

Monte Carlo Analysis in Academic Research

Paul E. Johnson

April 28, 2011

Department of Political Science and Center for Research Methods and Data Analysis,
University of Kansas

Monte Carlo analysis is a research strategy that incorporates randomness into the design, implementation or evaluation of theoretical models. It began in the 1940s, when the development of computer hardware and mathematical models made it possible to generate streams of random numbers. These random number streams are combined with mathematical models in order to create models and evaluate theories of random processes. This chapter attempts to tame this diverse, unmanageable collection of concepts and methods by dividing simulation projects into three types. The first, commonly called “Monte Carlo simulation,” is used to evaluate statistical estimators. When an estimation procedure is proposed, it is standard procedure to test it against a variety of simulated research problems. A second type of project, referred to as “Markov chain Monte Carlo” (or MCMC), helps researchers to draw conclusions about complicated probability models for which conventional research strategies do not yield insights. The third

type of project arises in the study of complex systems, which are characterized by a large number of loosely interconnected, autonomous elements. Commonly known as agent-based models, these simulations have found enthusiastic advocates in environmental and social sciences.

Keywords: Monte Carlo, Markov chain Monte Carlo (MCMC), pseudo random number generation (PRNG), Bayesian statistics, agent-based modeling

Monte Carlo (MC) analysis is a general term that refers to research that employs random numbers, usually in the form of a computer model (or simulation). Although this research began in the natural sciences, computer science, and mathematics, it is now widely applied in social science as well. This essay attempts to explain the fundamental ideas that spurred the creation of these new procedures as well as their eventual adaptation for use in social science research.

This chapter is not a “how to” guide for simulation, but rather it is a “what for” or “why you might want to” guide. Some of the difficulties that arise in MC research projects are considered as well. It begins with some background information on the development of computers and algorithms for random numbers. After that, the chapter takes up applications in the evaluation of proposed statistical estimators, the practice of Bayesian statistics via computer simulation, and investigation of complex systems through agent-based models. Some conclusions about the challenges that face the field are presented, along with a conclusion.

A significant part of the presentation is about the exciting developments that have occurred since 1990. Rapid improvements in hardware and software have opened up opportunities for scholars to work with models that were previously prohibited by conceptual and technical barriers. At the current time, we are able to conceptualize and implement models that were simply impossible just 10 years ago. The extremely rapid progress has been driven by a fruitful interaction of substantive researchers in the natural and social sciences as well as programmers and computer scientists.

A secondary theme in this presentation is that we face some troubles in the dissemination of these new research tools. The possibility that a computer simulation might approximate the solution of an otherwise intractable math problem quickly captures our imagination. The possibility, though, leaves in its wake a number of challenges in the creation of standardized tools and replicable results. The necessities of research have created a fruitful tension between our computing abilities and our conceptual models, a tension that has no doubt spurred the development of both. However, progress is usually found in solutions to particular problems, and we are then pressed to find out if those particular solutions can generalize to address the problems that we would like to solve in our various research projects.

1 Background

The key elements of modern modeling—computer hardware, mathematical models, and computer simulation—are inextricably interwoven. The physicists who studied atomic fission during World War II (the Manhattan Project at Los Alamos, New Mexico) had the support of some of the greatest mathematicians in the world. Nevertheless, there were mathematical problems that could not be solved without the imposition of strong simplifying assumptions, and some models could not be solved even then. In the usual usage, “solved” means that the answer to a question can be presented as an understandable formula that illustrates the roles played by all variables and unknowns.

Many of the problems with which they were confronted seemed to have uncertainty, or unpredictability, at their very core. The movement of atomic particles was described by probability models. Fixed inputs did not lead to the same output every time, so it appeared that trial and error would be inevitable. Testing on actual bombs was both expensive and dangerous. Where theoretical mathematics could not offer clear answers, it appeared that simulation experiments offered the only realistic hope.

The research team proposed a tool they created for this purpose, Monte Carlo simulation.

The quantum theory of physics holds that atomic particles move about in a way that appears random to the observer. Perhaps a particle's movement resembles a "random walk," which supposes that a particle positioned at point x will be at point $x + u$ at the next moment, where u is a realization of a random process. Analytical tools might describe that process "on the average," but a simulation may offer a richer view of the possible paths that will be traveled. These random numbers, which have might have been "drawn from a hat" or pulled from a roulette wheel, gave the models a quality of unpredictability ("Monte Carlo" is a reference to the most popular gambling location of that era). Computers were in their infancy at the time, little more than elaborate calculators. The sheer number of calculations required to generate random numbers and put them to use would stagger a team of scientists armed with pencils and calculators. Five hours of computer time would replace the full-time, year-long effort of 20 computational assistants (Baines, 1962).

Aside from the atomic bomb itself, the introduction of the conceptual framework for computer-based Monte Carlo analysis might have been the most important lasting contribution of the Manhattan Project. They created not only the working demonstration of the importance of random numbers in mathematical models, but also the fundamental framework of computing itself. The team proposed what we now call the "von Neumann architecture" as a framework for the design of computer hardware and operating systems, a design that is still in use today. (The framework bears this name because John von Neumann was the author of the "First Draft of a Report on the EDVAC" (1945), a report to the U.S. Army). The "contemplated device" would be able to keep data and command sequences—programs—in memory so as to allow repeated access to both. After proposing the architecture, von Neumann spent the rest of his life outlining a sequence of mathematical models that could be investigated with the computers that were still in development at the time of his death (Aspray, 1990).

Several publications appeared that outlined a sweeping set of new research strategies. In their famous article "The Monte Carlo Method," Los Alamos scientists Nicholas Metropolis and Stanislaw Ulam described the approach as a research strategy for "middle sized prob-

lems” (1949). The middle sized problems did not yield to mathematical strategies because they had too many separate parts, but the number of parts was not big enough to justify approximations that would overlook the importance of individual pieces. The hope was offered that sampling from a range of possibilities could allow us to appreciate the tendencies of unpredictable processes. These ideas were implemented in the most influential essay to emerge from that group, “Equation of state calculations by fast computing machines” (Metropolis, et al. 1953).

By the late 1950s, Monte Carlo simulation had been introduced in many scientific fields. The flavor of the applications that were appealing to physicists and mathematicians is quite clear in Bauer (1958). Difficult problems in integration and differential equations were approachable from an MC point of view. An applied role for simulation was foreseen by scholars in many fields, as scholars expected simulation to become an integral part of theory and model construction (Hammersley & Morton, 1954). The potential of simulations for the characterization of “real life” problems was recognized and put to use in the re-organization and design of manufacturing (Youle et al., 1959; Jessop, 1956), train yards (Crane et al., 1955), roads (Miller, 1961), landing control systems for airplanes (Blumstein, 1957), and air defense (Rich, 1955). Martin Shubik’s comprehensive review of Monte Carlo simulation projects showed that virtually no area of study had been left untouched (Shubik, 1960).

Monte Carlo simulation became more than just a last resort of the desperate mathematician. It became a way to build models that were more realistic. Where the formal approach would simplify a model in order to solve it, the simulation approach allowed scientists to implement models as theory intended. Simulation models were cropping up in areas where we might have least expected them, including political science (McPhee & Smith, 1962), ecology (Barnett, 1962), or even the great American pastime, baseball (Lindsey, 1961).

The remainder of the chapter is organized as follows. First, I explore the fundamental issue of random number generation. After that, I consider three types of applications of Monte Carlo analysis. These three methods are chosen so as to display the potential importance of

random number distributions in all stages of the research project. Simulation models can play vital roles in the creation, derivation, and evaluation of mathematical and statistical models or theories. Theories of subatomic particles, animals, trees, or people are thus seen in the same light. When the mathematical model represents the separate behaviors and interactive tendencies of these many parts, a simulation can project the tendencies of the whole system (the ensemble of particles, in the terminology of the Manhattan project scientists).

2 Where Do Random Numbers Come From?

In the 1940s and 1950s, programming expertise was necessary even to generate random integers. Today, random number generators are widely available, perhaps too much so. A leading researcher tested many common random number generators and concluded, “Do not trust the random number generators provided in popular commercial software such as Excel, Visual Basic, etc., for serious applications. Some of these [random number generators] give totally wrong answers for the two simple simulation problems...”(L’Ecuyer, 2001). A random number generator may fail if it repeats itself in a predictable pattern, or if there are sections in the stream that are compressed or trended.

I hasten to point out that *it is actually impossible to generate random numbers with a computer!* A program that generates a stream of random numbers today can generate the exact same stream tomorrow. Instead, computers use pseudo random number generator (PRNG) algorithms, procedures that will generate streams of numbers that appear to be unpredictable. The author of a simulation program must specify the starting values and parameters of a PRNG, thus causing the streams to differ. The resulting numbers appear random from the point of view of the observer who is not privy to that information; the pattern in the numbers cannot be deduced.

Before computers, one could buy books full of random numbers (I recall using these as late as 1980). There were algorithms to generate random numbers, such as rolling dice, but

computers made testing and development of these procedures much more feasible. There was quite a bit of trial and error as various randomization schemes were tried. An early review essay on computer PRNGs included 142 citations with a seemingly endless collection of proposed generators (Hull & Dobell, 1962)!

A pseudo random number generator aims to select values in an “equally likely” fashion from a set of integers, usually the range from 0 to the largest possible integer that the system can hold. On a 64-bit operating system, the integers range from 0 to 1.844674×10^{19} . To help the reader grasp the magnitude of that range, consider this. If one started counting, reading one number per second, she would be reading for 5,848,424,173 *centuries* before finishing. A good random generator will generate a long scramble of integer values with no discernible pattern. A fast algorithm is preferred, of course, since a project may require millions of random numbers.

At the current time, two random number generators are considered acceptable for researchers conducting Monte Carlo simulation (Lemieux, 2009, p. 24). The Mersenne Twister (Matsumoto & Nishimura, 1998), which is known as MT19937, does not repeat itself until it has dispensed $2^{19937} - 1$ values. Even among scientists who are accustomed to dealing with big numbers, that is a *huge* number. MT19937 is the default random generator in the R statistical program (R Development Core Team, 2010), Matlab, and the Swarm Simulation System (Minar et al., 1996). Also in widespread use is L’Ecuyer’s combined multiple-recursive generator, MRG32k3a (L’Ecuyer, 1999). The repetition period of that generator is 2^{191} , not so incredibly huge as the MT19937, but still impressive (3.1×10^{57} values can be drawn without repetition). Both of these approaches generate vectors of numbers that pass most tests for randomness. These have been the most widely accepted PRNGs for about 10 years, but there is always effort to improve upon them (see Panneton et al., 2006).

The stream of random integers is only the first stage in the typical simulation project. Researchers usually want to shape those random numbers into a statistical distribution, such as the normal, gamma, beta, binomial, or other distributions. Procedures to convert the

equally-likely stream of integers into a desired distribution have been the focus of much research (Knuth, 1968; von Neumann, 1951). A leading contributor has been George Marsaglia (to cite just a couple of his papers, Marsaglia, 1961; Marsaglia & Tsang, 1998). Procedures to generate continuous uniform and normal variates were available quite early in the computer era, but research on nonsymmetric, truncated, or multivariate distributions has been ongoing (Marsaglia & Tsang, 2000; Everson & Morris, 2000).

The generators that have been discussed so far are proposed as methods with which to draw a single long stream of numbers. Many simulation projects will require the creation of 100s or 1000s of separate random streams. This ability to create independent streams is especially important in the new era of parallel high performance computing where it is necessary to launch separate processes on many different compute cores.

In practice, many of us who work on simulation projects have not been too concerned with this problem. In many projects, seeds for separate generators have been set by more-or-less unpredictable events (e.g., the time, current weather). There were no practical, well documented methods for creating provably separate streams of numbers until quite recently. There are two especially prominent strategies to deal with the problem. The authors of MT19937 (Matsumoto & Nishimura, 2000), and a research team at Florida State University (Mascagni et al., 2000), have proposed schemes that would dynamically “spawn” new generators and their streams are kept separate because each new generator is controlled by a unique set of parameters. The intuition for this approach is very appealing. However, designing the program that can actually spawn those separate generators turns out to be a dicey problem. Some successful reports have been published (Srinivasan et al., 2003).

The other leading approach, due to L’Ecuyer et al. (2002), is to take the one long stream of numbers from the generator and then divide it into separate substreams. Their implementation uses the MRG32k3a. Most practitioners with whom I have discussed this issue believe the theory behind this approach is stronger than that of its competitors: since the whole vector meets the requirements of randomness, then one can “splice into it” at vari-

ous points and extract separate random sequences. This method is currently the preferred implementation in parallel processing packages that are used with R (Sevcikova & Rossini, 2009).

When computers were scarce and slow (say, before 1985, perhaps even 1990), practitioners of MC analysis had to be careful because computer time was expensive. Collecting observations from a computer simulation might have been as expensive as sampling human subjects at one time. Many early Monte Carlo researches were focused on efficiency, finding the smallest workable simulation experiment (Kahn & Marshall, 1953; Ehrenfeld & Ben-Tuvia, 1962). At the current time, the generation of random numbers can still be the major source of computational expense, but the rapid increase of the speed of central processing units and memory has relieved us of most concern about the cost of generating random numbers.

3 Applications of Monte Carlo Analysis

Monte Carlo analysis includes a broad array of research activities. In an effort to make this manageable, I've divided the research problems into three categories. First, I consider Monte Carlo experiments that evaluate statistical estimators. For social scientists, this will be the most familiar application. Second, the Markov chain Monte Carlo (MCMC) procedure for simulation of probability models is introduced. The MCMC procedure was pioneered in the late 1940s and was a primary research objective of the development team that invented modern computers. Third, I consider agent-based simulation modeling projects in the field of complex systems research. In these simulations, the random samples are used to perturb the small scale interactions of components in dynamical systems.

3.1 Understanding Sampling Distributions

In this section, we explore Monte Carlo simulation for testing and illustrating statistical estimators. In the mid 1960s, Yates proclaimed that the widespread availability of computers would constitute the “second revolution in statistics” (Yates, 1966). Statisticians generally prefer a formal proof, but a problem may not yield to analytical methods. Sometimes a simulation may have to do. Simulation is a way of forming an educated guess about the most likely outcomes or the range of possibilities. In this type of MC analysis, “from the point of view of a statistician, the problem is nothing more than to find the sampling distribution of an intricately and irregularly defined statistic” (Youle et al., 1959, p. 491).

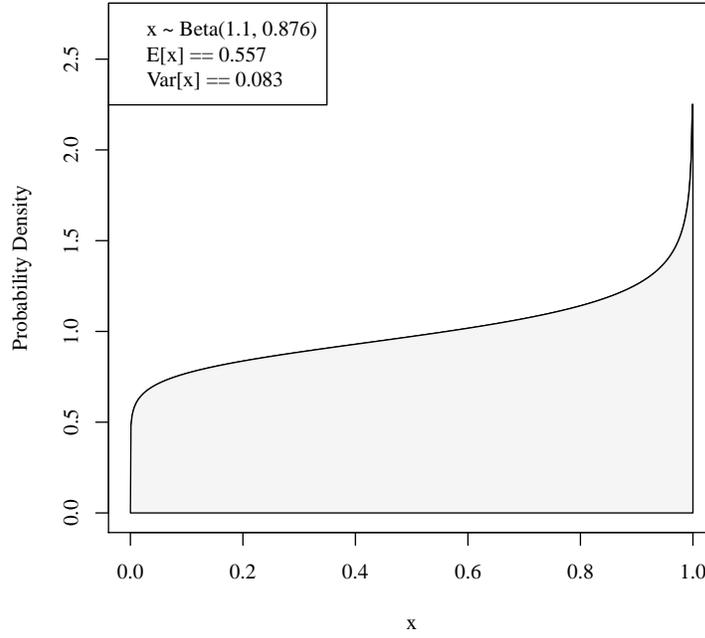
Researchers who conduct Monte Carlo experiments are usually aiming to compare several research procedures by applying them to randomly generated problems. For applied social scientists, this the most recognizable usage of the term “Monte Carlo.” The repeated application of a procedure to hundreds or thousands of simulated datasets will not constitute proof of a method’s superiority, but it will surely be serious evidence. To name just a few, this method of comparing procedures has been used in analysis of distributional tests (Thompson et al., 1967; D’Agostino & Rosman, 1974; Scott & Factor, 1981), regression (Huang & Bolch, 1974; McGee & Carleton, 1970; Royston & Thompson, 1995; Stefanski & Buzas, 1995), systems of equations (Foote, 1955; Wagner, 1958; Klein, 1960; Raj, 1980), comparison correlation estimators (Elston & Stewart, 1970; Kowalski, 1972; Srivastava & Keen, 1988), time series models (Beck & Katz, 1995; Granger & Hughes, 1968; Neave, 1972; Bhansali, 1973; Nelson & Schwert, 1982), multiple comparison procedures (Carmer & Swanson, 1973; Ramsey, 1978), and variance components (Boardman, 1974). Today, virtually every new statistical procedure is accompanied by a Monte Carlo simulation. The widespread use of this method for investigation of tools has brought calls for the creation of a more standardized methodology for the analysis and reporting of simulation tests Harwell (1992); Skrondal (2000); Paxton et al. (2001).

This kind of Monte Carlo simulation has shown itself to have strong benefits in the educational process. The old adage that “a single picture is worth a thousand words,” certainly applies. In their book *Statistical Methods for Social Scientists*, Hanushek & Jackson (1977) combined mathematical derivations of estimator properties with systematic Monte Carlo investigation. Experience indicates that students appreciate the power of mathematical proofs more meaningfully after they have seen evidence that a procedure “actually works.”

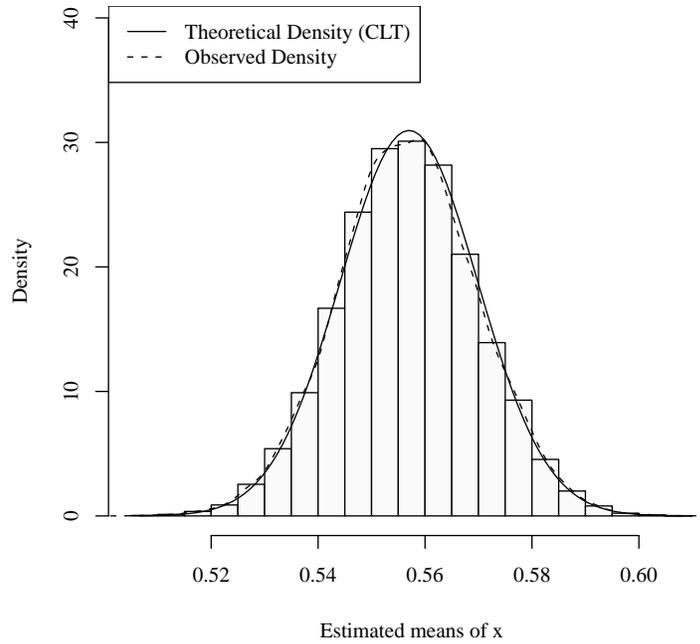
As a part of the educational role, Monte Carlo analysis is often used to demonstrate results for which we have formal derivations. Consider the Central Limit Theorem (CLT): the averages of repeated samples from a distribution (including non-normal distributions) will tend to be normally distributed. In Figure 1a, I illustrate the probability density of a variable following a beta distribution, a skewed, nonsymmetric distribution. Using the statistical software R (R Development Core Team, 2010), 10,000 samples of size 500 were drawn from the beta. The histogram of the means of those samples is presented in Figure 1b. Whereas the parent population is not symmetric or normal in the slightest, the means do appear to be normal. The Central Limit Theorem leads us to expect that the sampling distribution of the means will be normal in shape with a mean of about 0.557 and a variance of $0.00016 = 0.083/500$. The observed means match that prediction almost exactly. In Figure 1b, the solid line depicts the predicted normal probability that would correspond with those parameters and the dotted line is the observed “kernel density”. Note that the theoretical prediction of the CLT is almost exactly matched by the experimental means.

Monte Carlo simulation allows rapid exploration of informal conjectures that may be formalized later. Specific research problems may arise for which one has not yet found guidance in the literature. Suppose we are fitting a logistic regression model and one of the predictors is badly unbalanced. If a sample turns up many more women than men, for example, how reliable is the estimate of a “gender effect”? A hypothetical logistic model was constructed in which the “true” gender effect was 0.4. A collection of 1000 data sets was created in which males and females were equally represented, and then 1000 samples were drawn in which

Figure 1: The Sampling Distribution of Beta's Mean

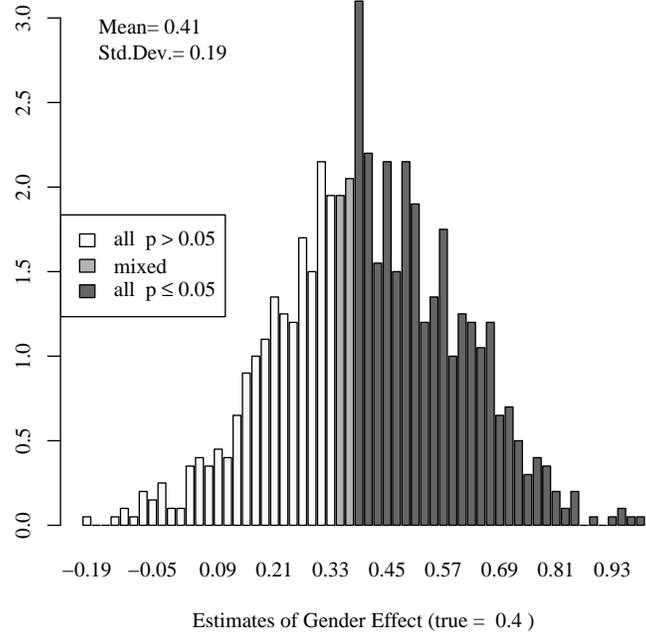


(a) A Beta Distribution

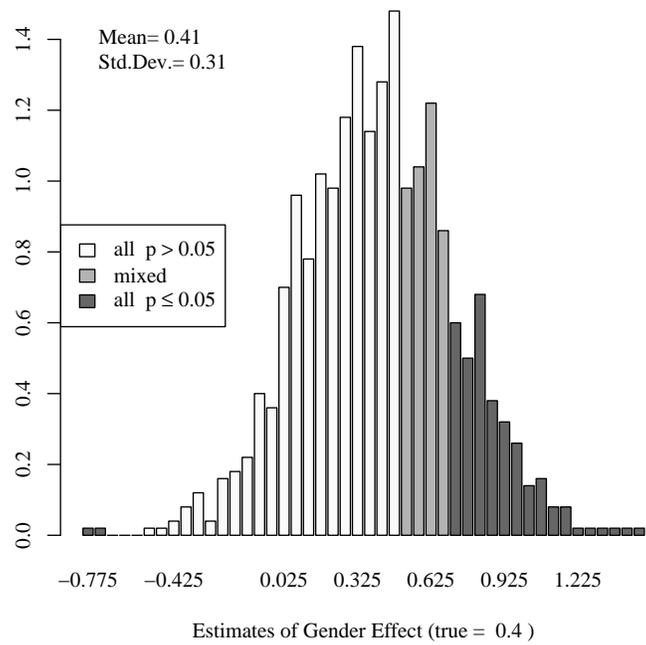


(b) Distribution of Means

Figure 2: The Impact of Imbalance in Logistic Regression



(a) Balanced Data: 50% female



(b) Unbalanced Data: 90% female

90% of the observations were females.

The effect of unbalanced samples is summarized in Figure 2, which compares the estimates when the gender split is 50-50 (balanced) with samples in which the split is 90-10 (badly unbalanced). The bars represent the density of estimates. At first glance, the estimates are encouraging. The “true gender effect” is 0.4, and the average of the estimates is close to 0.4 in both the balanced and unbalanced cases. However, when the sample is unbalanced, the distribution of estimates is more uncertain: the estimates of the gender effect are spread more widely and the standard errors estimated in the individual models are larger as well.

There is another serious consequence of the imbalance, one which I had not expected. The bars in the histogram are color-coded to summarize the “statistical significance” of the estimated coefficients in the runs. The dark gray bars indicate that all of the estimates in that range were deemed to be statistically significant, in the sense that $p \leq 0.05$ according to the Wald test. The white bars indicate that none of the estimates are statistically significant. Even in the balanced case, there are plenty of estimates that are not statistically significantly different from 0. Some textbooks indicate that when an estimate is “not significant,” no weight should be placed on its interpretation. One might be inclined to conclude that “gender doesn’t matter” and drop that variable from the model altogether. As a result, when gender is reported (i.e., when a case from a “dark bar” has manifested itself), the reported parameter estimates will tend to exaggerate the effect of gender. In the balanced case, the average of the significant coefficient estimates is 0.52, about 25% higher than the true value of 0.4. The mean of the significant estimates with unbalanced data is 0.78, *almost twice as large as the true value*. The fact that the estimates are, at the same time, both more uncertain and more biased presents us with a sobering assessment of the situation. After describing this finding to a colleague, I was directed to a now burgeoning literature on probable widespread bias in reported parameter estimates in published research (e.g., Dwan et al., 2008; Kyzas et al., 2007).

A final method that can be viewed as a member of this category is the so-called “Monte

Carlo hypothesis test.” Suppose there is no theoretical guidance on what to expect from a statistical estimator, but the process that is thought to generate the data can be simulated. Instead of treating the result in Figure 1b as an approximation of a sampling distribution, we now proceed as though it *actually is* the sampling distribution. If field data leads to the estimate of 0.99, far from the mean of 0.57, we would conclude that the data is probably not derived from the hypothesized process.

One might wonder how this MC hypothesis test is different from the well-known bootstrap estimation process (Efron & Tibshirani, 1993). Both of these tools are intended to solve the same problem: draw inferences when the sampling distribution of an estimator is unknown. However, they approach the problem from different directions. The bootstrap will repeatedly draw samples from a set of observations. The estimates from those “re-samples” are investigated in order to obtain an impression of the reliability of an estimator. When the estimates are clustered tightly in one part of the parameter space, one concludes that the standard error is low and thus a null hypothesis that is “far” from the estimate is probably wrong. The MC hypothesis test, on the other hand, only calculates one estimate from the observed data, but it calculates many possible estimates from random samples from the hypothesized model. If the one estimate appears to be grossly different than the simulated set of possibilities, then the null hypothesis is rejected.

The Monte Carlo hypothesis test can be thought of as an extension of the idea behind Fisher’s exact test (Fisher, 1922). The Fisher approach could exactly enumerate the full sample space and obtain the probability of each element, but only for small samples and specialized problems. For larger problems, the MC hypothesis test approximates that distribution by sampling. Algorithms have been developed to extend the exact test to some logistic regressions, for example, (Hirji et al., 1987; Mehta & Patel, 1995), and yet for larger problems, an approximation by simulation is necessary (Zamar et al., 2007). The MC hypothesis test is not discussed in most statistics texts, perhaps a signal that it is not considered necessary for most common statistical problems. Nevertheless, we can trace the use of this

tool back into the 1950s. Efforts to frame out a standard methodology have been offered from time to time (see Hope (1968); Jockel (1986); Besag & Clifford (1989)).

Some very well regarded applications of the MC hypothesis test and simulated sampling distributions are found in the analysis of spatial patterns. One recent stream of research follows the concepts proposed by Bartlett (1963) (see Besag & Diggle, 1977; Ripley, 1977; Marriott, 1979). Random processes are hypothesized to cause things (animals, plants, etc.) to be positioned across a space. After data is collected, one can check for clustering or un-predicted patterns by comparing observations against the hypothetical sampling distribution of various summary statistics. More recently, Manly (1997; 1995; Manly & Sanderson, 2002) has drawn the attention of researchers in ecology to this method by proposing a type of test for the distribution of features within a spatial environment (Raes & ter Steege, 2007; Lehsten & Harmand, 2006; Gotelli & Entsminger, 2003, 2001).

3.2 Markov Chain Monte Carlo: Approximating Solutions to Hard Problems

Nicholas Metropolis, the physicist who played such a prominent role in the first nuclear fission experiments at the University of Chicago and later in the Manhattan Project, is remembered most widely as the lead author on a paper that proposed the “Metropolis algorithm” (Metropolis et al., 1953; Hitchcock, 2003). The Metropolis algorithm is a simple idea with a very far-reaching set of implications. It is “the cornerstone of all Markov chain-based Monte Carlo methods” (Liu, 2001, p. 105) that have been at the forefront of methodological development in statistics and in many fields of science. It was recently called “one of the major contributions to theoretical chemistry of the twentieth century” (Jorgensen, 2000, 226).

The potential uses of calculations based on random numbers were anticipated by several mathematical developments in the 1920s and 1930s. Before the invention of the computer,

however, the actual use of these ideas was impractical. Stan Ulam, as Metropolis later recalled, felt that by the 1940s, “statistical sampling techniques had fallen into desuetude because of the length and tediousness of the calculations” (Metropolis, 1987).

To help the reader understand how the different pieces of the puzzle fit together, a thumbnail sketch of mathematical terminology is probably required. We might say there are 3 ways to “solve” for an unknown quantity in a mathematical problem.

1. Derive a closed form analytical solution.

Consider the quadratic equation:

$$y = ax^2 + bx + c \tag{1}$$

The values of x for which y is equal to 0 are known as “roots.” The famous solution for the roots is

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{2}$$

As another example, consider a simple statistical exercise: regression analysis. The theory is

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + e_i \tag{3}$$

where $\beta_j \in \mathbb{R}$ and $e_i \sim N(0, \sigma^2)$. In Ordinary Least Squares analysis, the unknown coefficients $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2)$ are found by minimizing the sum of squared errors, $\sum (y_i - \hat{y}_i)^2$, where the prediction formula is $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{1i} + \hat{\beta}_2 x_{2i}$. In matrix algebra, the solution is

$$\hat{\beta} = (X^T X)^{-1} X^T y \tag{4}$$

This formula is the famous solution that was discovered by Gauss in the late 18th century.

2. Calculate a numerical solution.

There are situations in which there is no closed formula with which to calculate an answer to a question. Nevertheless, there is a mathematical statement of an equation (or equations) that must hold exactly if a solution is to be found. Methods for finding numerical solutions are as old as the calculus itself; mathematicians have sought ways to approximate a function's slope, its roots, or the area under a curve.

The quadratic equation's roots can be found exactly. However, if the equation also includes higher powers, such as x^5 or higher, no such analytical solution exists. A numerical approach must be used to find the roots of the equation. Similarly, in the regression context, a change of the criterion for estimating $\hat{\beta}$ will generally prevent use of closed-form analytical solutions. Essentially all generalized linear models (McCullagh & Nelder, 1983) that do not use a normally distributed dependent variable will require numerical solution. Almost all models estimated by the principle of maximum likelihood require a numerical solution for the roots of complicated equations.

It is important to note that numerically derived estimates are not, in principle or interpretation, different from estimates that can be obtained analytically. They are simply more difficult to calculate. We act as though there's a number $\hat{\beta}$ and we calculate it.

3. Approximate a solution by Monte Carlo simulation.

Suppose that a problem cannot be solved directly or even numerically. Nevertheless, one might be able to derive a range of likely values and their probabilities. That was the situation in which Nicholas Metropolis and his colleagues found themselves when they introduced the Metropolis algorithm. To summarize the tendencies of a system, they sought to “average across” the many different positions in which the system could exist. The authors observed, “It is evidently impractical to carry out a several hundred-dimensional integral by the usual numerical methods, so we resort to the Monte Carlo method. The Monte Carlo method for many-dimensional integrals

consists simply of integrating over a random sampling of points instead of over a regular array of points”(Metropolis et al., 1953, 1088).

To understand the difference in this approach, note that we are no longer attempting to calculate the “one right number,” either analytically or numerically. Instead, we might need to derive hundreds or thousands of estimates of a number, and then draw conclusions that take our uncertainty into account.

As an effort at a simple explanation of this approach, I would offer the following. Recall from elementary statistics that the average of a random sample of scores,

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \tag{5}$$

is interpreted as an estimate of the “expected value” of a continuous probability distribution. The expected value is, of course, an integral. Let $\pi(x)$ represent the “true probability” of observing x . The expected value, $E[x]$, is defined as:

$$E[x] = \int \pi(x) \cdot x \, dx \tag{6}$$

The “Law of Large Numbers” asserts that, as N grows larger, the mismatch between $E[x]$ and \bar{x} shrinks.

The procedure known as “Monte Carlo integration” will have us reconsider that problem from the other direction. Theory leads us to believe there is a probability process, $\pi(x)$, that is generating data. We want to understand its properties, one of the ways we do so is to calculate an integral, such as expression (6). However, we have no analytical solution for that integral. If we can draw random observations from $\pi(x)$, we can approximate that integral by calculating the sample average. As long as we draw enough observations, we are

confident that the approximate solution is reasonably accurate.

This example does not seem so imposing because it has only one dimension under consideration. Numerical approximations will almost always out-perform Monte Carlo approximations in one dimension. However, when there are many dimensions, the Monte Carlo strategy can succeed where the numerical approach might fail altogether.

Consider a system that has, say, 10 characteristics.

$$(x_1, x_2, \dots, x_{10}) \tag{7}$$

We theorize that there a probability process that causes the system to “evolve” over time by skipping from one position to the next. The Monte Carlo model is intended to imitate that theoretical adjustment process. Begin at time 1 with a randomly selected position, $x^{(1)}$, and then repeat the Metropolis algorithm over and over:

$$\begin{aligned} \textit{time 1} \quad x^{(1)} &= (x_1^{(1)}, x_2^{(1)}, \dots, x_{10}^{(1)}) \\ \textit{time 2} \quad x^{(2)} &= (x_1^{(2)}, x_2^{(2)}, \dots, x_{10}^{(2)}) \\ &\vdots \\ \textit{time } k \quad x^{(k)} &= (x_1^{(k)}, x_2^{(k)}, \dots, x_{10}^{(k)}) \\ &\vdots \\ \textit{time stop} \quad x^{(\textit{stop})} &= (x_1^{(\textit{stop})}, x_2^{(\textit{stop})}, \dots, x_{10}^{(\textit{stop})}) \end{aligned} \tag{8}$$

As we repeat the process, we are exploring the space of possible system positions. After k steps, we believe our model has reached its equilibrium distribution. Once the equilibrium distribution is obtained, the chance of moving from one position to another is fixed (the probability model is “converged”), so sampled cases will reflect that system’s tendencies. After time k , we harvest a few thousand observations as the system moves from one position to another. Then the collection of vectors $x^{(k)}, x^{(k+1)}, \dots, x^{(\textit{stop})}$ gives us a sample of the

system's tendencies. The frequency of outcomes after k is an approximation of $\pi(x)$.

The true genius of the paper by Metropolis et al. (1953), of course, is that they proposed a way to make all of this actually work. The initial values of the system are $x_1^{(1)}, x_2^{(1)}, \dots, x_{10}^{(1)}$. A “proposal mechanism” suggests new values. The proposed mechanism is, more or less, a random walk. The Metropolis algorithm always accepts proposals that are “better” (according to the extent to which the change makes the system more closely approximate the theoretical model), and sometimes it accepts proposals that are “worse.” When the algorithm drops the system into an “unlikely” position, the next step will propose a random adjustment that will almost certainly be better, so the system will not stay in the bad region very long. This self-correcting aspect means that when the full history of the process is considered, the simulated system spends just a small amount of time in “unlikely” spots, and it spends more time in “good” spots. Metropolis, et al (1953) showed that the long run frequency of positions summarized in the chain is representative of the theoretical probability model $\pi(x)$. The system is forced to visit the “unlikely” spots only because we want to make sure they really are unlikely, and the fact that the system does not stay there is evidence that they are unlikely positions. The one-step proposal system is called a Markov chain in honor of Russian mathematician Andrei Markov, who pioneered the study of systems in which the move from $x^{(i)}$ to $x^{(i+1)}$ depends only on information available at time i .

The original Metropolis algorithm was concerned with the potential energy of a set of N particles. Proposals that have lower potential energy among all of their parts are “better” than others, and the simulation ends up generating a sample that is representative of the likely energy states of the system. They proved that there is some time k after which the simulation of the system generates numbers that match the theoretical distribution that they are seeking to understand. In other words, the collection of observed outcomes $x^{(k)}$ through $x^{(stop)}$ meaningfully represents the distribution of outcomes that would be observed if this system were re-created and re-run many times.

Everything else, as they say, is detail work. There have been many practical contributions

that improved the performance of the algorithm (perhaps most notably by Hastings (1970)). One can find many excellent comprehensive reviews of the Metropolis algorithm and the Markov chain modeling strategy that it inspired (Lemieux, 2009; Liu, 2001; Robert, 2010; Robert & Casella, 2009). Many new approaches have been suggested to improve the proposal mechanism, speedup calculations, make k smaller, and to enhance the statistical quality of the output.

In the early 1980s, applied research interest in Monte Carlo simulation of Markov chains was rekindled. By the end of that decade, most “research methodologists” in physical and social science had become aware of these applications. Two applications of the method, optimization via simulated annealing and the Markov chain Monte Carlo (MCMC) Bayesian parameter estimation, have had widespread impact.

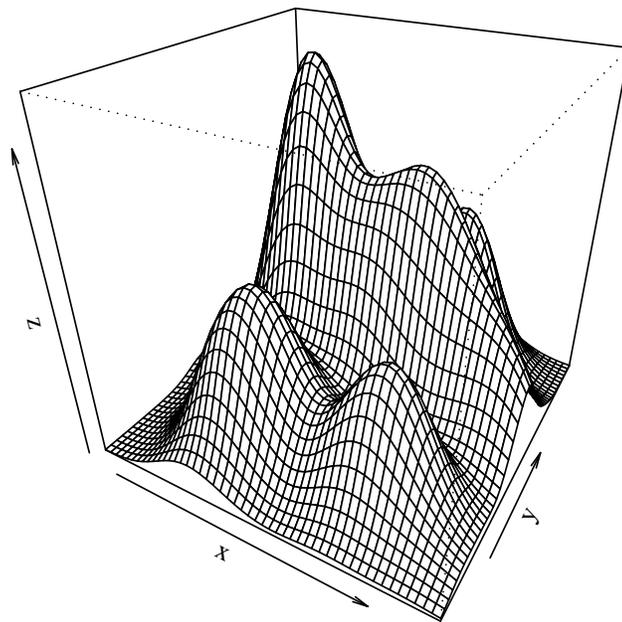
1. Optimization: Simulated Annealing

Since the calculus was invented, we have understood that the high and low points of a function are found where the slope is 0. If the function is “bumpy” or “rugged,” we are often uncertain about whether a solution is a “global maximum” or a “local maximum”.

To illustrate the problem, consider the irregular surface in Figure 3. Suppose we are assigned to find the (x, y) coordinates that correspond the maximum value of z . It is possible to imagine that we might wander about in the (x, y) plane, becoming trapped at the top of a small hill. A “hill climber” algorithm might reach the top of a mole hill and stop.

How can the Metropolis algorithm help? A paper by Kirkpatrick et al. (1983) showed the Metropolis proposal scheme can be used to improve the optimization process for these “bumpy” landscapes. The Metropolis algorithm generally goes up hill, but there is a chance that it will go downhill sometimes. Begin at some point, say $(x^{(i)}, y^{(i)})$, and then “tweak” one or two elements by adding a random value to create a new proposed position, $(x^{(i+1)}, y^{(i+1)})$. If the new proposed point is “better” according to the objective function, then it is accepted and becomes the system’s new position. The Metropolis algorithm will sometimes “walk

Figure 3: Irregular Surface



into a valley,” from whence the next random draw may lead it up a different hill toward a better outcome.

The adaptation of the Metropolis algorithm in this way is often called simulated annealing. It has been implemented as an optimization algorithm in many computer programs, including R’s optim function. The procedure has been widely investigated as a method of finding optimal solutions to problems in which there are many parameters (Vanderbilt & Louie, 1984;Suman & Kumar, 2006)

2. Bayesian Statistics: Markov chain Monte Carlo (MCMC)

Until the mid 1990s, many researchers (like me) thought that Bayesian statistics had a theoretically compelling foundation, but it was not useful. The math was too difficult. It was difficult not just in the sense that much careful mathematical study was required, but also in the sense that no amount of analytical mathematics would be likely to help. Even expert mathematical statisticians could not draw conclusions from many Bayesian models. Solutions were known to exist for only a small set of possible problems.

The fundamental Bayesian idea is that we ought to integrate our beliefs about the world with our statistical analysis of it. The competing view, dubbed the “frequentist” view, holds that a parameter is equal to a particular value (the “null” value) and if the sample estimate is “far enough” away from that value, we reject the original hypothesis completely. Despite the teaching of that method, most researchers will admit that they do not actually approach science in that way. If we believe that the average height of a male in the United States is 5’11”, and a sample estimate indicates that it is 6’4”, we don’t actually conclude that 5’11” was completely wrong. Rather we may think it is less likely to be correct. Our understanding of the world is not usually held as a “right” or “wrong” dichotomy. The Bayesian approach formally “updates” beliefs about parameters in light of observations. This approach appears to be both a more realistic description of what researchers actually do and also a better way to make decisions (DeGroot, 2004).

The Reverend Thomas Bayes's was probably not the first person to “discover” this principle, but his name is associated with it nevertheless (Stigler, 1983; Fienberg, 2006). Let $Pr(obs.)$ represent the probability of collecting a set of observations. Let $Pr(hyp.)$ be the probability that a particular hypothesis is correct. Usually $hyp.$ would be values for a set of parameters. Bayes's law holds that we can derive beliefs that reflect our observations, $Pr(hyp.|obs.)$ through this formula:

$$Pr(hyp.|obs.) = \frac{Pr(obs.|hyp.) \cdot Pr(hyp.)}{Pr(obs.)} \quad (9)$$

The left hand side, $Pr(hyp.|obs.)$, is the “posterior probability distribution,” which indicates how likely it is that a hypothesis is correct in light of the observed data. We don't intend to conclude that any particular hypothesis is correct. Instead, we want to be able to state how likely any each one is to be correct. $Pr(hyp.)$ is called a “prior” belief. It is a reflection of the researcher's experience. For example, a priori, we believe that the most likely height of a randomly drawn male is 5'11” and it is unlikely that we will find a 7' tall. Finally, $Pr(obs.|hyp.)$ is the “likelihood” that a given sample can occur if a given hypothesis is correct. The likelihood, of course, will be familiar to people who have conducted maximum likelihood analysis. Whereas a traditional maximum likelihood analysis would stop after finding a set of estimates that maximizes $Pr(obs.|hyp.)$, the Bayesian goes the extra step of blending that with previous beliefs.

This is the point at which Bayesian methodology becomes too difficult (or at least, it used to be). We would like to have a workable formula for calculating posterior probabilities, an analytical way of combining our beliefs with our sample. Some prior belief distributions do merge workably with the likelihood models (so-called conjugate distributions), but most do not. In practice, applied researchers quickly wander away from the safe path of workable models and into a forest of interesting but impractical models. This is true of maximum

likelihood analysis, of course, so it is not a uniquely Bayesian problem. But the practitioners of maximum likelihood analysis have learned to stay on the mathematically tractable path, whereas the Bayesian paradigm seems to invite us to wander away from it.

Consider as an example the so-called “hierarchical regression” or “mixed regression” model. Scholars are increasingly interested in taking the usual regression model, as in equation 3, and supposing that the parameters themselves are drawn from a random process. Suppose

$$y_{ijk} = \beta_0 + \beta_1 x1_i + \beta_2 x2_i + \beta_3 x3_i + e_i \quad (10)$$

A school student’s scores on a standardized test (y_{ijk}) reflect personal characteristics (the subscript i) as well as characteristics of the school (subscript j) and the city (subscript k). Other variables and random processes at those higher levels are thought to determine these other parameters

$$\beta_{2j} = \gamma_0 + \gamma_1 x4_j + u_j, \quad u_j \sim N(0, \sigma_u^2) \quad \beta_{3k} = \xi_0 + \xi_1 x5_k + v_k, \quad v_k \sim N(0, \sigma_v^2) \quad (11)$$

All of the unknowns are assumed to be normally distributed, so it is likely that this can be estimated by maximum likelihood as a mixed model with software such as lme4 (Bates & Maechler, 2010). For all practical purposes, it will simplify down to one equation.

Instead of assuming that there are normally distributed errors, suppose that there are random effects from some other distribution. Carlin et al. (2001) make a persuasive case in a study of smoking that the individual-level random effect needs to mix at least two distributions, one that may be normal, but another is concentrated near 0. Or, for another example, suppose a random effect has more extreme observations than the normal distribution will countenance. We might suppose that u_j is drawn from a t distribution, a distribution that has fatter tails (see Albert, 2007). Any wrinkle of that sort will probably turn this into a problem for which we do not have workable tools for maximum likelihood analysis. Maximum likelihood

calculations are prohibitively difficult, and, until recently, Bayesian analysis was unlikely to take us any further.

The MCMC approach gives the Bayesian statistician a workable strategy for this problem. The MCMC approach mirrors the Metropolis approach very closely. The vector of parameters to be estimated can be arranged like so :

$$\beta^{(1)} = (\beta_0^{(1)}, \gamma_0^{(1)}, \gamma_1^{(1)}, \sigma_e^{(1)}, \eta_0^{(1)}, \eta_1^{(1)}, \eta_2^{(1)}, \eta_3^{(1)}, \eta_4^{(1)}) \quad (12)$$

and then we would sample by creating a chain. We can calculate the probability that this vector is correct, then impose some random perturbations, and re-calculate. The so-called “burn in” period brings the model up to time k , after which it is said to have converged and the following samples are used to represent the posterior distribution.

This adaption of the Metropolis algorithm seems obvious in retrospect, but it was not recognized and put to use for about 40 years. Gelfand & Smith (1990) and Gilks & Wild (1992) were among the first to put the pieces of the puzzle together. Instead of the Metropolis algorithm, an update method known as Gibbs Sampling, which had been introduced for digital image reconstruction by Geman & Geman (1984), was incorporated by Gelfand and Smith. Gibbs sampling simplifies the problem of creating a proposed draw from the multivariate distribution by dividing the process into several one dimensional adjustments. We don't need to write down a probability model for the transition from the whole vector from one state to another. We only need to write down a shift for one parameter, taking all of the others as given. That is, we move from this starting position

$$(\beta_0^{(1)}, \gamma_0^{(1)}, \gamma_1^{(1)}, \sigma_e^{(1)}, \eta_0^{(1)}, \eta_1^{(1)}, \eta_2^{(1)}, \eta_3^{(1)}, \eta_4^{(1)}) \quad (13)$$

by drawing a new estimate of just one parameter:

$$(\beta_0^{(2)}, \gamma_0^{(1)}, \gamma_1^{(1)}, \sigma_e^{(1)}, \eta_0^{(1)}, \eta_1^{(1)}, \eta_2^{(1)}, \eta_3^{(1)}, \eta_4^{(1)}) \quad (14)$$

and then we draw an estimate of another parameter:

$$(\beta_0^{(2)}, \gamma_0^{(2)}, \gamma_1^{(1)}, \sigma_e^{(1)}, \eta_0^{(1)}, \eta_1^{(1)}, \eta_2^{(1)}, \eta_3^{(1)}, \eta_4^{(1)}). \quad (15)$$

This is possible because we can, more-or-less easily, because a conditional probability model for one parameter is relatively easy to derive (whereas a conditional model for all parameters is not). Gilks and Wild (1992) demonstrated that this conditional sampling strategy could be used reliably for complicated, hierarchical models. “We have shown that adaptive rejection sampling can be used as a black box routine for efficiently sampling from complex densities, in particular those arising in applications of Gibbs sampling to the analysis of hierarchical Bayesian models involving non-conjugacy” (p. 347). In other words, there is a meaningful approximation for the previously unsolvable problem. Around that same time, a lively debate following Geyer’s proposal (1992b) was evidence that many research teams were hard at work developing the theory of simulated chains (Gelman & Rubin, 1992; Tierney, 1994), diagnostics for the convergence of the process (Cowles & Carlin, 1996), working examples of applications to problems that researchers frequently encounter (Albert & Chib, 1993), and additional enhancements of the algorithms (Duffie & Glynn, 1995; Neal, 1994).

As great as they are, these insights would not have been so influential if they were not accompanied by high quality textbooks (Gelman et al., 2003; Gill, 2007; Jackman, 2009) and by computer software. The first widely available program, BUGS (“Bayesian Updating with Gibbs Sampling”) was circulated in the mid 1990s (Thomas, 1994; Gilks et al., 1994). It was accompanied by a thorough set of worked examples (Gilks et al., 1995). The implementation of WinBUGS (for Microsoft Windows operating system) made the Bayesian breakthrough widely accessible. The documentation included examples with discussion that educated the reader not only about WinBUGS, but Bayesian analysis more generally. The BUGS language

for model specification today lives on in the OpenBUGS project (Lunn et al., 2009). That language seems to have been accepted broadly in the community; it is also used in JAGS, Martyn Plummer’s new implementation (Plummer, 2010a,b), whose name is an acronym, Just Another Gibbs Sampler. For researchers who don’t want to learn the entire BUGS language framework to estimate basic models, there are several programs that have pre-packaged basic models with standard prior belief distributions (Martin et al., 2010; Hadfield, 2010; Rossi & McCulloch, 2008). A probit regression, for example, can be estimated with several R packages, including *MNP* (Imai & van Dyk, 2005a,b), *bayesm* (Rossi & McCulloch, 2008), or *MCMCpack* (Martin et al., 2010). These approaches typically allow one to adopt a simplified model of the prior, with the possibility that it can be uninformative, or “flat” (meaning it does not influence the posterior results very much). If one wants an “in between” approach, I’d recommend Albert’s R package *LearnBayes* (2010) and the associated textbook (2007). It supplies a workable set of building blocks for Bayesian estimation and provides a gateway to the more general BUGS modeling framework.

The most frequently asked question among my students has been, “what do I get in return for learning all of that Bayesian jargon and math?” It seems not persuasive to say, “you get to be a Bayesian!” That’s the correct answer, of course; one is freed from the limitations imposed by a certain way of thinking. If one adopts a Bayesian perspective, then models with unknown parameters, latent (unobserved) variables, and missing data all come into the focus of a single lens (Jackman, 2000a). One can fold the imputation of missing data into the MCMC analysis procedure, eliminating the need for a separate “multiple imputation” step that would ordinarily precede statistical analysis (Jackman, 2010).

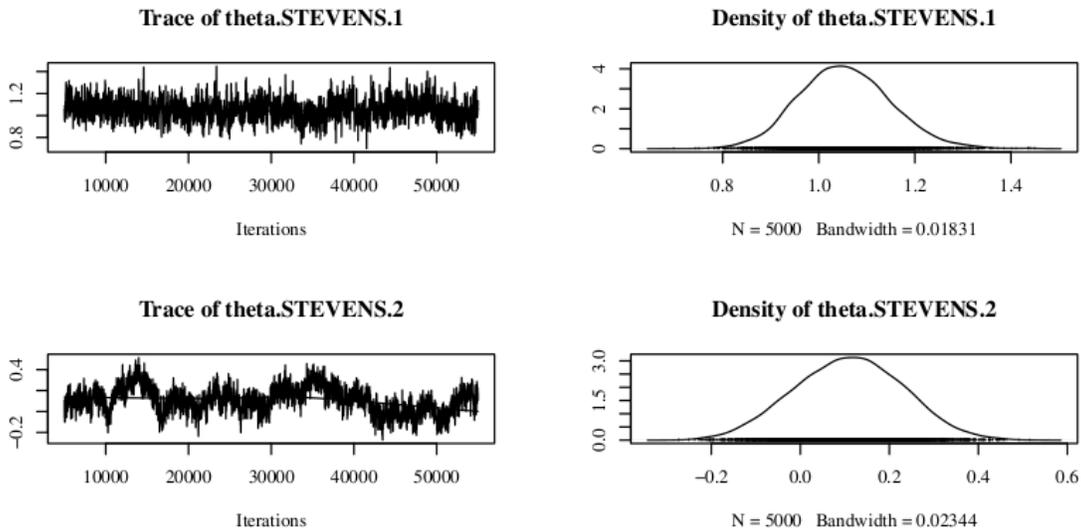
As an example of the MCMC experience, I have often presented a political science classic: the spatial voting model. Consider the problem of estimating the preferences of U.S. Senators from data on roll calls. From the “yeahs” and “nays”, we attempt to estimate each voter’s favorite position (ideal point) in an underlying (possibly multidimensional) space. Many political scientists will point to this as a foremost contribution of Bayesian analysis (Clinton

et al., 2004; Martin & Quinn, 2002, 2007; Jackman, 2000b). Adapting the concepts of Bayesian item response theory proposed in a path-breaking paper by (Albert, 1992), the multi-dimensional IRT estimation routines in *MCMCpack* can estimate either a one or a multidimensional preference model. (We might as well have used Jackman’s *ideal* estimator for R (2010), or the free-standing IRT package *MultiNorm* (Edwards, 2010)).

In Figure 4, a small bit of the output from a two dimensional model is presented. The figure represents just one Senator (Ted Stevens, Alaska), but any of the other Senators could have been selected for illustration. The estimation process included a 10,000 period burn-in, followed by 50,000 draws from the MCMC process. The plot on the left tracks all 60,000 estimates. The first 10,000 are thrown away, and then we “thin” the rest (a way of ameliorating autocorrelation). We keep only 1 in 10 estimates, leaving 5000 for the construction of the posterior density plots on the right side. However, we might not be finished, however. The chains, particularly the one on the bottom, may not have converged after 10,000 iterations, so we would apply some diagnostic checks. It may be necessary to throw away a much larger block of burn-in estimates. After a satisfactory set of results has been obtained, we might summarize the estimates for the individual voters by the modes or means of their posteriors.

In my experience, a few examples of problems that are otherwise unsolvable will go a long way to break down the resistance of the audience. Practitioners of “hierarchical models” are often framed in by their assumptions; MCMC offers a way out. In their leading textbook on hierarchical regression, Raudenbush & Bryk (2002) weave their way through the normal models, noting their limitations and pointing the reader in a Bayesian direction for the consideration of difficult cases. In his recent review of MCMC in psychology, Levy observes that, “A Bayesian framework ... supports the removal of historical boundaries that are likely to hinder the growth of substantively rich and methodological complex psychometric models” (Levy, 2009). One need only consider the range of examples provided with WinBUGS, or that which is surveyed in (Congdon, 2006), to gain the appreciation for the potential richness of

Figure 4: MCMC Estimation of Senator Ted Stevens Voting Tendencies



these models. Whereas the advocates for Bayesian tools may not have reached their audience before MCMC, they certainly have reached it today. Treatments of the Bayesian method have appeared the leading journals of many fields, including my field of political science (Western & Jackman, 1994; Jackman, 2000b, 2009).

The argument that Bayesian tools offer an “exact” view of the sampling distribution of parameter estimates is also persuasive. Typically, a frequentist will conduct t or Wald tests with the ratio $\hat{\theta}/s.e(\hat{\theta})$. When parameters have been estimated by maximum likelihood, those tests are not based on an exact characterization of the sampling distribution. Instead, they are based asymptotic (large sample) approximations. They are known to be correct only for infinite sample sizes. Proponents of the Bayesian MCMC claim their approach yields an “exact” representation of the sampling distribution, even if the sample is small (see Albert, 1992; McCulloch & Rossi, 1994). Estimates of the variance components in mixed models have unknown statistical properties, and only wishful thinking allows us to proceed by conducting ordinary hypothesis tests as if those parameters followed t distributions. Because of that problem, Baayen et al. (2008) suggest using MCMC to characterize the sampling

distributions of variance component estimates. A leading package for mixed models in R, `lme4`, implements that strategy (Bates & Maechler, 2010). That approach has also been proposed in ecological analysis (Clancy et al., 2010).

If the sales pitch for the Bayesian approach is still insufficient for the reader, I fall back to argue that simulation with MCMC may be helpful to frequentists who conduct maximum likelihood analysis. First, MCMC can facilitate the calculation of maximum likelihood estimates. Recall that the EM algorithm (Dempster et al., 1977) has been a staple in the calculation of maximum likelihood estimates. The E stands for Expectation, a procedure in which estimates for missing parameters are inserted to create a complete data set, and the M stands for Maximization. Wei & Tanner (1990) showed that MCMC simulation can be used to make the E step more practical. A number of similar approaches for the use of MCMC in the EM algorithm have been tested and found workable for particular classes of problems (Geyer, 1992a; Nielsen, 2000; Jank & Booth, 2003; Caffo et al., 2005; Marschner, 2001; Valpine, 2003). Second, very recent publications indicate that MCMC calculations can be used to derive ML estimates. Virtually the same algorithm was proposed in economics (Jacquier et al., 2007) and in ecological modeling (Lele et al., 2007). In the latter presentation, the procedure is given the memorable name “data cloning.” It is a blend of the “data augmentation” method for the EM algorithm (Tanner & Wong, 1987) and MCMC estimation. Both of the teams that propose this method claim it is fast and easy to use, portraying it as something of a magic bullet for difficult-to-estimate models. Lele et al. (2007) claim not only to produce ML estimates, but also a matrix of variance estimates that can be used to conduct the t or Wald hypothesis tests that frequentists usually employ (Ponciano et al., 2009).

3.3 Simulation Modeling and Hypothesis Construction

We have seen that Monte Carlo simulation can play a role in the evaluation of statistical procedures. It can also play the role of a connective tissue between complicated theoretical constructs that cannot otherwise talk to each other. In this section, we explore simulations that are used to derive theories and hypotheses. In this usage, MC simulation is not in principle different from mathematical formalization of a problem and the derivation of propositions from a model.

Suppose the research question is “How much money can a person earn by playing roulette a Casino?” We could hire a fleet of graduate students and bankroll them at the Flamingo Hotel in Las Vegas. This approach might be expensive, but that is not the worst problem. It leaves quite a few things to chance. Some students might bet carelessly, some might be distracted, some might take the money and play poker instead. If we could design a computer version of roulette, and then make a computer program that plays according to strategies we specify, then we might make some progress. Perhaps the authors would quibble with this characterization, but I’d say this is almost exactly what goes on at the genesis of projects like the Santa Fe Artificial Stock Market (Palmer et al., 1994; Johnson, 2002; Linn & Tay, 2007; Levy et al., 2000) or the so-called “minority game” (Challet et al., 2005), which flowed out of a whimsical story about Brian Arthur’s desire to hear Irish folk music in a not-too-crowded bar (Casti, 1996; Arthur, 1994).

When a computer program is designed so as to represent the behaviors of autonomous entities, it is often called an agent-based model (ABM) or an individual-based model (IBM). ABMs were originally developed (primarily) for the modeling of complicated environmental and natural systems (DeAngelis & Gross (1992); Grimm & Railsback (2005); Parker et al. (2003)), but social science usage has also resulted in some notable insights. The social science applications are surveyed in several textbooks (Gilbert & Troitzsch, 1999; Gilbert & Conte, 1995; Miller & Page, 2007). Apart from the economic study of markets and individualistic

decision-making (Luna & Stefansson, 2000), there are sociological approaches, with thematic applications, that can be found in *Growing Artificial Societies* (Epstein & Axtell, 1996) and *Turtles, Termites, and Traffic Jams* (Resnick, 1994).

Most ABMs rely on random numbers in two ways. First, the substance of the model might call for unpredictable events, such as changes in the weather, the stock market, or an election outcome, which are interpreted as exogenous shocks. Second, the researcher may use a sample from a statistical distribution to initialize the positions and characteristics of the agents. In either case, because the course of the simulation will reflect random input, it will be necessary to conduct a Monte Carlo analysis. The simulation will be run many times in order to ascertain the range of possibilities.

Social science simulation modeling has its roots in the study of cellular automata, the models on which von Neumann was working at the time of his death (Neumann & Burks, 1966). A cellular automaton is a “grid” or “lattice” of points that can be thought of as a checker board in which the squares can change color. Each cell will have transition rules, such as “if two of my neighbors are red, change my own color to black.” The grid’s main role is to determine the immediate neighbors of each cell. In a computer implementation of a cellular automaton, one can dispense with the grid concept altogether and simply define a neighborhood (a list of other cells) for each cell along with a status transition rule (Hegselmann, 1996; Nowak & Lewenstein, 1996).

Early social science applications of the cellular automata were not computer models, but rather they were conducted on a checker board or graph paper. Schelling’s model of neighborhood segregation was a pioneering effort. The squares on a board are homes, and markers of different colors represent the races of families that move about in order to find agreeable neighborhoods. A sharp separation of races can develop over time even if the families are relatively tolerant of each other (Schelling, 1971, 1978). This publication gave rise to a steady succession of studies of segregation (for example, Singh et al., 2009; Zhang, 2004; Aydinonat, 2007) and the “tipping models” of social behavior (Granovetter & Soong, 1988).

Tipping models are especially important in the history of simulation in social science because they appeal to the social scientist's intuition that interactive individual behaviors can have unexpected social consequences.

If we venture outside the confines of academic research, the most famous cellular automaton is *The Game of Life*, which was attributed to John Conway (Gardner, 1970). The Game is driven by simple rules that allow cells to remain lighted (alive) if they have a medium number of lighted neighbors. Cells can be turned off (die) if they are either too lonely or over-crowded. Some initial patterns can reproduce themselves endlessly, while others beget streams of strange, even bizarre patterns. Professionals and amateurs alike have been captivated by the seemingly endless variety. On the academic side, *Life* addresses the fundamental questions in computer science concerning the computational power of artificial machines. On the popular side, well, seeing is believing. The reader should do a Google search for "spaceship" in association with the Game of Life and play the interactive game at this Website: <http://www.conwaysgameoflife.net>.

Robert Axelrod's study of the Prisoner's Dilemma might be the most influential computer simulation in social science. The Prisoner's Dilemma (PD) is unique among two person games: each player has a dominant individual strategy to behave uncooperatively, and yet the payoffs of both players would be improved if they behaved differently. This conundrum, the apparent mismatch of individual incentives and social welfare, has fueled the study of the PD game. A computer tournament simulated social evolution by pitting strategies against each other and then rewarding successful strategies with more prevalence over time. The simulation led, somewhat unexpectedly, to the success of cooperative strategies (Axelrod, 1981, 1984). I'm more convinced now than ever that the Prisoner's Dilemma is the drosophila (fruit fly) of modern social science (Johnson, 1999). Love it, or hate it (Binmore, 1994), the simulations of the Prisoner's Dilemma in the last three decades outweigh any other topic, and by a considerable margin (see Hoffmann, 2000; Axelrod & D'Ambrosio, 1996; Gotts et al., 2003). Public opinion dynamics (to cite just a few, (Nowak et al., 1990; Latane, 1996;

Huckfeldt et al., 2004)) or competitive position taking by political parties Kollman et al., 1992, 1998; de Marchi, 1999; Laver, 2005; Laver & Schilperoord, 2007; Fowler & Laver, 2008 have also received a considerable amount of attention.

It seems certain that agent-based modeling has benefited from three developments in science. The first was the so-called “chaos theory,” which is often summarized by reference to the Lorenz model (1995), now commonly known as the “butterfly effect”(Gleick, 1988). Whereas scientists had assumed that a system that starts out in “roughly” the same position should generate “roughly” the same result, the chaos theorists found that virtually identical models could generate grossly different results (May, 1976). An especially highly prized result is a “bifurcation,” a “line in the parameter space” that separates systems that behave differently (for example, Nowak et al. (1994b); Nowak Martin & May Robert (1993); Nowak et al. (1994a)). The study of bifurcation is closely tied to the study of fractals, complicated geometrical designs that can evolve from simple mathematical expressions (Mandelbrot, 1983; Barnsley, 1993; *Wolfram, 2002*). Also highly prized, of course, is the opposite result which indicates that a system tends to evolve in a particular direction, regardless of where it starts or how it might be exogenously shocked. Perhaps the Schelling segregation model, or Axelrod’s culture model (Axelrod, 1997), would fall into this latter category.

The second development that dovetailed with the growth of ABM was the new science of “complexity” and the establishment of the Santa Fe Institute (Waldrop, 1994). A complex system has many loosely interconnected elements (Mitchell, 2009; Johnson, 2009). In most cases those elements include individual agents, such as models of people, animals, trees, etc. One main emphasis in this area of study is the development of “emergent” properties, which are defined as characteristics of systems that evolve without conscious guidance. Terms like “self-organized criticality” (Bak, 1999; Jensen, 1998), “hidden order” (Holland, 1996), “self-organization” (Camazine et al., 2003), “spontaneous order” (Kauffman, 1995, 1993) and “sync” (Strogatz, 2003) are all referring to this basic idea which, as one might expect, is open to many interpretations. Chris Langton, whose research on cellular automata (Langton,

1984, 1990) triggered the formation of a field of study called *Artificial Life* (Langton, 1995), contended that the individual pieces tend to adjust themselves over time to a position that he called “the edge of chaos.” In his model, systems that adapt well to stress are systems in which the individual components tend to position themselves close to the line of separation between stable and chaotic systems. Brian Arthur, an economist, found many examples of systems which seemed to defy the standard principles of his field (Arthur, 1999). A comprehensive collection of materials for economic applications is found at the Agent-Based Computational Economics Website (Tsfatsion, 2010).

A third development that dovetailed with the growth of simulation was a change in the field of computer science. The philosophy of object-oriented (OO) computer programming was introduced. The OO philosophy is almost exactly the same as the social science philosophy that motivates the agent-based model. OO endorses programs that separate information and functionality among types of objects. Information should be disclosed only through well defined protocols. Objects are thought of as representations of classes, which are conceptually organized from general to specific. Widely adopted languages such as C++ (Stroustrup, 1986), Objective-C (Cox, 1986), and later Java (Gosling et al., 2005), sought to make this a reality. The idea of having individual, autonomous agents in a simulation model could finally be implemented in a computer language that was based on the exact same idea. The introductory chapters in the Objective-C manual (Apple Computing, 2009) could as well be the introduction of a book on agent-based modeling.

Chris Langton, who was at the Santa Fe Institute, saw the potential of research with agent-based modeling, but was concerned that every simulation project was done “from scratch” using idiosyncratic concepts and code. There was no standard “workbench.” His team at the SFI proposed the Swarm Simulation System (Minar et al., 1996), a programming library, to address that problem. Some of the terminology of the Swarm project has filtered out to the research community, but it did not coalesce the community around a single tool. Instead, research teams sought to develop their own libraries. The Brookings Institution sponsored the

development of Ascape, the platform used by Epstein & Axtell (1996). Flowing out of the StarLogo framework (familiar to the readers of Resnick), software packages were made available from MIT (new variants of StarLogo) and Northwestern University (NetLogo Wilensky (1999)). The University of Chicago and Argonne National Laboratories sponsored RePast (REcursive Porous Agent Simulation Toolkit), and George Mason University’s Center for Social Complexity released MASON (Multi-Agent Simulation of Networks and Neighborhoods, (Luke et al., 2004)). This rendition includes only the most prevalently used libraries; through the years, quite a few other frameworks for software development have appeared. None of these has dominated the language or practice of agent-based modeling in the same way that the language of BUGS came to dominate MCMC research, or the way in which R has come to dominate development of statistical tools.

4 Practical Problems in the Immediate Future

The wave of development in Monte Carlo simulation has been driven by the urgency of the research questions and the ability of research teams to design programs that can get the job done. As those ideas and methods filter out to the broader class of academic practitioners, some problems are presenting themselves. In many of these cases, there is no simple or painless solution. It may be necessary to adopt significant changes in the way we conduct research and train students.

4.1 Replication

Replication has two meanings, one sharper and more demanding than the other. The looser meaning of replication is that we ought to be able to take someone’s MC project and re-write it in a different language (or on different computers) and the results should be comparable, “on average.” The sampling distributions of important estimators should be “about the

same.” The stricter standard of replication is that we should be able to reproduce results *exactly*, so that results coincide within the limits of precision in modern computers.

The looser standard for replication is important in practice. The value of a finding is made more certain when two different teams can design simulations in their own styles that produce roughly the same findings. Some computer models bring with them such a complicated combination of statistical and software concepts that we can never feel entirely confident that the results are completely understandable. That’s especially true in complex systems research, in which one objective is to design a system that produces unexpected results or emergent properties. Even if one has access to the code, it can be difficult to be sure that the unexpected result is substantively meaningful, rather than a glitch in the program.

The stricter standard for replication is also important, and yet it is almost universally ignored by practitioners. The ability to collect an exact set of records so as to re-run a model and reproduce the exact same results is one point of emphasis in John Chambers’s book, *Software for Data Analysis*. Chambers outlines some valuable strategies for management of micro-level details that facilitate precise replication. These steps are advocated as a part of his *Prime Directive* for developers of statistical research software. “The many computational steps between original data source and displayed results must all be truthful, or the effect of the analysis may be worthless, if not pernicious. This places an obligation on all creators of software to program in such a way that the computations can be understood and trusted” (Chambers, 2008, p. 3).

One of the problems that makes precise replication difficult is that researchers are sometimes unaware of the subtle differences between software implementations that can cause projects that are identical in “specification” to differ in practice. Recently, I noted that the same random number generator (Mersenne-Twister, MT19937) has been adopted as the default by SAS, R, Swarm, and countless other projects. On the surface, at least, that seems to imply that if one sets the same random seed to initialize the process, then one ought to be able to draw the exact same stream of random numbers. Documentation for most programs

is superficial, simply stating that the generator is MT19937 and referring the authors to the well known publication (Matsumoto & Nishimura, 1998). Many software users don't see the vagueness of that reference. I've done quite a bit of testing. Within SAS itself (or R, or any other project), one can repeatedly re-set the seed and then draw identical streams of numbers. However, one cannot set the seed to a given value in each program and then generate the same streams of random numbers. Theoretically, that should not happen, since the implementation of MT19937 is available directly from the developers on the project's Website (<http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/MT2002/emt19937ar.html>). The problem appears to be caused either by the usage of slightly different editions of the generator's C code, or (more discouragingly) poor implementations. In the open source projects, I've tracked down the differences to minor changes in the initialization of the random streams, but in the closed source programs, one can only guess about what causes the observed differences.

4.2 Making MC available to “the masses”.

For the sake of discussion, let's suppose that computer simulation is going to become an essential element in social science research. A significant overhaul in graduate training will be required. The graduate curriculum in American social science, at least if we judge by widely sold textbooks, remains under the control of the frequentists, not Bayesians. In order to make Bayesian MCMC estimation accessible for most students, a substantial amount of probability theory and mathematical statistics will have to be introduced. Apart from mathematical training, we also need training in computer programming. Most people earning Ph.D. degrees today in political science, sociology, or psychology have never written a program in a “low level” language like C or C++. Even though the implementations of BUGS have come closer to putting Bayesian statistics into a common, more-or-less workable language, the construction and interpretation of these models still requires a good deal of expertise and judgment. The BUGS Web page ends with this warning: “There

is, however, a need for caution. A knowledge of Bayesian statistics is assumed, including recognition of the potential importance of prior distributions, and MCMC is inherently less robust than analytic statistical methods. There is no in-built protection against misuse” (<http://www.mrc-bsu.cam.ac.uk/bugs>, April 21, 2011).

How is the danger ahead new and different? Commercial software. Until now, most cutting edge research software has been freely shared among research teams and considerable expertise is required to use those programs. The experts supervise each other, the rest of us benefit. That is changing, as the more user-friendly statistical packages like SAS and Mplus have begun to integrate some Bayesian options for MCMC simulation of parameter distributions. I am reminded of a warning offered by Andrew Hacker in *End of the American Era*(1970). He feared that the simplification of computer software packages, in combination with the ethos of “publish or perish,” would open the gates for a flood of silly research conducted by people who had not the slightest understanding of what they were doing. If that was a threat when SPSS made regression analysis widely accessible, one can only shudder at the danger from the dissemination of point-and-click simulation software.

Consider, for example, Mplus, a popular statistical package for structural equation modeling. The company offers an extensive user guide (<http://www.statmodel.com/ugexcerpts.shtml>) that has detailed instructions on how one might conduct a Monte Carlo simulation. The chapter on simulation explains how to set the seed of the program’s random generator, but there is no mention of what random generator algorithms are used or how those values are converted into statistical distributions. In the technical appendices and references, there are no citations to any random number generators or algorithms for the construction of statistical distributions. I understand that many researchers are using Mplus to conduct simulations, but I have to admit I’m concerned. Researchers who have purchased software feel, with some justification, that they have paid good money and they ought to be allowed to use the routines, even if they have no way of knowing what calculations are being conducted and there is no hope of replicating the results. If there ever was a violation of Chambers’s

Prime Directive, this is it. The warranty for Mplus offers users a refund if Mplus “does not perform in accordance with the accompanying documentation,” which is encouraging, except that the accompanying documentation is lacking in technical detail that might allow one to tell if the program is performing as documented.

There is a fairly persuasive argument that legitimate research software should be offered with code that is open for inspection. Seemingly small details, such as the algorithm for implementing MT19937 or calculating sample variance, can have a tremendous impact on the quality of the results. Commercial software companies do not agree, of course. Code and algorithms are trade secrets. Users are expected to trust the numbers they receive. The track record of some closed-source programs has been, well, poor (consider, for example, Microsoft Excel (McCullough & Heiser, 2008)). Access to the source code is most vital when we are on the “leading edge.” New software is most likely to have flaws, and researchers lack the breadth of experience that would help them guess that the code for a simulation package is mistaken.

4.3 Specification

A statistical model is a theoretical construct that approximates a data-generating process. What goes wrong if the data-generating conditions are different from the assumptions of the theoretical model? It is usually difficult to say. We don’t often ask, “what if I fit the wrong model?” In fact, when new procedures are proposed, they are usually accompanied by a Monte Carlo simulation that generates data according to a known process, and the statistical estimator is then shown to uncover the known properties of the data-generating process.

As time goes by, statistical models are often subjected to stresses so that we can find out what goes wrong when the theory that inspires the model does not match the data-generating process. The linear regression model would be a foremost example. We teach the additive

model with normal error:

$$y_i = \beta_0 + \beta_1 x_{1i} + e_i, \quad i \in \{1, \dots, n\}, \beta_j \in \mathbb{R}, j \in \{1, 2\}, e_i \sim N(0, \sigma^2). \quad (16)$$

After that, we consider the possible dangers of applying the estimator for that normal additive model to data that comes from other data-generating processes. What if the error's variance is not homogeneous? or the error is not normal? and so forth. We have a pretty good idea of the distortion that these things cause, and there are competing families of fixes for them. There is a growing set of robust estimators for regression models (Venables & Ripley, 2002). In that context, robust means that the estimate of β_1 , for example, is (by some standard) good, even if the assumptions about e_i are violated.

The major challenge for users of new statistical tools is that there is no powerful, universally applicable method to diagnose the mismatch between the theory and the data-generating process. We usually believe that new procedures work when they are fitted to the “right kind” of data. Otherwise, ambiguity reigns. In structural equation modeling, the proliferation of indices of “model fit” is a sure indicator of the problem. We do not agree on the kinds of mismatch that are most likely to arise in research and we do not agree on whether the mismatch causes harm to the parameter estimates. In the hierarchical, multi-equation models that are being explored with MCMC tools, the situation more problematic. Consider the Bayesian claim that the Markov chain converges to the exact distribution of the parameter estimates. That is true if the model as written matches the data-generating process. If the model does not match the data-generating process, we have, put bluntly, no idea what the posterior distributions mean. Of course, the same is true in maximum likelihood estimation. The claim that a parameter estimate is asymptotically normal pre-supposes that the assumed model is correctly specified. Many critics are quick to point out that the ratio of $\hat{\beta}/s.e.(\hat{\beta})$ is only distributed as a t statistic when the sample is infinitely large. Most have not been too concerned about the fact that, if wrong probability model is put to use, the distribution

that estimator is completely unknown, no matter how large the sample size might be.

In agent-based modeling, model misfit appears where the computer implementation of some details does not match one's substantive understanding of the problem. This is especially important in the effort to incorporate the passage of time in simulation models. The simulated agents behavior can affect the world, and the scholar's intuition about the passage of time and the inter-weaving of many separate actions into the time line may not match the computer implementation. Agents observe their world and adapt their behavior, but which agents and when? Albert Einstein is credited with the comment, "The only reason for time is so that everything doesn't happen at once." This is absolutely true in computer modeling. A computer's central processing unit manages instructions in a designated sequence; we attempt to simulate simultaneity by manipulating the model.

In the oldest tradition of computer simulation, the passage of time was represented as discrete steps at which all agents decided what to do at the exact same instant. In a cellular automaton, each cell has a "snapshot" of the world and each adjusts against it. That imposes synchronous patterns of action that are not generally reproduced if the cells update one-at-a-time. A theorist might suppose that individual actions are triggered by a dynamic, flexible system of triggers and the implementation of that idea turns out to require a great deal of care. We expect that the scheduling framework can matter, but most of the time we do not know what differences might be observed. One exception would be the spatial prisoner's dilemma (SPD) game. May and Nowak presented results for the SPD (1992; 1993); Huberman and Glance contended that the results were an artifact of the "everybody acts at once" model (1993). Follow-up studies have rebutted the largest part of the criticism by Huberman and Glance, but there are some contexts in which the scheduling framework does matter (Nowak et al. (1994b, 1996); see also Newth & Cornforth (2009) and Axtell et al. (1996)).

5 Conclusion

This essay has surveyed “Monte Carlo analysis,” a collection of the research methods that depend on computer generated random numbers. In an effort to convey the breadth of the potential applications, the use of pseudo random number generators has been explored in several phases of the research process.

To social scientists, the term “Monte Carlo analysis” refers to a procedure for evaluating statistical estimators. A Monte Carlo analysis involves application of estimators to many simulated data sets. One hopes to demonstrate that one procedure is more accurate or less uncertain than another.

On the other hand, to physicists and chemists in the mid 20th century, “Monte Carlo analysis” refers to a way of finding approximate solutions to intractable problems. Mathematical theories of matter and energy led to models that could not be solved. A Monte Carlo analysis draws a sequence of observations from that model to build a “map” of that system’s tendencies. That type of MC analysis was predominantly used in the natural sciences until the 1990s, when it found broad application in the Bayesian statistical approach known as Markov chain Monte Carlo (MCMC). Statistical models for which parameter estimates could not be derived by other approaches seemed more amenable to the Bayesian approach, but only after the introduction of MCMC did that potential become reality.

Finally, scholars in social sciences, ecology and land use were at the forefront of yet another type of “Monte Carlo analysis.” These computer models are often proposed as “realistic,” yet “mathematically unworkable” characterizations of “real world” processes. The growth of agent-based computer simulation models offered the hope of *A New Kind of Science* (Wolfram, 2002), one in which social (systemic) patterns were understood as an accumulation of individual behaviors. Because the interaction of animals (human or otherwise) and their environment can depend on many unpredictable events (weather, genetic mutation, etc.), computer generated pseudo random numbers have an obvious role. New scientific models

that incorporate nonlinearity and unpredictability (theories of chaos and complexity) found a natural expression in computer simulation. This new science, which seems to address the “really big questions,” such as the origin of life (e.g., Kauffman, 1995), captured the imaginations of many.

Even though there have been many accomplishments in the use of MC simulation, one should remember that the traditional approach was dominant for more than two centuries and, to a large extent, it still is. There will always be tension, or at least an uncomfortable interdependence, between traditional “mathematical solutions” and “simulation approximations.” Even though the mathematicians at Los Alamos championed the simulation approach, there’s no doubt they would rather have had “definite”, “predictable” answers for the problems with which they were presented. Some physical processes appear to be truly unpredictable, so computer generated random numbers were a realistic approach. Some mathematical problems could not be answered without simulation. Nevertheless, most scientists would rather have a formal theorem than a simulation.

Bauer’s early survey of Monte Carlo simulation focused most of its attention on mathematical problems with which sampling could help, but he held out the hope that “most fruitful application of the method” (1958, 449) would be found in the investigation of problems for which there was no “mathematical expression.” Simulation would not always be the last choice, or so it was hoped. The most widely accepted procedures based on random sampling, the Metropolis algorithm and MCMC, are situated at the ideal position: they have been shown to “approximately solve” an otherwise unsolvable problem and there is a formal proof that the approximation is meaningful. Probability theory leads us to expect that, if we did let the Markov Chain run “forever,” the draws would trace out the system’s tendencies with virtually complete accuracy. We do not have as much theoretical support for other applications of MC simulations, and for that reasons, conventional scientists are “withholding judgment” on simulation results that do not yet have theoretical grounding.

References

Albert, J. (2007). *Bayesian computation with R*. Springer.

Albert, J. (2010). *LearnBayes: Functions for Learning Bayesian Inference*. R package version 2.11.

Albert, J. H. (1992). Bayesian estimation of normal ogive item response curves using gibbs sampling. *Journal of Educational and Behavioral Statistics*, 17(3), 251–269.

Albert, J. H. & Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422), 669–679.

Apple Computing (2009). The Objective-C programming language: Introduction to the Objective-C programming language. <http://developer.apple.com/mac/library/documentation/Cocoa/Conceptual/ObjectiveC/Introduction/>

Arthur, W. B. (1994). Inductive reasoning and bounded rationality. *The American Economic Review*, 84(2), 406–411. ArticleType: primary_article / Issue Title: Papers and Proceedings of the Hundred and Sixth Annual Meeting of the American Economic Association / Full publication date: May, 1994 / Copyright © 1994 American Economic Association.

Arthur, W. B. (1999). Complexity and the economy. *Science*, 284(5411), 107–109. ArticleType: primary_article / Full publication date: Apr. 2, 1999 / Copyright © 1999 American Association for the Advancement of Science.

Aspray, W. (1990). *John Von Neumann and the Origins of Modern Computing*. History of computing. Cambridge, Mass: MIT Press.

Axelrod, R. (1981). The emergence of cooperation among egoists. *The American Political Science Review*, 75(2), 306–318. ArticleType: primary_article / Full publication date: Jun., 1981 / Copyright © 1981 American Political Science Association.

Axelrod, R. (1997). The dissemination of culture: A model with local convergence and global polarization. *The Journal of Conflict Resolution*, 41(2), 203–226. ArticleType:

primary_article / Full publication date: Apr., 1997 / Copyright © 1997 Sage Publications, Inc.

Axelrod, R. & D'Ambrosio, L. (1996). *Annotated Bibliography on The Evolution of Cooperation*.

Axelrod, R. M. (1984). *The Evolution of Cooperation*. New York: Basic Books.

Axtell, R., Axelrod, R., Epstein, J. M., & Cohen, M. D. (1996). Aligning simulation models: A case study and results. *Computational and Mathematical Organization Theory*, 1(2), 123–141.

Aydinonat, N. E. (2007). Models, conjectures and exploration: an analysis of schelling's checkerboard model of residential segregation. *Journal of Economic Methodology*, 14(4), 429.

Baayen, R., Davidson, D., & Bates, D. (2008). Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language*, 59, 390–412.

Baines, A. (1962). Statistics and computers. *Journal of the Royal Statistical Society. Series D*, 12(1), 32–38.

Bak, P. (1999). *How Nature Works: The Science of Self-Organized Criticality*. Springer, 1 edition.

Barnett, V. D. (1962). The Monte Carlo solution of a competing species problem. *Biometrics*, 18(1), 76–103.

Barnsley, M. F. (1993). *Fractals Everywhere*. Morgan Kaufmann Pub, 2 sub edition.

Bartlett, M. S. (1963). The spectral analysis of point processes. *Journal of the Royal Statistical Society. Series B*, 25(2), 264–296.

Bates, D. & Maechler, M. (2010). *lme4: Linear mixed-effects models using S4 classes*. R package version 0.999375-33.

- Bauer, W. F. (1958). The Monte Carlo method. *Journal of the Society for Industrial and Applied Mathematics*, 6(4), 438–451.
- Beck, N. & Katz, J. N. (1995). What to do (and not to do) with Time-Series Cross-Section data. *The American Political Science Review*, 89(3), 634–647.
- Besag, J. & Clifford, P. (1989). Generalized Monte Carlo significance tests. *Biometrika*, 76(4), 633–642.
- Besag, J. & Diggle, P. J. (1977). Simple Monte Carlo tests for spatial pattern. *Journal of the Royal Statistical Society. Series C*, 26(3), 327–333.
- Bhansali, R. J. (1973). A Monte Carlo comparison of the regression method and the spectral methods of prediction. *Journal of the American Statistical Association*, 68(343), 621–625.
- Binmore, K. (1994). Review of "the complexity of cooperation". *Journal of Artificial Societies and Social Simulation*, 1(1).
- Blumstein, A. (1957). A Monte Carlo analysis of the ground controlled approach system. *Operations Research*, 5(3), 397–408.
- Boardman, T. J. (1974). Confidence intervals for variance components – a comparative Monte Carlo study. *Biometrics*, 30(2), 251–262.
- Caffo, B. S., Jank, W., & Jones, G. L. (2005). Ascent-Based Monte Carlo Expectation-Maximization. *Journal of the Royal Statistical Society. Series B*, 67(2), 235–251.
- Camazine, S., Deneubourg, J., Franks, N. R., Sneyd, J., Theraula, G., & Bonabeau, E. (2003). *Self-Organization in Biological Systems*. Princeton University Press.
- Carlin, J. B., Wolfe, R., Brown, C. H., & Gelman, A. (2001). A case study on the choice, interpretation and checking of multilevel models for longitudinal binary outcomes. *Biostatistics*, 2(4), 397–416.
- Carmer, S. G. & Swanson, M. R. (1973). An evaluation of ten pairwise multiple comparison

- procedures by Monte Carlo methods. *Journal of the American Statistical Association*, 68(341), 66–74.
- Casti, J. (1996). Seeing the light at el farol. *Complexity*, 1(5).
- Challet, D., Marsili, M., & Zhang, Y. (2005). *Minority games*. Oxford University Press.
- Chambers, J. M. (2008). *Software for Data Analysis: Programming with R*. Statistics and computing. New York: Springer.
- Clancy, D., Tanner, J. E., McWilliam, S., & Spencer, M. (2010). Quantifying parameter uncertainty in a coral reef model using Metropolis-Coupled Markov Chain Monte Carlo. *Ecological Modelling*, 221(10), 1337–1347.
- Clinton, J., Jackman, S., & Rivers, D. (2004). The statistical analysis of roll call data. *The American Political Science Review*, 98(2), 355–370.
- Congdon, P. (2006). *Bayesian statistical modelling*. John Wiley & Sons.
- Cowles, M. K. & Carlin, B. P. (1996). Markov Chain Monte Carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association*, 91(434), 883–904.
- Cox, B. J. (1986). *Object-Oriented Programming ; an Evolutionary Approach*. Reading, Mass: Addison-Wesley.
- Crane, R. R., Brown, F. B., & Blanchard, R. O. (1955). An analysis of a railroad classification yard. *Journal of the Operations Research Society of America*, 3(3), 262–271.
- D’Agostino, R. B. & Rosman, B. (1974). The power of geary’s test of normality. *Biometrika*, 61(1), 181–184.
- de Marchi, S. (1999). Adaptive models and electoral instability. *Journal of Theoretical Politics*, 11(3), 393 –419.
- DeAngelis, D. L. & Gross, L. J. (1992). *Individual-Based Models and Approaches in Ecology: Populations, Communities, and Ecosystems*. New York: Chapman & Hall.
- DeGroot, M. H. (2004). *Optimal Statistical Decisions*. Wiley-Interscience, 1 edition.

- Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1), 1–38.
- Duffie, D. & Glynn, P. (1995). Efficient Monte Carlo simulation of security prices. *The Annals of Applied Probability*, 5(4), 897–905.
- Dwan, K., Altman, D. G., Arnaiz, J. A., Bloom, J., Chan, A., Cronin, E., Decullier, E., Easterbrook, P. J., Elm, E. V., Gamble, C., Ghersi, D., Ioannidis, J. P. A., Simes, J., & Williamson, P. R. (2008). Systematic review of the empirical evidence of study publication bias and outcome reporting bias. *PLoS ONE*, 3(8), e3081.
- Edwards, M. (2010). A Markov Chain Monte Carlo approach to confirmatory item factor analysis. *Psychometrika*, 75(3), 474–497.
- Efron, B. & Tibshirani, R. (1993). *An Introduction to the Bootstrap*. Number 57 in Monographs on statistics and applied probability. New York: Chapman & Hall.
- Ehrenfeld, S. & Ben-Tuvia, S. (1962). The efficiency of statistical simulation procedures. *Technometrics*, 4(2), 257–275.
- Elston, R. C. & Stewart, J. (1970). A new test of association for continuous variables. *Biometrics*, 26(2), 305–314.
- Epstein, J. M. & Axtell, R. (1996). *Growing Artificial Societies Social Science from the Bottom Up*. Complex adaptive systems. Washington, D.C: Brookings Institution Press.
- Everson, P. J. & Morris, C. N. (2000). Simulation from wishart distributions with eigenvalue constraints. *Journal of Computational and Graphical Statistics*, 9(2), 380–389.
- Fienberg, S. E. (2006). When did Bayesian inference become "Bayesian"? *Bayesian Analysis*, (pp. 1–41).
- Fisher, R. A. (1922). On the interpretation of χ^2 from contingency tables, and the calculation of p. *Journal of the Royal Statistical Society*, 85(1), 87–94.

- Foote, R. J. (1955). A comparison of single and simultaneous equation techniques. *Journal of Farm Economics*, 37(5), 975–990.
- Fowler, J. H. & Laver, M. (2008). A tournament of party decision rules. *Journal of Conflict Resolution*, 52(1), 68–92.
- Gardner, M. (1970). Mathematical games - the fantastic combinations of john conway's new solitaire game, life. *Scientific American*, (pp. 120–123).
- Gelfand, A. E. & Smith, A. F. M. (1990). Sampling-Based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410), 398–409.
- Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2003). *Bayesian Data Analysis, Second Edition*. Chapman & Hall, 2 edition.
- Gelman, A. & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4), 457–472.
- Geman, S. & Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the Bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-6(6), 721–741.
- Geyer, C. J. (1992a). On the convergence of Monte Carlo maximum likelihood calculations. *Journal of the Royal Statistical Society. Series B*, 56, 261–274.
- Geyer, C. J. (1992b). Practical Markov Chain Monte Carlo. *Statistical Science*, 7(4), 473–483.
- Gilbert, G. N. & Conte, R. (1995). *Artificial Societies: The Computer Simulation of Social Life*. London: UCL Press.
- Gilbert, G. N. & Troitzsch, K. G. (1999). *Simulation for the Social Scientist*. Taylor and Francis, Inc.
- Gilks, W., Richardson, S., & Spiegelhalter, D. (1995). *Markov Chain Monte Carlo in Practice: Interdisciplinary Statistics*. Chapman and Hall/CRC, 1 edition.

- Gilks, W., Thomas, A., & Spiegelhalter, D. (1994). A language and a program for complex Bayesian modelling. *The Statistician*, 43, 169–78.
- Gilks, W. R. & Wild, P. (1992). Adaptive rejection sampling for gibbs sampling. *Journal of the Royal Statistical Society. Series C*, 41(2), 337–348.
- Gill, J. (2007). *Bayesian Methods: A Social and Behavioral Sciences Approach, Second Edition*. Chapman and Hall/CRC, 2 edition.
- Gleick, J. (1988). *Chaos: Making a New Science*. Penguin Group USA.
- Gosling, J., Joy, B., Steele, G., & Bracha, G. (2005). *The Java Language Specification, Third Edition*. Amsterdam: Addison-Wesley Longman, 3 edition.
- Gotelli, N. & Entsminger, G. (2001). Swap and fill algorithms in null model analysis: rethinking the knight’s tour. *Oecologia*, 129(2), 281–291.
- Gotelli, N. J. & Entsminger, G. L. (2003). Swap algorithms in null model analysis. *Ecology*, 84(2), 532–535.
- Gotts, N. M., Polhill, J. G., & Law, A. N. R. (2003). Agent-Based simulation in the study of social dilemmas. *Artificial Intelligence Review*, 19(1), 3–92.
- Granger, C. W. J. & Hughes, A. O. (1968). Spectral analysis of short Series–A simulation study. *Journal of the Royal Statistical Society. Series A*, 131(1), 83–99.
- Granovetter, M. & Soong, R. (1988). Threshold models of diversity: Chinese restaurants, residential segregation, and the spiral of silence. *Sociological Methodology*, 18, 69–104. ArticleType: primary_article / Full publication date: 1988 / Copyright © 1988 American Sociological Association.
- Grimm, V. & Railsback, S. F. (2005). *Individual-Based Modeling and Ecology*. Princeton series in theoretical and computational biology. Princeton: Princeton University Press.
- Hacker, A. (1970). *The End of the American Era*. New York: Atheneum, 1st ed. edition.

- Hadfield, J. D. (2010). Mcmc methods for multi-response generalized linear mixed models: The MCMCglmm R package. *Journal of Statistical Software*, 33(2), 1–22.
- Hammersley, J. M. & Morton, K. W. (1954). Poor man’s Monte Carlo. *Journal of the Royal Statistical Society. Series B*, 16(1), 23–38.
- Hanushek, E. A. & Jackson, J. E. (1977). *Statistical Methods for Social Scientists*. New York: Academic Press.
- Harwell, M. R. (1992). Summarizing Monte Carlo results in methodological research. *Journal of Educational Statistics*, 17(4), 297–313.
- Hastings, W. K. (1970). Monte Carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1), 97–109.
- Hegselmann, R. (1996). Understanding social dynamics: The cellular automata approach. In *Social Science Microsimulation [Dagstuhl Seminar, May, 1995]* (pp. 282–306).: Springer-Verlag.
- Hirji, K. F., Mehta, C. R., & Patel, N. R. (1987). Computing distributions for exact logistic regression. *Journal of the American Statistical Association*, 82(400), 1110–1117.
- Hitchcock, D. B. (2003). A history of the Metropolis-Hastings algorithm. *The American Statistician*, 57(4), 254–257.
- Hoffmann, R. (2000). Twenty years on: The evolution of cooperation revisited. *Journal of Artificial Societies and Social Simulation*, 3.
- Holland, J. H. (1996). *Hidden order: how adaptation builds complexity*. Basic Books.
- Hope, A. C. A. (1968). A simplified Monte Carlo significance test procedure. *Journal of the Royal Statistical Society. Series B*, 30(3), 582–598.
- Huang, C. J. & Bolch, B. W. (1974). On the testing of regression disturbances for normality. *Journal of the American Statistical Association*, 69(346), 330–335.

- Huberman, B. A. & Glance, N. S. (1993). Evolutionary games and computer simulations. *Proceedings of the National Academy of Sciences*, 90(16), 7716–7718.
- Huckfeldt, R. R., Johnson, P. E., & Sprague, J. D. (2004). *Political Disagreement: The Survival of Diverse Opinions Within Communication Networks*. Cambridge, UK: Cambridge University Press.
- Hull, T. E. & Dobell, A. R. (1962). Random number generators. *SIAM Review*, 4(3), 230–254.
- Imai, K. & van Dyk, D. A. (2005a). A Bayesian analysis of the multinomial probit model using marginal data augmentation. *Journal of Econometrics*, 124(2), 311–334.
- Imai, K. & van Dyk, D. A. (2005b). MNP: R package for fitting the multinomial probit models. *Journal of Statistical Software*, 14(3), 1–32.
- Jackman, S. (2000a). Estimation and inference are missing data problems: Unifying social science statistics via Bayesian simulation. *Political Analysis*, 8(4), 307–332.
- Jackman, S. (2000b). Estimation and inference via Bayesian simulation: An introduction to Markov Chain Monte Carlo. *American Journal of Political Science*, 44(2), 375–404.
- Jackman, S. (2009). *Bayesian Analysis for the Social Sciences*. Chichester, UK: John Wiley & Sons, Ltd.
- Jackman, S. (2010). *pscl: Classes and Methods for R Developed in the Political Science Computational Laboratory, Stanford University*. Department of Political Science, Stanford University. R package version 1.03.5.
- Jacquier, E., Johannes, M., & Polson, N. (2007). MCMC maximum likelihood for latent state models. *Journal of Econometrics*, 137(2), 615–640.
- Jank, W. & Booth, J. (2003). Efficiency of Monte Carlo EM and simulated maximum likelihood in Two-Stage hierarchical models. *Journal of Computational and Graphical Statistics*, 12(1), 214–229.

- Jensen, P. H. J. (1998). *Self-Organized Criticality: Emergent Complex Behavior in Physical and Biological Systems*. Cambridge University Press.
- Jessop, W. N. (1956). Monte Carlo methods and industrial problems. *Journal of the Royal Statistical Society. Series C*, 5(3), 158–165.
- Jockel, K. (1986). Finite sample properties and asymptotic efficiency of Monte Carlo tests. *The Annals of Statistics*, 14(1), 336–347.
- Johnson, N. (2009). *Simply Complexity: A Clear Guide to Complexity Theory*. Oneworld Publications, reprint edition.
- Johnson, P. E. (1999). Simulation modeling in political science. *American Behavioral Scientist*, 42(10), 1509–1530.
- Johnson, P. E. (2002). Agent-Based modeling: What I learned from the artificial stock market. *Social Science Computer Review*, 20(2), 174–186.
- Jorgensen, W. L. (2000). Perspective on 'Equation of state calculations by fast computing machines'. *Theoretical Chemistry Accounts: Theory, Computation, and Modeling (Theoretica Chimica Acta)*, 103(3), 225–227.
- Kahn, H. & Marshall, A. W. (1953). Methods of reducing sample size in Monte Carlo computations. *Journal of the Operations Research Society of America*, 1(5), 263–278.
- Kauffman, S. A. (1993). *The origins of order: self organization and selection in evolution*. Oxford University Press US.
- Kauffman, S. A. (1995). *At home in the universe: the search for laws of self-organization and complexity*. Oxford University Press US.
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671–680.
- Klein, L. R. (1960). Single equation vs. equation system methods of estimation in econometrics. *Econometrica*, 28(4), 866–871.

- Knuth, D. E. (1968). *The Art of Computer Programming*. Addison-Wesley series in computer science and information processing. Reading, Mass: Addison-Wesley Pub. Co.
- Kollman, K., Miller, J. H., & Page, S. E. (1992). Adaptive parties in spatial elections. *The American Political Science Review*, 86(4), 929–937.
- Kollman, K., Miller, J. H., & Page, S. E. (1998). Political parties and electoral landscapes. *British Journal of Political Science*, 28(01), 139–158.
- Kowalski, C. J. (1972). On the effects of Non-Normality on the distribution of the sample Product-Moment correlation coefficient. *Journal of the Royal Statistical Society. Series C*, 21(1), 1–12.
- Kyzas, P. A., Denaxa-Kyza, D., & Ioannidis, J. P. (2007). Almost all articles on cancer prognostic markers report statistically significant results. *European Journal of Cancer*, 43(17), 2559–2579.
- Langton, C. G. (1984). Self-reproduction in cellular automata. *Physica D: Nonlinear Phenomena*, 10(1-2), 135–144.
- Langton, C. G. (1990). Computation at the edge of chaos. *Physica D*, 42, 134–144.
- Langton, C. G. (1995). *Artificial Life: an Overview*. Complex adaptive systems. Cambridge, Mass: MIT Press.
- Latane, B. (1996). Dynamic social impact: The creation of culture by communication. *Journal of Communication*, 46(4), 13–23.
- Laver, M. (2005). Policy and the dynamics of political competition. *American Political Science Review*, 99(02), 263–281.
- Laver, M. & Schilperoord, M. (2007). Spatial models of political competition with endogenous political parties. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1485), 1711–1721.

- L'Ecuyer, P. (1999). Good parameters and implementations for combined multiple recursive random number generators. *Operations Research*, 47(1), 159–164.
- L'Ecuyer, P. (2001). Software for uniform random number generation: distinguishing the good and the bad. In *Proceedings of the 33rd conference on Winter simulation* (pp. 95–105). Arlington, Virginia: IEEE Computer Society.
- L'Ecuyer, P., Simard, R., Chen, E. J., & Kelton, W. D. (2002). An Object-Oriented Random-Number package with many long streams and substreams. *Operations Research*, 50(6), 1073–1075.
- Lehsten, V. & Harmand, P. (2006). Null models for species co-occurrence patterns: assessing bias and minimum iteration number for the sequential swap. *Ecography*, 29(5), 786–792.
- Lele, S. R., Dennis, B., & Lutscher, F. (2007). Data cloning: easy maximum likelihood estimation for complex ecological models using Bayesian markov chain Monte Carlo methods. *Ecology Letters*, 10(7), 551–563.
- Lemieux, C. (2009). *Monte Carlo and Quasi-Monte Carlo Sampling*. Springer Series in Statistics. New York, NY: Springer-Verlag New York, 1 edition.
- Levy, M., Levy, H., & Solomon, S. (2000). *The microscopic simulation of financial markets: from investor behavior to market phenomena*. Academic Press.
- Levy, R. (2009). The rise of Markov chain Monte Carlo estimation for psychometric modeling. *Journal of Probability and Statistics*, 2009, 1–18.
- Lindsey, G. R. (1961). The progress of the score during a baseball game. *Journal of the American Statistical Association*, 56(295), 703–728.
- Linn, S. C. & Tay, N. S. P. (2007). Complexity and the character of stock returns: Empirical evidence and a model of asset prices based on complex investor learning. *Management Science*, 53(7), 1165–1180. ArticleType: primary_article / Issue Title: Complex Systems / Full publication date: Jul., 2007 / Copyright © 2007 INFORMS.

- Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing*. Springer series in statistics. New York: Springer.
- Lorenz, E. N. (1995). *The essence of chaos*. University of Washington Press.
- Luke, S., Cioffi-Revilla, C., Panait, L., & Sullivan, K. (2004). MASON: a new Multi-Agent simulation toolkit. In *Proceedings of the 2004 Swarmfest Workshop*.
- Luna, F. & Stefansson, B. (2000). *Economic simulations in Swarm: agent-based modelling and object oriented programming*. Springer.
- Lunn, D., Spiegelhalter, D., Thomas, A., & Best, N. (2009). The bugs project: Evolution, critique, and future directions. *Statistics in Medicine*, 28, 3049–3067.
- Mandelbrot, B. B. (1983). *The fractal geometry of nature*. Macmillan.
- Manly, B. & Sanderson, J. G. (2002). A note on null models: justifying the methodology. *Ecology*, 83(2), 580–582.
- Manly, B. F. (1997). *Randomization, Bootstrap and Monte Carlo Methods in Biology, Second Edition*. Chapman & Hall, 2 edition.
- Manly, B. F. J. (1995). A note on the analysis of species Co-Occurrences. *Ecology*, 76(4), 1109.
- Marriott, F. H. C. (1979). Barnard’s Monte Carlo tests: How many simulations? *Journal of the Royal Statistical Society. Series C*, 28(1), 75–77.
- Marsaglia, G. (1961). Expressing a random variable in terms of uniform random variables. *The Annals of Mathematical Statistics*, 32(3), 894–898.
- Marsaglia, G. & Tsang, W. W. (1998). The monty python method for generating random variables. *ACM Trans. Math. Softw.*, 24(3), 341–350.
- Marsaglia, G. & Tsang, W. W. (2000). A simple method for generating gamma variables. *ACM Trans. Math. Softw.*, 26(3), 363–372.

- Marschner, I. C. (2001). On stochastic versions of the EM algorithm. *Biometrika*, 88(1), 281–286.
- Martin, A. D. & Quinn, K. M. (2002). Dynamic ideal point estimation via Markov Chain Monte Carlo for the U.S. supreme court, 1953-1999. *Political Analysis*, 10(2), 134–153.
- Martin, A. D. & Quinn, K. M. (2007). Assessing preference change on the US supreme court. *Journal of Law, Economics, & Organization*, 23(2), 365–385.
- Martin, A. D., Quinn, K. M., & Park, J. H. (2010). *MCMCpack: Markov chain Monte Carlo (MCMC) Package*. R package version 1.0-7.
- Mascagni, M., Ceperley, D., & Srinivasan, A. (2000). SPRNG: a scalable library for pseudo-random number generation. *ACM Transactions on Mathematical Software*, 26, 436–461.
- Matsumoto, M. & Nishimura, T. (1998). Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, 8(1), 3–30.
- Matsumoto, M. & Nishimura, T. (2000). Dynamic creation of pseudorandom number generators. In H. Niederreiter & J. Spanier (Eds.), *Monte Carlo and Quasi-Monte Carlo methods* (pp. 56–69). Springer.
- May, R. M. (1976). Simple mathematical models with very complicated dynamics. *Nature*, 261(5560), 459–467.
- McCullagh, P. & Nelder, J. A. (1983). *Generalized Linear Models*. Number 37 in Monographs on statistics and applied probability. London: Chapman and Hall.
- McCulloch, R. & Rossi, P. E. (1994). An exact likelihood analysis of the multinomial probit model. *Journal of Econometrics*, 64(1-2), 207–240.
- McCullough, B. & Heiser, D. A. (2008). On the accuracy of statistical procedures in microsoft excel 2007. *Computational Statistics & Data Analysis*, 52(10), 4570–4578.

- McGee, V. E. & Carleton, W. T. (1970). Piecewise regression. *Journal of the American Statistical Association*, 65(331), 1109–1124.
- McPhee, W. & Smith, R. (1962). A model for analyzing voting systems. In W. McPhee & W. Glaser (Eds.), *Public Opinion and Congressional Elections* (pp. 123–179). New York: Free Press.
- Mehta, C. R. & Patel, N. R. (1995). Exact logistic regression: Theory and examples. *Statistics in Medicine*, 14(19), 2143–2160.
- Metropolis, N. (1987). The beginning of the Monte Carlo method. *Los Alamos Science*, 15, 125–130.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6), 1087.
- Metropolis, N. & Ulam, S. (1949). The Monte Carlo method. *Journal of the American Statistical Association*, 44(247), 335–341.
- Miller, A. J. (1961). A queueing model for road traffic flow. *Journal of the Royal Statistical Society. Series B*, 23(1), 64–90.
- Miller, J. H. & Page, S. E. (2007). *Complex Adaptive Systems: An Introduction to Computational Models of Social Life*. Princeton University Press, illustrated edition edition.
- Minar, N., Burkhart, R., Langton, C., & Askenazi, M. (1996). The swarm simulation system: A toolkit for building Multi-Agent simulations.
- Mitchell, M. (2009). *Complexity: A Guided Tour*. Oxford University Press, USA.
- Neal, R. (1994). Sampling from multimodal distributions using tempered transitions. *Statistics and Computing*, 6, 353–366.
- Neave, H. R. (1972). Observations on "Spectral analysis of short Series–A simulation study" by granger and hughes. *Journal of the Royal Statistical Society. Series A*, 135(3), 393–405.

- Nelson, C. R. & Schwert, G. W. (1982). Tests for predictive relationships between time series variables: A Monte Carlo investigation. *Journal of the American Statistical Association*, 77(377), 11–18.
- Neumann, J. V. & Burks, A. W. (1966). *Theory of Self-Reproducing Automata*. Urbana: University of Illinois Press.
- Newth, D. & Cornforth, D. (2009). Asynchronous spatial evolutionary games. *Biosystems*, 95(2), 120–129.
- Nielsen, S. F. (2000). The stochastic EM algorithm: estimation and asymptotic results. *Bernoulli*, 6(3), 457–489.
- Nowak, A. & Lewenstein, M. (1996). Modeling social change with cellular automata. In R. Hegselmann, U. Mueller, & K. G. Troitzsch (Eds.), *Modeling and Simulation in the Social Sciences from a Philosophical Point of View* (pp. 2249–285). Boston, Mass.: Kluwer.
- Nowak, A., Szamrej, J., & Latan, B. (1990). From private attitude to public opinion: A dynamic theory of social impact. *Psychological Review*, 97, 376, 362.
- Nowak, M., Bonhoeffer, S., & May, R. M. (1994a). More spatial games. *International Journal of Bifurcation and Chaos*, 4(1), 33–56.
- Nowak, M. A., Bonhoeffer, S., & May, R. M. (1994b). Spatial games and the maintenance of cooperation. *Proceedings of the National Academy of Sciences*, 91(11), 4877–4881.
- Nowak, M. A., Bonhoeffer, S., & May, R. M. (1996). Robustness of cooperation. *Nature*, 379(6561), 126.
- Nowak, M. A. & May, R. M. (1992). Evolutionary games and spatial chaos. *Nature*, 359(6398), 826–829.
- Nowak Martin, A. & May Robert, M. (1993). The Spatial Dilemmas of Evolution. *International Journal of Bifurcation and Chaos*, 3, 35–78.

- Palmer, R., Brianarthur, W., Holland, J., Lebaron, B., & Tayler, P. (1994). Artificial economic life: a simple model of a stockmarket. *Physica D*, 75(1-3), 264–274.
- Panneton, F., L'Ecuyer, P., & Matsumoto, M. (2006). Improved long-period generators based on linear recurrences modulo 2. *ACM Transactions on Mathematical Software*, 32(1), 1–16.
- Parker, D. C., Manson, S. M., Janssen, M. A., Hoffmann, M. J., & Deadman, P. (2003). Multi-Agent systems for the simulation of Land-Use and Land-Cover change: A review. *Annals of the Association of American Geographers*, 93(2), 314.
- Paxton, P., Curran, P. J., Bollen, K. A., Kirby, J., & Chen, F. (2001). Monte Carlo experiments: Design and implementation. *Structural Equation Modeling: A Multidisciplinary Journal*, 8(2), 287.
- Plummer, M. (2010a). JAGS - just another gibbs sampler. <http://www-fis.iarc.fr/~martyn/software/jags/>.
- Plummer, M. (2010b). *rjags: Bayesian graphical models using MCMC*. R package version 2.1.0-5.
- Ponciano, J. M., Taper, M. L., Dennis, B., & Lele, S. R. (2009). Hierarchical models in ecology: confidence intervals, hypothesis testing, and model selection using data cloning. *Ecology*, 90(2), 356–362.
- R Development Core Team (2010). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.
- Raes, N. & ter Steege, H. (2007). A null-model for significance testing of presence-only species distribution models. *Ecography*, 30(5), 727–736.
- Raj, B. (1980). A Monte Carlo study of Small-Sample properties of simultaneous equation estimators with normal and nonnormal disturbances. *Journal of the American Statistical Association*, 75(369), 221–229.

- Ramsey, P. H. (1978). Power differences between pairwise multiple comparisons. *Journal of the American Statistical Association*, 73(363), 479–485.
- Raudenbush, S. W. & Bryk, A. S. (2002). *Hierarchical linear models: applications and data analysis methods*. SAGE.
- Resnick, M. (1994). *Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds*. Complex adaptive systems. Cambridge, Mass: MIT Press.
- Rich, R. P. (1955). Simulation as an aid in model building. *Journal of the Operations Research Society of America*, 3(1), 15–19.
- Ripley, B. D. (1977). Modelling spatial patterns. *Journal of the Royal Statistical Society. Series B*, 39(2), 172–212.
- Robert, C. P. (2010). *Monte Carlo Statistical Methods*. Springer New York.
- Robert, C. P. & Casella, G. (2009). *Introducing Monte Carlo Methods with R*. Springer Verlag, 1 edition.
- Rossi, P. & McCulloch, R. (2008). *bayesm: Bayesian Inference for Marketing/Micro-econometrics*. R package version 2.2-2.
- Royston, P. & Thompson, S. G. (1995). Comparing Non-Nested regression models. *Biometrics*, 51(1), 114–127.
- Schelling, T. C. (1971). Dynamic models of segregation. *Journal of Mathematical Sociology*, 1, 143–186.
- Schelling, T. C. (1978). *Micromotives and Macrobehavior*. Fels lectures on public policy analysis. New York: Norton, 1st ed edition.
- Scott, D. W. & Factor, L. E. (1981). Monte Carlo study of three Data-Based nonparametric probability density estimators. *Journal of the American Statistical Association*, 76(373), 9–15.

- Sevcikova, H. & Rossini, T. (2009). *rlecuyer: R interface to RNG with multiple streams*. R package version 0.3-1.
- Shubik, M. (1960). Bibliography on simulation, gaming, artificial intelligence and allied topics. *Journal of the American Statistical Association*, 55(292), 736–751.
- Singh, A., Vainchtein, D., & Weiss, H. (2009). Schelling’s segregation model: Parameters, scaling, and aggregation. *Demographic Research*, 21, 341–366.
- Skrondal, A. (2000). Design and analysis of Monte Carlo experiments: Attacking the conventional wisdom. *Multivariate Behavioral Research*, 35(2), 137.
- Srinivasan, A., Mascagni, M., & Ceperley, D. (2003). Testing parallel random number generators. *Parallel Comput.*, 29(1), 69–94.
- Srivastava, M. S. & Keen, K. J. (1988). Estimation of the interclass correlation coefficient. *Biometrika*, 75(4), 731–739.
- Stefanski, L. A. & Buzas, J. S. (1995). Instrumental variable estimation in binary regression measurement error models. *Journal of the American Statistical Association*, 90(430), 541–550.
- Stigler, S. M. (1983). Who discovered bayes’s theorem? *The American Statistician*, 37(4), 290–296.
- Strogatz, S. H. (2003). *SYNC: The Emerging Science of Spontaneous Order*. Hyperion, 1 edition.
- Stroustrup, B. (1986). *The C++ Programming Language*. Addison-Wesley series in computer science. Reading, Mass: Addison-Wesley.
- Suman, B. & Kumar, P. (2006). A survey of simulated annealing as a tool for single and multiobjective optimization. *The Journal of the Operational Research Society*, 57(10), 1143–1160.

- Tanner, M. A. & Wong, W. H. (1987). The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82(398), 528–540.
- Tesfatsion, L. (2010). Agent-Based computational economics (Tesfatsion). <http://econ2.econ.iastate.edu/tesfatsi/ace.htm>.
- Thomas, A. (1994). BUGS: a statistical modelling package. *RTA/BCS Modular Languages Newsletter*, (pp. 36–38).
- Thompson, R., Govindarajulu, Z., & Doksum, K. A. (1967). Distribution and power of the absolute normal scores test. *Journal of the American Statistical Association*, 62(319), 966–975.
- Tierney, L. (1994). Markov chains for exploring posterior distributions. *The Annals of Statistics*, 22(4), 1701–1762.
- Valpine, P. D. (2003). Better inferences from Population-Dynamics experiments using Monte Carlo State-Space likelihood methods. *Ecology*, 84(11), 3064–3077.
- Vanderbilt, D. & Louie, S. G. (1984). A monte carlo simulated annealing approach to optimization over continuous variables. *Journal of Computational Physics*, 56(2), 259–271.
- Venables, W. N. & Ripley, B. D. (2002). *Modern Applied Statistics with S*. Statistics and computing. New York: Springer, 4th ed edition.
- von Neumann, J. (1951). Various techniques used in connection with random digits. *National Bureau Standards, Applied Mathematics Series*, 12, 36–38.
- Wagner, H. M. (1958). A Monte Carlo study of estimates of simultaneous linear structural equations. *Econometrica*, 26(1), 117–133.
- Waldrop, M. M. (1994). *Complexity: the emerging science at the edge of order and chaos*. Penguin.
- Wei, G. C. G. & Tanner, M. A. (1990). A Monte Carlo implementation of the EM algorithm

- and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(411), 699–704.
- Western, B. & Jackman, S. (1994). Bayesian inference for comparative research. *The American Political Science Review*, 88(2), 412–423.
- Wilensky, U. (1999). *NetLogo*. <http://ccl.northwestern.edu/netlogo/>, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.
- Wolfram, S. (2002). *A New Kind of Science*. Wolfram Media, 1 edition.
- Yates, F. (1966). Computers, the second revolution in statistics. *Biometrics*, 22(2), 233–251.
- Youle, P. V., Tocher, K. D., Jessop, W. N., & Musk, F. I. (1959). Simulation studies of industrial operations. *Journal of the Royal Statistical Society. Series A*, 122(4), 484–510.
- Zamar, D., McNeney, B., & Graham, J. (2007). elrm: Software implementing exact-like inference for logistic regression models. *Journal of Statistical Software*, 21(3).
- Zhang, J. (2004). A dynamic model of residential segregation. *The Journal of Mathematical Sociology*, 28(3), 147.