# Gibbs Sampler for Multistate Life Tables Software (GSMLT v.90)

Scott M. Lynch[1]

Princeton University

J. Scott Brown

Miami University



[1]Department of Sociology and Office of Population Research, Princeton University, Princeton NJ 08544. THIS SOFTWARE IS A BETA VERSION. WE WOULD VERY MUCH APPRECIATE ANY COMMENTS/FEEDBACK. Email slynch@princeton.edu to obtain the two .c program code files.

# 1 Overview

The software described in this manual generates distributions of multistate life table quantities for a specified covariate profile. Following a Bayesian perspective on probability, these distributions allow the generation of interval estimates of these quantities as well hypothesis testing.

This software/code consists of two programs: One is for multivariate (bivariate) hazard model estimation; one is for mulstistate life table construction. Specifically, `mstatehazard.c` estimates a multivariate hazard model for generating smoothed transition probabilities, and `mstatetables.c` generates multistate life tables using the output from `mstatehazard.c`.

In this version of the software, the multivariate hazard program (and life table program) is limited to 2 states plus death. The number of covariates, all of which may interact with age and all of which may interact with the starting state (in order to allow for state-dependent covariate effects), are unlimited. The general approach is (1) estimate the hazard model using the hazard model program, (2) use the output file the hazard model program generates as input to the multistate life table program, (3) summarize the output from this program as you like.

In addition to this software, we also have software available for construction of single and multiple decrement life tables. However, that software cannot be distributed as easily, because it involves the use of some copyrighted algorithms from *Numerical Recipes in C* . Contact us directly if you are interested in these additional programs.

The estimation algorithm for the multivariate hazard model is a Gibbs sampler, and thus the output file generated by the program consists of a sample from the joint posterior distribution of the hazard model parameters. The life table program uses the output from the hazard program to generate a series of life tables, and it outputs the state expectancies from these tables to an ASCII file for subsequent use. The series of life tables arises from the process of applying basic multistate life table calculations to the samples from the distribution of the hazard model parameters: Each sample from the joint distribution for the parameters yields a unique life table. Technical details of this approach are discussed in Lynch and Brown (in press), as well as throughout this manual. However, we note that the method discussed in Lynch and Brown (in press) involved using Metropolis-Hastings (MH) sampling (hence, version .8), whereas this software relies exclusively on Gibbs sampling (version .9). Gibbs sampling is much more efficient and faster than the MH approach described in our earlier work.

This software is distributed freely, subject to one constraint: Any published or presented work using this software *must* include the following reference:

Lynch, Scott M. and J. Scott Brown. (in press). "A New Approach to Estimating Life Tables with Covariates and Constructing Interval Estimates of Life Table Quantities." *Sociological Methodology.*

This short manual consists of five following sections. Section 2 discusses the multistate hazard model and its estimation. Section 3 discusses multistate life table generation. Section 4 describes exactly how to use the software. Section 5 provides a detailed example

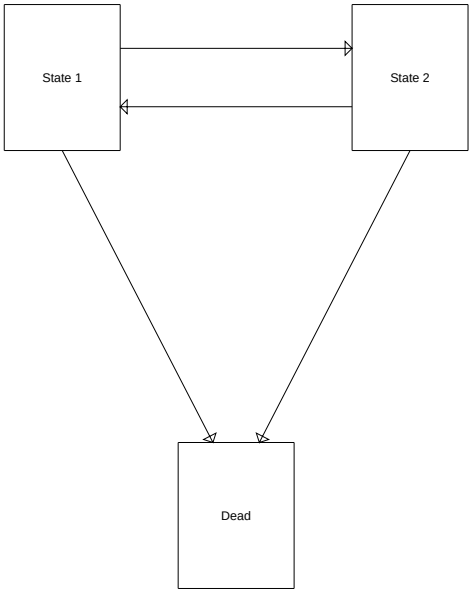


Figure 1: State Space for Two State Multistate Model

of the use of the software, including implementation and interpretation of results. Finally, Section 6 discusses some limitations of the software.

# 2 Hazard Model and Estimation

The hazard model state space can be viewed as in Figure 1. The hazard model estimated by the software is best-defined as a constrained multivariate probit model. Figure 2 depicts the multinomial space—in contingency-table format—involved in the multistate model. As an example, we depict a model estimating disability and death.

Here, there are two outcomes (ending states), $Y$ and $Z$, each of which is predicted by a linear combination of predictors $X$ and coefficients, $\beta$ for the equation for $Y$ and $\omega$ for the equation for $Z$ (note: one of these predictors must be the starting state). The model therefore can be written as:

$$p(Y_i = 1\ ,\ Z_i = 1) = \Phi_2(X_i^T\beta, X_i^T\omega, \rho),$$

where $\Phi_2$ is the cumulative standard bivariate normal distribution function with correlation $\rho$; in this model, $\rho$ is the error correlation between the equations for $Y$ and $Z$.

The model can be considered a constrained multivariate probit model, as opposed to a full multivariate probit model or a multinomial probit model for the following reason. In a true multinomial probit model, the outcome states are mutually exclusive; here the outcomes states are *not necessarily* mutually exclusive. That is, indiviuals may be assumed


Figure 2: Multinomial Space for Multistate Model

to die in a disabled state, and thus, it is possible for individuals to fall in any of the cells in the table. On the other hand, in a true multivariate probit model, individuals may fall in any cell of the table. However, the software allows the constraint to be imposed that individuals may be restricted, if dead, to *not* be considered disabled. This is a choice of the software user.

The program uses Gibbs sampling to generate samples from the joint posterior distribution of the model parameters. First, latent data are simulated for individuals (a step called 'data augmentation' in some literature), based on their observed outcome states and their covariate profiles. These latent trait distributions are truncated bivariate normal distributions, where the point of truncation is defined by the observed outcome states. For example, an individual who is observed alive and nondisabled at time 2 is assumed to possess a latent trait that falls below 0 in both dimensions of a bivariate normal distribution with mean vector $[X_i^T\beta\ X_i^T\omega]$ and covariance matrix $\Sigma$. In contrast, an individual who dies disabled is assumed to possess a latent trait that falls above 0 in both dimensions.

Once latent data have been simulated, the regression parameters are each simulated from their full conditional posterior distribution, which turn out each to be univariate normal. Finally, given the latent trait data and the regression parameters, the error covariance matrix is drawn from its full conditional distribution. This distribution turns out to be inverse Wishart. Here, we show the derivation of the conditional densities from which the regression parameters are sampled (not in the aforementioned article), as well as

the derivation of the distribution of the error covariance matrix.

The posterior distribution, conditional on latent simulated data for the observed outcome, is multivariate normal:

$$p(\beta, \omega, \Sigma \mid Y, Z) \propto$$
$$\prod_{i=1}^{n} \exp\left\{ -\frac{1}{2} \left[ (y_i - x_i^T\beta)\,(z_i - x_i^T\omega) \right] \begin{bmatrix} \frac{1}{1-\rho^2} & \frac{-\rho}{1-\rho^2} \\ \frac{-\rho}{1-\rho^2} & \frac{1}{1-\rho^2} \end{bmatrix} \begin{bmatrix} (y_i - x_i^T\beta) \\ (z_i - x_i^T\omega) \end{bmatrix} \right\}, \tag{1}$$

where $y_i$ and $z_i$ are the latent simulated scores, $\rho$ is the error correlation, and $\beta$ and $\omega$ are the vectors of regression coefficients for outcome state 1 and 2, respectively. In deriving the conditionals for each $\beta$ parameter, terms not involving $\beta$ can be extracted as irrelevant constants. Thus, $(z_i - x_i^T\omega)$ can be rewritten as $e_i(z)$. As another simplifying revision, for any particular $\beta$, say $\beta_k$, we can rewrite $(y_i - x_i^T\beta)$ as $(e_i(y)^* - \beta_k x_{ik})$, where $e_i(y)^*$ is the error term omitting the combination of the $\beta$ of interest with its associated $x$. After carrying-out the matrix multiplication, we have:

$$p(\beta, \omega, \Sigma \mid Y, Z) \propto$$
$$\prod_{i=1}^{n} \exp\left\{ \left[ -\frac{1}{2(1-\rho^2)} \right] \left[ (e_i(y)^* - \beta_k x_{ik})^2 - 2\rho(e_i(y)^* - \beta_k x_{ik}) e_i(z) + e_i(z)^2 \right] \right\}.$$

Given that $e_i(z)$ is constant with respect to $\beta_k$, using the fact that $e^{a+b} = e^a e^b$, we can eliminate $e_i(z)^2$ as an irrelevant constant. After expanding the quadratic and the other term involving $(e_i(y)^* - \beta_k x_{ik})$, we obtain (suppressing the conditioning on $\omega$, $Y$, and $Z$):

$$p(\beta_k) \propto \prod_{i=1}^{n} \exp\left\{ \left[ -\frac{1}{2(1-\rho^2)} \right] \right.$$
$$\left. \left[ e_i(y)^{*2} - 2e_i(y)^*\beta_k x_{ik} + (\beta_k x_{ik})^2 - 2\rho e_i(y)^* e_i(z) + 2\rho e_i(z)\beta_k x_{ik} \right] \right\}.$$

Here, $e_i(y)^{*2}$ and $-2\rho e_i(y)^* e_i(z)$ are constant with respect to $\beta_k$ and can be removed. After rearranging terms, we have:

$$p(\beta_k) \propto \prod_{i=1}^{n} \exp\left\{ \left[ -\frac{1}{2(1-\rho^2)} \right] \left[ (\beta_k x_{ik})^2 - 2(e_i(y)^* - \rho e_i(z))(\beta_k x_{ik}) \right] \right\}.$$

At this point, we need to take the product over all observations; doing so yields:

$$p(\beta_k) \propto \exp\left\{\left[-\frac{1}{2(1-\rho^2)}\right]\left[\left(\sum_{i=1}^{n} x_{ik}^2\right)\beta_k^2 - 2\beta_k \sum_{i=1}^{n}\left(e_i(y)^* - \rho e_i(z)\right)x_{ik}\right]\right\}.$$

If we divide this interior expression by $(\sum_{i=1}^{n} x_{ik}^2)$ to isolate $\beta_k^2$ and complete the square in $\beta_k$ we obtain:

$$p(\beta_k) \propto \exp\left\{\left[-\frac{1}{2(1-\rho^2)/\sum_{i=1}^{n} x_{ik}^2}\right]\left(\beta - \frac{\sum_{i=1}^{n}\left(e_i(y)^* - \rho e_i(z)\right)x_{ik}}{\sum_{i=1}^{n} x_{ik}^2}\right)^2\right\}.$$

Thus, the full conditional distribution for $\beta_k$ is recognizable as normal:

$$\beta_k \sim N\left(\frac{\sum_{i=1}^{n}\left(e_i(y)^* - \rho e_i(z)\right)x_{ik}}{\sum_{i=1}^{n} x_{ik}^2},\ \frac{(1-\rho^2)}{\sum_{i=1}^{n} x_{ik}^2}\right)$$

A similar result can be obtained for each $\beta$ and each $\omega$. Our Gibbs sampling routine for drawing each regression coefficient samples the coefficients sequentially, updating each regression coefficient one at a time.

Once the regression parameters have been selected, the error covariance matrix can be updated. Returning to Equation 1, we can rewrite this posterior as:

$$p(\Sigma) \propto exp\left\{(-1/2)tr(S\Sigma^{-1})\right\},$$

where $S = (1/v)\sum_{i=1}^{n}[e(y)_i\ e(z)_i]^T[e(y)_i\ e(z)_i]$ is the observed error covariance matrix obtained under a particular set of regresion parameters. This is recognizable as an inverse wishart form: $\Sigma \sim IW(S, v)$, with $v$ degrees of freedom (*conditionally* on fixed values of $\beta$ and $\omega$, $v = n - 1$).

Once we have selected a draw for the covariance matrix of the errors, given that the latent data are assumed to have a variance of 1 as an identification constraint, we simply divide the error covariance by the square root of the newly sampled error variance elements to obtain the error correlation. (The error correlation and error covariance are identical if the variances are standardized).

# 3 Multistate Life Table Generation

The life table calculations performed in the life table program are very basic multistate calculations. We use a linear assumption for person-years lived in a state in a given age interval (individuals who transition do so in the middle of an interval). This assumption becomes less tenable as the width of the age interval becomes larger, but we have obtained

reasonable results—consistent with others' estimates—with this assumption. We carry-out our life table calculations to age 105, and we assume that the remaining individuals die at the end of the next age interval (thus, $l^{\omega} = 0$).

Thus, our flow equation for the number of persons, $l$, in state $i$ at age $a$ is:

$$l_i^a = l_i^{a-m} + l_{(-i)i}^{(a-m),a} - l_{i(-i)}^{(a-m),a},$$

where $m$ is the measured time interval between ages in the data, $l_{(-i)i}^{(a-m),a}$ is the number of individuals who transition into state $i$ from another state in the interval $[(a-m), a]$, and $l_{i(-i)}^{(a-m),a}$ is the number of individuals who transition out of state $i$ over the interval (either transitioning to the other state or to death).

Our calculation for the number of person years, $L$, lived in each state between age $a$ and $a+m$ is:

$$L_i^{a,(a+m)} = \frac{l_i^a + l_i^{(a+m)}}{2}.$$

# 4 Using the Software

In general, here are the steps you should follow in using the software for analyses:

1. Compile the programs.
2. Construct your data appropriately.
3. Run the hazard model program.
4. Monitor the output of the hazard program to determine whether you have run the program long enough.
5. Run the life table program.
6. Conduct whatever analyses you like with the output.

## 4.1 Compiling the software

The software is fairly flexible, but the requirements for the input data and command line arguments, etc. are tedious, making it seem not-so-user-friendly. The software code is written in unix-based C and must be compiled on the user's platform in order to run (the executable files are NOT platform-independent). The following steps will compile the programs:

1. On the unix command line in the directory in which the .c files are located, type

```
gcc -lm mstatehazard.c -o mstatehazard.x
```

"gcc" calls the gnu compiler. "-lm" attaches the math library. "-o" is required to name the output (executable) file. This step compiles the hazard model program and creates the executable file.

2. To compile the life table program, type

```
gcc -lm mstatetables.c -o mstatetables.x
```

Once these programs are compiled, they are ready to be used. They are used in the order in which they were compiled: First, the hazard model must be estimated; then the life tables are constructed using the output of the hazard model program.

## 4.2 Constructing data appropriately

In order for us to make the programs flexible enough to accomodate various user needs, it was necessary to develop the programs to take a large number of command-line arguments. These arguments *must* be made in the order discussed in the subsequent sections, and *no arguments can be omitted.* Furthermore, the data must be structured in a very specific way in order for the programs to read it correctly and function correctly in general.

The data must be structured as a free-field ASCII file with spaces as the delimiters between variables. Each row represents a case in the data [1] This can be done quite easily using the "put" statement in SAS. Alternatively, using STATA, the "outfile" command can be used (with some non-default specifications). The order of the variables must be as follows:

1. A column of ones (for the intercept).
2. **Age.** Regardless of the actual starting age, the first age must be coded 0. Thus, if your first actual age is 20, then subtract 20 from every individual's age in the data. If you are using 5-year age intervals, they should be numbered sequentially starting at 0.
3. **Starting State.** As it stands, individuals can be in one of two states at the starting point. These must be coded 0 and 1.
4. **Age×Starting State interaction.** This variable is the multiple of the *adjusted* age variable by the starting state.
5. **Additional Covariates.** The number of these is unlimited, and they may follow any reasonable coding scheme that you would use in any other regression analysis. If any of these variables are to be interacted with age, they must be listed *first.* For

[1] More specifically, each row represents a *transition* . That is, the starting and ending state over the time interval are both included as variables. If more than two waves—that is, more than one transition—of data is to be used, you will need multiple lines for each individual, just as in a discrete time logit model. In that case, you will have the starting and ending state for each time interval. For example, suppose an individual is observed at years 1, 2, and 3. This individual will require 2 person/transition records in the data file. The first will capture the starting state at time 1 and the ending state at time 2; the second will use the ending state at time 2 as the starting state and the ending state at time 3 as the ending state.

example, if sex, race, and region are your additional covariates, and sex and region are to be interacted with age, the order of covariates *must* be sex, region, and then race (or region, sex, race).

6. **Additional Age×Covariate interactions.** You may have as many such interactions as you like, as long as there is a main effect associated with them. *HOWEVER,* these interactions must be in the same order as the covariates with which age is interacted. For example, if, in the previous step you have the covariates in the order of sex, region, race, your interaction variables must be in the order sex×age, region×age.

7. **Additional Starting State×Covariate interactions.** You may have as many of these interactions to capture state-dependent effects as you like, as long as there is a main effect associated with them. *HOWEVER,* as with the age-dependence interactions, these must be in the same order as the covariates with which the starting state is interacted. For example, if, in the previous step you have the covariates in the order of sex, region, race, your interaction variables must be in the order sex×starting state, region×starting state.

8. **Outcome State.** The outcome variable must be coded as 0,1, or 2. The outcome state that is coded as 0 should be the same state coded as 0 in the starting state variable; similarly, the outcome state that is coded as 1 should be the same state coded as 1 in the starting state variable. Death must be coded as 2 on the outcome.

As an example that follows these rules, suppose we would like to include sex (Sex), race (R), education (E), and income (Inc) as covariates in the model, as well as interactions between age and education (AE), and age and race (AR), and interactions between the starting state and race (RStart). Recall that we also need a starting state variable (ST), a column with intercepts (Int), age (A), an interaction between age and the starting state (AST), and an outcome variable (OUT). In that case, the data file must be structured as:

Int A Start AST R E Sex Inc AR AE RStart OUT

Order is important in the structuring of the data. It is irrelevant how the covariates are ordered, but once they are ordered, the interactions must follow the same order, beginning with the first covariate listed. It is also important to keep in mind that the order must be consistent for both types of interactions (for capturing age dependence and state dependence). If, in the example above, we did not want a state dependent effect of race, but instead education, the data would need to be ordered as:

Int A Start AST E R Sex Inc AE AR EStart OUT

Notice that, in this structure, Education and Race have exchanged places. This has happened so that education is listed first, because it is the only variable that interacts with starting state. If we kept the order the way it was previously, the program would assume that the starting state interaction applied to race.

As it stands, in this version of the software, this issue of ordering the covariates and both sets of interactions presents a slight limitation: there are some cases in which you may need to include either interactions to capture age dependence or interactions to capture state dependence that you may not want to. For example, suppose your covariates were A, B, C, and D, and you wanted to capture age dependence in the effect of A and C and state dependence in the effect of B and D. There is no way, in this version of the software, to do this; the data cannot be ordered to satisfy this desire. Suppose, for example, we ordered the data A C B D. In that case, we could capture the desired age dependence effects, but we would need to include state dependence effects for A and C as well as B and D, given that the state dependence effects must be ordered beginning with the first covariate. The other alternative would be to order the data as B D A C, but in this case, we would need to include two more age-dependence effects than we'd like (for B and D).

This limitation may seem important, but in practice, we generally want most (if not all) effects to be state dependent, and we often want many effects to be age dependent.

## 4.3 Running the multistate hazard model program

Once the data file has been constructed appropriately, you are ready to estimate the model. As stated earlier, the program takes a number of command line arguments (7), which allows a great amount of flexibility in usage. To run the program, type the following on the unix command line (the bracketed numbers are the arguments discussed below):

```
mstatehazard.x [1] [2] [3] [4] [5] [6] [7]
```

(note: you may need to check with your unix system administrator to make sure the path for the executable is correct. for example, we often need to type "./" before the executable name)
The arguments are as follows:

1. Name of the input data file (e.g., mydata.dat)
2. Name of the output data file (e.g., myhazardresults.out)
3. The sample size (e.g., 3076)
4. The number of covariates *including the intercept and all interactions* (e.g., 13).
5. the number of iterations to allow the program to run (e.g., 200000).
6. the number of iterations to be skipped when writing to the output file—the extent of the 'thinning' to be done (e.g., 50).

7. an indicator for whether a structural constraint for the (1,1) outcome state should be imposed.

The first four arguments are self-explanatory. The last three are more technical. The MCMC algorithm draws samples from the joint posterior distribution of the hazard model parameters using a Gibbs sampler. These draws are not independent: they depend on the immediately previous sampled value and are therefore generally autocorrelated at a number of lags. Thinning the output to every $L^{th}$ iteration, where $L$ is the number of lags at which significant autocorrelation ceases to exist, increases the independence of the sampled values and leads to better inference. In practice, we have found that saving every $10^{th}$ iteration is more than sufficient to reduce autocorrelation, although this is only a rough guideline.

The number of iterations the algorithm should be run depends on (1) the number of lags at which the chain is thinned, (2) the number of life tables you want to generate (one per saved iteration *after* the initial "burn-in" iterations are discarded), and (3) the length of the burn-in period before convergence is reached. With a moderate number of covariates, convergence is generally obtained very rapidly, but with highly-correlated covariates or a very large number of them, convergence may be slower.

Finally, the indicator for the structural constraint can take a value of 1 (impose the constraint) or 0 (do not impose the constraint). As part of the Gibbs sampler, the multinomial data indicating the outcome state are augmented by drawing latent bivariate normal data for each observation. If the outcome states are (1) alive, nondisabled, (2) alive, disabled, and (3) dead, the two outcome equations that are estimated are (1) disabled versus not and (2) dead versus not. It is not immediately clear whether individuals who are dead at the time the outcome is measured should be simulated a value of 'disabled' or 'not disabled' in the first equation. If the structural constraint is imposed, decedents' scores will be forced to be simulated as 'not disabled' in this dimension of the outcome; if the constraint is not imposed, decedents' scores will be allowed to be simulated as either 'nondisabled or 'disabled.' For multistate tables, we recommend not imposing the constraint. The implication of not imposing the constraint is that the error correlation parameter will generally be close to 0. The reason for this is that the unobserved factors that would be associated with both disability and death will 'come out' in the data augmentation step.

## 4.4 Monitoring the hazard model output

As immediately discussed above, determining the number of iterations to allow the program to run and the number of iterations to 'skip' in thinning the output relies on technical knowledge of MCMC methods. If you are completely unfamiliar with these methods, email us and we can recommend some literature (e.g., the first author has a forthcoming book detailing these methods). In general, there are two general issues to be concerned about with MCMC sampling: convergence and mixing. Before making inference regarding parameters from MCMC output, the algorithm must have converged to the appropriate probability density and must have sampled well throughout it (i.e., it must have mixed well). Convergence and thorough mixing are ultimately impossible to definitively assess when faced with analytically untractable densities. However, a number

of ways exist to provide evidence that convergence and thorough mixing have probably occurred. A thorough discussion of these is beyond the scope of these instructions, but you should be convinced that these two criteria are met before proceeding further. One approach that is relatively simple is to compare the means and standard deviations of the parameters (after discarding the "burn-in" iterations) that predict the death outcome to univariate probit model results obtained via maximum likelihood estimation. If the posterior means and standard deviations of each parameter are close to the MLEs and associated standard deviations, the model has probably converged and mixed well. Another approach is to construct time series (trace) plots of the parameters and see whether the trace lines level off and show no sign of trending.

The remaining issue is to determine whether you have generated enough post-burn-in iterations to generate a sufficient number of life tables to provide an adequate summary of state expectancies. A rule of thumb is that you need at least 1,000 samples from a distribution to summarize it; thus, you need at least 1,000 post-burn-in iterations after thinning.

The output of the hazard model program will be a free-field ASCII file that contains (a) the iteration number, (b) the error correlation, (c) the parameters for all of the covariates for each equation. The order of (c) is such that the intercept parameter predicting outcome state 1 is listed first, followed by the intercept parameter predicting outcome state 2, followed by the age parameter for outcome state 1, followed by the age parameter for outcome state 2, and so on. Do not tamper with the order of columns in this file, because they are read-in by the life table program in this specific order.

## 4.5 Running the life table program

Once the hazard model output file is finalized, you are ready to generate the multistate life tables from the hazard model output file. This program is structured to produce an output file containing the state expectancies (for states 0 and 1) at every age for a given specification of covariates. In other words, you will need to decide the values at which to specify the covariates (e.g., do you want a distribution of state expectancies for black men? white men? white women? etc.). For every covariate (except age and the starting state), you will need to specify a value.

As stated earlier, these programs take a number of command line arguments; the life table program takes *more* arguments than the hazard model program, but allows a tremendous amount of flexibility in usage. To run the program, type the following on the unix command line (the bracketed numbers are the arguments discussed below):

```
mstatetables.x [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13 . . .]
```

The arguments are as follows:

1. The input file name. This should be the output file from the hazard model (e.g., myhazardresults.out).

2. The output file name. This is the file in which the state expectancies will be placed. (e.g., mylifetables.out).

3. The number of burn-in iterations in the hazard output file to discard (e.g., 3000). NOTE: Remember that the number of lines actually stored in the hazard output depends on the extent to which you thinned the MCMC chain and not just the number of iterations the program was run! This step tells the program to discard the first $m$ lines of results in the hazard output file.

4. The number of post-burn-in iterations for which to generate life tables (e.g., 1000). The program will, after discarding the first $m$ lines as the burn-in, then compute $n$ life tables.

5. The radix for starting state 0 (e.g., 1500).

6. The radix for starting state 1 (e.g., 800). These two radices should be chosen carefully; ideally, for a population-based table they should be the number of persons in each state in the youngest age category in the data. Using a proportion, as opposed to the raw numbers themselves, will yield identical results (e.g., using the 1500 and 800 above, we could also use 1500/2300 and 800/2300).

7. The starting age of the sample. Although the age coding in the original data used by the hazard model program must be 0, this value corresponds to some age (e.g., 20, 65, etc.). This argument requires this value so that the program can determine the number of age intervals to estimate.

8. The number of years of age represented by a single age code (e.g., 1 or 5—Is the table to be done by single year of age or is it to be abridged?) This choice must be consistent with the hazard model: You cannot generate an unabridged table if age in the original data used in the hazard model was in 5-year intervals.

9. An indicator for whether the structural constraint was imposed in the hazard model.

10. The number of *main effects* covariates NOT counting the intercept, age, the starting state, or the age-by-starting-state interaction. For example, if sex, race, region, education, and income are the additional covariates you should enter 5, regardless of whether any of these are interacted with age.

11. The number of age-by-X interactions. In the example just discussed, if sex, race, and region are interacted with age, you should enter 3.

12. The number of starting state-by-X interactions.

13. From argument 13 on, list the values at which the covariates are to be set. There should be as many numbers listed here as there are covariates listed in argument 10. You need only set the values for the main effects: the age-by-X and starting state-by-X interaction variables are automatically set to the values for their corresponding main effects. *Remember the ordering constraint discussed above under*

*"constructing data appropriately."* In the example involving sex, race, region, education, and income, with sex, race, and region interacting with age, the program assumes that the three interactions correspond to the first three main effects variables. Thus, your value chosen for sex will be the value chosen for the variable in the first interaction term.

### 4.6 Conducting analyses with the results

The output that is produced by the life table program is a data file that consists of state expectancies for each of the two states in the data. The output from the life table program is written to this file so that each row is a sample life table. The state expectancies are written in age-ascending order. For example, if le(ab) is the life expectancy at age a in state b, and k is the maximum age, the data are written as:

```
0 le(00) le(01) 1 le(10) le(11) 2 le(20) le(21) . . . k le(k0) le(k1)
```

These data can be summarized just as one would summarize data on individuals. For example, the mean of le(00) would be the expected numbers of years a person at age 0 will live in state 0. The standard deviation of this variable can be used to construct an interval estimate for this expectancy. It is important to note that this output file represents the state expectancies for the particular combination of covariates you selected in the life table program. If you want to compare state expectancies across levels of covariates, you will have to rerun the life table program using a different set of covariate values. Then the two output files can be merged and comparisons can be made. For example, independent samples t-tests, regression models, or whatever other type analysis can be conducted.

## 5 An Example

### 5.1 Constructing the Data

Our example examines sex and race differences in active life expectancy (ALE). The data for this example are from the 1987 and 1992 followups—the National Epidemiologic Followup Survey (NHEFS)—to the 1971 National Health and Nutrition Examination Survey (NHANES). We restrict the sample to individuals who were black or white and for whom disability status in 1987 and final status (nondisabled/disabled/dead) in 1992 were known. Given that these waves were 5 years apart, we estimate abridged tables in 5-year age groups beginning at age 65.

For the purposes of the example, the variables we include are (in this order): (1) an intercept term, (2) age (in 5-year groups; range 0-4), (3) the starting state (0=nondisabled; 1=disabled), (4) an interaction between agegroup and starting state, (5) sex (female=1), (6) race (black=1), (7) an interaction between sex and starting state, (8) an interaction between race and starting state, (9) an interaction between age and sex, (10) an interaction between age and race, (11) and the 1992 outcome variable (0=nondisabled, 1=disabled,

2=dead). The variables are arranged in this order as per the instructions in Section 2. The first few lines of the data (5 observations) look like:

```
1 2 0 0 0 0 0 0 0 0 1
1 3 0 0 1 0 3 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0
1 0 0 0 1 0 0 0 0 0 0
1 0 0 0 1 0 0 0 0 0 2
```

Each line consists of 11 numbers separated by a space. The first number is the intercept; the second is the age group; the third is the starting state (0=healthy/nondisabled); the fourth is the interaction of age with the starting state; the fifth is sex (female=1); the sixth is race (black=1); the seventh and eigth are the interaction of age with sex and race; the ninth and tenth are the interactions of starting state with sex and race; the eleventh is the 1992 outcome (0=nondisabled; 1=disabled; 2=dead).

## 5.2 Running the Hazard Model Program

After compiling the programs and creating two executable files (the hazard program—mstatehazard.x—and the life table program—mstatetables.x), the following was entered on the unix command line:

```
./mstatehazard.x example.dat example.out 3056 10 10000 5 0
```

The "./mstatehazard.x" calls the hazard executable file. "example.dat" is the input data. 3056 is the sample size. 10 is the number of covariates including the intercept (intercept, age, starting state, age×starting state interaction, age×sex interaction, age×race interaction, sex×starting state interaction, and race×starting state interaction). 10000 is the number of iterations the program will run. 5 is the number of iterations to skip when writing to the output file (every fifth iteration will be saved). 0 indicates that the structural constraint on the (1,1) cell of the multinomial space will not be imposed in the data augmentation step.

As the program runs, some output is printed to the screen. First, the program prints part of every $100^{th}$ case as it is read from the data file. Specifically, the first and second variables (the intercept and age variable) are printed, and the outcome variable value are printed. For example:

```
reading case:  100  x1=1.00000 x2=4.00000 out=2
reading case:  200  x1=1.00000 x2=0.00000 out=2
reading case:  300  x1=1.00000 x2=1.00000 out=1
reading case:  400  x1=1.00000 x2=4.00000 out=2
```

After the data is read from the file, the program will print "read in all the data" and then begin printing the iteration number and some of the sampled parameter values at

each iteration. Specifically, the intercept (0), age effect (1), and starting state effect (2) for each outcome state ("b" for state 1 versus "o" for death) are printed to the screen, as well as the error correlation (s). The program will print every $k^{th}$ iteration to the screen, where $k$ is the argument you supplied for the thinning of the chain. In this example, the following was printed:

```
it5 b0=-0.8233 b1=0.1032 b2=1.0529 b3=-0.1366 o0=-0.6288 o1=0.1930 o2=0.4458 o3=0.0838 s=-0.0324
it10 b0=-1.0966 b1=0.1618 b2=1.3728 b3=-0.2387 o0=-0.8845 o1=0.2465 o2=0.2366 o3=0.1538 s=0.0762
it15 b0=-1.1213 b1=0.1650 b2=1.5496 b3=-0.3073 o0=-0.8320 o1=0.2324 o2=0.2491 o3=0.1303 s=-0.0842
it20 b0=-1.2416 b1=0.1625 b2=1.8062 b3=-0.3333 o0=-0.8243 o1=0.2703 o2=0.2482 o3=0.0888 s=0.0144
it25 b0=-1.2289 b1=0.1662 b2=1.7662 b3=-0.3479 o0=-0.8809 o1=0.2898 o2=0.4633 o3=-0.0373 s=0.0071
```

You should be able to watch this output as it is printed to the screen and observe that each of the parameter values rapidly stabilize in a fairly narrow region.

The output file, "example.out," is an ASCII file like the input data, with columns separated by spaces. The program prints the iteration number first and the error correlation second. The parameters printed after are printed in the order of the covariates by pairs—one for each of the two equations. For example, below is an excerpt from the output file "example.out." The parameter -.846601 is the sampled intercept for the disability equation, while -.663696 is the sampled intercept for the death equation; .103955 is the sampled parameter for age in the disability equation, while .184689 is the sampled parameter for age in the death equation (etc.).

```
5 -0.088451 -0.846601 -0.663696 0.103955 0.184689 0.873567 0.426736 -0.028613 0.130125 -0.192224 -0.588139 0.108206 0.05
8436 0.055638 0.059658 -0.013769 0.019358 0.107436 -0.125020 0.115285 -0.197990
10 0.000715 -0.959597 -0.847072 0.108199 0.257474 0.887270 0.510417 -0.042794 0.088428 -0.178089 -0.533007 0.109902 -0.0
46339 0.063310 0.068697 0.015666 0.006379 0.252050 -0.077508 0.090404 0.027993
15 0.064626 -1.073040 -0.853372 0.134817 0.273028 1.073578 0.371396 -0.113468 0.106751 -0.094432 -0.594757 0.079711 -0.0
13627 0.060905 0.062297 -0.024973 0.034237 0.291364 0.065307 -0.367364 0.132762
20 0.096486 -1.199079 -0.919208 0.174481 0.303285 1.437240 0.363027 -0.161308 0.031628 -0.070070 -0.478484 0.154506 -0.1
81024 0.045726 0.031050 -0.036987 0.043077 0.025456 0.126938 -0.396758 0.155327
```

This output may be read into any statistical software package to obtain summary statistics for the hazard model parameters. The mean and standard deviation of each column should be close to the values obtained via maximum likelihood estimation for the parameters and standard errors. NOTE: technically, with uniform prior distributions (which we use for the hazard model), the posterior *mode* of each column should correspond to the maximum likelihood estimates; however, because the posterior distributions should be approximately symmetric, the mean and mode will be similar.

As we said earlier, we should examine the hazard model results to determine whether the algorithm was run long enough to obtain convergence. Since Gibbs sampling generates samples from distributions for the hazard model parameters, the algorithm should converge to a distribution, and not a point like maximum likelihood algorithms. We can examine for convergence by constructing time series plots of the parameters. For example, below are four plots of parameters from this sample hazard model output file (example.out).

As the plots show, the algorithm rapidly converged to a narrow region for all four parameters and then randomly walked around that region for each parameter.[2] If we take the last 1000 of the 2000-iteration sequence for these parameters and examine the distributions in histogram format, they appear as in the figure below.

[2]A large literature in Bayesian statistics exists discussing how to formally evaluate convergence. We can point you to literature discussing this, or we can forward our forthcoming paper which has several citations.

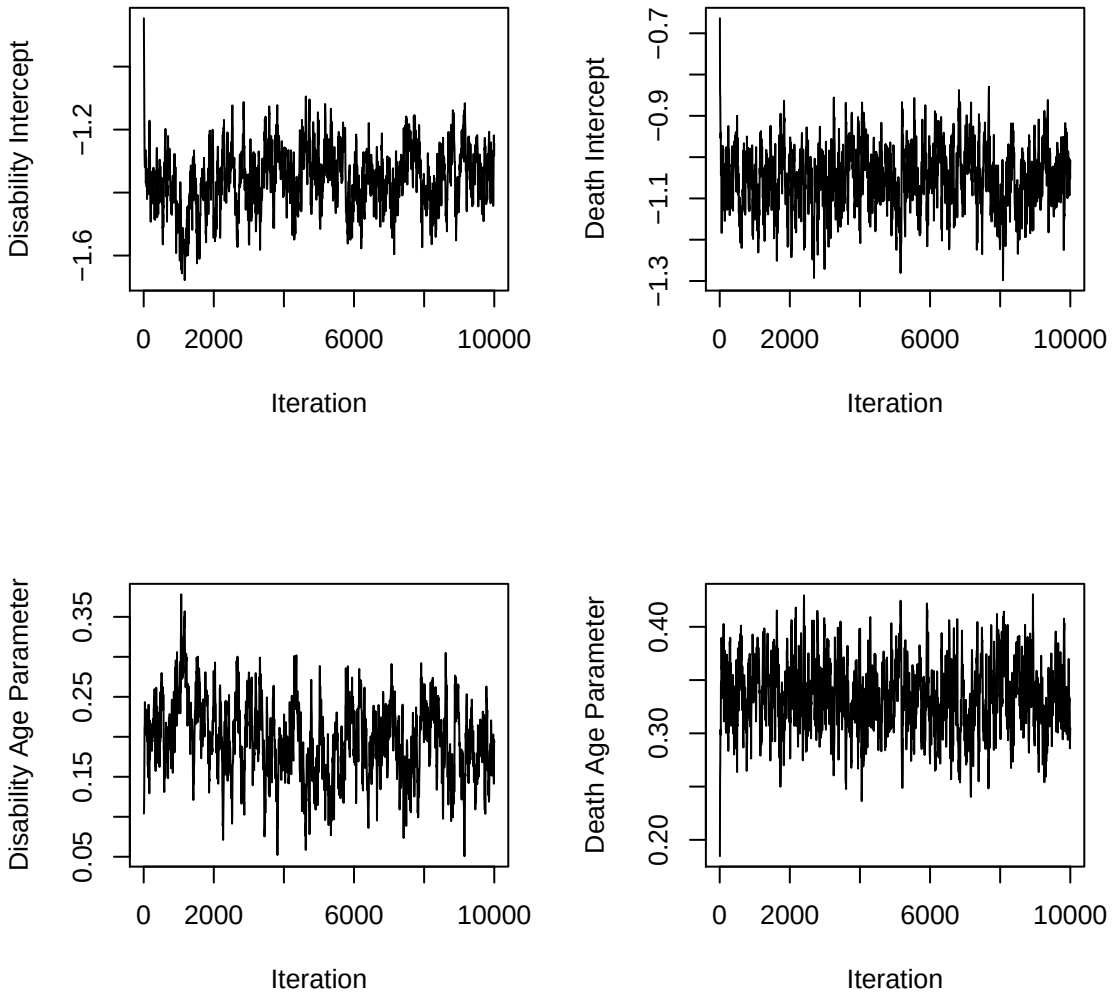


Figure 3: Time series (trace) plots of four parameters from the example multivariate hazard model.

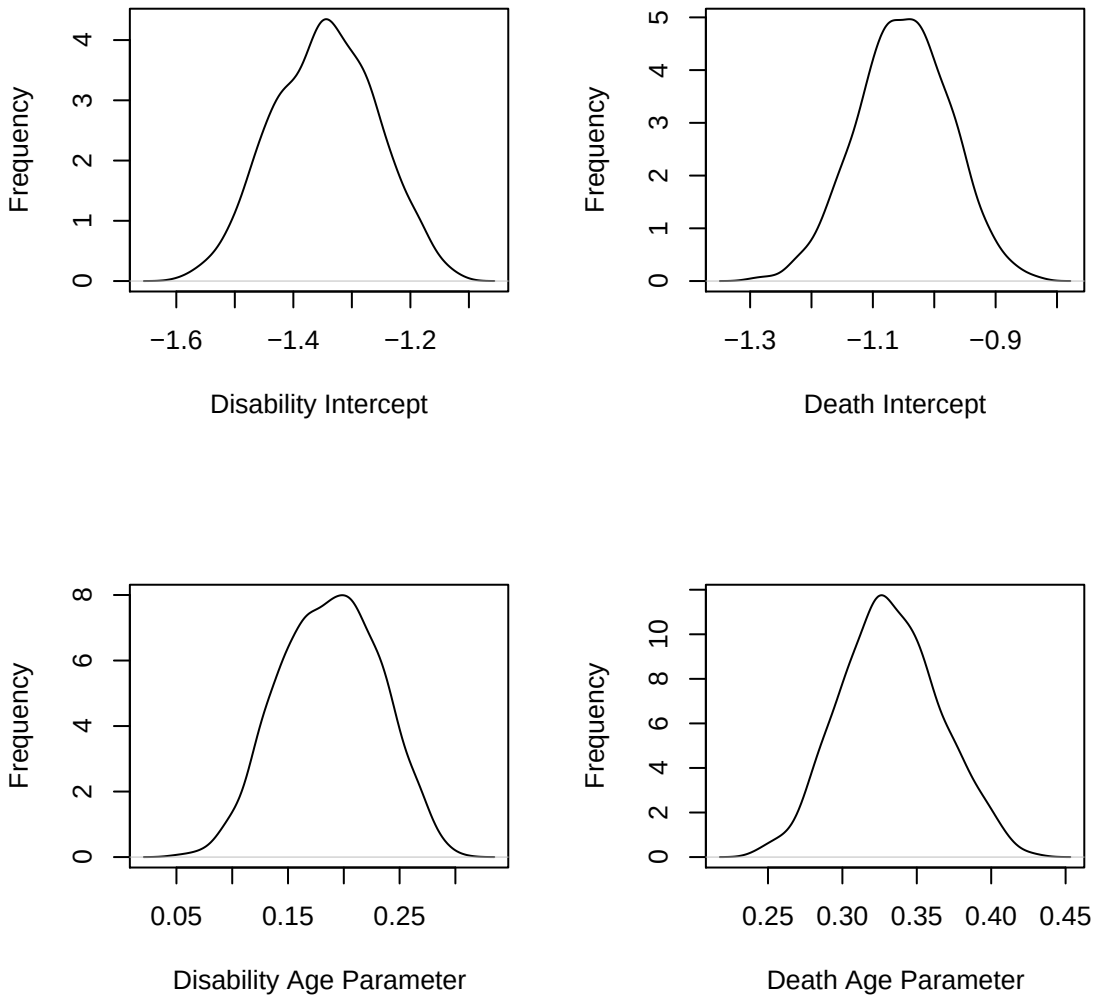


Figure 4: Histograms of the last 1000 sampled values from of four parameters from the example multivariate hazard model.

The parameters, as expected, all appear fairly normally distributed. We recommend, in addition to examining trace plots for determining convergence, that you compare summary statistics of the distributions of parameters samples (e.g., the mean and standard deviation) to those obtained via maximum likelihood estimation (e.g., the MLE and standard error). The results should be similar, but probably will not be exact, especially in the equation for predicting disability. They will not be the same in the disability equation because the ML-estimated model will not be able to include decedents (they are missing on disability at Time 2).

Below is a table summarizing the probit model results obtained via the software along with results obtained via maximum likelihood estimation in SAS. As the table shows, the results obtained via Gibbs sampling are nearly identical to those obtained via ML estimation.

| | Posterior Means (s.d.) | | MLE (s.e.) | |
|---|---|---|---|---|
| Variable | Disability | Death | Disability | Death |
| Intercept | -1.35(.09) | -1.05(.08) | -1.35(.09) | -1.06(.07) |
| Age | .19(.04) | .33(.03) | .19(.04) | .34(.03) |
| Start Disabled | 1.73(.20) | .69(.17) | 1.72(.21) | .70(.15) |
| Age×Start | -.20(.06) | .01(.05) | -.20(.07) | .01(.05) |
| Female | .03(.11) | -.41(.10) | .04(.11) | -.40(.10) |
| Black | .36(.17) | .16(.15) | .35(.17) | .16(.15) |
| Age×Female | .06(.05) | .01(.04) | .05(.05) | .004(.04) |
| Age×Black | -.04(.07) | -.05(.06) | -.04(.07) | -.05(.06) |
| Start×Female | -.17(.21) | -.03(.14) | -.17(.21) | -.02(.14) |
| Start×Black | -.36(.24) | -.11(.18) | -.37(.24) | -.11(.18) |
| n | 3056 | | 3056 | 2097 |

## 5.3 Running the Life Table Program

After we were convinced that the Gibbs sampler had converged, we then ran the multistate life table program. Again, the life table program requires the selection of a particular covariate profile. In this example, the only main-effects covariates we included in the model were sex and race. In order to make comparisons across sex and race, we ran the life table program four times—once for each sex-race combination. We generally run the program multiple times (once for each desired covariate profile) by stacking multiple command lines into a script file (like a DOS batch file). Below is the first line from such a script file:

```
./mstatetables.x example.out example.wm 1000 1000 259 27 65 5 0 2 2 2 0 0
```

The first argument listed after the program name (`mstatetables.x` ) is the input file—`example.out` . This, of course, is the hazard model output discussed in the previous section. The second argument is the output file for the life table results—`example.wm` .

The third argument—1000—is the number of burn-in iterations in the input file (example.out) that we will discard before beginning to compute life tables. This determination was made by examining the trace plots of the hazard model parameters: the model appeared to converge after a few hundred iterations, but to be conservative, we dropped the first 1000. The fourth argument—1000—is the number of life tables to generate. Given that the hazard model ran for 10,000 iterations, with every $5^{th}$ iteration saved, this left us with 2,000 sampled parameter values. After discarding the first 1,000 of these, we were left with 1,000 with which to construct life tables.

The next two arguments—259 and 27—are the number of white males who began the first age interval (65-69) nondisabled and disabled, respectively. The seventh argument—65—is the starting age, while the eighth argument—5—tells the program the number of years represented in each age group. The ninth argument—0—tells the program that the structural zero constraint (constraining individuals not to be both disabled and dead in the data augmentation step of the Gibbs sampler) was NOT imposed in the hazard model. The tenth, eleventh and twelfth arguments—2, 2, and 2–tell the program there are two main effects covariates (not counting age, the starting state, and the age-by-starting state interaction, which *must* be included), two age-by-covariate interactions, and two starting state-by-covariate interactions. Finally, the last two arguments—0 and 0—establish the covariate profile. Since the first covariate is sex (female=1) and the second is race (black=1), and we set these to 0 and 0, respectively, this instance of the program will construct tables for white males.

The following few lines are from the `example.wm` file:

```
0 10.77 1.84 1 8.04 1.62 2 5.95 1.41 3 4.42 1.23 4 3.35 1.11 5 2.61 1.03 6 2.11 1.01 7 1.76 1.05 8 1.47 1.10 9 0.00 0.00
0 10.91 2.19 1 7.94 1.95 2 5.67 1.71 3 4.05 1.51 4 2.93 1.36 5 2.20 1.28 6 1.71 1.28 7 1.38 1.34 8 1.12 1.43 9 0.00 0.00
0 10.54 2.14 1 7.67 1.92 2 5.49 1.70 3 3.92 1.51 4 2.85 1.38 5 2.14 1.31 6 1.67 1.31 7 1.33 1.38 8 1.06 1.48 9 0.00 0.00
0 11.00 2.00 1 8.14 1.81 2 5.86 1.66 3 4.15 1.54 4 2.96 1.45 5 2.16 1.41 6 1.62 1.43 7 1.24 1.51 8 0.95 1.60 9 0.00 0.00
```

As discussed earlier, these files contain the active life and disabled life expectancies for each age group implied by each sampled parameter set from the Gibbs sampling program. For example, in the `example.wm` data set, the first set (row) of parameters from the Gibbs sampler implies an ALE at age 65-69 of 10.77 years and a DLE of 1.84 years. The implied total life expectancy (TLE) is therefore $10.77 + 1.84 = 12.61$ years, and the implied proportion of life nondisabled is $10.77/12.61 = .854$. Similar calculations can be made for each successive age group, completing a row in the output file. Thus, each row in the data set contains the life table implied by the sampled hazard model parameters.

Figure 5.3 shows histograms of the distribution of ALE, DLE, TLE and the ratio ALE/TLE for white males at age 65. Superimposed over the histograms are the mean (solid line) and lower and upper bounds of a 95% empirical confidence interval. These limits are found by sorting the appropriate column in the data set and selecting the $25^{th}$ and $975^{th}$ sampled value (the $2.5^{th}$ percentile lowest and highest values).

Given that our output files consist of 1000 samples of ALE and DLE for each sex-race combination, we can compare these quantities across groups. Figure 5.3 shows the decline in mean ALE, TLE, and the proportion of life remaining nondisabled across age by sex and race. The first plot shows that years of active life decline for all four groups across age and

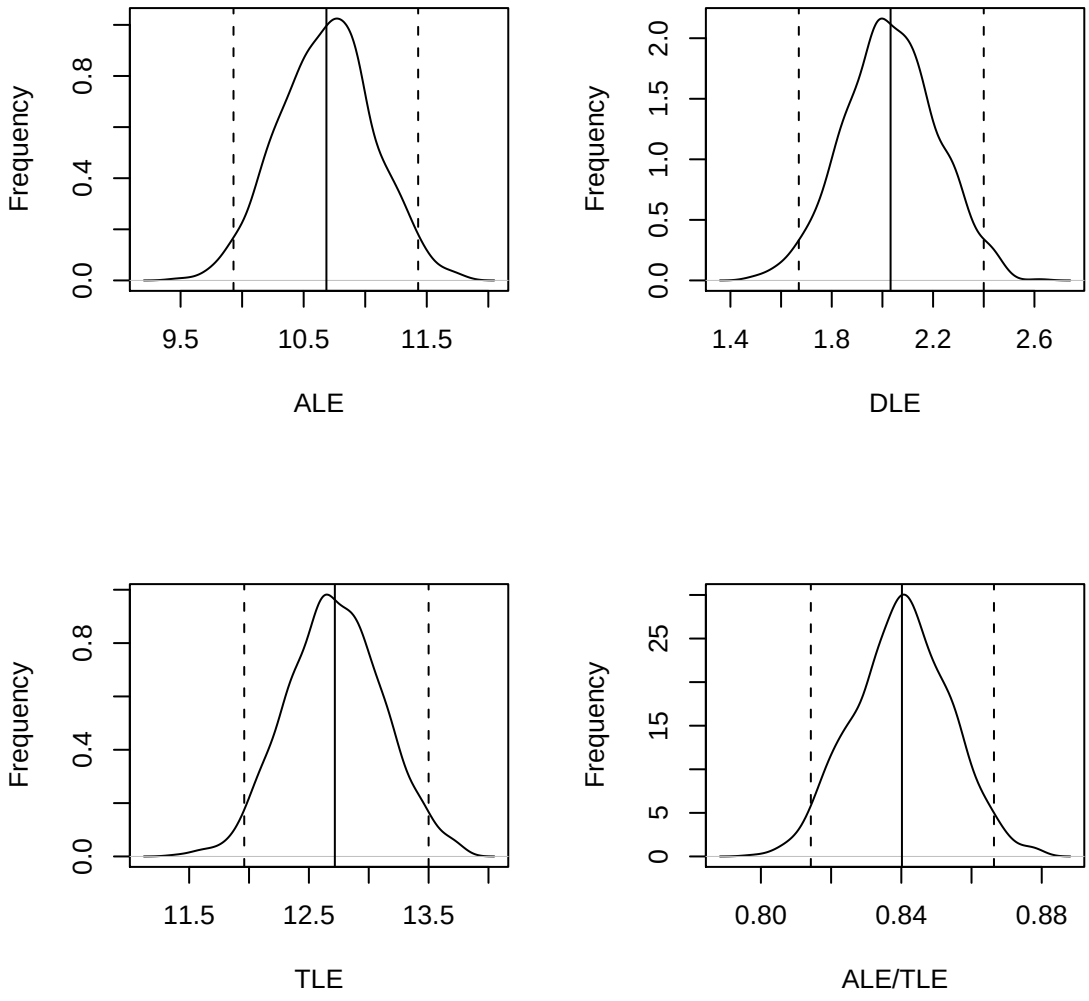


Figure 5: Histograms of the ALE, DLE, TLE and ALE/TLE for White Males at Age 65.

appear to converge. The second plot shows that years of total life also decline for all four groups across age and also appear to converge. The third plot shows that the proportion of life remaining active also declines across age for all four groups, but it also shows that, although these proportions converge within sex between races, the proportions diverge between sexes within races. In other words, white males and females evidence divergence across age in the proportion of life remaining active, while white and black males evidence convergence.

Aside from these comparisons of point estimates across age, statistical comparisons can also be made. For the sake of brevity, we keep these comparisons here limited to visual comparisons. Figure 5.3 shows histograms of the ratio ALE/TLE by sex and race at ages 65 and 85. As the figure indicates, there are some clear differences in the distributions of this ratio at age 65, but these differences are largely gone by age 85. Formal statistical comparison can be made much as one would do using classical methods. For example, one can consider the output file data to be a sample from the distribution of ALE and DLE (and their combinations) and conduct t-tests, regression models, etc. to determine whether, once sampling and other uncertainty is considered, there are between group differences in these quantities.

# 6 Limitations

There are several limitations of the software in its current form. First, age dependence is limited to a probit specification. This limitation is not particularly problematic—mortality risk is known to follow an s-shaped (and not exponential) curve across age. However, for other types of transitions (e.g., to/from disabled states) this specification may not be ideal.

Second, the software currently limits the state space to two non-absorbing states. We expect to expand this to three states soon.

Third, as discussed earlier, there are some combinations of age and state dependent effects that may be impossible to incorporate, given the data structure requirements. We expect to remedy this limitation soon; however, for this first version, we decided not to incorporate the additional command-line arguments necessary to fix this.

Fourth, testing convergence of the model parameters is best performed by running multiple instances of the model using different starting values. As it stands, this version starts all parameters at 0. We expect to automate testing convergence in subsequent versions of the software.

Fifth, the unix platform-based code is perhaps not ideal. We would like to produce more user-friendly versions (e.g., in MS Windows) and expect to do so when funding becomes available.

Sixth, as written the software assumes that only two time points are observed. If you are working with more than two time points, the person-year data file will contain multiple transitions per observation in the data set. These observations are not independent, given that they represent the same individual measured repeatedly. Ideally, a random effect should be included to compensate for the lack of independence, but we have not incorporated this into the software yet. The implication is that the standard deviations of the parameter distributions will be underestimated, just as they would be in any other

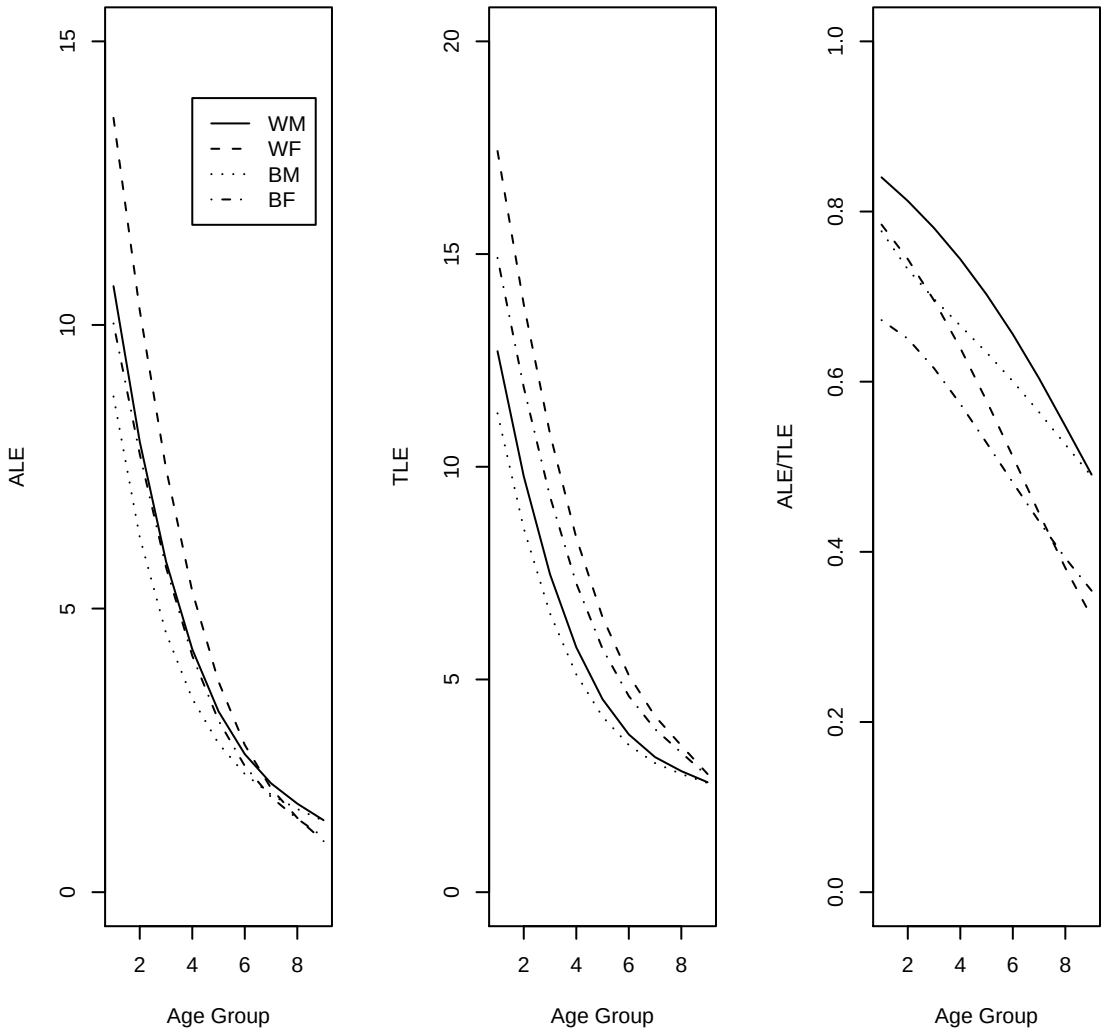


Figure 6: Age Patterns of ALE, TLE, and ALE/TLE by Sex and Race.

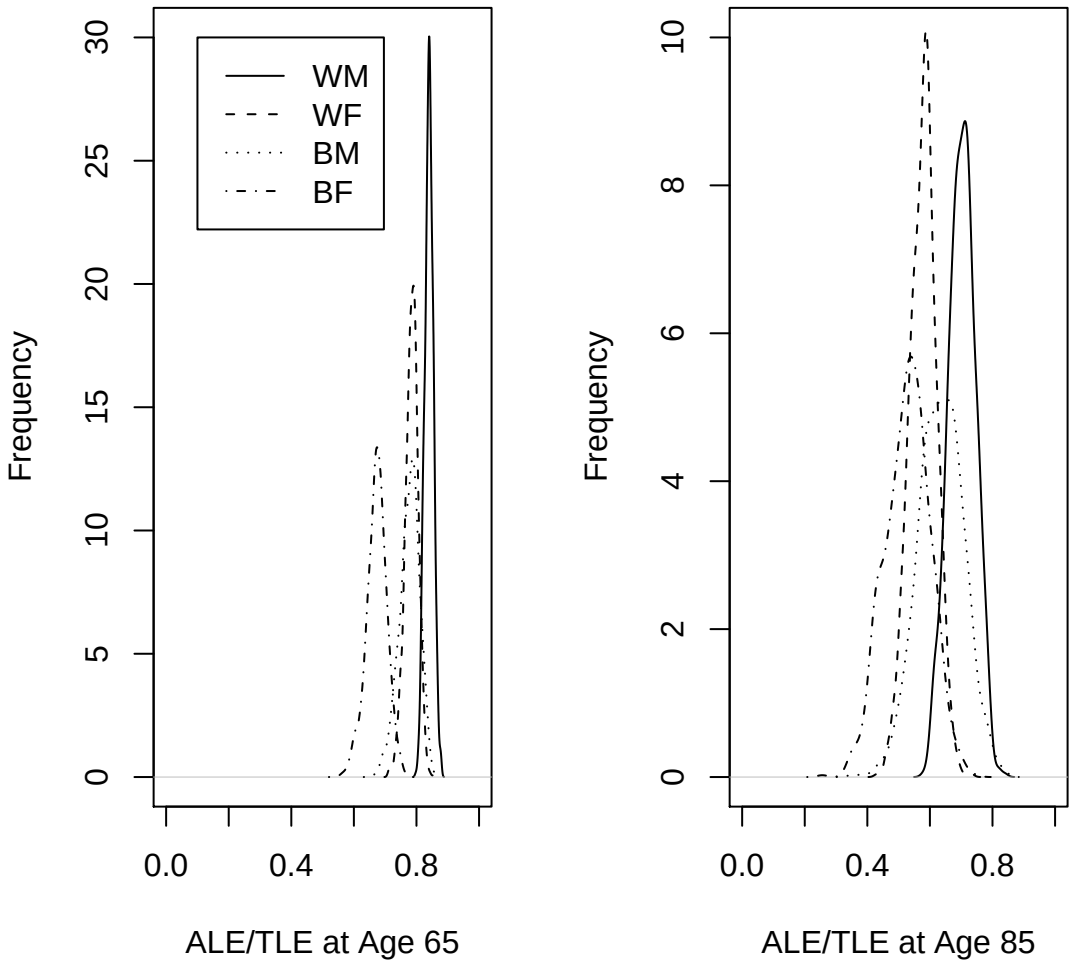


Figure 7: Histograms of the ALE/TLE by Sex and Race at Ages 65 and 85.

type of analysis.

If you have any difficulty, questions, comments, or suggestions, please contact us via email and let us know. We expect to update the software regularly.