STATISTICAL COMPUTING

J.H. Maindonald
Centre for Bioinformation Science, Australian National University, Australia

Keywords: Computing, Statistical analysis, Biometry, Expert system, Computer language, Database, Statistical package, Document preparation, Markup system.

Contents

Summary

This chapter describes the large gains made in statistical computing in the past 50 years. These advances allow a huge range of statistical calculations that were formerly difficult or impossible to handle satisfactorily. They have affected both the style and the content of statistical computing. The effects are most immediately obvious in the work of statistical analysis professionals, but will in due course affect all who plan quantitative studies and undertake statistical analyses. Much of the advance that can be expected in the future will follow the same general pattern. Connectivity between statistical computing systems and other computing systems is a major focus for current research. To date, there has been very limited success in incorporating, within statistical analysis systems, the expertise that skilled professionals bring to their work. This is a major challenge for future research.

1. Introduction

1.1. Content and Style

Experimentation with the use of electronic digital computers for statistical analysis started in the 1950s. Since that time, advances in the methodology that is available to handle problems, and associated advances in computing technology, have greatly widened the range of problems that are amenable to statistical analysis. This chapter will discuss the technical capabilities of systems and environments for statistical computing, the human interface, and the professional skills that may be needed for effective statistical analysis. It will speculate on directions of future development.

1.2. Computing as a Stimulus to Innovation in Statistical Methodology

There are several ways in which advances in computing technology have stimulated innovation in statistical methodology. The encoding of algorithms in computer languages has encouraged and assisted ongoing review of earlier methods, often leading to huge improvements. Thus, regression diagnostics have become a standard and necessary adjunct to the use of regression methods. Computing advances have allowed the use of methods that, because of their heavy resource demands, could not previously have been contemplated. Such changes have then influenced both the analyses that are performed and the way that statistical professionals think about methodology.

Additionally, there have been large conceptual advances that have brought formerly disparate methods within the reach of a unified conceptual framework.

Four types of calculation will be noted that have become relatively commonplace. First, many of the newer methods require calculations that are inherently iterative – for example a least squares type of calculation may be iterated many times. Such problems arise in time series analysis, in comparing the time profiles of two or more groups of patients in a clinical trial, in accounting for spatial effects in the analysis of agricultural yield data, in modeling climate data, and so on. In all these problems there can be complex patterns of statistical variation, demanding models whose "error" structures are very different from the independently and identically distributed error structures of elementary statistical courses. All these applications typically require methods that take account of correlations that are strongest for points that are close together in space and/or time. Thus, in the modeling of climate data, regions that are close geographically may have relatively similar climate patterns while regions that are widely separated may have patterns of variation that are very nearly independent. To be effective, modeling must reflect this complex pattern of dependence.

Second, Bayesian approaches to thinking about statistical problems, and to the calculations that arise in this context, seem more appealing to statistical professionals than a decade ago. For some problems, a Bayesian approach offers the only available computationally tractable framework. The *Markov Chain Monte Carlo* (MCMC) and related methods that are now in wide use are highly computationally intensive. Until recently, most practical implementations of Bayesian methods have used specially written software. This may change, as the WinBUGS system gains wider acceptance and as routines that implement Bayesian approaches become available for major software systems such as R, S-PLUS, SAS and SPSS. Already, R has several packages that implement Bayesian approaches.

Third, there is widespread use of *resampling* methods. These repeat what is essentially the same calculation on many different sub-samples (e.g., ~1000) from the same data. As some methods require several different levels of sub-sampling, the calculations have the potential to be heavily computationally intensive. Such methods can be useful when standard theory fails or is in doubt, or when the distributional properties of a sample statistic are not known.

Finally, various forms of automatic data capture allow the creation of data sets that are of unprecedented size. Such data sets arise in climate modeling, in various other spatial modeling applications, in astronomy, in commerce, in industry, and in molecular biology. In work with large data sets, a first step is to process and perhaps summarize the data into a form where it is suitable for analysis. This may itself be a challenging task. The size of the data set that results, even if substantially reduced, may nevertheless offer a severe challenge to the analyst. In general, the challenge arises from the combination of modest size with an error structure that is far from independently and identically distributed and normal.

The above discussion has reflected on methodological advances, i.e., on advances in the approaches that are available to handle the analysis of data. There have been other

changes also, just as important in their own way. There have been large improvements to the computing environment in which analyses are performed, so that the computer handles tasks that would formerly have been done manually. Such tasks include: access to data bases that hold primary or ancillary data; the collection and maintenance of a list of references; routine checks on data and on analysis results; maintaining and accessing a record of the steps involved in an analysis; the integration of data summaries, graphs and analysis results into the text of a report or paper; mechanisms that ensure that any draft of a paper immediately reflects changes made to the code used for analysis or to the data; and the provision on a web site of information and data that are ancillary to the published paper.

1.3. Improvements to the Statistical Computing Environment

Automation of calculations, and the building of new abilities on top of earlier carefully tested abilities, makes it possible to do highly complex calculations reliably and in reasonable time. Routine checks, of a kind that would have been impossible when calculations were done by hand, have become an established part of good statistical practice. Whereas in the early 1970s most statistical systems either gave no graphs or gave poor quality graphs, the best modern systems include high quality graphs as an integral part of their analysis output. In the 1970s, high quality graphics systems were in general totally separate from statistical systems, and were designed to work with specific hardware devices. The best modern statistical systems make it easy to obtain high quality graphical representations of results from statistical analysis; data analysis and graphics are tightly linked.

It is now clear that an effective working environment for statistical analysis requires other tight linkages – with database systems, editors, web interfaces, software that assists in incorporating results from analyses into reports and published papers, and so on. Most statistical software systems now make it possible to work within an editor-like environment, so that it is straightforward to recall and execute previous commands in a modified form, and to edit output as its appears. A generic solution has advantages, where a general purpose editor is tailored for use with one or more statistical systems. The ESS ("Emacs speaks statistics") environment allows the use of the Emacs editor as an environment for a wide variety of statistical software systems.

1.4. Graphical User Interfaces

Graphical User Interfaces (GUIs) are one of several areas where different relatively advanced systems offer substantially different environments. This is in part a result of different histories, with some systems retaining vestiges from the punched card era. There are important differences that reflect a lack of consensus on the best approach. Graphical user interfaces (GUIs) obviously have a place; they work well for word processors, and for some statistical computing system tasks such as those connected with data input and output. Wide agreement on a style of GUI that is suitable for use in statistical analysis environments seems still some way off. There are large differences in the styles of GUI that are available in different statistical computing systems. If and when a consensus does emerge, professionals may still prefer, for most tasks, a command line style of communication.

It is often important to examine a record of the sequence of calculations that have been performed. While it is possible to store and replay the operations that were performed by the user, such a replay may, unless supplemented by verbal description, have many of the characteristics of a mime, requiring the user to fill in the words. The difficulty is increased where there are many false steps that have not contributed to the final analysis. A verbal record of the successive steps seems a necessary adjunct to any replay of the point and click steps. Such a verbal record becomes even more important when the steps that were performed must be justified and documented.

1.5. Efficiency Considerations

Code for an interpreted language, such as R, is inherently slower to execute than code that has been compiled from a low level language such as FORTRAN or C. For many R functions however, such as linear model calculations, or a cluster analysis, the main part of the work is done by a single call to code that has been compiled from C, and there is little loss of efficiency relative to use of a free-standing C, Fortran, C^{++} or Java program.

In the writing of computer programs, it is almost always a good strategy to begin by coding the calculations in a high level language such as R, then identifying the code segments that take the greater part of the time. These can then be coded in C or FORTRAN, and the compiled code called from R.

1.6. Parallel Processing

Parallel multi-processor systems offer a way to speed up throughput for calculations that lend themselves to efficient parallel implementations. There are two broad types of task – "embarrassingly parallel" tasks that split easily into parallel subtasks, and tasks where skill is required to organize calculations in a way that takes advantage of the parallel processors. Where the same embarrassingly parallel task will be run repeatedly, but with different inputs, it is straightforward for an expert to set up the calculation so that subsequent runs are relatively automatic. There may however be a substantial set-up cost for each new type of task. Moreover the set-up is likely to be specific to the statistical software system that will be used. Packages that are available for assisting parallel computing from within the environment of the R system include *Rmpi*, *snow*, and *qlapply*.

1.7. Unrealized Promises

The overview chapter described the potential for statistical expert systems, able to replace statistical experts over a wide range of problems. A really effective expert system would learn and continue learning in the way that human experts learn, by engaging in dialogue with skilled professionals. This is especially necessary because expert statistical professionals still have a long way to go in harnessing the full power of software tools that are currently available, or will appear in the next few years.

More limited goals of this general type are realistic, and need more attention than they have received. It is entirely possible to provide step by step guidance as users proceed

through an analysis, indicating what to look for and making suggestions for the steps that should follow. Any such system must be carefully researched, both to ensure that the advice is appropriate and to ensure that there is a balance between overly cryptic language and tedious detail. The developers will need to combine, though almost certainly not in the same individuals, a high level of relevant computing skill with the skills needed to expound and justify good statistical practice. The skills are similar to those required to write a high quality expository book, based on extensive experience with a wide range of statistical analysis problems, on modern statistical practice.

2. Advances in Routines Used for Statistical Computation

2.1. Numerical versus Non-numerical Routines

Numerical computing, although of central importance, is not the whole of statistical computing. Tasks that are not inherently numerical include text string manipulation, sorting for calculating ranks and order statistics, and so on. I note them here because they are important for some types of data analysis. They have a crucial role in the translation of high level language code into machine language, in the writing of interfaces to computer systems, in operating systems, and in data manipulation. Many of the developments that will be described in later sections of this chapter rely heavily on computations that are largely non-numerical.

Detailed discussion, e.g., of algorithms for handling such computations, is beyond the scope of this chapter.

2.2. The Structuring of Numerical Routines

Numerical analysts have built up many of the numerical computing abilities that are needed for statistical software in a highly structured way. There are three layers that are important in the context of this chapter, with each layer depending on any earlier layers. In the first layer are arithmetic operations and the evaluation of standard mathematical functions, in the second are relatively low-level routines that are widely important for other numerical computations, and in the third are least squares and related matrix calculations, and eigenvalue calculations. Brief comments will be made on each of these layers in turn.

In the first layer are arithmetic operations and the evaluation of standard mathematical functions. The release of the first of the IEEE standards in 1985 was a watershed for calculations at this level. The great majority of computing systems now implement this standard, providing a basis on which implementers of numerical computational methods can build.

In the second layer are a variety of simple calculations that occur frequently in numerical computation, for which there are good and bad ways to handle the calculation. An example is the calculation of the square root of $x_1^2 + x_2^2 + \ldots + x_n^2$. Perhaps surprisingly for novices, there are good and bad ways to do this calculation. Another is matrix multiplication. Specifications for sets of lower-level routines have been established in the numerical analysis literature, where they are known as BLAS

(basic linear algebra subroutines). The BLAS, or other such lower-level routines, are then used as building blocks in creating higher-level routines.

In the third layer are least squares and related calculations, and eigenvalue calculations. The LAPACK library provides fast, accurate and reliable code for handling such calculations. Numerical calculations at all these levels have been the subject of extensive research. The available software is well tested, mature and a reliable foundation on which to build other routines. Additionally, there are a variety of other libraries for numerical computing tasks. Particularly important is the commercial IMSL collection of routines. Routines that handle numerical statistical computations sit on top of these three layers, in ways that will be described below.

2.3. Non-numerical Routines

These include routines for sorting, for searching, for text processing, for parsing computer language inputs, for manipulation of images and so on. They provide important connecting links between the artificial languages used to communicate with computing systems, and the numerical routines that may finally do much of the work. They manipulate the contextual information (row and column labels are a simple example) needed for the interpretation of computer output. They help process computer output into a form that the user can assimilate. They are important for the building of interfaces to other systems, such as a database systems and the internet. Such interfacing is often preferable to the direct incorporation of the necessary features into the statistical system, and is the subject of much current statistical computing research.

2.4. Numerical Statistical Computing

Numerical statistical routines can now build very directly on the routines that are available in LAPACK and related libraries. Thus, in the scheme that was described in Section 2.1, they belong in layer 4 or higher. Once further numerical statistical routines have been developed and thoroughly tested, they can then themselves be a basis on which to build further routines. This adding of complexity, layer upon layer, can continue indefinitely, and is responsible for much of the power of modern statistical computing systems. In practice however, layers at level 4 and above are rarely clearly separated. For packages for the R system, there is a mechanism for specifying dependencies on other packages. Some routines ("functions") will call routines from other packages, while other routines will be independent of routines in other packages.

2.5. Calculations that Challenge Current Programs

Least squares and weighted least squares calculations build very directly on the routines that are available in packages such as LAPACK. Users can be assured that calculations are fast, reliable and accurate. Least squares calculations are now possible with huge data sets. Where data sets really are too large for direct use of least squares, several devices are available that, depending on the structure of the data, can reduce the calculation to a size that is manageable. These include carrying out calculations on samples from the data, and working with a reduced data set in which values of all variables are averaged over suitably chosen subsets of the data.

There are slightly greater complications for calculations that rely on the iteration of weighted least squares calculations. These include Generalized Linear Models in the style of Nelder and Wedderburn. Again, devices are often available that will reduce very large problems to a size that is manageable.

Standard forms of least squares calculations, and generalized linear model calculations, are appropriate for data where errors are independent. Often however, there is a structure to the random part of the model ("error"), and the analysis should have regard to it. Especially common is a hierarchical error structure. For example, in a sample survey there may be data in which streets are nested within a random sample of neighborhoods, houses within streets, and individuals within houses. If the aim is to generalize to other neighborhoods, then it may be necessary to take account of variation at four levels – neighborhoods, streets, houses and individuals. Multi-level models have been developed to handle problems of this type. A further level of complexity is added if individuals are followed through time. The use of a repeated measures model, allowing for a sequential correlation structure in the values at successive times, may then be appropriate.

Multi-level models, repeated measures models and spatial models test the limits of currently available software. Large data sets are especially likely to have the type of error structure that makes such models necessary. At the same time, the computational demands of such large data sets may be too severe for current software and hardware. In special cases, there are devices that can simplify the calculations. Many such problems currently require purpose-written programs, in order to get the necessary efficiency. The numerical computing challenges of these types of problem require more attention than they have so far received.

-
-
-

Bibliography

Anderson E., Bai Z., Bischof C., Blackford S., Demmel J., Dongarra J., Du Croz, J., Greenbaum, A., Hammarling S., McKenney, A. and Sorensen, D. (1999). LAPACK Users' Guide, Third Edition. Society for Industrial and Applied Mathematics. [LAPACK has many of the numerical routines that are fundamental in statistical computation.]

Goldberg, D. (1991). What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys* 23, 5–48. [Describes the implications of the IEEE 754 standard for computer arithmetic.]

Goosens, M., Rahtz, S., Gurari, E.M., Moore, R., and Sutor, R.S. (1999). The LaTeX Web Companion: Integrating TeX, HTML and XML. Addison-Wesley. [The TeX markup language was designed for use in the preparation of mathematical manuscripts. The much more recent HTML and XML markup languages

were designed for use with documents that may appear as web pages. This book describes how to use these three systems in an integrated manner.]

Ihaka, R. and Gentleman, R. 1996. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics* 5: 299-314. [Describes the R system and language. See also the web site http://www.r-project.org/]

Maindonald, J.H. and Braun, W.J. (2003). Data Analysis and Graphics Using R: An Example-Based Approach. Cambridge University Press. [This emphasizes modern statistical analysis approaches. It makes extensive use of graphs. A book of this type would not have been possible two decades ago. It has a brief introduction to the R system.]

Nelder, J.A. and Wedderburn, R.W.M. (1972). Generalized linear models. *Journal of the Royal Statistical Society, Series A* 135: 370-384. [This important paper presents a unified theoretical and computational approach to a wide range of models that had, prior to the appearance of this paper, been treated as distinct.]

Parmigiani, G., Garrett, E.S., Irizarry, R.A. and Zeger, S.L. 2003. The Analysis of Gene Expression Data. [This multi-author volume discusses methods for the analysis of gene expression data.]

Payne, R.W., Lane, P.W., Digby, P.G.N., Harding, S.A., Leech, P.K., Morgan, G.W., Todd, A.D., Thompson, R., Tunnicliffe Wilson, G., Welham, S.J. and White, R.P. 1993. *Genstat 5 Release 3 Reference Manual*. Oxford University Press: Oxford. [The Genstat statistical system has been widely used in agricultural research. It pioneered the use of several innovative statistical computing ideas, including the use of the Wilkinson and Rogers syntax for specifying models. It remains the system of preference for the analysis of balanced experimental designs.]

Senn, S. (2003). A conversation with John Nelder. Statistical Science 18: 118-131. [This interview gives interesting insights into the history of statistical computing.]

Spiegelhalter, D.J., Thomas, A. and Best, N.G. (1999). WinBUGS Version 1.2 User Manual. MRC Biostatistics Unit, Cambridge UK. [This is one of very few systems, intended for wide general use, that implement Bayesian approaches to statistical analysis.]

Wilkinson, G.N. & Rogers, C.E. (1973). Symbolic description of models for analysis of variance. *Applied Statistics* 22, 392-399. [This describes the model formula syntax that is now widely used across many different statistical systems.]

Biographical Sketch

After earlier experience as a schoolteacher and University lecturer, **John Maindonald** worked for 25 years as an applied statistician in publicly funded science in New Zealand. In 1996 he moved to Australia, first to the University of Newcastle and then to Australian National University. He has had wide experience in the use of statistical methodology in many different application areas, including entomology, horticulture and clinical medicine. He has coauthored numerous papers with application area specialists. His first book - Statistical Computation - was published by Wiley in 1984. He is the senior author of a second book - Data Analysis and Graphics Using R: An Example-Based Approach - that was published by Cambridge University Press in 2003. Currently, he is employed by the Centre for Bioinformation Science at the Australian National University, working on methods for the analysis of microarray and other gene expression data.