*Ecology*, 91(2), 2010, pp. 610–620


# Replicated sampling increases efficiency in monitoring biological populations

Brian Dennis,[1,4] José Miguel Ponciano,[2] and Mark L. Taper[3]

[1]*Department of Fish and Wildlife Resources and Department of Statistics, University of Idaho, Moscow, Idaho 83844-1136 USA*
[2]*Centro de Investigación en Matemáticas, CIMAT A.C. Calle Jalisco s/n, Colonia Valenciana, A.P. 402, C.P. 36240 Guanajuato, Guanajuato, México*
[3]*Montana State University, Department of Ecology, 301 Lewis Hall, Bozeman, Montana 59717-3460 USA*

*Abstract*. Observation or sampling error in population monitoring can cause serious degradation of the inferences, such as estimates of trend or risk, that ecologists and managers frequently seek to make with time-series observations of population abundances. We show that replicating the sampling process can considerably improve the information obtained from population monitoring. At each sampling time the sampling method would be repeated, either simultaneously or within a short time. In this study we examine the potential value of replicated sampling to population monitoring using a density-dependent population model. We modify an existing population time-series model, the Gompertz state-space model, to incorporate replicated sampling, and we develop maximum-likelihood and restricted maximum-likelihood estimates of model parameters. Depending on sampling protocols, replication may or may not entail substantial extra cost. Some sampling programs already have replicated samples, but the samples are aggregated or pooled into one estimate of population abundance; such practice of aggregating samples, according to our model, loses considerable information about model parameters. The gains from replicated sampling are realized in substantially improved statistical inferences about model parameters, especially inferences for sorting out the contributions of process noise and observation error to observed population variability.

*Key words: Breeding Bird Survey; density dependence; environmental noise; Gompertz growth model; Kalman filter; measurement error; observation error; process noise; profile likelihood; sampling error; state-space model; stochastic population model.*

## Introduction

The sizes of ecological populations are often estimated rather than censused, producing variability in the observed population abundances. This variability exists over and above the random population fluctuations due to environmental processes, and is variously termed "measurement error," "sampling error," or "observation error." A growing body of research has indicted observation error in population monitoring as a cause of serious degradation of the inferences, such as estimates of density-dependence strength, population trend, or extinction risk, that ecologists and managers typically want to make with such data (Solow 1990, 2001, Shenk et al. 1998, Holmes 2001, Holmes and Fagan 2002, Staples et al. 2004, Freckleton et al. 2006). The degradation takes two main forms. First, if observation error is not properly accounted for in the statistical models of population growth, then the estimates of model parameters (and functions of parameters such as probability of extinction) can be seriously biased. Second, if observation error is somehow adequately incorporated into statistical population models, then the observation variance and process variance parameters can be nearly nonidentifiable and difficult to estimate separately, producing ridge-shaped likelihoods and large confidence limits (Dennis et al. 2006, Knape 2008).

"State space" population models incorporating both process noise and observation error have great potential for improving population studies (de Valpine and Hastings 2002, Clark and Bjørnstad 2004). The inferential approaches in previous applications of state-space models have been largely Bayesian (e.g., Clark and Bjørnstad 2004). Recent developments, both analytical (Staples et al. 2004, Dennis et al. 2006) and computational (de Valpine 2003, 2004, Ionides et al. 2006, Lele et al. 2007), are helping to make statistical inferences possible for state-space models in the conventional frequentist sense. However, the problem remains for Bayesian and frequentist approaches alike that variability due to observation error is often difficult to disentangle from the natural population fluctuations (process noise) in the time-series abundance data.

One idea for improving the information obtained in population monitoring is to replicate the sampling

Manuscript received 10 June 2008; revised 23 January 2009; accepted 26 January 2009; final version received 22 April 2009. Corresponding Editor: K. Newman.
[4] E-mail: brian@uidaho.edu

process. Instead of one observation per time unit, the sampling method would be repeated, either simultaneously or within a short time, at each sampling time. Such replication could take such forms as another transect of river snorkeled, another mark–recapture sample conducted, another backcountry road driven and spotlighted at night, or another night of light trapping. Ideally, the replication would be designed to draw from all sources of variability that go into the observation error in the monitoring data. Although such replication might potentially improve the information that can be obtained from population monitoring, in fact such replication is almost nonexistent in reported ecological studies.

In a recent review of the sampling-error problem, Freckleton et al. (2006) offered the idea of replicated sampling and discussed results from two such studies in which the sampling-error component could be estimated separately. One was a study of shorebirds by Spearpoint et al. (1988), who estimated the variability in counts between two observers measuring the same populations simultaneously, as well as long-term (1981–1988) variability in numbers. The other was by Cunningham et al. (1999) who analyzed replicated censuses of 65 species of Australian woodland birds. In these studies, however, the sampling replication was a one-time study and not a routine part of the long-term population monitoring.

In a large study featuring many bird species, Link et al. (1994) analyzed what they termed "within-site variability" of bird-count data. They designed and carried out replicated sampling at a variety of sites in the North American Breeding Bird Survey (BBS; Peterjohn 1994), and they constructed variance-components models to estimate the amount of variability attributable to sampling. They found that in at least 14 out of 98 species more than half of the variation in bird counts was attributable to sampling variability. However, like the studies by Spearpoint et al. (1988) and Cunningham et al. (1999), the Link et al. (1994) study did not encompass multiple years, so that combining the information gleaned from the replicated sampling with time-series modeling of population abundances is not straightforward.

Indeed, we have not been successful in locating an example data set in which the sampling process has been intentionally replicated through time. Spatially distinct populations have been sampled simultaneously, but such data must usually be used to estimate separate process- and sampling-error parameters (albeit with possible covariation among the populations). Yet, given the degradation of parameter estimation associated with the presence of observation error, the idea of replicated sampling is intriguing. After all, population monitoring consumes much time, personnel, and resources of the agencies involved; is the information thereby gained useful for its intended purposes? How much can the information from population monitoring be improved by replicating the sampling procedures?

In this paper we introduce a state-space population model featuring observations from replicated sampling. The model can be used for analyzing replicated sampling data and for studying the potential value of replicated sampling in population monitoring. We modified an existing time-series population model, the Gompertz state-space model, to include replicated sampling. We describe maximum-likelihood and restricted maximum-likelihood parameter estimation for the model. Our computer simulations of parameter estimation reveal that substantial gains in parameter precision can be obtained with replicated sampling; such gains can be weighed by managers against the costs of the extra sampling. Some sampling programs already have replicated samples, but the samples are aggregated or pooled into one estimate of population abundance; such practice of aggregating samples, according to our results, loses considerable information about model parameters. We explain the appropriate statistical procedures for using the model, and we provide an R program for such use. We suggest that ecologists and managers might find replicated sampling worthwhile to undertake in a wide variety of settings.

## Methods

### *Replicated sampling model*

We first briefly review the Gompertz state-space (GSS) model for a single time series with unreplicated observations. Additional details can be found in Dennis et al. (2006). The GSS model takes a discrete-time, stochastic Gompertz model to represent the density-dependent growth of the population:

$$N_t = N_{t-1}\exp[a + b\,\ln(N_{t-1}) + E_t].$$

Here $N_t$ is population abundance at time $t$ ($t = 0, 1, 2, \ldots$), and $E_t \sim \text{normal}(0, \sigma^2)$, with $E_1, E_2, \ldots$ uncorrelated. The noise process $E_t$ represents environmentally induced fluctuations in the logarithmic population growth rate. The stochastic Gompertz assumes population abundance is known without error, with the random fluctuations being driven by ecological processes. The stochastic Gompertz forms the basis of density-dependence tests (Pollard et al. 1987, Dennis and Taper 1994), models of multiple sites (Langton et al. 2002), and models of multiple interacting species (Ives et al. 2003).

The GSS model further takes population abundance to be estimated with error, with the estimated or observed population $O_t$ at time $t$ given by

$$O_t = N_t\exp(F_t)$$

where $F_t \sim \text{normal}(0, \tau^2)$, and $F_1, F_2, \ldots$ are uncorrelated with each other and uncorrelated with $E_1, E_2, \ldots$. The lognormal error $\exp(F_t)$ is a model of population estimation error under heterogeneous ob-

serving or sampling conditions (Dennis et al. 2006). For data analysis, the model cast on the logarithmic scale is more convenient:

$$X_t = a + cX_{t-1} + E_t \qquad Y_t = X_t + F_t$$

in which $X_t = \ln N_t$, $c = b + 1$, and $Y_t$ is an observed or estimated value of $X_t$. It should be noted that $Y_t$ is an unbiased estimate of $X_t$. The GSS model has four parameters: $\sigma^2$ is the process-noise variance, $\tau^2$ is the sampling-error variance, $c$ is inversely related to the strength of density dependence, and $a$ scales the stationary mean (given by $a/(1 - c)$) of log-population abundance ($\sigma^2 > 0$, $\tau^2 > 0$, $-1 < c < +1$, $a > 0$).

Data-analysis methods for a single time series using the GSS model are based on a multivariate normal distribution (see Dennis et al. 2006). Let $Y_0, Y_1, \ldots, Y_q$ represent the (random) values of $Y_t$ in successive times, and let $y_0, y_1, \ldots, y_q$ be a particular realization of the process (i.e., the data). It can be shown that $Y_0, Y_1, \ldots, Y_q$ have a joint multivariate normal distribution with a mean vector and variance–covariance matrix that are functions of the model parameters and time. The probability density function (pdf) for the multivariate normal distribution of $Y_0, Y_1, \ldots, Y_q$ can be decomposed into a product of univariate normal pdfs, with each pdf conditioned on the previous history of the time series. The likelihood function for the unknown parameters $a$, $c$, $\sigma^2$, $\tau^2$ becomes

$$L(a, c, \sigma^2, \tau^2) = f(y_0)f(y_1 \mid y_0)f(y_2 \mid y_0, y_1) \times \cdots \times f(y_q \mid y_0, y_1, \ldots, y_{q-1}).$$

Here $f(y_t \mid y_0, y_1, \ldots, y_{t-1})$ is a normal distribution pdf with mean $m_t$ and variance $v_t^2$ that are calculated with two simultaneous recursion equations containing the parameters and previous observations:

$$m_t = a + c\left[m_{t-1} + \frac{v_{t-1}^2 - \tau^2}{v_{t-1}^2}(y_{t-1} - m_{t-1})\right]$$

$$v_t^2 = c^2\tau^2\frac{v_{t-1}^2 - \tau^2}{v_{t-1}^2} + \sigma^2 + \tau^2.$$

If the process is assumed to be stationary when the initial value $y_0$ was recorded, then the recursions are initiated at $m_0 = a/(1 - c)$, $v_0^2 = [\sigma^2/(1 - c^2)] + \tau^2$. If the initial population, however, was far from equilibrium, the recursions are initiated at $m_0 = x_0$, $v_0^2 = \tau^2$, with the underlying initial population size $x_0$ becoming another unknown parameter. The recursions are a special case of four matrix equations for conditional moments known as the "Kalman filter" (for instance, Harvey 1993), used for multivariate Gaussian time series. The parameter values that jointly maximize the likelihood function are the ML (maximum-likelihood) estimates. The maximization must be done numerically (SAS program given in Dennis et al. [2006]).

A special case of the model representing density-independent population growth occurs with $c = 1$. The model becomes a stochastic version of exponential growth or decay, with process noise and observation error (see Holmes 2001, Staples et al. 2004, Dennis et al. 2006). The likelihood function for the density-independent case is multivariate normal, but can be calculated using the above Kalman recursion equations with the value of $c$ fixed at 1. Population abundance under the density-independent model does not have stationary behavior, so the recursions must be initiated at $m_0 = x_0$, $v_0^2 = \tau^2$, with $x_0$ treated as an unknown parameter.

The extension of the GSS model to replicated sampling is as follows. Suppose at sampling time $t$, the sampling process is replicated $p_t$ times, producing observations $Y_{1t}, Y_{2t}, \ldots, Y_{p_t t}$. Denote by $\mathbf{Y}_t$ the $p_t \times 1$ column vector $[Y_{1t}, Y_{2t}, \ldots, Y_{p_t t}]'$ of the observations (as random variables) at time $t$, and denote by $\mathbf{y}_t$ the $p_t \times 1$ column vector $[y_{1t}, y_{2t}, \ldots, y_{p_t t}]'$ of the recorded outcomes (data values) of the random variables in the vector $\mathbf{Y}_t$ at time $t$. The number of replications $p_t$ can vary for different sampling times, but to apply the methods reported here, $p_t$ must be at least 1 for each sampling time. We write $\mathbf{j}_t$ for a $p_t \times 1$ column vector of 1's, $\mathbf{J}_t$ for a $p_t \times p_t$ matrix of 1's, and $\mathbf{I}_t$ for a $p_t \times p_t$ identity matrix. The Gompertz state space, replicated sampling (GSS-RS) model consists of the underlying population process joined with a multivariate sampling process:

$$X_t = a + cX_{t-1} + E_t \qquad \mathbf{Y}_t = \mathbf{j}_t X_t + \mathbf{F}_t.$$

Here $\mathbf{F}_t$ is a $p_t \times 1$ random vector having a multivariate normal distribution with a mean vector of 0's and a variance–covariance matrix $\tau^2\mathbf{I}_t$. The form of the variance–covariance matrix corresponds to the assumption that the observations at each sampling period are independent replicates (given the value of $X_t$) with constant variance $\tau^2$. The elements in $\mathbf{F}_t$ are additionally assumed to be uncorrelated through time and uncorrelated with $E_t$.

### *Maximum-likelihood estimation*

The basic result needed to form the likelihood function is the joint distribution of $\mathbf{Y}_t$ given $\mathbf{Y}_{t-1} = \mathbf{y}_{t-1}$, $\mathbf{Y}_{t-2} = \mathbf{y}_{t-2}$, $\ldots$, $\mathbf{Y}_0 = \mathbf{y}_0$. That distribution is a multivariate normal (MVN) distribution, with a mean vector $\mathbf{m}_t$ and a variance–covariance matrix $\mathbf{V}_t$ that are calculated recursively from the previous data, similar to the univariate case. We write

$$\mathbf{Y}_t \mid \{\mathbf{Y}_{t-1} = \mathbf{y}_{t-1}, \mathbf{Y}_{t-2} = \mathbf{y}_{t-2}, \ldots, \mathbf{Y}_0 = \mathbf{y}_0\} \sim \text{MVN}(\mathbf{m}_t, \mathbf{V}_t).$$

For this replicated sampling model, the Kalman recursion equations do not simplify much, and all four equations are needed:

$$\mu_t = a + c[\mu_{t-1} + \mathbf{j}'_{t-1}\varphi^2_{t-1}\mathbf{V}^{-1}_{t-1}(\mathbf{y}_{t-1} - \mathbf{m}_{t-1})]$$

$$\varphi^2_t = c^2\varphi^2_{t-1}[1 - \varphi^2_{t-1}\mathbf{j}'_{t-1}\mathbf{V}^{-1}_{t-1}\mathbf{j}_{t-1}] + \sigma^2$$

$$\mathbf{m}_t = \mathbf{j}_t\mu_t$$

$$\mathbf{V}_t = \mathbf{J}_t\varphi^2_t + \mathbf{I}_t\tau^2.$$

Here the recursions are started at $\mu_0 = a/(1 - c)$, $\varphi^2_0 = \sigma^2/(1 - c^2)$, $\mathbf{m}_0 = \mathbf{j}_0 a/(1 - c)$ for the stationary case, and $\mu_0 = x_0$, $\varphi^2_0 = 0$ for the non-stationary case. The scalar quantities $\mu_t$ and $\varphi^2_0$ are, respectively, the conditional mean and variance of the underlying process $X_t$, given the previous observations $\mathbf{Y}_{t-1} = \mathbf{y}_{t-1}$, $\mathbf{Y}_{t-1} = \mathbf{y}_{t-1}$, ..., $\mathbf{Y}_0 = \mathbf{y}_0$. The joint pdf for $\mathbf{Y}_t$ given the history $\mathbf{Y}_{t-1} = \mathbf{y}_{t-1}$, $\mathbf{Y}_{t-1} = \mathbf{y}_{t-1}$, ..., $\mathbf{Y}_0 = \mathbf{y}_0$ is that of a multivariate normal distribution with mean vector $\mathbf{m}_t$ and variance–covariance matrix $\mathbf{V}_t$:

$$f(\mathbf{y}_t \mid \mathbf{y}_{t-1}, \mathbf{y}_{t-2}, \ \ldots, \ \mathbf{y}_0) = (2\pi)^{-p_t/2}|\mathbf{V}_t|^{-1/2} \times \exp\left[-\frac{1}{2}(\mathbf{y}_t - \mathbf{m}_t)'\mathbf{V}^{-1}_t(\mathbf{y}_t - \mathbf{m}_t)\right]$$

where $\mathbf{m}_t$ and $\mathbf{V}_t$ are calculated from parameters and observation history using the Kalman recursion relationships. In parallel with the univariate case, the distribution of the initial observation vector $\mathbf{Y}_0$ is multivariate normal, with mean vector $\mathbf{m}_0$ and variance–covariance matrix $\mathbf{V}_0$.

The likelihood function under the GSS-RS model is formed as the product of the conditional multivariate normal pdfs:

$$L(a, c, \sigma^2, \tau^2) = f(\mathbf{y}_0)f(\mathbf{y}_1 \mid \mathbf{y}_0)f(\mathbf{y}_2 \mid \mathbf{y}_1, \ \mathbf{y}_0) \times \cdots \times f(\mathbf{y}_q \mid \mathbf{y}_{q-1}, \ \mathbf{y}_{q-2}, \ \ldots, \ \mathbf{y}_0).$$

The log-likelihood, used for the statistical calculations, then becomes

$$\ln L(a, c, \sigma^2, \tau^2) = -\frac{r}{2}\ln(2\pi) - \frac{1}{2}\sum_{t=0}^{q}\ln|\mathbf{V}_t| - \frac{1}{2}\sum_{t=0}^{q}(\mathbf{y}_t - \mathbf{m}_t)'\mathbf{V}^{-1}_t(\mathbf{y}_t - \mathbf{m}_t).$$

Here $r = p_0 + p_1 + \cdots + p_q$ is the total number of observations, and $q + 1$ is the total number of times at which the population has been sampled. The likelihood function contains the additional unknown parameter $x_0$ when the non-stationary assumption is used.

The recursion equations and likelihood function under replicated sampling remain valid for the density-independent case with $c = 1$. The recursions are initiated at the nonstationary conditions $\mu_0 = x_0$, $\varphi^2_t = 0$, with $x_0$ treated as an unknown parameter.

The likelihood function $L(a, c, \sigma^2, \tau^2)$ can be shown to be that of a single multivariate normal distribution for the vector formed by stacking the vectors $\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_q$ of observations. The above Kalman filter representation represents a decomposition of that likelihood. The multivariate normal distribution follows from linearly transforming the $X_t$'s and $F_t$'s in the definition of the $\mathbf{Y}_t$'s and is derived and displayed in the Appendix.

The ML estimates of unknown parameters for the GSS-RS model are the values that jointly maximize the likelihood function. There are no closed-form formulas for such estimates, because the likelihood is too complex for simple calculus maximization. Numerical optimization routines available in R, MATLAB, and various other computational software packages are easy to use (Appendix). Although the ML calculations for the univariate GSS model can be performed with mixed-effects analysis of variance programs (such as PROC MIXED in SAS; see Dennis et al. 2006), there appears to be no such computational resource for the GSS-RS model.

### *Restricted maximum-likelihood estimation*

While ML estimates are asymptotically unbiased in statistical theory, for random-effects models it is known that ML parameter estimates can often retain a substantial bias for seemingly large yet finite samples. The recommended improvement of restricted maximum-likelihood (REML) estimates has now become standard practice (Searle et al. 1992). REML estimation transforms the data linearly so as to remove the fixed-effects parameters, leaving the variance components to be estimated through the covariance structure of the transformed data.

REML estimates are formed by using a linear transformation of the observations that has as a mean a vector of 0's. The procedure eliminates uncertainty in the estimate of the mean vector from the estimation of the variance components; the information in the data is concentrated toward estimating parameters in a variance–covariance matrix. One REML transformation of the observations in replicated sampling can be defined as follows. Multiply each element in the vector $\mathbf{y}_t$ ($t = 1, 2, \ldots, q$) by $p_{t-1}$ (the size of the previous vector), and then subtract the sum of the elements in the previous vector $\mathbf{y}_{t-1}$, that is,

$$w_{it} = p_{t-1}y_{it} - (y_{1t-1} + y_{2t-1} + \cdots + y_{p_{t-1}t-1}).$$

Let $\mathbf{w}_t = [w_{1t}, w_{2t}, \ldots, w_{p_t t}]'$. Then the transformed observations in $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_q$, stacked into a column vector, are from a multivariate normal distribution with a mean vector of 0 and a variance–covariance matrix having elements that depend on the unknown parameters (derived and displayed in the Appendix). Numerical maximization is required but is straightforward with matrix-based programming languages (Appendix).

ML estimates remain important, for calculating confidence intervals based on profile likelihoods, and for likelihood-based model selection. In particular,

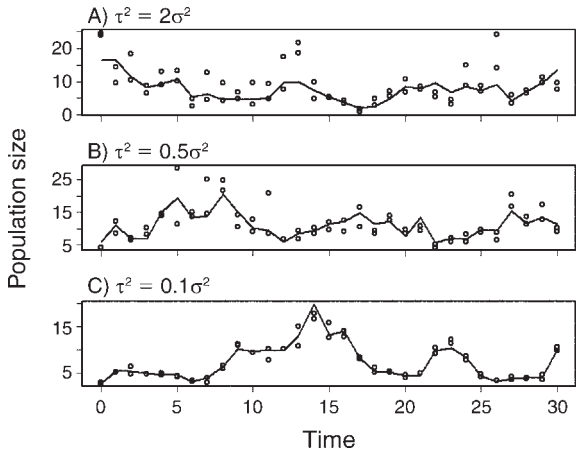


FIG. 1. Three simulated time-series data sets for the Gompertz state-space model with replicated sampling. The solid lines represent the underlying true population abundances; circles are the sampled/estimated population abundances, with two sampling replications each time period. Here $\sigma^2$, $\tau^2$, $c$, and $\alpha$ are the four parameters in the Gompertz state-space model: $\sigma^2$ is the process noise variance, $\tau^2$ is the sampling error variance, $c$ is inversely related to the strength of density dependence, and $\alpha$ scales the stationary mean (given by $\alpha/[1 - c]$) of log(population abundance) ($\sigma^2 > 0$, $\tau^2 > 0$, $-1 < c < +1$, $\alpha > 0$). Parameter values used were: (A) $a = 0.4$, $c = 0.8$, $\sigma^2 = 0.1$, $\tau^2 = 0.2$; (B) $a = 0.4$, $c = 0.8$, $\sigma^2 = 0.1$, $\tau^2 = 0.05$; and (C) $a = 0.4$, $c = 0.8$, $\sigma^2 = 0.1$, $\tau^2 = 0.01$. In panel (A) the parameters are maximum-likelihood estimates for an example time series in the North American Breeding Bird Survey calculated by Dennis et al. (2006).

model selection with the Akaike information criterion (AIC; Akaike 1973, Sakamoto et al. 1986) and its variants requires maximized log-likelihood functions for all the models under consideration, using the same data. The likelihood used for REML estimates applies to the transformed observations ($w_{it}$'s) and is not comparable to likelihoods for other models fitted to the untransformed observations ($y_{it}$'s).

We compared ML estimation and REML estimation with computer simluation (see *Simulations*, below).

### *Disaggregating data*

Although replication of samples might seem onerous or prohibitively costly, it might not be in practice depending on sampling protocols. In some situations such replication might already have been accomplished unwittingly, and no extra data need be collected. A not-infrequent practice in population-monitoring studies is to aggregate subsamples, say of areas or transect portions, into one overall estimate of population abundance. The subsamples may be legitimate replicates. Other studies, such as the BBS in North America or the Common Birds Census (CBC) in the United Kingdom, feature simultneous estimates from spatially separated samples of possibly the same populations. Whether such samples can be considered replicates is a scientific judgment to be made case by case, but if the samples can be analyzed as replicates, considerable gains in parameter estimation might be realized. The question is, how much information about model parameters might be gained by retaining and analyzing the subsamples as replicated samples rather than aggregating them?

The GSS-RS model can be used to study the effects of disaggregating data into replicated samples. Suppose for example there are the same number, $p$, of samples each sampling time, and suppose that the sample observations $Y_{1t}, Y_{2t}, \ldots, Y_{pt}$ arise from a GSS-RS model with parameters $a$, $c$, $\sigma^2$, and $\tau^2$. With lognormal sampling error, it would be natural to combine data on the logarithmic scale in order to obtain an unbiased estimate of $X_t$, and so we define the aggregated population estimate as the sample mean of the log-abundances in the subsamples:

$$\bar{Y}_t = \frac{1}{p}\sum_{i=1}^{p} Y_{it}.$$

It is straightforward to show that $\bar{Y}_0, \bar{Y}_1, \ldots, \bar{Y}_q$ follow a univariate GSS model with parameters $a$, $c$, $\sigma^2$, and reduced sampling variance $\tau^2/p$. Thus, parameter estimates obtained with the disaggregated observations can be contrasted with the estimates of the same parameters obtained with the aggregated observations. We view results of a computer simulation in the next section.

## SIMULATIONS

Data sets simulated with the Gompertz state-space, replicated sampling (GSS-RS) model show the striking effects of sampling variability (Fig. 1). Each simulation in Fig. 1 has two observations per sampling time. The numerical values of the parameters $a$, $c$, $\sigma^2$, and $\tau^2$ used in the first simulation (Fig. 1A) were maximum-likelihood (ML) estimates for a time series in the Breeding Bird Survey, and so the values reflect an actual field setting (see Dennis et al. 2006: Table 1 and Fig. 1). The original time series, consisting of counts of American Redstart (*Setophaga ruticilla*) at a BBS location, had one observation at each sampling time. The ML estimates suggest that the larger proportion of the variability in the counts was due to observation error; the estimate of the sampling-variability parameter $\tau^2$ is twice as large as that of the process variability parameter $\sigma^2$. Although the simulated true population abundances (solid line) show considerable variability, the observations (circles) in the top panel of Fig. 1A are by comparison widely scattered. Decreasing the value of $\tau^2$ to half of the value of $\sigma^2$ (Fig. 1B) reduces the sampling-based scattering, but the graph still gives the visual impression that observation error is obscuring the population signal. The observations resemble the true population abundances when the value of $\tau^2$ is decreased to one-tenth the value of $\sigma^2$ (Fig. 1C).

The likelihood functions for the univariate GSS model tend to be narrow, ridge-shaped, and multi-

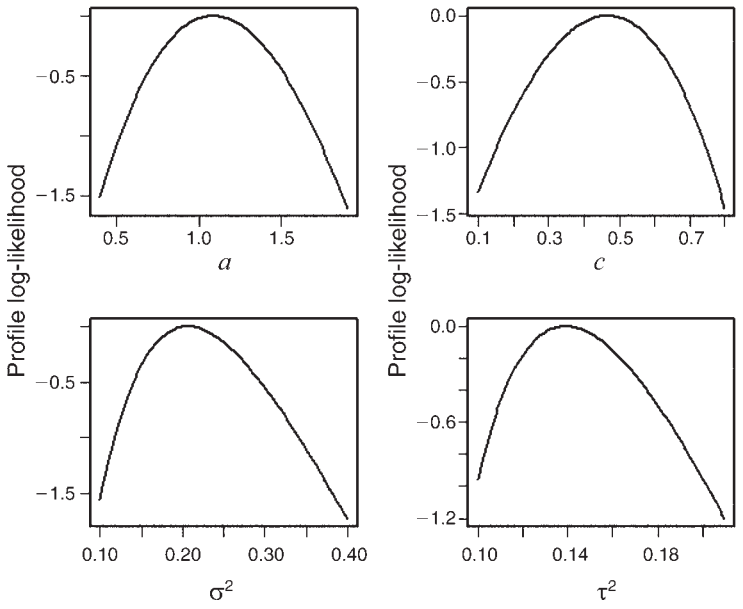


FIG. 2. Profile log-likelihoods (with maximized log-likelihood values subtracted) for the parameters $a$, $c$, $\sigma^2$, and $\tau^2$ in the Gompertz state-space model, calculated for a simulated data set having 31 sampling times ($q = 30$ = number of sampling times minus 1), with two replicated observations per sampling time. Parameter values used in the simulation were $a = 0.4$, $c = 0.8$, $\sigma^2 = 0.1$, and $\tau^2 = 0.2$.

modal (Dennis et al. 2006: Fig. 3). The likelihood shapes reflect the near-nonidentifiability of the parameters $\sigma^2$ and $\tau^2$, that is, different combinations of $\sigma^2$ and $\tau^2$ values produce nearly the same likelihood function heights. Despite the similarity in likelihood values these different parameter combinations can represent profoundly different underlying population dynamics. The Dennis et al. (2006) study reported that such likelihood functions were common in simulations of the GSS model, even with time-series lengths of 100 or more.

What can replicated sampling contribute to parameter identifiability? Profile log-likelihoods for parameters of the GSS-RS (replicated sampling) model suggest that replicated sampling essentially fixes the estimation problems (Fig. 2). The profile log-likelihoods in Fig. 2, calculated using the simulated data plotted in Fig. 1A, are unimodal and approximately parabolic. The parabolic shapes indicates that the distributions of the ML estimates are nearly normal, suggesting that the estimates are converging swiftly to the theoretical asymptotic normal distributions (e.g., Pawitan 2001) for ML estimates. The parabolic shapes also suggest that the parameters are statistically identifiable. In addition, the parabolic shapes render the ML estimates easy to calculate with numerical optimization techniques. The locations of the profile peaks are somewhat far from the true values of the parameters, however, suggesting the presence of a bias in the ML estimates. The bias likely could be corrected with parametric bootstrapping (Ospina et al. 2006) or penalized likelihood (Pawitan 2001); alternatively restricted maximum-likelihood (REML) estimates can be used (simulations area described later in this section). The R program in the Supplement to this paper produces ML and REML estimates as well as profile log-likelihood plots for the GSS model; the program contains the example data and reproduces the plots in Fig. 2.

How much information is gained toward parameter estimation from replicated sampling? We calculated profile log-likelihoods for the GSS model for two simulated data sets, one set having 20 sampling times ($q = 19$ = the number of sampling times minus 1) with two replicated observations per sampling time, and the other set having 40 sampling times ($q = 39$) with just one replication per sampling time (Fig. 3). The data sets conceptually represent the same amount of expended sampling effort. The difference is striking. The longer univariate time series (dashed line) produced wider profiles with extra shoulders or maxima, while the shorter time series with replicated samples (solid line) produced profiles with regular parabolic shapes. Thus, the same total sampling effort spent in replicated sampling can produce better information about the parameters in less time.

We simulated ML and REML estimation under replicated sampling (Fig. 4). The simulations incorporated two replicate samples for each sampling time, with lengths of 31 ($q = 30$; Fig. 4A) and 101 ($q = 100$; Fig. 4B) sampling times. The 101 sampling times, while unrealistically long in current ecological research, allow assessment of statistical convergence. The parameters from Fig. 1A were used as reference values in the simulations. In each simulation, 2000 data sets were

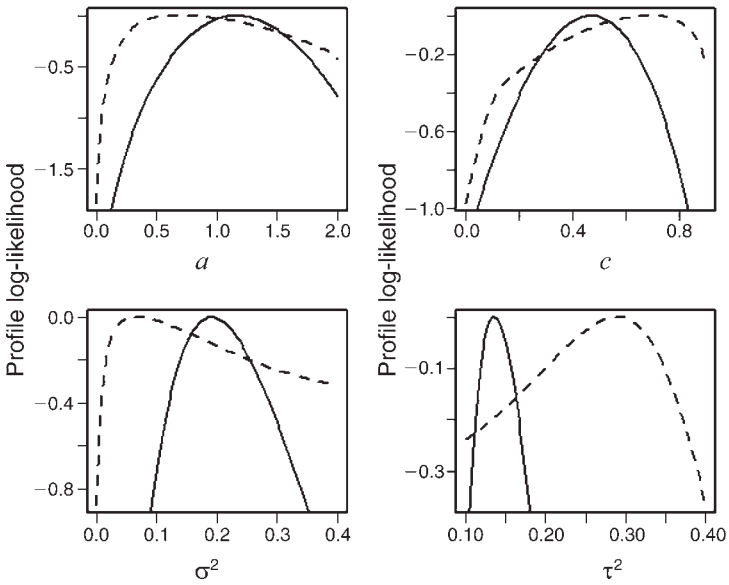


FIG. 3. Profile log-likelihoods (with maximized log-likelihood values subtracted) for the parameters $a$, $c$, $\sigma^2$, and $\tau^2$ in the Gompertz state-space model. The solid lines represent profiles calculated for a simulated data set having 20 sampling times ($q = 19$), with two replicated observations per sampling time; the dashed lines represent profiles for 40 sampling times ($q = 39$), with one observation per sampling time. Parameter values used in the simulation were $a = 0.4$, $c = 0.8$, $\sigma^2 = 0.1$, and $\tau^2 = 0.2$.

generated from the GSS-RS model, and 2000 sets of parameter estimates were obtained by fitting the model (i.e., numerically maximizing the likelihood or restricted likelihood) to the generated data. Each box plot depicts 2000 parameter estimates divided by the reference parameter value, so that relative variability and bias can be compared across different parameters.

For 31 sampling times, substantial median biases were evident in the ML estimates of $a$ and $c$ (Fig. 4A). The ML estimates of $a$ were biased upward and were substantially more variable than the ML estimates of the other parameters. The ML estimates of $c$ were biased downward but had relatively small variability. The biases in ML estimates of $a$ and $c$ persisted for 101 sampling times (Fig. 4B). It has been noted (Dennis et al. 2006) that the GSS model is a type of random-effects model, and ML estimation in such models often has persistent finite-sample bias due to the dependence of the observations. However, the near-parabolic shapes of the log-likelihoods in the GSS-RS model (Fig. 2) indicate statistical regularity and suggest that the biases could be routinely corrected by bootstrapping. In addition, ML estimates of certain functions or transformations of the parameters in the GSS model, such as the stationary mean $a/(1 - c)$, turn out to be nearly unbiased (Dennis et al. 2006).

REML estimates, in contrast to ML estimates, were well centered. Little median bias was evident in REML estimates of any parameters for either 31 sampling times (Fig. 4A) or for 101 sampling times (Fig. 4B). The variability of the REML estimates for 101 sampling times was of course reduced from that of 31 sampling times, although the relatively modest amount of reduction seems disproportionate to the large increase in sample size. Such slow gains from increasing sample sizes are common in dependent data problems. The greatest gains tend to occur by altering sampling protocols, such as in the present case by taking two samples instead of one at each sample time.

If samples have been aggregated into one population estimate, what is the effect of disaggregating them into replicated samples? Note that disaggregation is a much different question than the question of whether to undertake additional samples. Aggregating samples holds out the promise of reducing the sampling variability in a univariate time series; recall that the sampling-variability parameter becomes $\tau^2/p$ (where $p$ is the number of samples in each sampling time) in the GSS model. Can better parameter estimates instead be obtained by disaggregating, that is, by simply analyzing the data differently? We generated data under the GSS-RS model, with two replicate samples per sampling time, using the parameter values from Fig. 1A. Each of the replicate sample pairs were then aggregated as sample averages on the logarithmic scale, forming a univariate time series. We compared parameter estimates obtained by fitting the GSS-RS model (parameters $a$, $c$, $\sigma^2$, and $\tau^2$), to the replicated sampled with estimates obtained by fitting the GSS model (parameters $a$, $c$, $\sigma^2$, and $\tau^2/p$, where $p = 2$) to the aggregated samples. Fig. 5 shows log-profile likelihoods calculated for $a$, $c$, $\sigma^2$, and $\tau^2$ under both the GSS-RS (solid curves) and GSS (dashed curves) approaches, for a representative simulated data set. For each parameter, using the replicated samples

and the full GSS-RS model produces steeper profiles, indicating greater information in the data. In the cases of $a$ and $c$, the GSS-RS profiles are only slightly steeper. Note, however, that the GSS profiles have "shoulders" indicating the possibility of another local maximum. The shoulders in the GSS profiles for $\sigma^2$ and $\tau^2$ are pronounced. The GSS-RS profiles for $\sigma^2$ and $\tau^2$ are considerably steeper than the GSS profiles, showing that disaggregating into replicated samples is especially advantageous for sorting out the sources of variability in population data. Fig. 5 and Fig. 3 portray different concepts, although they look similar. While Fig. 3 contrasts taking one sample rather than two, the second contrasts pooling two replicated samples rather than keeping them disaggregated.

## Discussion

### *Improvement in inferences*

Replicating the samples potentially offers a substantial gain of information in return for the costs involved. Disentangling sampling variability and process variability in unreplicated time series is difficult (Dennis et al. 2006). Very long time series are necessary for clean separation of the two variance components. The ridge-shaped, multimodal likelihood functions require case-by-case attention, and the analyses are hard to automate for bootstrapping or processing multiple data sets. Estimation of the other parameters in population models, and functions of parameters, depends on adequate estimation of sampling variability. Thus, the inferences about the populations, such as whether they are declining, growing, or stable, and whether changes in the population processes have occurred, are degraded without adequate information about the contribution of sampling variability. In particular, estimates of first-passage properties such as persistence probabilities in population viability analysis are known to be sensitive to sampling variability (Holmes 2001, Holmes and Fagan 2002).

Most noteworthy about our replicated-sampling simulations was that they were numerically well behaved. Even for 31 sampling occasions, the profile log-likelihood functions were smooth and unimodal (Fig. 2). The log-likelihood shapes indicate that replicated samples contain sharper information for estimating the parameters, in particular for separating the contributions of process noise and observation error.

### *Alternative model forms*

Many alternatives to the Gompertz state-space, replicated sampling (GSS-RS) model exist that could accommodate replicated sampling. Alternative state-space models would incorporate changes in either the sampling component, or the population abundance component, or both.

A state-space model could use, for instance, a Poisson sampling model, in which counts $Y_{1t}$, $Y_{2t}$, ..., $Y_{p,t}$ at time $t$ arise from independent Poisson distributions,

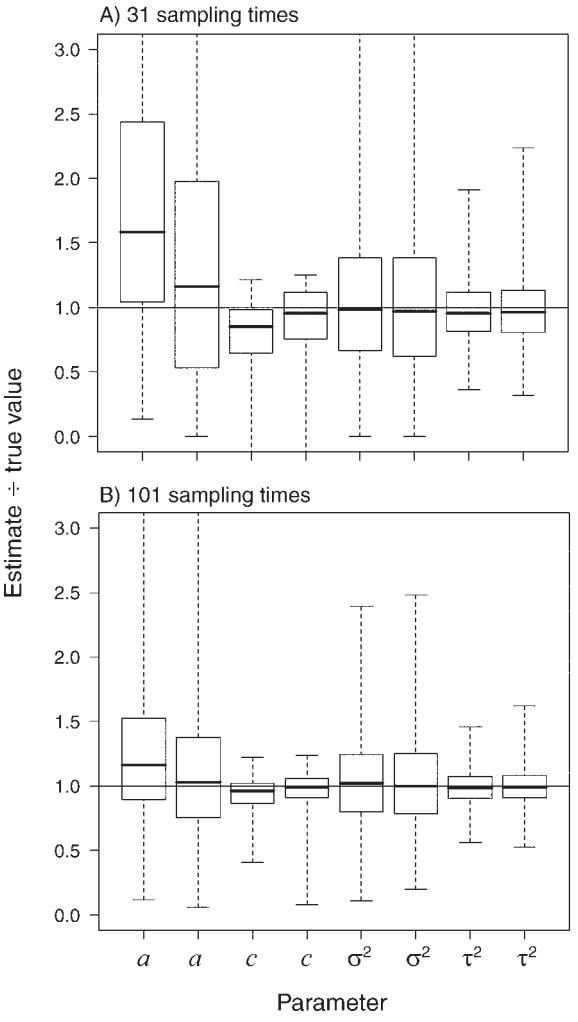


FIG. 4. Box plots of maximum-likelihood (left) and restricted maximum-likelihood (right) parameter estimates divided by their true values, for the parameters $a$, $c$, $\sigma^2$, and $\tau^2$ in the Gompertz state-space model with replicated sampling, calculated for 2000 simulated time-series data sets with (A) 31 sampling times ($q = 30$) and (B) 101 sampling times ($q = 100$) and two replicated observations per sampling time. Data sets were simulated using the parameter values from Fig. 1A. The box plots show (bottom to top): minimum, 1st quartile, median, 3rd quartile, and maximum.

each with mean $\lambda N_t$, where $\lambda$ is a parameter and the underlying population abundance $N_t$ is governed by some stochastic population model. The probability of a count $y$ under Poisson sampling is given by

$$\mathrm{P}[Y_{it} = y] = e^{-\lambda N_t}(\lambda N_t)^y/(y!)$$

for outcomes $y = 0, 1, 2, \ldots$. An attractive aspect of the Poisson model is that 0's in the data are accommodated as events having positive, even substantial, probability.

The Poisson model classically applies to sampling a well-mixed laboratory microbial population and could in some circumstances approximate the variability in standard closed-population mark–recapture estimation,

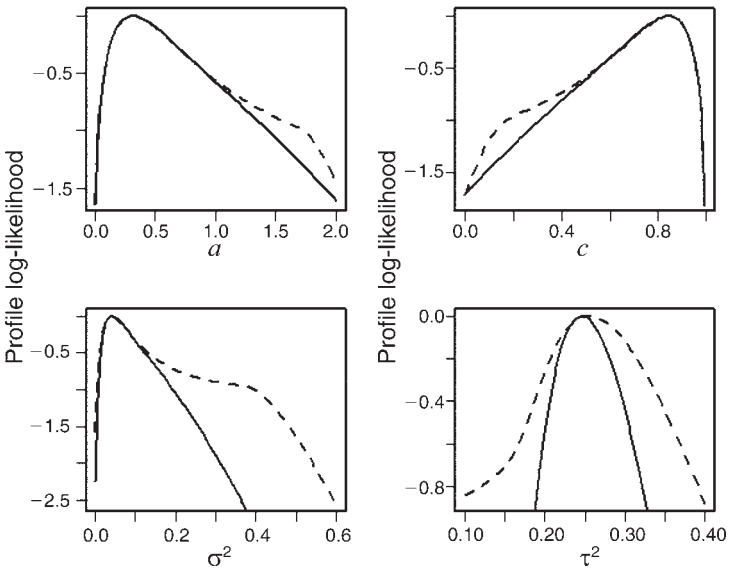


FIG. 5. Profile log-likelihoods (with maximized log-likelihood values subtracted) under replicated sampling (solid curves) and under aggregation of replicates by averaging (dashed curves), for the parameters $a$, $c$, $\sigma^2$, and $\tau^2$ in the Gompertz state-space model. Data were simulated using 30 sampling times ($q = 29$), two replicated observations per sampling time, and parameters set at $a = 0.4$, $c = 0.8$, $\sigma^2 = 0.1$, and $\tau^2 = 0.2$.

transect sampling, or population indexes such as sightings and light-trap captures. Ver Hoef and Frost (2003) for instance develop a Poisson sampling model for aerial surveys of harbor seals in Alaska, although they do not connect the means with a population-dynamics model. Muhlfeld et al. (2006) experimentally demonstrated the validity of Poisson sampling to redd (identifiable fish oviposition sites) count data.

However, many field situations are characterized by heterogeneous sampling conditions leading to overdispersion in the observations. The lognormal sampling component in the GSS-RS model is a model of overdispersed sampling, but it is a continuous distribution in which 0's are not allowable outcomes. An alternative overdispersed sampling model for count data is the negative binomial distribution. One formulation of the negative binomial for state-space modeling takes the probability of a count of $y$ to be

$$\mathrm{P}[Y_{it} = y] = \frac{\Gamma(\alpha + y)}{\Gamma(y+1)\Gamma(\alpha)} \left(\frac{N_t}{N_t + \beta}\right)^y \left(\frac{\beta}{N_t + \beta}\right)^\alpha$$

for outcomes $y = 0, 1, 2, \ldots$, where $\alpha$ and $\beta$ are positive parameters and $\Gamma(\cdot)$ is the gamma function. This negative binomial model differs from the above-mentioned Poisson sampling model by allowing $\lambda$ to have a gamma distribution with probability density function $g(\lambda) = \beta^\alpha \lambda^{\alpha-1} e^{-\beta\lambda}/\Gamma(\alpha)$. Link and Sauer (1998) apply the negative binomial sampling model toward estimating population trends in various Breeding Bird Survey time series, although they do not describe the underlying process with a dynamic population model.

Alternatives for the population-abundance component include various forms of density dependence and process noise, such as stochastic versions of the Ricker, Beverton-Holt, theta-Ricker, or Hassell growth models (Bellows 1981). While extensive analyses undertaken with thousands of time-series data sets (Woiwod and Hanski 1992, Zeng et al. 1998, Sibly et al. 2005, Brook and Bradshaw 2006) have suggested that particular model forms describe nature more often, none of these analyses have adequately accounted for sampling variability in the data. Certainly, no replicated sampling studies are yet known that might allow careful evaluation of different forms for both the sampling component and the population-abundance component.

One potential advantage of some alternative models is that the weak identifiabilities of the sampling and process variabilities might be sidestepped. For instance, if one can justify using the above Poisson sampling model with $\lambda = 1$, then an entire parameter is eliminated. Similar structures can be built into process variability to bypass its estimation (e.g., Sullivan 1992, Newman 1998, Besbeas et al. 2002). However, such approaches if used should be based firmly on the biological and methodological properties of the system at hand, and not contrived for convenience. Too few parameters can be as misleading as too many parameters, in that bias can be large when models are misspecified.

### *Computing for alternative models*

Alternative models to the GSS-RS, in which either the population-abundance component or the sampling component is altered, pose additional computing problems for parameter estimation. The essential problem is that the likelihood function in such "hierarchical models" (models with random components, in this case,

population abundance $N_t$) is a multidimensional integral that generally cannot be written down in closed form. The Gompertz growth model and the lognormal sampling model conveniently combine on the logarithmic scale to produce a multivariate normal likelihood, but no such consolidation occurs for nonnormal models. The recent "data cloning" algorithm (Lele et al. 2007) is a promising method for calculating ML estimates in hierarchical models, including state-space models. The data cloning method re-directs the computer-intensive Bayesian MCMC algorithms, ordinarily used for calculating posterior distributions in Bayesian statistics, toward calculating ML estimates and asymptotic frequentist confidence intervals. Other frequentist approaches for state-space models (de Valpine and Hastings 2002, de Valpine 2003, 2004, Ionides et al. 2006) are computer intensive as well. Alternatively, full Bayesian inferences can be undertaken (see Clark and Bjørnstad 2004), provided one understands the substantial conceptual differences between the Bayesian and frequentist approaches (Dennis 1996, 2004, Lele and Dennis 2009). We point out that the references cited in this subsection all contain numerically computed examples of alternative, realistic state-space models.

### *Correlated observations*

Situations exist for which the observations in replicated samples might be correlated. Some examples are: (1) if each year a different observer performs the sampling, but within each year replicated observations receive a bias from that observer's technique; (2) observation conditions vary substantially from year to year, but are similar for the within-year replicated samples performed close together in time; and (3) sampling protocols vary from year to year. The main feature of such correlation is that a different sampling bias is added each year essentially as a random effect. If the replications are modeled as independent when sampling correlation is present, the result could be spuriously high accuracy in estimation. One potential advantage of aggregation might be to avoid such sampling correlation. Alternatively, the sampling correlation could be modeled. Modeling correlation between replicated samples is straightforward, but estimation has not been studied. One might anticipate that difficulties with parameter identifiability could arise. Investigators at present should try to avoid sampling correlation by designing the replications to avoid such within-year biases. Ideally, all possible sources of variability in sampling should be embodied in each replicate.

### *Concluding remarks*

The importance of biological monitoring data cannot be over emphasized. A high-quality data set recording a population's abundances, long-term, can be a canary in the mine. The responses of ecological communities to global climate change are projected to be large, fast, and extensive (for instance, Rehfeldt et al. 2006). Protecting species and communities from extinction, and sustaining the ecological services derived by humans from the earth's biota, will require major modifications to human economic activities. Policy decisions about economic activities will hinge on the changes observed in biological monitoring data and on the scientifically inferred causes of those changes.

Yet, inadequate attention to the design of a monitoring program risks losing the very signal that the program exists to monitor. Ecologists now know, for instance, that conventional ecological sampling methods can contribute more than enough variability to cloud or bias conclusions about population abundance and dynamics. Replicating samples may be expensive or not, depending on whether existing sampling designs can be disaggregated. In either case sample replication should be considered as a survey design issue where the goal is to maximize the amount of useful information gained within the constraints of allowable cost. As managers, let us not waste money or the future of the species in our charge. We suggest checking the canary again, using replicated sampling.

## Acknowledgments

This work was supported in part by Montana Fish, Wildlife, and Parks contract 060327 to M. L. Taper. Additional partial support to B. Dennis came from the Strategic Environmental Research and Development Program (Project SI-1477). We are grateful for the numerous helpful suggestions for manuscript revision provided by Rob Freckleton and an anonymous referee.

## Literature Cited

Akaike, H. 1973. Information theory and an extension of the maximum likelihood principle. Pages 267–281 *in* B. N. Petrov and F. Csaki, editors. Second International Symposium on Information Theory. Akademiai Kiado, Budapest, Hungary.

Bellows, T. S. 1981. The descriptive properties of some models for density dependence. Journal of Animal Ecology 50:139–156.

Besbeas, P., S. N. Freeman, B. J. T. Morgan, and E. A. Catchpole. 2002. Integrating mark–recapture–recovery and census data to estimate animal abundance and demographic parameters. Biometrics 58:540–547.

Brook, B. W., and C. J. A. Bradshaw. 2006. Strength of evidence for density dependence in abundance time series of 1198 species. Ecology 87:1445–1451.

Clark, J. S., and O. N. Bjørnstad. 2004. Population time series: process variability, observation errors, missing values, lags, and hidden states. Ecology 85:3140–3150.

Cunningham, R. B., D. B. Lindenmayer, H. A. Nix, and B. D. Lindenmayer. 1999. Quantifying observer heterogeneity in bird counts. Australian Journal of Ecology 24:270–277.

Dennis, B. 1996. Discussion: should ecologists become Bayesians? Ecological Applications 6:1095–1103.

Dennis, B. 2004. Statistics and the scientific method in ecology (with commentary). Pages 327–378 *in* M. L. Taper and S. R. Lele, editors. The nature of scientific evidence: statistical, philosophical, and empirical considerations. University of Chicago Press, Chicago, Illinois, USA.

Dennis, B., J. M. Ponciano, S. R. Lele, M. L. Taper, and D. F. Staples. 2006. Estimating density dependence, process noise, and observation error. Ecological Monographs 76:323–341.

Dennis, B., and M. L. Taper. 1994. Density dependence in time series observations of natural populations: estimation and testing. Ecological Monographs 64:205–224.

de Valpine, P. 2003. Better inferences from population-dynamics experiments using Monte Carlo state-space likelihood methods. Ecology 84:3064–3077.
de Valpine, P. 2004. Monte Carlo state-space likelihoods by weighted posterior kernel density estimation. Journal of the American Statistical Association 99:523–536.
de Valpine, P., and A. Hastings. 2002. Fitting population models incorporating process noise and observation error. Ecological Monographs 72:57–76.
Freckleton, R. P., A. R. Watkinson, R. E. Green, and W. J. Sutherland. 2006. Census error and the detection of density dependence. Journal of Animal Ecology 75:837–851.
Harvey, A. C. 1993. Time series models. Second edition. MIT Press, Cambridge, Massachusetts, USA.
Holmes, E. E. 2001. Estimating risks in declining populations with poor data. Proceedings of the National Academy of Sciences (USA) 98:5072–5077.
Holmes, E. E., and W. F. Fagan. 2002. Validating population viability analysis for corrupted data sets. Ecology 83:2379–2386.
Ionides, E. L., C. Bretó, and A. A. King. 2006. Inference for nonlinear dynamical systems. Proceedings of the National Academy of Sciences (USA) 103:18438–18443.
Ives, A. R., B. Dennis, K. L. Cottingham, and S. R. Carpenter. 2003. Estimating community stability and ecological interactions from time-series data. Ecological Monographs 73: 301–330.
Knape, J. 2008. Estimability of density dependence in models of time series data. Ecology 89:2994–3000.
Langton, S. D., N. J. Aebischer, and P. A. Robertson. 2002. The estimation of density dependence using census data from several sites. Oecologia 133:466–473.
Lele, S. R., and B. Dennis. 2009. Bayesian methods for hierarchical models: Are ecologists making a Faustian bargain? Ecological Applications 19:581–584.
Lele, S. R., B. Dennis, and F. Lutscher. 2007. Data cloning: easy maximum likelihood estimation for complex ecological models using Bayesian Markov chain Monte Carlo methods. Ecology Letters 10:551–563.
Link, W. A., R. J. Barker, J. R. Sauer, and S. Droege. 1994. Within-site variability in surveys of wildlife populations. Ecology 75:1097–1108.
Link, W. A., and J. R. Sauer. 1998. Estimating population change from count data: application to the North American Breeding Bird Survey. Ecological Applications 8:258–268.
Muhlfeld, C. C., M. L. Taper, D. F. Staples, and B. B. Shepard. 2006. Observer error structure in bull trout redd counts in Montana streams: implications for inference on true redd numbers. Transactions of the American Fisheries Society 135:643–654.
Newman, K. 1998. State-space modeling of animal movement and mortality with application to salmon. Biometrics 54:274–297.
Ospina, R., F. Cribari-Neto, and K. L. P. Vasconcellos. 2006. Improved point and interval estimation for a beta regression model. Computational Statistics and Data Analysis 51:960–981.
Pawitan, Y. 2001. In all likelihood: statistical modeling and inference using likelihood. Oxford University Press, Oxford, UK.
Peterjohn, B. G. 1994. The North American breeding bird survey. Birding 26:386–398.
Pollard, E., K. H. Lakhani, and P. Rothery. 1987. The detection of density dependence from a series of annual censuses. Ecology 68:2046–2055.
Rehfeldt, G. E., N. L. Crookston, M. V. Warwell, and J. S. Evans. 2006. Empirical analyses of plant–climate relationships for the western United States. International Journal of Plant Sciences 167:1123–1150.
Sakamoto, Y., M. Ishiguro, and G. Kitagawa. 1986. Akaike information criterion statistics. KTK Scientific, Tokyo, Japan.
Searle, S. R., G. Casella, and C. E. McCulloch. 1992. Variance components. John Wiley and Sons, New York, New York, USA.
Shenk, T. M., G. C. White, and K. P. Burnham. 1998. Sampling-variance effects on detecting density dependence from temporal trends in natural populations. Ecological Monographs 68:445–463.
Sibly, R. M., D. Barker, M. C. Denham, J. Hone, and M. Pagel. 2005. On the regulation of populations of mammals, birds, fish, and insects. Science 309:607–610.
Solow, A. R. 1990. Testing for density dependence: a cautionary note. Oecologia 83:47–49.
Solow, A. R. 2001. Observation error and the detection of delayed density dependence. Ecology 82:3263–3264.
Spearpoint, J. A., B. Every, and L. G. Underhill. 1988. Waders (Charadrii) and other shorebirds at Cape Recife, Algoa Bay, South Africa: seasonality, trends, conservation, and reliability of surveys. Ostrich 59:166–177.
Staples, D. F., M. L. Taper, and B. Dennis. 2004. Estimating population trend and process variation for PVA in the presence of sampling error. Ecology 85:923–929.
Sullivan, P. J. 1992. A Kalman filter approach to catch-at-length analysis. Biometrics 48:237–257.
Ver Hoef, J. M., and K. J. Frost. 2003. A Bayesian hierarchical model for monitoring harbor seal changes in Prince William Sound, Alaska. Environmental and Ecological Statistics 10: 201–219.
Woiwod, I. P., and I. Hanski. 1992. Patterns of density dependence in moths and aphids. Journal of Animal Ecology 61:619–629.
Zeng, Z., R. M. Nowierski, M. L. Taper, B. Dennis, and W. P. Kemp. 1998. Complex population dynamics in the real world: modeling the influence of time-varying parameters and time lags. Ecology 79:2193–2209.

## APPENDIX

Multivariate-normal likelihood function for replicated sampling (*Ecological Archives* E091-044-A1).

## SUPPLEMENT

R program to calculate maximum-likelihood or restricted maximum-likelihood estimates of model parameters, as well as to calculate and plot profile log-likelihoods, for the Gompertz state-space models with replicated sampling (*Ecological Archives* E091-044-S1).

**Brian Dennis, José Miguel Ponciano, Mark L. Taper. 2010. Replicated sampling increases efficiency in monitoring biological populations.** ***Ecology*** **91:610-620.**

***Ecological Archives*** **E091-044**

Appendix A. Multivariate-normal likelihood function for replicated sampling.

Here we construct the multivariate-normal likelihood function for replicated sampling under the Gompertz state-space (GSS) model, for the stationary case. The likelihood function allows maximum-likelihood (ML) estimation of model parameters through numerical maximization. In addition, we obtain a multivariate-normal likelihood for use in obtaining restricted maximum-likelihood (REML) estimates with transformed observations.

We assume the sampling process is replicated $p_t$ times at sampling time $t$, producing observations $Y_{1t}$, $Y_{2t}$, ..., $Y_{p_t t}$. Denote by $\mathbf{Y}_t$ the $p_t \times 1$ column vector [$Y_{1t}$, $Y_{2t}$, ..., $Y_{p_t t}$]′ of the observations (as random variables) at time $t$, and denote by $\mathbf{y}_t$ the $p_t \times 1$ column vector [$y_{1t}$, $y_{2t}$, ..., $y_{p_t t}$]′ of the recorded outcomes (data values) of the random variables in the vector $\mathbf{Y}_t$ at time $t$. We write **j** for a column vector of ones, **o** for a column vector of zeros, **J** for a matrix of ones, **O** for a matrix of zeros, and **I** for an identity matrix, with the sizes determined by context (or clarified with subscripts, when necessary) so as to be consistent with the matrix operations. The GSS model consists of the underlying population process joined with the multivariate sampling process:

$$X_t = a + cX_{t-1} + E_t$$

$$\mathbf{Y}_t = \mathbf{j}X_t + \mathbf{F}_t$$

where $E_t \sim$ normal(0, $\sigma^2$) and $\mathbf{F}_t \sim$ MVN($\mathbf{o}_t$, $\tau^2 \mathbf{I}$), with $\mathbf{F}_t$ assumed independent of $E_t$ and $X_t$, and no autocorrelation of the noise processes $E_t$ and $\mathbf{F}_t$. Also, denote by $\mathbf{X}$ the $(q+1)\times 1$ column vector $[X_0, X_1, \ldots, X_q]'$, and denote by $\mathbf{Y}$ the $r \times 1$ column vector ($r = p_0 + p_1 + \cdots + p_q$) formed by stacking the vectors $\mathbf{Y}_0$, $\mathbf{Y}_1$, …, $\mathbf{Y}_q$. Similarly denote by $\mathbf{F}$ the $r \times 1$ column vector formed by stacking the vectors $\mathbf{F}_0$, $\mathbf{F}_1$, …, $\mathbf{F}_q$.

*ML estimation*

The essential idea is that $\mathbf{X}$ has a multivariate-normal distribution, and $\mathbf{Y}$ is the sum of $\mathbf{F}$ and a linear transformation of $\mathbf{X}$. First, a well-known property of the stationary AR(1) process $X_t$ is that

$$\mathbf{X} \sim \text{MVN}(\mathbf{j}a\,(1-c)\,,\,\mathbf{\Sigma})$$

with all the main diagonal elements of the variance--covariance matrix $\mathbf{\Sigma}$ equal to the stationary variance $\text{V}(X_t) = \sigma^2/(1-c^2)$, and the other elements giving the stationary covariances $\text{CV}(X_t, X_{t+s}) = |c|^s \sigma^2/(1-c^2)$ (see for instance, Dennis et al. 2006, Harvey 1993). Second, let a matrix $\mathbf{C}$ be defined by stacking column vectors of ones and zeros in the following manner:

$$\mathbf{C} = \begin{bmatrix} \mathbf{j}_0 & \mathbf{o}_0 & \cdots & \mathbf{o}_0 \\ \mathbf{o}_1 & \mathbf{j}_1 & \cdots & \mathbf{o}_1 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{o}_q & \mathbf{o}_q & \cdots & \mathbf{j}_q \end{bmatrix}.$$

Here $\mathbf{j}_t$ and $\mathbf{o}_t$ are $p_t \times 1$ (the size of $\mathbf{y}_t$). One can see that $\mathbf{Y}$ is a linear transformation of $\mathbf{X}$:

$$\mathbf{Y} = \mathbf{C}\mathbf{X} + \mathbf{F}.$$

Therefore,

$$\mathbf{Y} \sim \text{MVN}(\ \mathbf{C}\mathbf{j}\, a/(1-c)\ ,\ \mathbf{C}\boldsymbol{\Sigma}\mathbf{C}' + \tau^2\,\mathbf{I}\ )\ .$$

The log-likelihood function for a vector of data **y** thus is the log-probability density for a multivariate-normal distribution:

$$\ln L\left(a, c, \sigma^2, \tau^2\right) = -\frac{r}{2}\ln(2\pi) - \frac{1}{2}\ln|\mathbf{V}| - \frac{1}{2}\,(\mathbf{y}-\boldsymbol{\mu})'\,\mathbf{V}^{-1}\,(\mathbf{y}-\boldsymbol{\mu})\ ,$$

where $\mathbf{V} = \mathbf{C}\boldsymbol{\Sigma}\mathbf{C}' + \tau^2\,\mathbf{I}$, and $\boldsymbol{\mu}$ is a $r \times 1$ vector with all elements equal to $a/(1-c)$.

*REML estimation*

A REML transformation of the observations in replicated sampling can be defined as follows. Multiply each element in the vector $\mathbf{y}_t$, ($t$ = 1, 2,…, $q$) by $p_{t-1}$ (the size of the previous vector), and then subtract the sum of the elements in the previous vector $\mathbf{y}_{t-1}$, that is,

$$w_{i\,t} = p_{t-1}y_{i\,t} - \left(y_{1\,t-1} + y_{2\,t-1} + \ldots + y_{p_{t-1}\,t-1}\right).$$

Because all the observations have the same mean (the stationary mean $a/(1-c)$), each $w_{i\,t}$ arises from a distribution with mean zero. Let $\mathbf{w}_t = [\,w_{1\,t}\,,\ w_{2\,t}\,,\ \ldots,\ w_{p_t\,t}\,]'$. Then $\mathbf{w}_t$ can be represented by

$$\mathbf{w}_t = p_{t-1}\,\mathbf{y}_t - \mathbf{j}_t\,\mathbf{j}_{t-1}'\,\mathbf{y}_{t-1}\ .$$

Furthermore, let **w** be the vectors $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_q$ stacked into a column vector, and denote by **W** the random vector version (of which **w** is a particular realization). Using the transformation matrix given by

$$\mathbf{D} = \begin{bmatrix} -\mathbf{J}_{10} & p_0\mathbf{I}_{11} & \mathbf{O}_{12}\cdots & \mathbf{O}_{1q} \\ \mathbf{O}_{20} & -\mathbf{J}_{21} & p_1\mathbf{I}_{22}\cdots & \mathbf{O}_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{O}_{q0} & \mathbf{O}_{q1} & \cdots & p_{q-1}\mathbf{I}_{qq} \end{bmatrix}$$

where $\mathbf{J}_{st}$, $\mathbf{O}_{st}$ and $\mathbf{I}_{st}$ are $p_s \times p_t$ matrices, we establish that $\mathbf{W} = \mathbf{DY}$ is a $(r - p_0) \times 1$ vector that has a multivariate-normal distribution with a mean vector of zero and a variance--covariance matrix given by $\mathbf{\Phi} = \mathbf{D}[\mathbf{C\Sigma C}' + \tau^2 \mathbf{I}]\,\mathbf{D}' = \mathbf{DVD}'$. The restricted log-likelihood function for **w** is then:

$$\ln L(c, \sigma^2, \tau^2) = -\frac{r - p_0}{2} \ln(2\pi) - \frac{1}{2} \ln|\mathbf{\Phi}| - \frac{1}{2} \mathbf{w}' \mathbf{\Phi}^{-1} \mathbf{w} .$$

The parameter *a* does not appear in the restricted log-likelihood. The restricted log-likelihood is maximized numerically over the values of the parameters *c*, $\sigma^2$, and $\tau^2$. The parameter *a* is then estimated as

$$a = (1 - c)\frac{\mathbf{j}'\mathbf{V}^{-1}\mathbf{y}}{\mathbf{j}'\mathbf{V}^{-1}\mathbf{j}}$$

where everything on the right-hand side of the equation is evaluated at the REML estimates for *c*, $\sigma^2$, and $\tau^2$.

## LITERATURE CITED

Dennis, B., J. M. Ponciano, S. R. Lele, M. L. Taper, and D. F. Staples. 2006. Estimating density dependence, process noise, and observation error. Ecological Monographs 76:323–341.

Harvey, A. C. 1993. Time series models. Second edition. MIT Press, Cambridge, Massachusetts, USA.

**Brian Dennis, José Miguel Ponciano, and Mark L. Taper. 2010. Replicated sampling increases efficiency in monitoring biological populations. *Ecology* 91:610–620.**

---

## Supplement

**R program to calculate maximum-likelihood or restricted maximum-likelihood estimates of model parameters, as well as to calculate and plot profile log-likelihoods, for the Gompertz state-space model with replicated sampling. *Ecological Archives* E091-044-S1.**



---

Authors
File list (downloads)
Description

---

## Author(s)

Brian Dennis
Department of Fish and Wildlfe Resources and Department of Statistics
University of Idaho
Moscow, Idaho 83844-1136 USA
E-mail: brian@uidaho.edu

José Miguel Ponciano
Centro de Investigación en Matemáticas, CIMAT A. C. Calle Jalisco s/n
Col. Valenciana, A.P. 402
C.P. 36240 Guanajuato, Guanajuato, México

Mark L. Taper
Department of Ecology
Montana State University
301 Lewis Hall
Bozeman, Montana 59717-3460

---

## File list

Dennis_etal_Gompertz_state_space_model_with_replicated_sampling.R

## Description

The computer program, in the open-source R language (R Core Development Team. 2006. R: a

language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria), uses numerical maximization of a multivariate-normal log-likelihood to compute maximum-likelihood and restricted maximum-likelihood parameter estimates for the Gompertz state-space model of density dependent population growth, for time series abundance data with replicated samples at each observation time. The program also computes and plots profile log-likelihoods for the four model parameters. The data included in the program as an example are the simulated observations appearing in the top panel of Fig. 1 of the main article, and the program was used to produce Fig. 2 in the same paper.

`Dennis_etal_Gompertz_state_space_model_with_replicated_sampling.R` contains annotated code for fitting a stochastic, density-dependent population growth model to time series abundance data in which samples have been replicated at each sampling time.

---

```
#           GOMPERTZ STATE SPACE WITH REPLICATED SAMPLING
#
#  GSS-RS model:  program to calculate ML or REML estimates & profile
#  log-likelihoods.  The program assumes an equal number (p) of
#  sampling replicates per sampling occasion.
#
#  The model is:
#       X(t) = a + c*X(t-1) + E(t),
#       Y(t) = j*X(t) + F(t).
#  Here:
#       X(t) is log-abundance of a population at time t
#         (t=0,1,2,...,q).
#       E(t) ~ normal with mean 0, variance ssq, and no
#          autocorrelation.
#       Y(t) is a pX1 vector of estimates of X(t) obtained by
#          replicated sampling.
#       j is a pX1 vector of ones.
#       F(t) ~ multivariate normal with mean vector 0 (pX1) and var-cov
#          matrix tsq*I, where I is a pXp identity matrix.
#       a>0, ssq>0, tsq>0, -1<c<+1 are model parameters.
#
#  Be patient.  R is slow.

#-----------------------------------------------------------------------
#        USER INPUT SECTION
#-----------------------------------------------------------------------
#  User must supply initial parameter values here.
a0=.4;             #  Value of "a" to be used.
c0=.8;             #  Value of "c" to be used.
ssq0=.1;            #  Value of "ssq" to be used.
tsq0=.2;            #  Value of "tsq" to be used.

#  User specifies number of sampling replicates here.
p=2;     #  Number of replicates

#  User provides data here.  Statements can be altered to input data
#  from a file.  Result must be a pX(q+1) matrix YP.t, containing
#  replicated observations (log-scale) as columns.
O.t=c(24,24,9,14,10,18,7,9,9,13,13,10,5,3,13,4,10,4,5,7,3,10,5,9,
  17,8,22,19,5,10,5,5,3,4,1,2,3,5,7,6,7,11,8,8,5,7,4,3,9,15,7,9,
  14,24,3,6,6,7,11,10,8,9);  #  (24,24) are the observations at
                             #    time 0, (9,14) are the observations
                             #    at time 1, etc.
qplus1=length(O.t)/p;
q=qplus1-1;
YP.t=matrix(log(O.t),p,qplus1,byrow=FALSE);  #  Columns are replicated
                                             #    samples.

#  User sets parameter intervals for profile plots and total
#  number of horizontal axis increments here.
aalo=0.4;     #  Low value of "a"
aahi=1.9;     #  High value of "a", etc.
cclo=0.1;
cchi=0.8;
ssqlo=0.1;
ssqhi=0.4;
tsqlo=0.1;
tsqhi=0.21;
nincs=100;   #  Number of increments for the profile plots.
```

```
#-----------------------------------------------------------------------
#        PROGRAM INITIALIZATION SECTION
#-----------------------------------------------------------------------
library(MASS);  #  loads miscellaneous functions (ginv, etc.).

#  Sets parameter values for profile likelihoods.
aavals=seq(aalo,aahi,by=((aahi-aalo)/nincs));
ccvals=seq(cclo,cchi,by=((cchi-cclo)/nincs));
ssqvals=seq(ssqlo,ssqhi,by=((ssqhi-ssqlo)/nincs));
tsqvals=seq(tsqlo,tsqhi,by=((tsqhi-tsqlo)/nincs));

#  These vectors will eventually hold the profiles.
profileRS.aa=aavals;
profileRS.cc=ccvals;
profileRS.ssq=ssqvals;
profileRS.tsq=tsqvals;

#  Matrices for calculating multivariate normal likelihood for REML
#  estimates.
YP.reml=matrix(YP.t,p*qplus1,1);  #  Replicated samples "stacked" in a vector.
J.p=matrix(1,p,p);          #  pXp matrix of ones.
I.p=diag(qplus1);           #  (q+1)X(q+1) identity matrix.
D.reml=kronecker(I.p,J.p);  #  kronecker product (block diagonal).
I.temp=cbind(matrix(0,q*p,p),diag(q*p));
I.temp=rbind(I.temp,matrix(0,p,p*qplus1));
D.reml=-D.reml+2*I.temp;
D.reml=D.reml[1:(q*p),];    #  This is the D transformation matrix
                            #   for REML.
WP.t=D.reml%*%YP.reml;      #  The REML-transformed observations.
j.p=matrix(1,p,1);          #  pX1 column vector of ones.
j.pXqp1=matrix(1,p*qplus1,1);  #  p*(q+1) X 1 col;umn vector of ones.
C.ml=kronecker(I.p,j.p);    #  C matrix in the multivariate normal
                            #    distribution of YP.reml.

#  Log-likelihood for ML estimation.
negloglikeRS.ml=function(theta,yt)
{
   p=nrow(yt);
   aa=exp(theta[1]);              #  Constrains a > 0.
   cc=2*exp(-exp(theta[2]))-1;    #  Constrains -1 < c < 1.
   sigmasq=exp(theta[3]);         #  Constrains ssq > 0.
   tausq=exp(theta[4]);           #  Constrains tsq > 0.
   q=ncol(yt)-1;
   jj=matrix(1,p,1);
   JJ=matrix(1,p,p);
   Itausq=matrix(0,p,p);
   diag(Itausq)=rep(tausq);
   mu.t=aa/(1-cc);
   psq.t=sigmasq/(1-cc*cc);
   mm.t=jj*mu.t;
   VV.t=JJ*psq.t+Itausq;
   VVinv=ginv(VV.t);
   lnf=(p/2)*log(2*pi)+0.5*log(det(VV.t))+
      0.5*(yt[,1]-t(mm.t))%*%VVinv%*%t(yt[,1]-t(mm.t));
   ofn=lnf;
   for (tt in 1:q)
   {
      mu.t=aa+cc*(mu.t+psq.t*t(jj)%*%VVinv%*%t(yt[,tt]-t(mm.t)));
      psq.t=cc*cc*psq.t*(1-psq.t*t(jj)%*%VVinv%*%jj)+sigmasq;
      mm.t=jj*mu.t[1];
```

```
      VV.t=JJ*psq.t[1]+Itausq;
      VVinv=ginv(VV.t);
      lnftemp=(p/2)*log(2*pi)+0.5*log(det(VV.t))+
         0.5*(yt[,tt+1]-t(mm.t))%*%ginv(VV.t)%*%t(yt[,tt+1]-t(mm.t));
      ofn=ofn+lnftemp;
   }
   return(ofn);
}

#  Log-likelihoods for profiles:  first the parameter "a" is
#    fixed, then "c" is fixed, and so on.

#  "a" is a vector of fixed values.
negloglikeRS.a.ml=function(theta,parval,yt)
{
   p=nrow(yt);
   aa=parval;
   cc=2*exp(-exp(theta[1]))-1;   #  Constrains -1 < c < 1.
   sigmasq=exp(theta[2]);      #  Constrains ssq > 0.
   tausq=exp(theta[3]);        #  Constrains tsq > 0.
   q=ncol(yt)-1;
   jj=matrix(1,p,1);
   JJ=matrix(1,p,p);
   Itausq=matrix(0,p,p);
   diag(Itausq)=rep(tausq);
   mu.t=aa/(1-cc);
   psq.t=sigmasq/(1-cc*cc);
   mm.t=jj*mu.t;
   VV.t=JJ*psq.t+Itausq;
   VVinv=ginv(VV.t);
   lnf=(p/2)*log(2*pi)+0.5*log(det(VV.t))+
      0.5*(yt[,1]-t(mm.t))%*%VVinv%*%t(yt[,1]-t(mm.t));
   ofn=lnf;
   for (tt in 1:q)
   {
      mu.t=aa+cc*(mu.t+psq.t*t(jj)%*%VVinv%*%t(yt[,tt]-t(mm.t)));
      psq.t=cc*cc*psq.t*(1-psq.t*t(jj)%*%VVinv%*%jj)+sigmasq;
      mm.t=jj*mu.t[1];
      VV.t=JJ*psq.t[1]+Itausq;
      VVinv=ginv(VV.t);
      lnftemp=(p/2)*log(2*pi)+0.5*log(det(VV.t))+
         0.5*(yt[,tt+1]-t(mm.t))%*%ginv(VV.t)%*%t(yt[,tt+1]-t(mm.t));
      ofn=ofn+lnftemp;
   }
   return(ofn);
}

#  "c" is a vector of fixed values
negloglikeRS.c.ml=function(theta,parval,yt)
{
   p=nrow(yt);
   aa=exp(theta[1]);           #  Constrains a > 0.
   cc=parval;
   sigmasq=exp(theta[2]);      #  Constrains ssq > 0.
   tausq=exp(theta[3]);        #  Constrains tsq > 0.
   q=ncol(yt)-1;
   jj=matrix(1,p,1);
   JJ=matrix(1,p,p);
   Itausq=matrix(0,p,p);
   diag(Itausq)=rep(tausq);
```

```
   mu.t=aa/(1-cc);
   psq.t=sigmasq/(1-cc*cc);
   mm.t=jj*mu.t;
   VV.t=JJ*psq.t+Itausq;
   VVinv=ginv(VV.t);
   lnf=(p/2)*log(2*pi)+0.5*log(det(VV.t))+
      0.5*(yt[,1]-t(mm.t))%*%VVinv%*%t(yt[,1]-t(mm.t));
   ofn=lnf;
   for (tt in 1:q)
   {
      mu.t=aa+cc*(mu.t+psq.t*t(jj)%*%VVinv%*%t(yt[,tt]-t(mm.t)));
      psq.t=cc*cc*psq.t*(1-psq.t*t(jj)%*%VVinv%*%jj)+sigmasq;
      mm.t=jj*mu.t[1];
      VV.t=JJ*psq.t[1]+Itausq;
      VVinv=ginv(VV.t);
      lnftemp=(p/2)*log(2*pi)+0.5*log(det(VV.t))+
         0.5*(yt[,tt+1]-t(mm.t))%*%ginv(VV.t)%*%t(yt[,tt+1]-t(mm.t));
      ofn=ofn+lnftemp;
   }
   return(ofn);
}

#  "sigma-squared" is a vector of fixed values.
negloglikeRS.ssq.ml=function(theta,parval,yt)
{
   p=nrow(yt);
   aa=exp(theta[1]);             #  Constrains a > 0.
   cc=2*exp(-exp(theta[2]))-1;   #  Constrains -1 < c < 1.
   sigmasq=parval;
   tausq=exp(theta[3]);          #  Constrains tsq > 0.
   q=ncol(yt)-1;
   jj=matrix(1,p,1);
   JJ=matrix(1,p,p);
   Itausq=matrix(0,p,p);
   diag(Itausq)=rep(tausq);
   mu.t=aa/(1-cc);
   psq.t=sigmasq/(1-cc*cc);
   mm.t=jj*mu.t;
   VV.t=JJ*psq.t+Itausq;
   VVinv=ginv(VV.t);
   lnf=(p/2)*log(2*pi)+0.5*log(det(VV.t))+
      0.5*(yt[,1]-t(mm.t))%*%VVinv%*%t(yt[,1]-t(mm.t));
   ofn=lnf;
   for (tt in 1:q)
   {
      mu.t=aa+cc*(mu.t+psq.t*t(jj)%*%VVinv%*%t(yt[,tt]-t(mm.t)));
      psq.t=cc*cc*psq.t*(1-psq.t*t(jj)%*%VVinv%*%jj)+sigmasq;
      mm.t=jj*mu.t[1];
      VV.t=JJ*psq.t[1]+Itausq;
      VVinv=ginv(VV.t);
      lnftemp=(p/2)*log(2*pi)+0.5*log(det(VV.t))+
         0.5*(yt[,tt+1]-t(mm.t))%*%ginv(VV.t)%*%t(yt[,tt+1]-t(mm.t));
      ofn=ofn+lnftemp;
   }
   return(ofn);
}

#  "tau-squared" is a vector of fixed values
negloglikeRS.tsq.ml=function(theta,parval,yt)
{
```

```
   p=nrow(yt);
   aa=exp(theta[1]);             #  Constrains a > 0.
   cc=2*exp(-exp(theta[2]))-1;   #  Constrains -1 < c < 1.
   sigmasq=exp(theta[3]);        #  Constrains ssq > 0.
   tausq=parval;
   q=ncol(yt)-1;
   jj=matrix(1,p,1);
   JJ=matrix(1,p,p);
   Itausq=matrix(0,p,p);
   diag(Itausq)=rep(tausq);
   mu.t=aa/(1-cc);
   psq.t=sigmasq/(1-cc*cc);
   mm.t=jj*mu.t;
   VV.t=JJ*psq.t+Itausq;
   VVinv=ginv(VV.t);
   lnf=(p/2)*log(2*pi)+0.5*log(det(VV.t))+
      0.5*(yt[,1]-t(mm.t))%*%VVinv%*%t(yt[,1]-t(mm.t));
   ofn=lnf;
   for (tt in 1:q)
   {
      mu.t=aa+cc*(mu.t+psq.t*t(jj)%*%VVinv%*%t(yt[,tt]-t(mm.t)));
      psq.t=cc*cc*psq.t*(1-psq.t*t(jj)%*%VVinv%*%jj)+sigmasq;
      mm.t=jj*mu.t[1];
      VV.t=JJ*psq.t[1]+Itausq;
      VVinv=ginv(VV.t);
      lnftemp=(p/2)*log(2*pi)+0.5*log(det(VV.t))+
         0.5*(yt[,tt+1]-t(mm.t))%*%ginv(VV.t)%*%t(yt[,tt+1]-t(mm.t));
      ofn=ofn+lnftemp;
   }
   return(ofn);
}

#  Log-likelihood for REML estimation.
negloglikeRS.reml=function(theta,wt,pt)
{
   cc=2*exp(-exp(theta[1]))-1;   #  Constrains -1 < c < 1.
   sigmasq=exp(theta[2]);        #  Constrains ssq > 0.
   tausq=exp(theta[3]);          #  Constrains tsq > 0.
   q=length(wt)/pt;
   qp1=q+1;
   Sigma.mat=(sigmasq/(1-cc*cc))*toeplitz(abs(cc)^seq(0,q,1));
   J=matrix(1,pt,pt);          #  pXp matrix of ones.
   I=diag(qp1);           #  (q+1)X(q+1) identity matrix.
   D=kronecker(I,J);     #  kronecker product (block diagonal).
   I.2=cbind(matrix(0,q*pt,pt),diag(q*pt));
   I.2=rbind(I.2,matrix(0,pt,pt*qp1));
   D=-D+2*I.2;
   D=D[1:(q*pt),];  #  This is the D transformation matrix for REML.
   j=matrix(1,pt,1);          #  pX1 column vector of ones.
   j.2=matrix(1,pt*qp1,1);  #  p*(q+1) X 1 column vector of ones.
   C=kronecker(I,j);    #  C matrix in the multivariate normal
                        #    distribution of YP.reml.
   V=C%*%Sigma.mat%*%t(C)+tausq*diag(qp1*pt);  #  Var-cov matrix of
                                               #    YP.reml.
   Phi.mat=D%*%V%*%t(D);    #  Var-cov matrix of WP.t.
   Phiinv.mat=ginv(Phi.mat);
   llikew=-(p*q/2)*log(2*pi)-(1/2)*log(det(Phi.mat))-
      (1/2)*t(wt)%*%Phiinv.mat%*%wt;  #  REML log-likelihood.
   ofn=-llikew;
   return(ofn);
```

```
}

#-----------------------------------------------------------------------
#        SECTION FOR CALCULATING PROFILE LIKELIHOODS
#-----------------------------------------------------------------------

for (ii in 1:(nincs+1))
{
#  Calculate profile for "a".
GSSRSaa=optim(par=c(log(-log((c0+1)/2)),log(ssq0),log(tsq0)),
   negloglikeRS.a.ml,NULL,method="Nelder-Mead",parval=aavals[ii],yt=YP.t);
profileRS.aa[ii]=-GSSRSaa$value;

#  Calculate profile for "c".
GSSRScc=optim(par=c(log(a0),log(ssq0),log(tsq0)),
   negloglikeRS.c.ml,NULL,method="Nelder-Mead",parval=ccvals[ii],yt=YP.t);
profileRS.cc[ii]=-GSSRScc$value;

#  Calculate profile for "ssq".
GSSRSssq=optim(par=c(log(a0),log(-log((c0+1)/2)),log(tsq0)),
   negloglikeRS.ssq.ml,NULL,method="Nelder-Mead",parval=ssqvals[ii],yt=YP.t);
profileRS.ssq[ii]=-GSSRSssq$value;

#  Calculate profile for "tsq".
GSSRStsq=optim(par=c(log(a0),log(-log((c0+1)/2)),log(ssq0)),
   negloglikeRS.tsq.ml,NULL,method="Nelder-Mead",parval=tsqvals[ii],yt=YP.t);
profileRS.tsq[ii]=-GSSRStsq$value;
}

#  Sets highest profile value at zero.
profileRS.aa=profileRS.aa-max(profileRS.aa);
profileRS.cc=profileRS.cc-max(profileRS.cc);
profileRS.ssq=profileRS.ssq-max(profileRS.ssq);
profileRS.tsq=profileRS.tsq-max(profileRS.tsq);

#  Profiles plotted here.
par(cex.lab=1.5, cex.axis=1.5, lwd=2);
layout(matrix(1:4, 2, 2));
plot(aavals, profileRS.aa, type="l", lty=1, ylab="profile
   log-likelihood", xlab=expression(a));
plot(ssqvals, profileRS.ssq, type="l", lty=1, ylab="profile
   log-likelihood", xlab=expression(sigma^2));
plot(ccvals, profileRS.cc, type="l", lty=1, ylab="profile
   log-likelihood", xlab=expression(c));
plot(tsqvals, profileRS.tsq, type="l", lty=1, ylab="profile
   log-likelihood",  xlab=expression(tau^2));


#-----------------------------------------------------------------------
#        SECTION FOR CALCULATING ML & REML ESTIMATES
#-----------------------------------------------------------------------

GSSRSml=optim(par=c(log(a0),log(-log((c0+1)/2)),log(ssq0),log(tsq0)),
   negloglikeRS.ml,NULL,method="Nelder-Mead",yt=YP.t);
GSSRSml=c(exp(GSSRSml$par[1]),2*exp(-exp(GSSRSml$par[2]))-
   1,exp(GSSRSml$par[3]),exp(GSSRSml$par[4]),-GSSRSml$value);
a.ml=GSSRSml[1];            # These are the ML estimates.
c.ml=GSSRSml[2];            #          --
ssq.ml=GSSRSml[3];          #          --
tsq.ml=GSSRSml[4];          #          --
```

```
lnlike.ml=GSSRSml[5];       #  This is the maximized log-likelihood
                            #    value.

GSSRSreml=optim(par=c(log(-log((c0+1)/2)),log(ssq0),log(tsq0)),
   negloglikeRS.reml,NULL,method="Nelder-Mead",wt=WP.t,pt=p);
GSSRSreml=c(2*exp(-exp(GSSRSreml$par[1]))-1,exp(GSSRSreml$par[2]),
   exp(GSSRSreml$par[3]),-GSSRSreml$value);

c.reml=GSSRSreml[1];            #  These are the REML estimates.
ssq.reml=GSSRSreml[2];          #          --
tsq.reml=GSSRSreml[3];          #          --
lnlike.reml=GSSRSreml[4];       #  This is the maximized log-likelihood
                                #    value.

#  Calculate REML estimate of the parameter "a".
Sigma.mat=(ssq.reml/(1-c.reml*c.reml))*toeplitz(c.reml^seq(0,q,1));
V.mat=C.ml%*%Sigma.mat%*%t(C.ml)+tsq.reml*diag(qplus1*p);
Vinv.mat=ginv(V.mat);
a.reml=(1-c.reml)*t(j.pXqp1)%*%Vinv.mat%*%YP.reml/
   (t(j.pXqp1)%*%Vinv.mat%*%j.pXqp1);

#  Gather up stuff for printing.
estimates.ml=c(a.ml,c.ml,ssq.ml,tsq.ml,lnlike.ml);
estimates.reml=c(a.reml,c.reml,ssq.reml,tsq.reml,lnlike.reml);
names.ml=c("a.ml","c.ml","ssq.ml","tsq.ml","lnlike.ml");
names.reml=c("a.reml","c.reml","ssq.reml","tsq.reml","lnlike.reml");
values.ml=data.frame(names.ml,estimates.ml);
values.reml=data.frame(names.reml,estimates.reml);

#  Print the ML results.
values.ml;

#  Print the REML results.
values.reml;
```