768200 0.8368161 0.9482054
39:-2log Likelihood 5077.2822 7.7659076 5076.7461 5075.4617
-2log Likelihood for means of beta estimates = 5054.9900
DIC = 293.66354

```

We see that the estimate of the correlation, based on the mean of the posterior, is $\hat{\rho} = 0.7905$ (the estimated median was only marginally higher). Of note, here, is that the 95% credibility interval [0.332, 0.991] doesn’t bound 0, which suggests that our visual intuition concerning a positive covariance between survival and recruitment for these moth data might be correct (or at least somewhat more defensible) after all.

However, we need to be careful here. Earlier, we noted that the back-transformation of ρ from the logit scale to the real probability scale was generally an identity transformation (i.e., that the value of ρ on one scale was the same as the value on the other). However, we also noted that this was strictly true only if (i) the transformation was linear, or nearly so, (ii) monotonic, and (importantly), (iii) was the same transformation applied to both parameters included in the multivariate hyperdistribution. Clearly, this

is not will not always be the case.

What about for the present example, where we've applied a logit transform to φ , and a log transform to f ? While both transformations are monotonic, the log transform is clearly non-linear. Can we take our estimate of $\hat{\rho} = 0.7905$ and interpret it 'as is'? As you might expect, the answer is 'no'. A simple simulation will demonstrate the problem. If we take our MCMC results for μ and σ (above), and simulate 1,000 random samples from a bivariate normal logit-log distribution (φ on logit scale, f on log scale), with $\hat{\rho} = 0.7905$ (from above), our scatterplot looks something like Fig. (E.11a) – typical of bivariate normal data. The estimated correlation of our simulated data ($\hat{\rho} = 0.796$) is close to the value of 0.7905 estimated from the MCMC analysis.

However, when we back-transform the bivariate data from the logit scale and the log scale, respectively, to the real scale, the distribution is clearly no longer bivariate normal on the real scale (Fig. E.11b), to the point where even reporting the estimated bivariate correlation from the back-transformed data ($\hat{\rho} = 0.618$) verges on silly. Clearly, considerable care must be taken in interpreting the MCMC estimate of ρ , probably generally, but particularly if the two parameters are subjected to different transformations, especially if one or both transformations is highly non-linear over the range of the data.

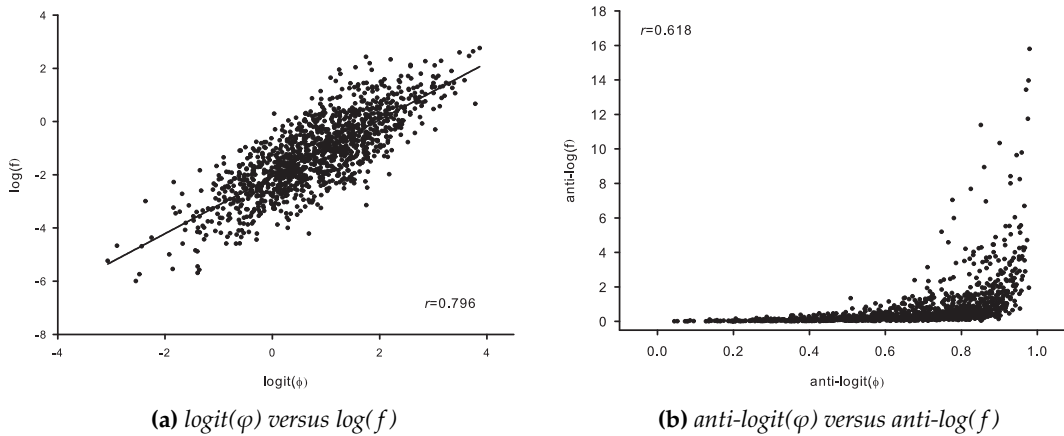


Figure E.11: Scatterplots of random samples from a bivariate normal logit-log distribution for $\text{logit}(\varphi)$ and $\log(f)$ with $\hat{\rho} = 0.7905$, and transformed and back-transformed scales. Values of μ and σ for each parameter are given in the text.

Despite this complication in evaluating ρ for time-symmetric models, at this point, you might imagine building a model where λ is in the likelihood, and looking for a correlation between λ on one of the vital rates (say, apparent survival, φ), as an approach to partitioning variation in λ due to variation in a particular vital rate. We'll leave this and other interesting questions for you to pursue as an exercise.

E.3. caveats, warnings, and general recommendations

Using random effects generally, and MCMC in particular, as a basis for modeling collections of related parameters is a relatively long-standing approach in statistics and one that can, in many cases, be very effective. Use of the random effects approach in what we refer to generally as 'capture-recapture' is relatively new – in the nearly 10 years since the publication of the seminal paper by Burnham & White (2002) on the 'method of moments' approach, and despite the recent astronomic rise in the application of MCMC to this and related problems in estimation, there have been relatively few applications of these models to real data, despite what we believe are several interesting opportunities made available

by these methods.

However, we also believe that both methodologies need to be better understood as to any potential pitfalls and as to its operating characteristics. The following is a summary of our experience to date with random effects models, particularly as implemented in **MARK**, and with MCMC (also as implemented in **MARK**). This material is abstracted from the equivalent section in Appendix D, supplemented with experience with such models since the time of that publication:

1. The ‘method of moments’ described in Appendix D, and as implemented in **MARK**, has been shown to perform well, especially when $\sigma^2 > 0.025$. This method may not do so well if $\sigma^2 \rightarrow 0$. However, we think it reasonable to believe that for a worthwhile study yielding good data, process variation, σ^2 , will generally not be too small, relative to average sampling variation and it is for these conditions (of ‘good data’) that we need effective random effects inference methods.

It is less clear how critical these issues are for MCMC estimation of σ^2 , but a recent paper by White, Burnham & Barker (2009) suggests that MCMC may not be a complete solution. Their general conclusion was that MCMC did very well for estimating the parameter mean μ , but performance was mixed with respect to estimating process variance σ^2 : for sparse data (i.e., ≤ 10 occasions), or when process variation was low, performance was poor. When there were sufficient data, and larger variance, performance was much improved.

2. Another issue to be aware of, as regards to estimation of the parameter σ^2 , is the matter of unequal, rather than equal length, time intervals. Let the time interval i have length Δ_i . Then we should parameterize the model as $S_i = (\psi_i)^{1/\Delta_i}$ where now each survival probability ψ_i is on the same unit time basis. It may then make biological sense to consider parameters that are a mean and variation for ψ_1, \dots, ψ_k . But this may just as well not make sense, because the time intervals are intrinsically not comparable as they may be in very different times of the annual cycle. It becomes a subject matter judgement as to whether random effects analysis will be meaningful with unequal time intervals. For the moment, don’t apply random effects models or variance components analysis – or MCMC – to situations where the intervals between sampling occasions are unequal.
3. A key design feature to focus on to meet the criterion of ‘having good data’ when applying random effects – or MCMC – is simply k , the number of estimable random effects parameters (time intervals, locations, etc.). The sample size for estimating σ^2 is k . Therefore, one must not have k too small; < 10 is too small. Even if we knew all the underlying S_i a sample of size $k < 10$ is too small for reliable inference about the variation of these parameters (even if we had a random sample of them, which is not required here). Inference performance has been shown to be acceptable when $k > 15$. The benefits (includes shrinkage estimates) of random effects models become greater as the number of underlying parameters, k , increases. And ability to estimate univariate and multivariate hyperparameters with MCMC also benefits significantly from longer time series.
4. A potential technical issue is the ‘boundary effect’, at least under what is basically a likelihood approach. As discussed in Burnham & White (2002), if one enforces the constraint $S < 1$ when the unbounded MLE $\hat{S} \geq 1$, then standard numerical methods used in **MARK** to get the observed information matrix fails. As a result, the estimated information matrix is incorrect for any terms concerning the \hat{S} that is at the bound of 1 (and the inverse information matrix is likely wrong in all elements). Experience shows that, in this case, the resultant point estimate of σ^2 can be very different from what one gets when the survival parameter MLE’s are allowed to be unbounded.

The difference can be substantial. Using an identity link, Burnham & White found $\hat{\sigma}^2$ to be unbiased in many cases. With good data we rarely observe an unbounded MLE of S

that exceeds 1. This might be explored in a Bayesian context, where it is easy (in a MCMC analysis) to allow S to have its distribution over an interval such as 0 to 2 (rather than the usual interval of 0 to 1). Burnham & White considered this, and found a strong effect of the upper-bound on the point estimate (and entire posterior distribution) for σ^2 , and for that particular S .

5. Another technical issue is accounting for the link function (transformation) when interpreting the estimates for various hyperparameters (μ, σ, ρ) , which are generated on the appropriate transformed scale. For μ and σ , the back-transformation is relatively straightforward. The only complication is that you need to use the appropriate back-transformation for σ – either via the Delta method, or a numerical simulation.

For multivariate hyperparameters (say, the correlation ρ between two structural parameters in a given model), you need to pay particular attention to whether or not both parameters are subjected to the same transformation, and whether the transformation is linear or not (at least over the range of the data). For situations where both parameters are estimated via MCMC on the logit scale, then $\hat{\rho}_{\text{logit}} = \hat{\rho}_{\text{real}}$. If, however, the transformations differ, and are non-linear for one or both parameters, interpretation is more complicated, and it may not be generally possible to come up with an acceptable interpretation of the correlation on the real scale.

E.4. Summary

This appendix has considered using Markov Chain Monte Carlo (MCMC) to estimate (i) mean and variance of a set of parameters, and (ii) covariation between sets of parameters. The former can also be accomplished using the ‘method of moments’ approach introduced in Appendix D, such that MCMC and ‘method of moment’ are best considered as complimentary approaches. In contrast, estimating the covariance among parameters requires something like MCMC.

At present, application of MCMC in **MARK** is limited to these two objectives. For more general application of MCMC to data from marked individuals, you will need to consider **BUGS**, or the equivalent. Nonetheless, the ability in **MARK** to quickly and easily explore patterns of variation and covariation among structural parameters for the large number of different data types available in **MARK** is a significant advance.

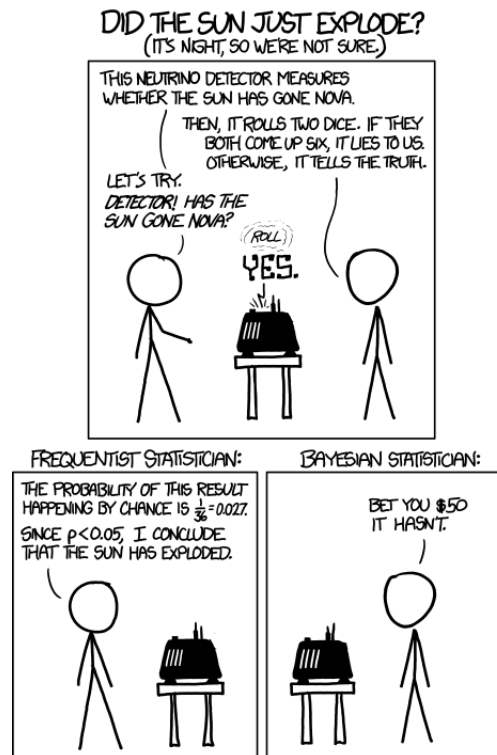
E.5. References

- Burnham, K. P., and White, G. C. (2002) Evaluation of some random effects methodology applicable to bird ringing data. *Journal of Applied Statistics*, **29**, 245-264.
- Celeux, G., Forbes, F., Robert, C. P., and Titterton, D. M. (2006) Deviance information criteria for missing data models. *Bayesian Analysis*, **1**, 651-674.
- Hooten, M. B., and Cooch, E. G. (2019) Comparing ecological models. Pages 63-76, in L. A. Brennan, A. N. Tri, and B. C. Marcot, editors. *Quantitative Analyses in Wildlife Science*. Johns Hopkins University Press.
- Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and van der Linde, A. (2002) Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society, Series B*, **64**, 583-639.
- Watanabe, S. (2013) A widely applicable Bayesian information criterion. *Journal of Machine Learning Research*, **14**, 867-897.
- White, G. C., Burnham, K. P., and Barker, R. J. (2009) Evaluation of a Bayesian MCMC random effects

inference methodology for capture-mark-recapture data. Pages 1119-1127, in D.L. Thomson, E. G. Cooch, and M.J. Conroy, editors. *Modeling Demographic Processes in Marked Populations*. Springer, Berlin.

Addendum – mechanics + basic principles of MCMC

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent urna sem, suscipit id aliquam id, mollis quis tellus. Phasellus varius velit a varius laoreet. Curabitur ullamcorper ante in lorem rhoncus, a sagittis orci venenatis. Quisque eu porttitor velit. Nulla auctor pharetra enim, nec feugiat eros adipiscing in. Donec lacinia neque vehicula, accumsan erat eget, interdum nibh. Aenean rutrum nec nulla nec tempor. Morbi mollis porttitor tortor. Curabitur at vestibulum ligula. Maecenas tincidunt dignissim nunc, quis cursus magna dictum et. Vestibulum sodales velit et mauris vehicula, non malesuada turpis pulvinar.



(This addendum still under construction....stay tuned.)