# STATISTICAL SOFTWARE – OVERVIEW

JAN DE LEEUW

## 1. 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, micro-arrays, 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 data sets 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 data sets 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

*Date*: Wednesday 23rd December, 2009 — 13h 14min — Typeset in TIMES ROMAN.

*Key words and phrases.* Statistical Software, R, Reproducibility, Open Source.

detailed feature matrices for the various packages and libraries are given. This does not seem to be a useful approach any more, 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.

## 2. 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 is still sold in Europe by Statistical Solutions. It is now, however, no longer a serious contender. In 2009 SPSS itself was bought by IBM, where it now continues as PASW (Predictive Analytics Software). As the name change indicates, the emphasis in SPSS has shifted from social science data analysis to business analytics. The same development is going on at SAS, which was originally the Statistical Analysis System. Currently SAS is not an acronym any more. Its main products are SAS Analytics and SAS Business Intelligence, indicating that the main client base is now in the corporate and business community. Both SPSS (now PASW) and SAS continue to have their statistics modules, but the keywords have definitely shifted to analytics, forecasting, decision, and marketing.

## 3. DATA DESK, JMP, STATA

The second generation of statistics packages started appearing in the 1980's, 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 co-founders 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 data sets. 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.

## 4. 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 VAX'es and then to PC's. 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 build 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. Luke 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 extend 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.

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. There are thousands of add-on packages, with enormous redundancies, and often with code that is not very good and documentation that is poor. 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.

## 5. 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, preprint servers, and open access. Electronic publishing makes both open source and reproducibility more easy to realize. The Journal of Statistical Software, at `http://www.jstatsoft.org`, the only journal that publishes and reviews statistical software, insists on complete code and completely reproducible examples. Literate Programming systems such as Sweave, at `http://www.stat.uni-muenchen.de/~leisch/Sweave/`, are becoming more popular ways to integrate text and computations in statistical publications.

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.

## References

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

J.M. Chambers. *Software for Data Analysis: Programming with R*. Statistics and Computing. Springer Verlag, New York, N.Y., 2008.

J.M. Chambers and T.J. Hastie, editors. *Statistical Models in S*. Wadsworth, 1992.

I. Francis. *A Comparitive Review of Statistical Software*. International Association for Statistical Computing, Voorburg, The Netherlands, 1979.

R. Gentleman and D. Temple-Lang. Statistical Analyses and Reproducible Research. Bioconductor Project Working Papers 2, 2004. URL `http://www.bepress.com/cgi/viewcontent.cgi?article=1001&context=bioconductor`.

R. Ihaka and R. Gentleman. R: a Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics*, 5:299–314, 1996.

B.D. Ripley. Statistical Methods *Need* Software: A View of Statistical Computing. Presentation RSS Meeting, September 2002. URL `http://www.stats.ox.ac.uk/~ripley/RSS2002.pdf`.

L. Tierney. *LISP-STAT. An Object-Oriented Environment for Statistical Computing and Dynamic Graphics*. Wiley, 1990.

P.M. Valero-Mora and F. Udina. Special issue: Lisp-stat: Past, present and future. *Journal of Statistical Software*, 13, 2004. URL `http://www.jstatsoft.org/v13`.

W.N. Venables and B.D. Ripley. *Modern Applied Statistics with S*. Springer, first edition, 1994.

W.N. Venables and B.D. Ripley. *S Programming*. Statistics and Computing. Springer Verlag, New York, N.Y., 2000.

DEPARTMENT OF STATISTICS, UNIVERSITY OF CALIFORNIA, LOS ANGELES, CA 90095-1554

*E-mail address*, Jan de Leeuw: `deleeuw@stat.ucla.edu`

*URL*, Jan de Leeuw: `http://gifi.stat.ucla.edu`