CHAPTER 1

Introduction: Consequences of Numerical Inaccuracy

1.1 IMPORTANCE OF UNDERSTANDING COMPUTATIONAL STATISTICS

How much pollution is bad for you? Well-known research conducted from 1987 to 1994 linked small-particle air pollution to health problems in 90 U.S. cities. These findings were considered reliable and were influential in shaping public policy. Recently, when the same scientists attempted to replicate their own findings, they produced different results with the same data—results that showed a much weaker link between air pollution and health problems. "[The researchers] re-examined the original figures and found that the problem lay with how they used off-the-shelf statistical software to identify telltale patterns that are somewhat akin to ripples from a particular rock tossed into a wavy sea. Instead of adjusting the program to the circumstances that they were studying, they used standard default settings for some calculations. That move apparently introduced a bias in the results, the team says in the papers on the Web" (Revkin, 2002).

Problems with numerical applications are practically as old as computers: In 1962, the *Mariner I* spacecraft, intended as the first probe to visit another planet, was destroyed as a result of the incorrect coding of a mathematical formula (Neumann 1995), and five years later, Longley (1967) reported on pervasive errors in the accuracy of statistical programs' implementation of linear regression. Unreliable software is sometimes even expected and tolerated by experienced researchers. Consider this report on the investigation of a high-profile incident of academic fraud, involving the falsification of data purporting to support the discovery of the world's heaviest element at Lawrence Berkeley lab: "The initial suspect was the analysis software, nicknamed Goosy, a somewhat temperamental computer program known on occasion to randomly corrupt data. Over the years, users had developed tricks for dealing with Goosy's irregularities, as one might correct a wobbling image on a TV set by slapping the side of the cabinet" (Johnson 2002).

In recent years, many of the most widely publicized examples of scientific application failures related to software have been in the fields of space exploration

Numerical Issues in Statistical Computing for the Social Scientist, by Micah Altman, Jeff Gill, and Michael P. McDonald
ISBN 0-471-23633-0

and rocket technology. Rounding errors in numerical calculations were blamed for the failure of the Patriot missile defense to protect an army barracks in Dhahran from a Scud missile attack in 1991 during Operation Desert Storm (Higham 2002). The next year, the space shuttle had difficulties in an attempted rendezvous with *Intelsat 6* because of a round-off error in the routines that the shuttle computers used to compute distance (Neumann 1995). In 1999, two Mars-bound spacecraft were lost, due (at least in part) to software errors—one involving failure to check the units as navigational inputs (Carreau 2000). Numerical software bugs have even affected our understanding of the basic structure of the universe: highly publicized findings suggesting the existence of unknown forms of matter in the universe, in violation of the "standard model," were later traced to numerical errors, such as failure to treat properly the sign of certain calculations (Glanz 2002; Hayakawa and Kinoshita 2001).

The other sciences, and the social sciences in particular, have had their share of less publicized numerical problems: Krug et al. (1988) retracted a study analyzing suicide rates following natural disasters that was originally published in the *Journal of the American Medical Association*, one of the world's most prestigious medical journals, because their software erroneously counted some deaths twice, undermining their conclusions (see Powell et al. 1999). Leimer and Lesnoy (1982) trace Feldstein's (1974) erroneous conclusion that the introduction of Social Security reduced personal savings by 50% to the existence of a simple software bug. Dewald et al. (1986), in replicating noted empirical results appearing in the *Journal of Money, Credit and Banking*, discovered a number of serious bugs in the original authors' analyses programs. Our research and that of others has exposed errors in articles recently published in political and social science journals that can be traced to numerical inaccuracies in statistical software (Altman and McDonald 2003; McCullough and Vinod 2003; Stokes 2003).

Unfortunately, numerical errors in published social science analyses can be revealed only through replication of the research. Given the difficulty and rarity of replication in the social sciences (Dewald et al. 1986; Feigenbaum and Levy 1993), the numerical problems reported earlier are probably the tip of the iceberg. One is forced to wonder how much of the critical and foundational findings in a number of fields are actually based on suspect statistical computing.

There are two primary sources of potential error in numerical algorithms programmed on computers: that numbers cannot be perfectly represented within the limited binary world of computers, and that some algorithms are not guaranteed to produce the desired solution.

First, small computational inaccuracies occur at the precision level of all statistical software when digits beyond the storage capacity of the computer must be rounded or truncated. Researchers may be tempted to dismiss this threat to validity because measurement error (miscoding of data, survey sampling error, etc.) is almost certainly an order of magnitude greater for most social science applications. But these small errors may propagate and magnify in unexpected ways in the many calculations underpinning statistical algorithms, producing wildly erroneous results on their own, or exacerbating the effects of measurement error.

Second, computational procedures may be subtly biased in ways that are hard to detect and are sometimes not guaranteed to produce a correct solution. Random number generators may be subtly biased: random numbers are generated by computers through non-random, deterministic processes that mimic a sequence of random numbers but are not genuinely random. Optimization algorithms, such as maximum likelihood estimation, are not guaranteed to find the solution in the presence of multiple local optima: Optimization algorithms are notably susceptible to numeric inaccuracies, and resulting coefficients may be far from their true values, posing a serious threat to the internal validity of hypothesized relationships linking concepts in the theoretical model.

An understanding of the limits of statistical software can help researchers avoid estimation errors. For typical estimation, such as ordinary least squares regression, well-designed off-the-shelf statistical software will generally produce reliable estimates. For complex algorithms, our knowledge of model building has outpaced our knowledge of computational statistics. We hope that researchers contemplating complex models will find this book a valuable tool to aid in making robust inference within the limits of computational statistics.

Awareness of the limits of computational statistics may further aid in model testing. Social scientists are sometimes faced with iterative models that fail to converge, software that produces nonsensical results, Hessians that cannot be inverted, and other problems associated with estimation. Normally, this would cause researchers to abandon the model or embark on the often difficult and expensive process of gathering more data. An understanding of computational issues can offer a more immediately available solution—such as use of more accurate computations, changing algorithmic parameters of the software, or appropriate rescaling of the data.

1.2 BRIEF HISTORY: DUHEM TO THE TWENTY-FIRST CENTURY

The reliability of scientific inference depends on one's tools. As early as 1906, French physicist and philosopher of science Pierre Duhem noted that every scientific inference is conditioned implicitly on a constellation of background hypotheses, including that the instruments are functioning correctly (Duhem 1991, Sec. IV.2). The foremost of the instruments used by modern applied statisticians is the computer.

In the early part of the twentieth century the definition of a *computer* to statisticians was quite different from what it is today. In antiquated statistics journals one can read where authors surprisingly mention "handing the problem over to my computer." Given the current vernacular, it is easy to miss what is going on here. Statisticians at the time employed as "computers" *people* who specialized in performing repetitive arithmetic. Many articles published in leading statistics journals of the time addressed methods by which these calculations could be made less drudgingly repetitious because it was noticed that as tedium increases linearly, careless mistakes increase exponentially (or thereabouts). Another rather prescient development of the time given our purpose here was the attention paid to creating self-checking procedures where "the computer" would at regular intervals have a clever means to check calculations against some summary value as a way of detecting errors (cf. Kelley and McNemar 1929). One of the reasons that Fisher's normal tables (and therefore the artificial 0.01 and 0.05 significance thresholds) were used so widely was that the task of manually calculating normal integrals was time consuming and tedious. Computation, it turns out, played an important role in scholarship even before the task was handed over to machines.

In 1943, Hotelling and others called attention to the accumulation of errors in the solutions for inverting matrices in the method of least squares (Hotelling 1943) and other matrix manipulation (Turing 1948). Soon after development of the mainframe computer, programmed regression algorithms were criticized for dramatic inaccuracies (Longley 1967). Inevitably, we improve our software, and just as inevitably we make our statistical methods more ambitious. Approximately every 10 years thereafter, each new generation of statistical software has been similarly faulted (e.g., Wampler 1980; Simon and LeSage 1988).

One of the most important statistical developments of the twentieth century was the advent of *simulation* on computers. While the first simulations were done *manually* by Buffon, Gosset, and others, it was not until the development of machine-repeated calculations and electronic storage that simulation became prevalent. In their pioneering postwar work, von Neumann and Ulam termed this sort of work *Monte Carlo simulation*, presumably because it reminded them of long-run observed odds that determine casino income (Metropolis and Ulam 1949; Von Neumann 1951). The work was conducted with some urgency in the 1950s because of the military advantage of simulating nuclear weapon designs. One of the primary calculations performed by von Neumann and his colleagues was a complex set of equations related to the speed of radiation diffusion of fissile materials. This was a perfect application of the Monte Carlo method because it avoided both daunting analytical work and dangerous empirical work. During this same era, Metropolis et al. (1953) showed that a new version of Monte Carlo simulation based on Markov chains could model the movement of atomic particles in a box when analytical calculations are impossible.

Most statistical computing tasks today are sufficiently routinized that many scholars pay little attention to implementation details such as default settings, methods of randomness, and alternative estimation techniques. The vast majority of statistical software users blissfully point-and-click their way through machine implementations of noncomplex procedures such as least squares regression, cross-tabulation, and distributional summaries. However, an increasing number of social scientists regularly use more complex and more demanding computing methods, such as Monte Carlo simulation, nonlinear estimation procedures, queueing models, Bayesian stochastic simulation, and nonparametric estimation. Accompanying these tools is a general concern about the possibility of knowingly or unknowingly producing invalid results.

In a startling article, McCullough and Vinod (1999) find that econometric software packages can still produce "horrendously inaccurate" results (p. 635) and that inaccuracies in many of these packages have gone largely unnoticed (pp. 635–37). Moreover, they argue that given these inaccuracies, past inferences are in question and future work must document and archive statistical software alongside statistical models to enable replication (pp. 660–62).

In contrast, when most social scientists write about quantitative analysis, they tend not to discuss issues of accuracy in the implementation of statistical models and algorithms. Few of our textbooks, even those geared toward the most sophisticated and computationally intensive techniques, mention issues of implementation accuracy and numerical stability. Acton (1996), on the other hand, gives a frightening list of potential problems: "loss of significant digits, iterative instabilities, degenerative inefficiencies in algorithms, and convergence to extraneous roots of previously docile equations."

When social science methodology textbooks and review articles in social science do discuss accuracy in computer-intensive quantitative analysis, they are relatively sanguine about the issues of accurate implementation:

- On finding maximum likelihood: "Good algorithms find the correct solution regardless of starting values. . . . The computer programs for most standard ML estimators automatically compute good starting values." And on accuracy: "Since neither accuracy nor precision is sacrificed with numerical methods they are sometimes used even when analytical (or partially analytical) solutions are possible" (King 1989, pp. 72–73).
- On the error of approximation in Monte Carlo analysis: "First, one may simply run ever more trials, and approach the infinity limit ever more closely" (Mooney 1997, p. 100).
- In the most widely assigned econometric text, Greene (2003) provides an entire appendix on computer implementation issues but also understates in referring to numerical optimization procedures: "Ideally, the iterative procedure should terminate when the gradient is zero. In practice, this step will not be possible, primarily because of accumulated rounding error in the computation of the function and its derivatives" (p. 943).

However, statisticians have been sounding alarms over numerical computing issues for some time:

- Grillenzoni worries that when confronted with the task of calculating the gradient of a complex likelihood, software for solving nonlinear least squares and maximum likelihood estimation, can have "serious numerical problems; often they do not converge or yield inadmissible results" (Grillenzoni 1990, p. 504).
- Chambers notes that "even a reliable method may perform poorly if not careful checked for special cases, rounding error, etc. are not made" (Chambers 1973, p. 9).
- "[M]any numerical optimization routines find local optima and may not find global optima; optimization routines can, particularly for higher dimensions,

'get lost' in subspaces or in flat spots of the function being optimized" (Hodges 1987, p. 268).

- Beaton et al. examine the famous Longley data problem and determine: "[T]he computationally accurate solution to this regression problem—even when computed using 40 decimal digits of accuracy—may be a very poor estimate of regression coefficients in the following sense: small errors beyond the last decimal place in the data can result solutions more different than those computed by Longley with his less preferred programs" (Beaton et al. 1976, p. 158). Note that these concerns apply to a *linear model!*
- The `BUGS` and `WinBUGS` documentation puts this warning on page 1 of the documentation: "**Beware—Gibbs sampling can be dangerous!**"

A clear discrepancy exists between theoreticians and applied researchers: The extent to which one should worry about numerical issues in statistical computing is unclear and even debatable. This is the issue we address here, bridging the knowledge gap difference between empirically driven social scientists and more theoretically minded computer scientists and statisticians.

1.3 MOTIVATING EXAMPLE: RARE EVENTS COUNTS MODELS

It is well known that binary rare events data are difficult to model reliably because the results often greatly underestimate the probability of occurrence (King and Zeng 2001a). It is true also that rare events counts data are difficult to model because like binary response models and all other generalized linear models (GLMs), the statistical properties of the estimations are conditional on the mean of the outcome variable. Furthermore, the infrequently observed counts are often not temporally distributed uniformly throughout the sample space, thus produce clusters that need to be accounted for (Symons et al. 1983).

Considerable attention is being given to model specification for binary count data in the presence of overdispersion (variance exceeding the mean, thus violating the Poisson assumption) in political science (King 1989; Achen 1996; King and Signorino 1996; Amato 1996; Londregan 1996), economics (Hausman et al. 1984; Cameron and Trivedi 1986, 1990; Lee 1986, Gurmu 1991), and of course, statistics (McCullagh and Nelder 1989). However, little has been noted about the numerical computing and estimation problems that can occur with other rare events counts data.

Consider the following data from the 2000 U.S. census and North Carolina public records. Each case represents one of 100 North Carolina counties, and we use only the following subset of the variables.

- **Suicides by Children.** This is (obviously) a rare event on a countywide basis and refers almost strictly to teenage children in the United States.
- **Number of Residents in Poverty.** Poverty is associated directly with other social ills and can lower the quality of education, social interaction, and opportunity of children.

- **Number of Children Brought Before Juvenile Court.** This measures the number of first-time child offenders brought before a judge or magistrate in a juvenile court for each of these counties.

Obviously, this problem has much greater scope as both a sociological question and a public policy issue, but the point here is to demonstrate numerical computing problems with a simple but *real* data problem. For replication purposes these data are given in their entirety in Table 1.1.

For these we specified a simple Poisson generalized linear model with a log link function:

$$\underbrace{g^{-1}(\boldsymbol{\theta})}_{100\times 1} = g^{-1}(\mathbf{X}\boldsymbol{\beta}) = \exp[\mathbf{X}\boldsymbol{\beta}]$$
$$= \exp[1\beta_0 + \text{POV}\beta_1 + \text{JUV}\beta_2]$$
$$= E[\mathbf{Y}] = E[\text{SUI}]$$

in standard GLM notation (Gill 2000). This basic approach is run on five commonly used statistical packages and the results are summarized in Table 1.2. Although there is some general agreement among `R`, `S-Plus`, `Gauss`, and `Stata`, `SAS` (Solaris v8) produces estimates substantively different from the other four.[1] Although we may have some confidence that the results from the four programs in agreement are the "correct" results, we cannot know for sure, since we are, after all, estimating unknown quantities. We are left with the troubling situation that the results are dependent on the statistical program used to generate statistical estimates.

Even among the four programs in agreement, there are small discrepancies among their results that should give pause to researchers who interpret t-statistics strictly as providing a measure of "statistical significance." A difference in the way `Stata` handles data input explains some of the small discrepancy between `Stata`'s results and `R` and `S-Plus`. Unless specified, `Stata` reads in data as single precision, whereas the other programs read data as double precision. When we provide the proper commands to read in data into `Stata` as double precision, the estimates from the program lie between the estimates of `R` and `S-Plus`. This does not account for the difference in the estimates generated by `Gauss`, a program that reads in data as double precision, which are in line with `Stata`'s single-precision estimates.

This example highlights some of the important themes to come. Clearly, inconsistent results indicate that there are some sources of inaccuracy from these data. All numerical computations have limited accuracy, and it is possible for particular characteristics of the data at hand to exacerbate these effects; this is the focus of Chapter 2. The questions addressed there are: What are the sources of inaccuracy associated with specific algorithmic choices? How may even a small error propagate into a large error that changes substantive results?

[1]Note that `SAS` issued warning messages during the estimation, but the final results were not accompanied by any warning of failure.

Table 1.1 North Carolina 2000 Data by Counties

County	Suicide	Poverty	Juvenile/ Court	County	Suicide	Poverty	Juvenile/ Court
Alamance	0	14,519	47	Johnston	1	15,612	45
Alexander	0	2,856	70	Jones	0	1,754	81
Alleghany	0	1,836	26	Lee	0	6,299	87
Anson	0	4,499	49	Lenoir	0	9,900	17
Ashe	0	3,292	56	Lincoln	0	5,868	14
Avery	0	2,627	58	Macon	0	4,890	70
Beaufort	0	8,767	71	Madison	0	3,756	58
Bertie	0	4,644	26	Martin	0	3,024	74
Bladen	0	6,778	66	McDowell	1	5,170	86
Brunswick	1	9,216	19	Mecklenburg	0	63,982	1
Buncombe	0	23,522	52	Mitchell	1	2,165	50
Burke	0	9,539	33	Montgomery	0	4,131	69
Cabarrus	0	9,305	36	Moore	0	8,524	25
Caldwell	0	8,283	29	Nash	1	11,714	22
Camden	0	695	60	New Hanover	0	21,003	62
Carteret	0	6,354	13	Northampton	1	4,704	54
Caswell	0	3,384	67	Onslow	1	19,396	42
Catawba	0	12,893	51	Orange	0	16,670	6
Chatham	0	4,785	79	Pamlico	0	1,979	26
Cherokee	0	3,718	68	Pasquotank	2	6,421	74
Chowan	0	2,557	46	Pender	0	5,587	10
Clay	0	1,000	20	Perquimans	0	2,035	35
Cleveland	0	12,806	41	Person	0	4,275	82
Columbus	0	12,428	2	Pitt	0	27,161	27
Craven	1	11,978	12	Polk	0	1,851	20
Cumberland	2	38,779	73	Randolph	1	11,871	42
Currituck	0	1,946	61	Richmond	0	9,127	9
Dare	0	2,397	75	Robeson	1	28,121	64
Davidson	1	14,872	55	Rockingham	0	11,767	4
Davie	0	2,996	72	Rowan	0	13,816	44
Duplin	0	9,518	69	Rutherford	0	8,743	32
Durham	2	29,924	53	Sampson	1	10,588	71
Edgecombe	0	10,899	34	Scotland	0	7,416	18
Forsyth	1	33,667	57	Stanly	0	6,217	83
Franklin	1	5,955	84	Stokes	0	4,069	16
Gaston	0	20,750	59	Surry	0	8,831	24
Gates	0	1,788	15	Swain	1	2,373	56
Graham	0	1,559	37	Transylvania	0	2,787	78
Granville	0	5,674	85	Tyrrell	0	967	11
Greene	0	3,833	40	Union	1	10,018	38
Guilford	1	44,631	77	Vance	0	8,806	7
Halifax	1	13,711	8	Wake	5	48,972	80
Harnett	0	13,563	39	Warren	0	3,875	48
Haywood	1	6,214	21	Washington	0	2,992	43
Henderson	0	8,650	30	Watauga	0	7,642	63
Hertford	1	4,136	56	Wayne	0	15,639	42
Hoke	0	5,955	76	Wilkes	0	7,810	23
Hyde	0	897	81	Wilson	0	13,656	31
Iredell	1	10,058	28	Yadkin	2	3,635	3
Jackson	0	5,001	5	Yancey	0	2,808	65

Table 1.2 Rare Events Counts Models in Statistical Packages

		R	S-Plus	SAS	Gauss	Stata
Intercept	Coef.	−3.13628	−3.13678	0.20650	−3.13703	−3.13703
	Std. err.	0.75473	0.75844	0.49168	0.76368	0.76367
	t-stat.	−4.15550	−4.13585	0.41999	−4.10788	−4.10785
Poverty/1000	Coef.	0.05264	0.05263	−1.372e-04	0.05263	0.05269
	Std. err.	0.00978	0.00979	1.2833-04	0.00982	0.00982
	t-stat.	5.38241	5.37136	−1.06908	5.35881	5.36558
Juvenile	Coef.	0.36167	0.36180	−0.09387	0.36187	0.36187
	Std. err.	0.18056	0.18164	0.12841	0.18319	0.18319
	t-stat.	2.00301	1.99180	−0.73108	1.97541	1.97531

In this example we used different software environments, some of which required direct user specification of the likelihood function, the others merely necessitating menu direction. As seen, different packages sometimes yield different results. In this book we also demonstrate how different routines within the same package, different version numbers, or even different parameter settings can alter the quality and integrity of results. We do not wish to imply that researchers who do their own programming are doing better or worse work, but that the more responsibility one takes when model building, the more one must be aware of issues regarding the software being used and the general numerical problems that might occur. Accordingly, in Chapter 3 we demonstrate how proven benchmarks can be used to assess the accuracy of particular software solutions and discuss strategies for consumers of statistical software to help them identify and avoid numeric inaccuracies in their software.

Part of the problem with the example just given is attributable to these data. In Chapter 4 we investigate various data-originated problems and provide some solutions that would help with problems, as we have just seen. One method of evaluation that we discuss is to check results on multiple platforms, a practice that helped us identify a programming error in the `Gauss` code for our example in Table 1.2.

In Chapter 5 we discuss some numerical problems that result from implementing Markov chain Monte Carlo algorithms on digital computers. These concerns can be quite complicated, but the foundational issues are essentially like those shown here: numerical treatment within low-level algorithmic implementation. In Chapter 6 we look at the problem of a non-invertible Hessian matrix, a serious problem that can occur not just because of collinearity, but also because of problems in computation or data. We propose some solutions, including a new approach based on generalizing the inversion process followed by importance sampling simulation.

In Chapter 7 we investigate a complicated modeling scenario with important theoretical concerns: ecological inference, which is susceptible to numerical inaccuracies. In Chapter 8 Bruce McCullough gives guidelines for estimating general nonlinear models in economics. In Chapter 10 Paul Allison discusses numerical issues in logistical regression. Many related issues are exacerbated with spatial data, the topic of Chapter 9 by James LeSage. Finally, in Chapter 11 we provide a summary of recommendations and an extended discussion of methods for ensuring replicable research.

1.4 PREVIEW OF FINDINGS

In this book we introduce principles of numerical computation, outline the optimization process, and provide tools for assessing the sensitivity of subsequent results to problems that exist in these data or with the model. Throughout, there are real examples and replications of published social science research and innovations in numerical methods.

Although we intend readers to find this book useful as a reference work and software guide, we also present a number of new research findings. Our purpose is not just to present a collection of recommendations from different methodological literatures. Here we actively supplement useful and known strategies with unique findings.

Replication and verification is not a new idea (even in the social sciences), but this work provides the first replications of several well-known articles in political science that show where optimization and implementation problems affect published results. We hope that this will bolster the idea that political science and other social sciences should seek to recertify accepted results.

Two new methodological developments in the social sciences originate with software solutions to historically difficult problems. Markov chain Monte Carlo has revolutionized Bayesian estimation, and a new focus on sophisticated software solutions has similarly reinvigorated the study of ecological inference. In this volume we give the first look at numerical accuracy of MCMC algorithms from pseudo-random number generation and the first detailed evaluation of numerical periodicity and convergence.

Benchmarks are useful tools to assess the accuracy and reliability of computer software. We provide the first comprehensive packaged method for establishing standard benchmarks for social science data input/output accuracy. This is a neglected area, but it turns out that the transmission of data across applications can degrade the quality of these data, even in a way that affects estimation. We also introduce the first procedure for using *cyclical redundancy checks* to assess the success of data input rather than merely checking file transfer. We discuss a number of existing benchmarks to test numerical algorithms and to provide a new set of standard benchmark tests for distributional accuracy of statistical packages.

Although the negative of the Hessian (the matrix of second derivatives of the posterior with respect to the parameters) must be positive definite and hence invertible in order to compute the variance matrix, invertible Hessians do not exist for some combinations of datasets and models, causing statistical procedures to fail. When a Hessian is non-invertible purely because of an interaction between the model and the data (and not because of rounding and other numerical errors), this means that the desired variance matrix does not exist; the likelihood function may still contain considerable information about the questions of interest. As such, discarding data and analyses with this valuable information, even if the information cannot be summarized as usual, is an inefficient and potentially biased procedure. In Chapter 6 Gill and King provide a new method for applying generalized inverses to Hessian problems that can provide results even in circumstances where it is not usually possible to invert the Hessian and obtain coefficient standard errors.

Ecological inference, the problem of inferring individual behavior from aggregate data, was (and perhaps still is) arguably once the longest-standing unsolved problem in modern quantitative social science. When in 1997 King provided a new method that incorporated both the statistical information in Goodman's regression and the deterministic information in Duncan and Davis's bounds, he garnered tremendous acclaim as well as persistent criticism. In this book we report the first comparison of the numerical properties of competing approaches to the ecological inference problem. The results illuminate the trade-offs among correctness, complexity, and numerical sensitivity.

More important than this list of new ideas, which we hope the reader will explore, this is the first general theoretical book on statistical computing that is focused purely on the social sciences. As social scientists ourselves, we recognize that our data analysis and estimation processes can differ substantially from those described in a number of (even excellent) texts.

All too often new ideas in statistics are presented with examples from biology. There is nothing wrong with this, and clearly the points are made more clearly when the author actually cares about the data being used. However, we as social scientists often *do not* care about the model's implications for lizards, beetles, bats, coal miners, anchovy larvae, alligators, rats, salmon, seeds, bones, mice, kidneys, fruit flies, barley, pigs, fertilizers, carrots, and pine trees. These are actual examples taken from some of our favorite statistical texts. Not that there is anything wrong with studying lizards, beetles, bats, coal miners, anchovy larvae, alligators, rats, salmon, seeds, bones, mice, kidneys, fruit flies, barley, pigs, fertilizers, carrots, and pine trees, but we would rather study various aspects of human social behavior. This is a book for those who agree.