model (3). The main aim to introduce the chirp-like model (4) is that it behaves very similarly to the chirp model (3). It has been observed that the chirp like model exhibits same type of behavior as the chirp model and is capable of modelling similar physical phenomena. It has been observed by Grover et al. (2021) that it is very difficult to distinguish the signals generated by the two different models. Moreover, to analyze a nearly periodic data, less number of parameters may be required for a chirp-like model than a chirp model. On the other hand it has been shown that analytically as well as computationally the chirp-like model is easier to handle than the chirp model. Grover et al. (2021) established a very efficient estimation procedures of the unknown parameters, and also establish their asymptotic properties. There are several open problems associated with this model, see for example Kundu and Nandi (2021), more work is needed along this direction.

About the Author

Professor Debasis Kundu received his Ph.D in Statistics in 1989 from the Pennsylvania State University under the guidance of Prof. C.R. Rao. He has (co-)authored four books and about 350 peer reviewed papers. He is the Editor-in-Chief of the *Journal of the Indian Society of Probability and Statistics*. He is a Fellow of the National Academy of Sciences, India. He has received Distinguished Teacher's Award from the Indian Institute of Technology Kanpur and Distinguished Statistician's Award from the Indian Society of Probability and Statistics.

References

Bose, N.K., Rao, C.R.: Signal Processing and its Applications. Handbook of Statistics, vol. 10. North-Holland, Amsterdam (1993)

Djurić, P.M., Kay, S.M.: Parameter estimation of chirp signals. IEEE Trans. Acoust. Speech Signal Process. **38**, 2118–2126 (1990)

Fisher, R.A.: Tests of significance in Harmonic analysis. Proc. R. Soc. London Ser. A **125**, 54–59 (1929)

Grover, R., Kundu, D., Mitra, A.: Asymptotic properties of least squares estimators and sequential least squares estimators of a chirp-like signal model parameters. Circ. Syst. Signal Process. **40**, 5421–5467 (2021)

Grover, R., Sharma, A., Delcourt, T., Kundu, D.: Computationally efficient algorithm for frequency estimation of a two-dimensional sinusoidal model. Circ. Syst. Signal Process. **41**, 346–371 (2022)

Hannan, E.J.: The estimation of frequencies. J. Appl. Probab. **10**, 510–519 (1973)

Jennrich, R.I.: Asymptotic properties of non-linear least squares estimation. Ann. Math. Stat. **40**, 633–643 (1969)

Kay, S.M.: Modern Spectral Estimation. Prentice Hall, New York (1987)

Kundu, D.: Asymptotic theory of the least squares estimators of sinusoidal signal. Statistics **30**, 221–238 (1997)

Kundu, D., Nandi, S.: On chirp and some related signals analysis: a brief review and some new results. Sankhya Ser. A **83**, 844–890 (2021)

Lahiri, A., Kundu, D.: On parameter estimation of two-dimensional polynomial phase signal model. Stat. Sinica **27**, 1779–1792 (2017)

Nandi, S., Kundu, D.: Statistical Signal Processing: Frequency Estimation, 2nd edn. Springer, Singapore (2000)

Nandi, S., Kundu, D.: Asymptotic properties of the least squares estimators of the parameters of the chirp signals. Ann. Instit. Stat. Math. **56**, 529–544 (2004)

Quinn, B.G., Hannan, E.J.: The Estimation and Tracking of Frequency. Cambridge University Press, Cambridge (2001)

Rihaczek, A.W.: Principles of High Resolution Radar. McGraw-Hill, New York (1969)

Wu, C.F.J.: Asymptotic theory of non-linear least squares estimation. Ann. Stat. **9**, 501–513 (1981)

Statistical Software: Overview

Jan de Leeuw[1] and Miodrag Lovric[2]
[1]Department of Statistics, University of California, Los Angeles, CA, USA
[2]Department of Mathematics and Statistics, Radford University, Radford, VA, USA

Introduction

It is generally acknowledged that the most important changes in statistics in the last 50 years are driven by technology, more specifically, by the development and universal availability of fast computers and of devices to collect and

store ever-increasing amounts of data. Satellite remote sensing, large-scale sensor networks, continuous environmental monitoring, medical imaging, microarrays, the various genomes, and computerized surveys have not just created a need for new statistical techniques. These new forms of massive data collection also require efficient implementation of these new techniques in software. Thus, development of statistical software has become more and more important in the last decades.

Large datasets also create new problems of their own. In the early days, in which the *t*-test reigned, including the data in a published article was easy, and reproducing the results of the analysis did not take much effort. In fact, it was usually enough to provide the values of a small number of sufficient statistics. This is clearly no longer the case. Large datasets require a great deal of manipulation before they are ready for analysis, and the more complicated data analysis techniques often use special-purpose software and some tuning. This makes *reproducibility* a very significant problem. There is no science without replication, and the weakest form of replication is that two scientists analyzing the same data should arrive at the same results.

It is not possible to give a complete overview of all available statistical software. There are older publications, such as Francis (1979), in which detailed feature matrices for the various packages and libraries are given. This does not seem to be a useful approach anymore; there simply are too many programs and packages. In fact, many statisticians develop ad hoc software packages for their own projects.

We will give a short historical overview, mentioning the main general purpose packages and emphasizing the present state of the art. Niche players and special purpose software will be largely ignored. There is a well-known quote from Brian Ripley (2002): "Let's not kid ourselves: the most widely used piece of software for statistics is Excel." This is surely true, but it is equally true that only a tiny minority of statisticians have a degree in statistics. We have to distinguish between "statistical software" and the much wider terrain of "software for statistics." Only the first type is of interest to us here—we will go on kidding ourselves.

BMDP, SAS, SPSS

The original statistical software packages were written for IBM mainframes. BMDP was the first. Its development started in 1957, at the UCLA Health Computing Facility. SPSS arrived second, developed by social scientists at the University of Chicago, starting around 1968. SAS was almost simultaneous with SPSS, developed since 1968 by computational statisticians at North Carolina State University. The three competitors differed mainly in the type of clients they were targeting. And of course health scientists, social scientists, and business clients all needed the standard repertoire of statistical techniques, but in addition some more specialized methods important in their field. Thus, the packages diverged somewhat, although their basic components were very much the same.

Around 1985, all three packages added a version for personal computers, eventually developing WIMP (window, icon, menu, pointer) interfaces. Somewhat later, they also added matrix languages, thus introducing at least some form of extensibility and code sharing.

As in other branches of industry, there has been some consolidation. In 1996, SPSS bought BMDP and basically killed it, although BMDP-2009 was still sold in Europe by Statistical Solutions. However, as of 2017, BMDP is no longer available. In 2009, SPSS itself was bought by IBM and subsequently rebranded as IBM SPSS Statistics. It continues to be widely used in academia, business, and research for statistical analysis, data management, and reporting. The software has evolved to include advanced analytics and machine learning capabilities. SAS continued to innovate and adapt to changing technological landscapes. The company expanded its focus on business intelligence, data mining, and predictive analytics.

SAS was founded at North Carolina State University by Anthony James Barr, James Goodnight, John Sall, and Jane Helwig. The original purpose was to analyze agricultural research data. In the 2000s, SAS introduced cloud-based solutions and embraced open-source technologies to stay competitive in the evolving analytics market. The software suite became more modular, allowing users to choose specific components based on their needs. At present, SAS remains a prominent player in the analytics and business intelligence industry, providing a comprehensive suite of tools and solutions for data analysis, advanced analytics, machine learning, and artificial intelligence.

Data Desk, JMP, Stata

The second generation of statistics packages started appearing in the 1980s, with the breakthrough of the personal computer. Both Data Desk (1985) and JMP (1989) were, from the start, written for Macintosh, i.e., for the WIMP interface. They had no mainframe heritage and baggage. As a consequence, they had a much stronger emphasis on graphics, visualization, and exploratory data analysis.

Data Desk was developed by Paul Velleman, a former student of John Tukey. JMP was the brain child of John Sall, one of the cofounders and owners of SAS, although it existed and developed largely independent of the main SAS products. Both packages featured dynamic graphics and used graphical widgets to portray and interactively manipulate datasets. There was much emphasis on brushing, zooming, and spinning. Both Data Desk and JMP have their users and admirers, but both packages never became dominant in either statistical research or statistical applications. They were important, precisely because they emphasized graphics and interaction, but they were still too rigid and too difficult to extend.

Stata, another second-generation package for the personal computer, was an interesting hybrid of a different kind. It was developed since 1985, like BMDP starting in Los Angeles, near UCLA. Stata had a CLI (command line interface) and did not get a GUI until 2003. It emphasized, from the start, extensibility and user-contributed code. Stata did not get its own matrix language Mata until Stata-9, in 2007.

Much of Stata's popularity is due to its huge archive of contributed code and a delivery mechanism that uses the Internet to allow for automatic downloads of updates and new submissions. Stata is very popular in the social sciences, where it attracts those users that need to develop and customize techniques, instead of using the more inflexible procedures of SPSS or SAS. For such users, a CLI is often preferable to a GUI.

Until Stata developed its contributed code techniques, the main repository had been CMU's statlib, modeled on netlib, which was based on the older network interfaces provided by ftp and email. There were no clear organizing principles, and the code generally was FORTRAN or C, which had to be compiled to be useful. We will see that the graphics from Data Desk and JMP, and the command line and code delivery methods from Stata, were carried over into the next generation.

S, LISP-STAT, R

Work had on the next generation of statistical computing systems had already started before 1980, but it mostly took place in research labs. Bell Laboratories in Murray Hill, N.J., as was to be expected, was the main center for these developments.

At Bell, John Chambers and his group started developing the S language in the late seventies. S can be thought of as a statistical version of MATLAB, as a language and an interpreter wrapped around compiled code for numerical analysis and probability. It went through various major upgrades and implementations in the eighties, moving from mainframes to VAXes and then to PCs. S developed into a general purpose language, with a strong compiled library of linear algebra, probability and optimization, and with implementations of both classical and modern

statistical procedures. The first fifteen years of S history are ably reviewed by Becker (1994), and there is a thirty year history of the S language in Chambers (2008, Appendix A). The statistical techniques that were implemented, for example, in the *White Book* (Chambers and Hastie 1992), were considerably more up to date than techniques typically found in SPSS or SAS. Moreover, the S system was built on a rich language, unlike Stata, which until recently just had a fairly large number of isolated data manipulation and analysis commands. Statlib started a valuable code exchange of public domain S programs.

For a long time, S was freely available to academic institutions, but it remained a product used only in the higher reaches of academia. AT&T, later Lucent, sold S to the Insightful Corporation, which marketed the product as S-plus, initially quite successfully. Books such as Venables and Ripley (1994, 2000) effectively promoted its use in both applied and theoretical statistics. Its popularity was increasing rapidly, even before the advent of R in the late nineties. S-plus has been quite completely overtaken by R. Insightful was recently acquired by TIBCO, and S-plus is now TIBCO Spotfire S+. We need not longer consider it as a serious contender.

There were two truly exciting developments in the early nineties. Tierney (1990) developed LISP-STAT, a statistics environment embedded in a Lisp interpreter. It provided a good alternative to S, because it was more readily available, more friendly to personal computers, and completely open source. It could, like S, easily be extended with code written in either Lisp or C. This made it suitable as a research tool, because statisticians could rapidly prototype their new techniques and distribute them along with their articles. LISP-STAT, like Data Desk and JMP, also had interesting dynamic graphics capabilities, but now the graphics could be programmed and extended quite easily. Around 2000 active development of LISP-STAT stopped, and R became available as an alternative (Valero-Mora and Udina 2004).

R was written as an alternative implementation of the S language, using some ideas from the world of Lisp and Scheme (Ihaka and Gentleman 1996). The short history of R is a quite unbelievable success story. It has rapidly taken over the academic world of statistical computation and computational statistics and to an ever-increasing extent the world of statistics teaching, publishing, and real-world application. SAS and SPSS, which initially tended to ignore and in some cases belittle R, have been forced to include interfaces to R, or even complete R interpreters, in their main products. SPSS has a Python extension, which can run R since SPSS-16. The SAS matrix language SAS/IML, starting at version 3.2., has an interface to an R interpreter.

An important contribution to the increased usability and popularity of R was the Rstudio integrated development environment (IDE), developed by Posit PBC (formerly Rstudio PBC and RStudio Inc) since 2010, with the 1.0 release in 2016. The Rstudio desktop application runs on all major operating systems and is available for free.

It is, among other things, a graphical user interface for running and maintaining R, for developing, testing, and packaging R code, for writing and publishing dynamic literate programming books and articles containing graphics and executable R code, for using cloud-based R programming, and for organizing collaborative scientific work into Rstudio projects that are also with Github repositories. Rstudio, like R, has promoted open-source software and open access publishing, and their staff has contributed a great many R packages and R publications to the public domain.

R is many things to many people: a rapid prototyping environment for statistical techniques, a vehicle for computational statistics, an environment for routine statistical analysis, and a basis for teaching statistics at all levels. Or, going back to the origins of S, a convenient interpreter to wrap existing compiled code. R, like S, was never designed for this all-encompassing role, and the basic engine is straining to support the rate of change in the size and nature of data and the developments in hardware.

The success of R is both dynamic and liberating. But it remains an open-source project, and nobody is really in charge. One can continue to tag on packages extending the basic functionality

of R to incorporate XML, multicore processing, cluster and grid computing, web scraping, and so on. But the resulting system is in danger of bursting at the seams. There are now four ways to do (or pretend to do) object-oriented programming, four different systems to do graphics, and four different ways to link in compiled C code. Currently (November 2023), CRAN (Comprehensive R Archive Network) package repository features 20026 available packages. New statistical methods are quickly implemented in R. Many statisticians, and many future statisticians, learn R as their first programming language, instead of learning real programming languages such as Python, Lisp, or even C and FORTRAN. It seems realistic to worry at least somewhat about the future and to anticipate the possibility that all of those thousands of flowers that are now blooming may wilt rather quickly.

Statistical Software Packages for Bayesian Analysis

In recent years, there has been a remarkable increase in the prominence of Bayesian computation, as highlighted in the comprehensive historical review by Martin et al. (2020). Cameletti and Gómez-Rubio (2021) provided a comparison of a number of software packages for Bayesian inference on different topics such as general packages for hierarchical linear model fitting, survival models, clinical trials, missing values, time series, hypothesis testing, priors, approximate Bayesian computation, and others.

SAS software includes Bayesian methods through its PROC MCMC procedure. It enables users to perform Bayesian statistical modeling and inference using Markov Chain Monte Carlo (MCMC) methods. Users can specify parameters, define prior distributions, and model likelihood functions to conduct Bayesian analyses, making it applicable to a wide range of statistical modeling tasks. The procedure allows for the estimation of posterior distributions, providing insights into parameter uncertainty.

SPSS offers support for the limited number of Bayesian methods including one sample and paired sample t-tests, binomial proportion tests, Poisson distribution analysis, linear regression, one-way ANOVA, and similar. Users can execute Bayesian modeling and analysis using external tools or programming languages (such as R of Python) and then import the results back into SPSS for further data manipulation, visualization, and reporting.

STATA enables Bayesian analysis through its dedicated suite of commands, collectively referred to as the "bayes" prefix. The bayes suite in Stata provides users with a range of tools for conducting Bayesian statistical modeling and inference. Users can specify Bayesian models, set prior distributions, and perform analyses using Bayesian estimation methods. Currently (November 2023), STATA offers over 60 likelihood models and incorporates many prior distributions (including normal, lognormal, multivariate normal, gamma, beta, and Wishart), adaptive and hybrid Metropolis-Hastings, and Full Gibbs sampling for some models, and also calculation of the deviance information criterion and Bayes factor.

R offers many packages for Bayesian statistics including (a) *general purpose model-fitting* packages (such as arm, BACCO, bayesm, BayesianTools, LaplaceDemon, MCMCpack), *application-specific* packages (e.g., bayesanova, BayesFactor, BMA, bridgesampling, and RoBMA), *Bayesian tree models* (including dbarts, bartBMA, bartCause and bartcs), *causal inference* (such as, bama, bartCause, BayesTree, BDgraph, and blavaan), packages for computational methods (including abc, bayesian, EntropyMCMC, etc.), and many other packages for discrete data, contingency tables, meta-analysis, graphics, hierarchical models, machine learning methods, factor analysis, handling of missing data, mixture models, network models, shrinkage/variable selection/Gaussian process, spatial models, survival models, time series models, post-estimation tools, Bayesian model for specific disciplines, packages for learning Bayesian statistics, and packages that link R to other sampling engines (including JAGS, OpenBUGS, Stan, WinBUGS).

Python has several packages for Bayesian statistics such as PyMC3, PyStan (interface to Stan), and arviz.

Finally, for the overview of the software packages exclusively devoted to the Bayesian analysis, see the entry **Software packages for Bayesian analysis**.

Open Source and Reproducibility

One of the consequences of the computer and Internet revolution is that more and more scientists promote open-source software and reproducible research. Science should be, per definition, both open and reproducible. In the context of statistics (Gentleman and Temple-Lang 2004), this means that the published article or report is not the complete scientific result. In order for the results to be reproducible, we should also have access to the data and to a copy of the computational environment in which the calculations were made.

Publishing is becoming more open, with E-journals and preprint servers providing open access. Electronic publishing makes both open source and reproducibility more easy to realize. The Journal of Statistical Software, at http://www.jstatsoft.org, was for a long time the only journal that published and reviewed statistical software, insisting on complete code and completely reproducible examples. Literate programming systems are becoming more popular ways to integrate text and computations in statistical publications. Using the R Markdown language, created and promoted by RStudio, with the corresponding knitr, bookdown, blogdown, and Quarto packages makes it possible and straightforward for anyone to write articles and books and to publish them on open access sites such as https://rpubs.com or https://bookdown.org or on preprint servers such as https://arxiv.org.

We started this overview of statistical software by indicating that the computer revolution has driven much of the recent development of statistics, by increasing the size and availability of data. Replacement of mainframes by minis, and eventually by powerful personal computers, has determined the directions in the development of statistical software. In more recent times, the Internet revolution has accelerated these trends, and is changing the way scientific knowledge, of which statistical software is just one example, is disseminated.

About the Author

Dr. Jan de Leeuw is a distinguished professor emeritus and founding chair of the Department of Statistics, UCLA. He has a 1973 Ph.D. in Social Sciences from the University of Leiden, Netherlands. He came to UCLA in 1987, after leading the Department of Data Theory at the University of Leiden for about 10 years. He is elected fellow, Royal Statistical Society (1984), elected member, International Statistical Institute (1986), member, Royal Netherlands Academy of Sciences (1987), and elected fellow, Institute of Mathematical Statistics (2001) and American Statistical Association (2001). Dr. de Leeuw is the founding editor and from 1996 until 2014 editor in chief of *Journal of Statistical Software* and the editor in chief of the *Journal of Multivariate Analysis* (1997–2014). He is a former president of the *Psychometric Society* (1987). Professor de Leeuw has (co-)authored over 550 papers, book chapters, and reviews, including *Introducing Multilevel Modeling* (with Ita Kreft, Sage Publications Ltd, 1998) and *Handbook of Multilevel Analysis* (edited with Erik Meijer, Springer-Verlag New York, 2007).

References

Becker, R.A.: A Brief History of S. Technical report, AT&T Bell Laboratories, Murray Hill (1994). http://www2.research.att.com/areas/stat/doc/94.11.ps

Cameletti, M., Gómez-Rubio, V.: J. Stat. Softw. **100**(1), 1–7 (2021). https://doi.org/10.18637/jss.v100.i01.

Chambers, J.M.: Software for Data Analysis: Programming with R. Statistics and Computing. Springer, New York (2008)

Chambers, J.M., Hastie, T.J. (eds.): Statistical Models in S. Wadsworth (1992)

Francis, I.: A Comparitive Review of Statistical Software. International Association for Statistical Computing, Voorburg, The Netherlands (1979)
Gentleman, R., Temple-Lang, D.: Statistical Analyses and Reproducible Research. Bioconductor Project Working Papers 2 (2004). http://www.bepress.com/cgi/viewcontent.cgi?article=1001&context=biocond%uctor.
Ihaka, R., Gentleman, R.: R: a Language for Data Analysis and Graphics. J. Comput. Graph. Stat. **5**, 299–314 (1996)
Martin, G.M., Frazier, D.T., Robert, C.P.: Computing Bayes: Bayesian computation from 1763 to the 21st Century (2020). https://arxiv.org/abs/2004.06425
Ripley, B.D.: Statistical Methods Need Software: A View of Statistical Computing. Presentation RSS Meeting (2002). http://www.stats.ox.ac.uk/~ripley/RSS2002.pdf
Tierney, L.: LISP-STAT. An Object-Oriented Environment for Statistical Computing and Dynamic Graphics. Wiley, New York (1990)
Valero-Mora, P.M., Udina, F.: Special issue: Lisp-stat: past, present and future. J. Stat. Softw. **13** (2004) http://www.jstatsoft.org/v13
Venables, W.N., B.D. Ripley: S Programming. Statistics and Computing. Springer, New York (2000)
Venables, W.N., Ripley, B.D.: Modern Applied Statistics with S, 1st edn. Springer, Berlin (1994)

Statistical View of Information Theory, A

Adnan M. Awad
University of Jordan, Amman, Jordan

Information Theory has origins and applications in several fields such as: thermodynamics, communication theory, computer science, economics, biology, mathematics, probability and statistics. Due to this diversity, there are numerous information measures in the literature. Kullback (1978), Sakamoto (1986), and Pardo (2006) have applied several of these measures to almost all statistical inference problems.

According to The Likelihood Principle, all experimental information relevant to a parameter θ is mainly contained in the likelihood function $L(\theta)$ of the underlying distribution. Bartlett's information measure is given by $-\log(L(\theta))$. Entropy measures are expectations of functions of the likelihood. Divergence measures are also expectations of functions of likelihood ratios. In addition, Fisher-like information measures are expectations of functions of derivatives of the log-likelihood. DasGupta (2008, ch 2) reported several relations among members of these information measures. In sequential analysis, Wald (1947, p53) showed earlier that the average sample number depends on a divergence measure of the form

$$E_\theta\left[\log\frac{f(X,\theta_1)}{f(X,\theta_0)}\right]$$

where θ_0 and θ_1 are the assumed values of the parameter θ of the density function f of the random variable X under the null and the alternative hypothesis, respectively.

It is worth noting that, and from the point of view of decision making, the expected change in utility can be used as a quantitative measure of the worth of an experiment. In this regard Bayes' rule can be viewed as a mechanism that processes information contained in data to update the prior distribution into the posterior probability distribution.

Furthermore, according to Jaynes' Principle of Maximum Entropy (1957), information in a probabilistic model is the available moment constraints on this model. This principle is in fact a generalization of Laplace's Principle of Insufficient Reason.

From a statistical point of view, one should concentrate on the statistical interpretation of properties of entropy-information measures with regard to the extent of their agreement with statistical theorems and to their degree of success in statistical applications.

The following provides a discussion of preceding issues with particular concentration on Shannon's entropy. For more details, the reader can consult the list of references.

1. Consider a discrete random variable X taking a finite number of values $\vec{X} = (x_1, \ldots, x_n)$ with probability vector $P = (p_1, \ldots, p_n)$. Shannon's entropy (information) of P or of X (1948) is given by

$$H(X) = H(P) = -\sum_{i=1}^{n} p_i \log(p_i).$$