

Monte Carlo Simulation

Monte Carlo simulation is useful for understanding the properties of a model, either under the assumptions of the model, or under other assumptions (i.e., under a different model). In addition, such simulation can be useful in the design of studies or experiments. Program RELEASE (see Burnham et al. 1987, Part 9) is very useful in the context of open Capture-recapture models because it offers a range of options that are easy to use.

The idea underlying simulated data is fairly simple. Under the multinomial model one has $E(y_4) = np_4$, the expectation of the number of times a "4" is observed after rolling a die n times. However, the observed value of y_4 will vary over k studies, whereby n throws are conducted on each study. Clearly, y_4 is a random variable. Monte Carlo simulation produces a set of such random variables, where the distribution and the parameters are both known. The procedure produces "simulated" data where the generating model and its parameters are known.

In generating simulated data, one usually fixes the binomial parameter p_4 to a numerical value (e.g., say $1/6 = 0.166666\dots$). Then, one draws from a uniform distribution from zero to one (i.e., $U(0,1)$) a variate. If the value of this variate is ≤ 0.166666 , then a 1 is recorded in the frequency count (e.g., y_4 is augmented by 1). The procedure is carried out n times, each time drawing a new variate from a $U(0,1)$ distribution, checking to see if it is \leq the parameter in question (in this case, p_4), and augmenting the count if the condition is satisfied. Such computations can be done by hand if the model is as simple as die throwing.

In the open capture-recapture models, one has probabilities such as $\{1001101\}$ and, under model $\{\phi_t, p_t\}$ has probability

$$(\phi_1 q_2)(\phi_2 q_3)(\phi_3 p_4)(\phi_4 p_5)(\phi_5 q_6)(\phi_6 p_7).$$

Now, given the numerical values of the apparent survival and recapture probabilities, one could compute the probability of observing $\{1001101\}$, generate R_1 $U(0,1)$ variates, check for the inequality, and increment the count m_{17} each time the inequality is satisfied. Of course, m_{12} , m_{13} , m_{14} , m_{15} , m_{16} , m_{17} , m_{18} and *never* are also being incremented as outcomes from releases at occasion 1. This simulated data set is generated for the other cohort releases also: R_2 , R_3 ,

The approach actually used in Program RELEASE and Program MARK is to model the fate of an individual animal. Suppose the animal is released on occasion 1, i.e., it is part of R_1 . The first event is whether the animal survives the first interval, i.e., whether it survives to occasion 2. The probability that the animal dies during interval 1 is $(1 - \phi_1)$, or with probability ϕ_1 that it survives interval 1. A random uniform variate is generated. If this value is greater than ϕ_1 , then the animal died, resulting in the encounter history $\{1000000\}$. If the random variate is less than ϕ_1 , then the animal lived. So now, the animal is available for capture on occasion 2. The probability of capture is p_2 . A new random variate is generated. If the value is less than p_2 , then the animal was captured. The

encounter history would now have the value {11} at the start, indicating this capture. The process is continued until the animal either dies, giving its encounter history, or else it survives through all 7 intervals, also giving its encounter history. Each animal in the study is simulated in the same way, to give the final set of encounter histories.

The approach used in Programs RELEASE and MARK allows for individual heterogeneity, age-specific effects, and thus provides a very general simulation capability. Using the m_{ij} array to perform the simulations is more efficient, but not nearly as general as simulating the individual animals.

Using thousands of draws from a uniform random number generator, one can generate a simulated data set where the model and its underlying parameters are known. Such Monte Carlo data sets can then be used for a wide variety of purposes. Typically, 1000 simulated data sets are used for many studies, however, 10,000 is not uncommon and is often required for studies of achieved confidence interval coverage. Each simulated data set is different, just as the outcomes (the y_i) after n throws of a die would be different.

Some reasons for conducting Monte Carlo studies:

Data Have Been Generated under Model A and Studied under the Same Model:

1. Small sample bias in the MLEs $\hat{\theta}$ of parameters
2. Compare SD of the MLEs from the 1000 reps vs. the average $\hat{se}(\hat{\theta})$
3. Confidence interval width and achieved coverage.

Data Have Been Generated under Model A and Studied under a Different Model:

1. Model bias in the MLEs $\hat{\theta}$ of parameters (robustness?)
2. Compare SD of the MLEs from the 1000 reps vs. the average $\hat{se}(\hat{\theta})$
3. Confidence interval width and achieved coverage.
4. Model selection issues

Design of Studies and Experiments

1. What is the effect of doubling the number released or precision?
2. What is the effect of increasing the duration of the study (k) on precision?
3. What is the effect of increasing p on precision?
4. What are the effects of increasing the level of the treatment?

Using Expected Values as "Data"

A shortcut procedure can often be used to obtain quick and dirty insights into some of the above issues without running 1000 or 10,000 simulations. In this procedure, one computes (only once) the

expectations of each encounter history or m_{ij} cell and rounds to the nearest integer. These values are then used as if they were "data." These values are not random variables.

Running such expectations through an estimation routine (e.g., *MARK* or *RELEASE*) gives rough ideas about model bias, standard errors, confidence interval width. This information is very useful in design of studies and formal experiments. For example, this simple method will give quick insights into the relation between the p_i and precision of some $\hat{\theta}$, effects of increasing study duration or precision, relative allocation of R_{ti} vs. R_{ci} , etc.

Once the design has been fine-tuned one may often want to go to some Monte Carlo simulations to obtain confidence interval coverage and comparisons between the standard deviation of the estimates over the reps vs. the average model-based standard error of the estimates.

Program RELEASE

Program *RELEASE* is very useful in simulation of Monte Carlo data sets for open population capture-recapture and band recovery data sets. It has a friendly user interface and allows the user to easily define parameters for simulation of data sets. Running this interactive portion of the program builds an input file (RELEAS00.INP). This can be run by simply entering the following as a DOS command,

```
RELEASE i=RELEAS00.INP o=RELEAS00.OUT
```

The first word calls program *RELEASE*. The second provides the filename for the input (i.e., i=) file while the third allows the user to specify the filename for the output file.

After the first run, one can merely edit the input file (i.e., call this new input file RELEAS01.INP) in DOS and make additional runs to define the design. This is a very useful program and is documented in Burnham et al. (1987: Part 9).

Program *RELEASE* computes test power for goodness of fit tests and various other tests for a specific set of nested models. These are still of some use.

Program MARK

The design of studies or experiments can be fine-tuned in program *MARK* as the models that can be run in program *RELEASE* is somewhat limited. In addition, *MARK* has its own simulation options that are being developed.