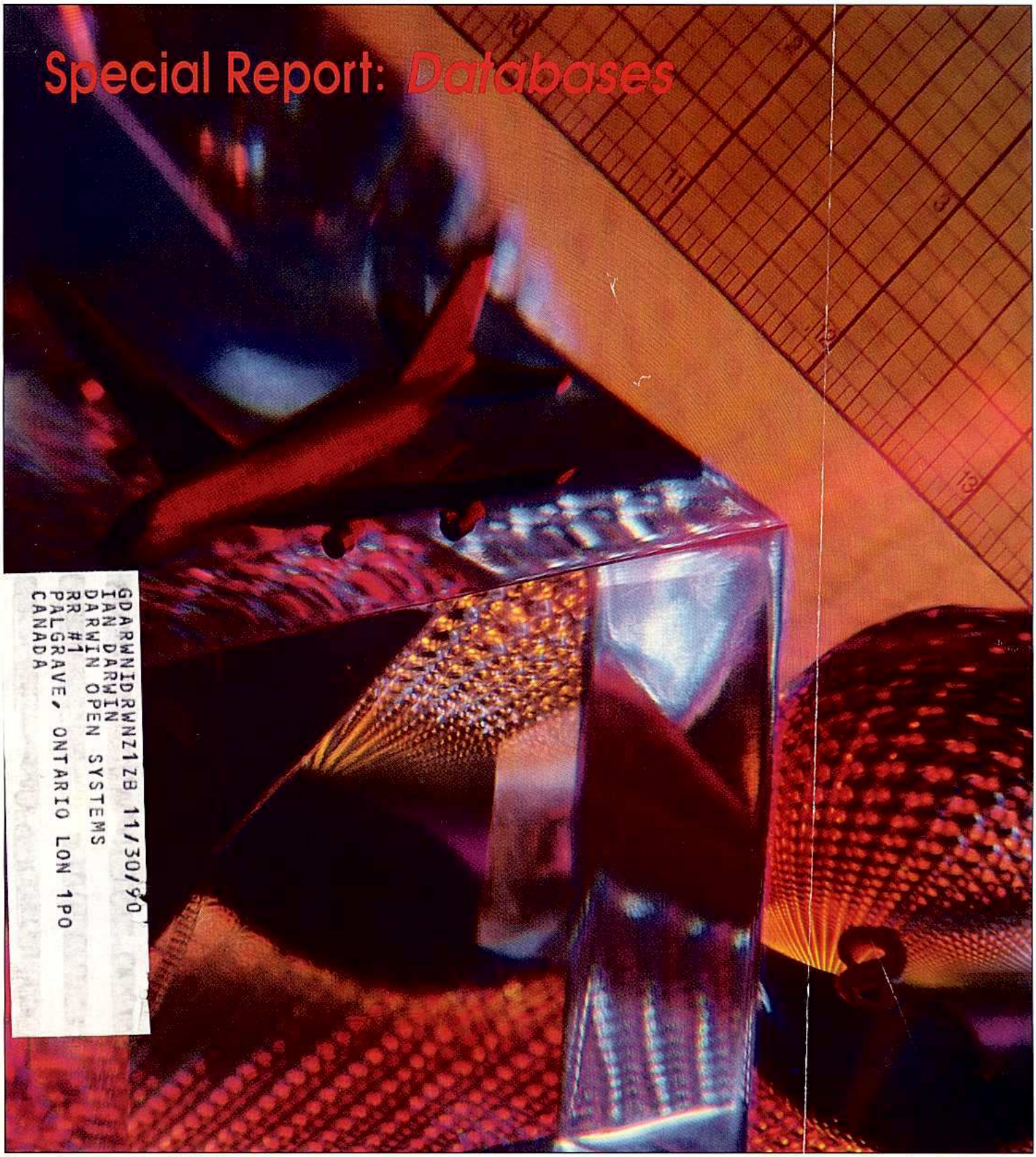
SunExpert

An Independent Forum for Open Systems

MAY 1991 Vol. 2 Num. 5 $4.50

Special Report: *Databases*



Review: S-PLUS

News: Sun's MP Plan

S-PLUS:

An Interactive Programming Environment for Data Analysis and Graphics

by IAN F. DARWIN

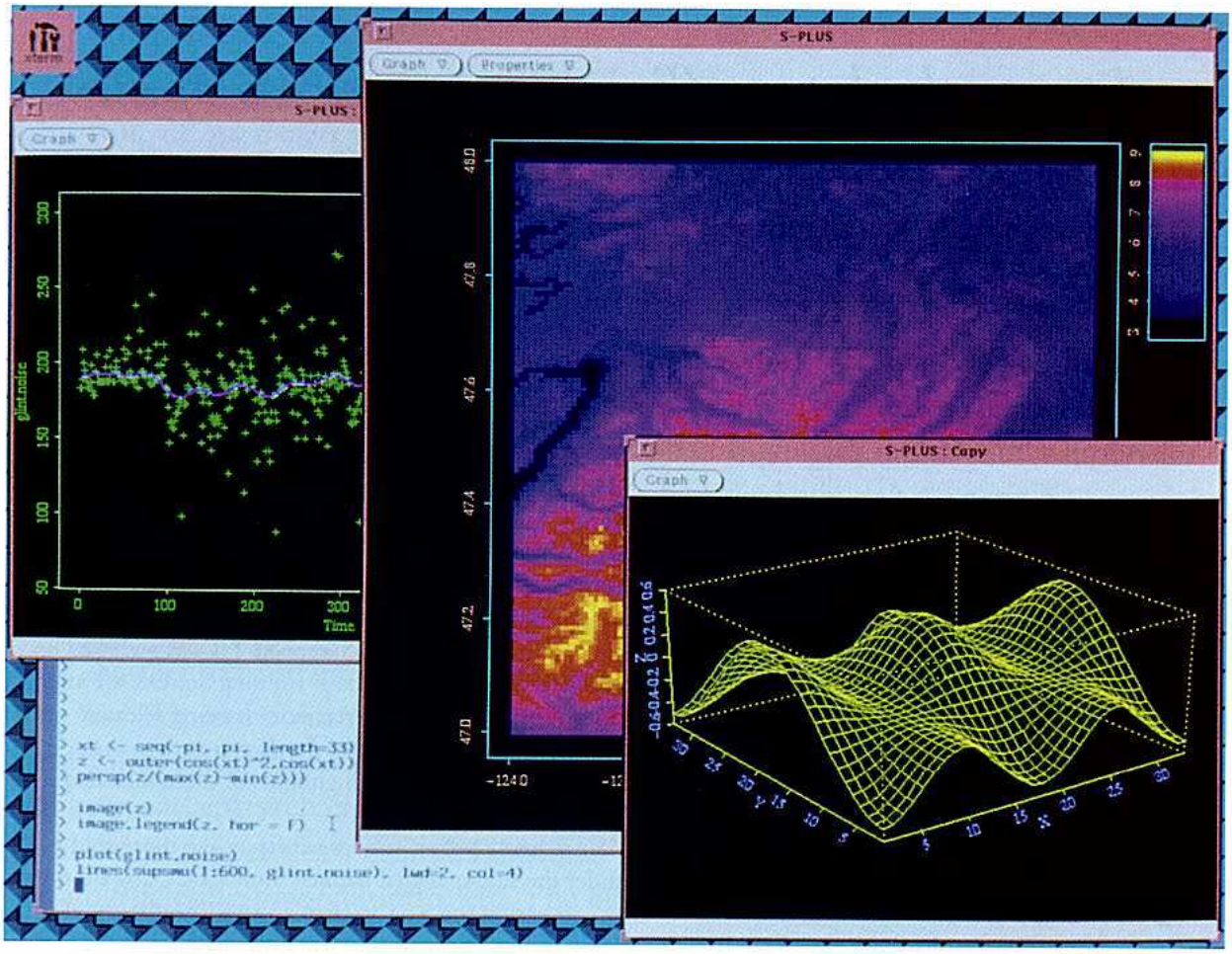
If you were originally attracted to UNIX because of the power it derives from careful combinations of a few well-chosen primitives, then the S language is for you. S is to spreadsheets as the UNIX shell is to simple-but-limiting, menu-driven canned applications. Rather than blather on about it, though, I thought I'd start with a real-world example, that of trying to get rich with your computer. Let's say we have last year's winning numbers from the "Pick 3" lottery (run by the Ontario Lottery Corp.) in a file called `pick3.raw`, and we want to look for regularities (or irregularities) that will help us pick winning tickets. You can start up the S-PLUS package by typing the command `S-PLUS`. Users of older versions of S will likely add a link name of S, which I did.

Pick3 is the name of an S variable in which we'll hold the data. (S uses "<-" or "_" as its normal assignment operator, since "=" means something else.) The `read` function reads data from a UNIX file, while `ts` makes the variable data a time series, starting at the beginning of 1990, with daily (365) data. I printed the array to proof-read some of the leading data points (see Listing, opposite page). The first seven points are NA (not available) just because I don't have data for them. S allows "NA," for missing values, in most places that data is allowed. This is to account for the real-world nature of statistical data.

Once we have this data read-in to S, we can plot it using a high-level graphing language:

```
> X11()
> hist(pick3)
> title(main="Ontario Pick3 Lottery — Jan 8, 1990 - Oct 6, 1990")
```

The first line tells S which graphics device to use–others include `suntools()`, `pscript()`, `hplj()` and various graphics terminals–and starts up the display if it's an online device like X11. The parentheses are needed because everything in S is a function call. (Indeed, my only problem in getting used to the syntax of S was remembering the parenthesis. The overall syntax is otherwise suggestive of a fully interactive version of `awk`.)



Although S-PLUS makes light work of plotting data, its real strength is statistical analysis.

The second line plots the histogram shown in Figure 1, and the last line sets a top title on the graph. What can we infer from this graph? There are three digits in each number; if they were evenly distributed, and the sample were large enough, you'd expect a straight line across the top. But in this sample, winning numbers beginning with "9" are substantially more likely than those beginning with "8," for example.

If S were just a plotting package, you'd never have heard about it, for I wouldn't review it on that basis alone. S also harnesses the power of a modern computer to tame the tasks of analyzing and graphing data. For this reason, S is often viewed as a statistical-analysis language, though there is much more to it than that.

S includes a rich library of statistical operations, as well as a simple extension language, which lets you write S functions by building on the work of others, exactly the way shell programs leverage on the availability of other programs. The S-PLUS statistical functions provide almost all statistical analyses in common use, including simple descriptive statistics, classic multivariate statistics, time-series analysis and ARIMA modeling, survival analysis, linear regression and more. In addition, S-PLUS contains an emphasis not present in most statistics packages, an emphasis on "robust statis-

Listing

```
% S
S-PLUS : Copyright (c) 1988, 1989, 1990 Statistical Sciences Inc. S : Copyright AT&T.
Version 2.3 Release 1 for Sun3 under SunOS 4.0.x : 1990
Working data will be in /home/darian/ian/.Data
> pick3 <- ts(read("pick3.raw"), 1990, freq=365) Read 245 items
> pick3

      1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
1990: NA  NA  NA  NA  NA  NA  NA 892 319 967 455 672 543  NA 286 419 700 777

     19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
1990:641 729 777 715 369 936 840  35  NA  37 347  39 380 878  61  NA 482 743
... etc ...
```

According to the latest official figures, 43% of all statistics are totally worthless.
–Berkeley fortune file

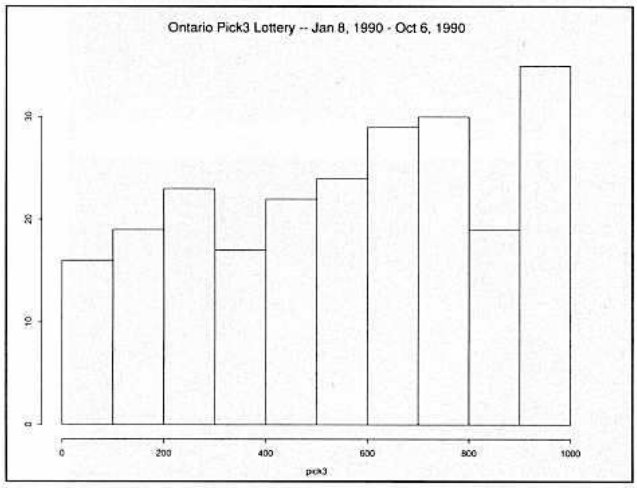


Figure 1

tics." Robust statistics is a relatively new field, and Statistical Sciences Inc. (StatSci) has been aggressive in bringing the latest in robust statistics into its package.

It's been about a decade since the S language first appeared. Written by Richard Becker and John Chambers of AT&T Bell Laboratories, S was sold by AT&T in source form only. Most of the purchasers were universities, who (as with UNIX itself) got the source at a radical discount. This led to a widespread but informal "S community" of users, primarily in universities, research institutes and the like.

Recently the second major version of S, called "The New S Language," was put together by Becker, Chambers and Allan Wilks, along with a textbook of the same name. AT&T decided to allow binary sublicensing of S to make it more widely useful in the UNIX community. Enter StatSci, which began enhancing, debugging and supporting the S language in 1987. Its product, S-PLUS, is a complete superset of the S package from AT&T, but it is a less expensive, binary-only package available for most UNIX platforms (and even some DOS platforms). AT&T continues to enhance its version of S, and StatSci continues to track and incorporate new versions from AT&T. S-PLUS Version 2.3 was reviewed on an 8-MB monochrome Sun-3 running SunOS 4.1 and OpenWindows. S-PLUS Version 3 is being released around the time this issue of *SunExpert* goes to press; it was not reviewed but the highlights of it are included in "S-PLUS Release 3."

Since many people use spreadsheets like those sold under the suit-spangled banner "Lotus 1-2-3," we'll start by comparing the S package to a spreadsheet's way of doing things. We'll also look at how the S language and the S-PLUS implementation fit in with UNIX and with OpenWindows. And finally there's an overall evaluation of the strengths and weaknesses of S and S-PLUS.

It's Not a Spreadsheet

It's not a spreadsheet. It's a line-based interpreter, and it's open-ended. Nowadays spreadsheets claim to be programmable, what with having a macro processor and all. But S is really programmable; you can (on most versions of UNIX) dynamically draw in functions from C or FORTRAN ".o" files, and make them part of the language. This is used, for example, in some of the device-

Figure 2

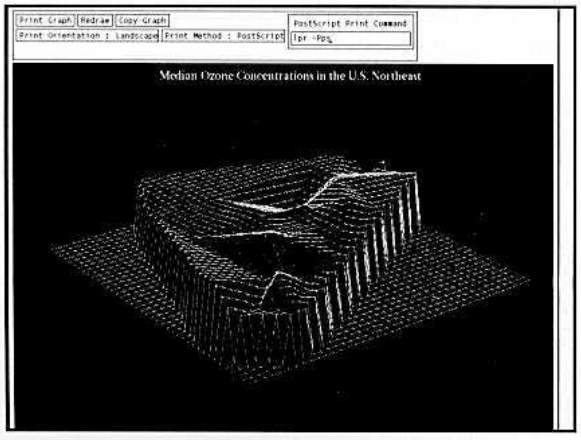
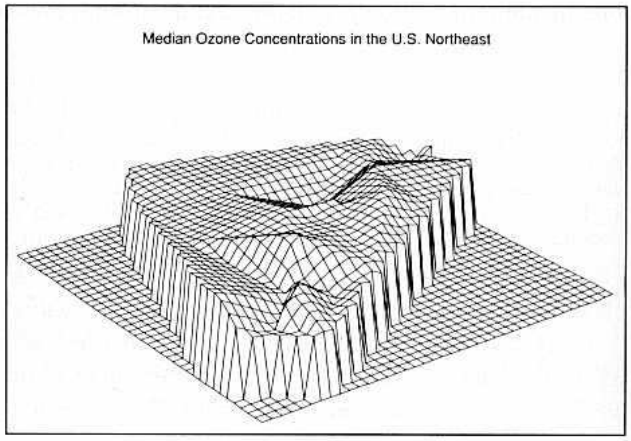


Figure 3



96.37% of all statistics are made up.

–attributed to Kevin D. Quitt,

kdq@demott.com

specific graphics routines, and many of the mathematical operators. Thus the vast library of routines written over the years in C and FORTRAN can be considered to be directly available in S. There is no need to pipe data out to a program nor to leave your environment.

On the other hand, there is no full-screen, X-Y-based spreadsheet data editor included with S. The S community might argue that none is needed, since you can edit a file with `vi` or `emacs` and read it in. Further, if you really need the spreadsheet-like data-editing capability, you can always write one. In fact, StatSci has recently done so–again, see "S-PLUS Release 3." S is a very "open" system, not in the marketing sense, but in the sense that you can interconnect it with a large variety of existing software. And the standard textbook describes what you need to know to interface your own C-language functions. To demonstrate, consider this simple bit of C code (function `tse`, stored in file `tse.c`) that takes two longs and multiplies them. All arguments are passed as pointers since they are usually arrays. The normal convention in S is that the answer is returned as part of the argument list, so we store it in place of "x."

```
/*
* Try out the .C interface in S.
*/ tse(x, y)
long *x, *y; /* integer in S */
{
*x *= *y;
}
```

To use this with no fancy packaging, I need only `make tse.o` in a shell window, then go back into S and type

```
dyn.load("tse.o")       # load tse.o into S
# tse(4, 8);
.C("tse," x=as.integer(4), y= as.integer(8))
```

The function .C calls an external C function named by its first argument, with the remaining arguments passed to the C function. This particular call prints out both the "32" (result stored in place of the 4) and the "8" that was input. To get just the answer, I use the S convention of appending a dollar sign and the component (analogous to C's use of -> and field names):

```
.C("tse," x=as.integer(4), y= as.integer(8))$x
```

which prints 32. For anything more realistic, I would probably write a `tse` function in S, like this:

```
tse <- function(x, y)
   .C("tse," x=as.integer(x), as.integer(y))$x
```

Then I can just say `tse(4,8)` and get the same answer.

As this example just begins to show, the programming language is very versatile, much like the UNIX shell. On the other hand, it does not do the same level of hand-holding a spreadsheet does. Just as with the UNIX shell, you are expected to learn to use the system's features if you want to make good use of it and develop new functions.

Again, as with the shell, you can package up functions for others to use, and your function becomes just another function that is part of the environment for others to use without having to learn the inner workings. Thus both S and UNIX encourage users and programmers to build on the work of those who have gone before.

But there's more. S comes from a background of "exploratory data analysis." This suggests that you are willing to think about statistical analysis. Now I know a little about statistics, but I am not a statistician. I do know some professional statisticians who use it in preference to other packages they have access to. And to help you out in learning to use the analytical tools, S comes with a library of 50 or so datasets. These can be used to try out the examples in the documentation.

No Rough Edges

Unlike some mainframe statistics packages, the S package originated in the UNIX environment: It fits in well and feels like a part of UNIX. It can read or write UNIX files, call UNIX processes from the S command line (or within an S function), etc. It uses the UNIX editor (default `vi` in this version) to edit functions, to edit command recall and so on.

The textbook is full of examples of clever uses of S, and many of these demonstrate methods for using the rest of UNIX to advantage. Page 188 of the textbook, for example, features the source for a `tbl` function that writes objects of any shape into a file suitable for processing by the `troff` pre-processor `tbl`, to make neatly printed tables in a formatted document.

The graphics modes provide a very convenient way of quickly visualizing data and also producing publication-quality graphics. The screen modes (X11 and Suntools) cooperated just fine with OpenWindows 2.0. The X version uses X resources as it should, and includes an app-defaults file that the installation procedure installs. The X window shows a spiffy little graph icon when you close it, under `olwm` and `twm` at least. The two common hardcopy modes (hplj, pscript) can be accessed directly from within S-PLUS, or can also be called from within one of the two screen modes. Since S keeps graphics objects as a display list, a drawing can be rescaled, drawn to a different device, and so on. The pscript version emits conforming Encapsulated PostScript that `pageview` was happy with. Other hardcopy devices supported include HPGL for Hewlett-Packard HP-GL plotters and pic for the `troff` preprocessor of the same name. The only problems with the drivers were minor. The first time I tried X11 under OpenWindows, I got this flurry of flames:

```
> X11()
Client is not authorized to connect to server
X11 Toolkit Error : Can't Open display
Bailing out!
X11 driver could not start
Try again after changing something
Dumped
>
```

This is basically "permission denied" for opening the X connection, apparently because the X11 function (`splus/x/dev.X11`) is not linked with an X library supporting the `MIT_MAGIC_COOKIE` authorization protocol, which OpenWindows and other modern versions of X11 use. And it's linked statically against the X libraries, so it won't take advantage of any newer X libraries you may have. A simple bypass is to either use `xhost` to enable clients from your host to access its display without Xauth, or (if you are very brave) use `openwin -noauth`. Later releases of S-PLUS will certainly fix this. Later releases will also offer the Open Look and Motif interfaces under X.

White On Black

In both X11 and Suntools graphics modes, the default screen is white lines on black background, which is fine. But, in Suntools, the cursor in the graphics window is only printed in black, so to see the cursor you have to flip to white on black. Then you get a lovely "S+" cursor, with the hot spot on the + sign, of course. In X11, you always get an X11-ish arrow cursor that is visible because of cursor masks. This is a non-trivial buglet for some users, who wish to use the cursor for, say, pointing at particular points and getting their coordinates. S calls this "identifying the outliers."

The screen graphics have an extra provided by Statistical Sciences: You can print a graph just by clicking on a button. For example, this simple perspective graph from the persp help file was previewed on the X11 driver:

```
> i <-
interp(ozone.xy$x,ozone.xy$y,ozone.median)
> # set NA's to zero
> i$z <- ifelse(is.na(i$z),0,i$z)
> persp(i$z/200)        # plot it
> title(main="Median Ozone Concentrations in
the U.S. Northeast")
```

Figure 2 shows the graph, as it looked on my screen. When I clicked the "Print Graph" button, I got the PostScript output in Figure 3, which uses PostScript fonts, line drawing, etc. The S graphics language is extensible. If you have some odd-ball plotter or other display, you can write your own driver for it (though you would likely lose the "click-to-print" feature). Several drivers are available from the S mail list server.

Finally, StatSci has added some significant value in visual analysis of multivariate data. Two functions that only work on X11 or similar devices are brush and spin. Brush takes a matrix of multivariate data and displays a matrix of all possible 2D scatter plots. It can optionally have histograms and a 3D spinning plot. Once the data display is up on your screen, you can use the mouse to select datapoints (or names), and the corresponding data is immediately highlighted in all the scatterplots, histograms, etc. It is very impressive! Spin is used to rotate multivariate data so that the 3D interrelationships between the variables can be seen and analysed. Unfortunately, these two graphics devices do not currently support hardcopy under X11. And anyway, they are interactive graphics, so they have to be seen to be appreciated.

Overall Evaluation

The S-PLUS package measures up to expectations. It is reliable, robust, self-documenting and comprehensive, yet extensible.

The documentation is comprehensive: You get a copy of the Becker, *et al*, textbook, which is a good introduction both to interactive exploratory analysis and to S. The original S package was "self-documenting" in the same sense as UNIX and `emacs`. Its documentation takes the form of "help files" that are remarkably similar to the traditional UNIX man-page format. These are reprinted in Appendix 1 of the textbook. With S-PLUS, you also get two binders, one full of the updated help files to supplant the Appendix of help files printed in the Becker textbook. The other binder provides some introductory material and examples written by StatSci, information on the X11 interface and on some contributed statistical routines. A final note gives you the address of an email archive server for S routines. The archive is run by the Statistics Department at Carnegie-Mellon University for

more

S-PLUS, Release 2.3

System Requirements: 8 MB of memory; Sun-3, SunOS 3.2X or higher; SPARC, SunOS 4.X

Pricing: Single user, commercial – $2,800; substantial academic, non-profit and quantity discounts are available.

For availability on other UNIX platforms, contact the vendor.

Statistical Sciences Inc.
1700 Westlake Ave. N., Ste. 500
Seattle, WA 98109
Circle 171

the S community, not by StatSci. StatSci's inclusion of documentation on it is a sign that StatSci is in touch with the rest of the S community, and that the company's not afraid of having users extend the software using code contributed by others.

Competition?

Of course, anyone who reads computer magazines knows there is no end of excellent statistical analysis packages available. Two commercial offerings regularly advertised in UNIX magazines are SAS and PV-Wave.

SAS from SAS Institute Inc., Cary, NC, is a mainframe statistical analysis system that has been ported to MS-DOS PCs and to some UNIX platforms. SAS includes most of the classical statistical routines as well as reasonable graphics. One surprise for people in the UNIX community is that you cannot buy SAS. You can only rent it. That is, you are required to pay a yearly fee in order to continue to use the software. With this in mind, SAS is substantially more expensive than S-PLUS.

PV-Wave Point-&-Click, from Precision Visuals Inc., Boulder, CO, is a less programmatic, more pre-packaged package for visual data analysis that might be easier to get started with. Effective May 1, 1991, PV-Wave Point-&-Click runs under SunView (i.e., no NeWS or X11 support) on Sun-3 and Sun-4/SPARC, and costs $4,500, roughly double the cost of S-PLUS. It supports PostScript, HPGL, PCL and several other output formats. ⊸

Ian Darwin has been using UNIX since 1979. He has developed publishing software for SoftQuad Inc., and has taught UNIX and C for the U of T (that's Toronto) Department of Computer Science and for Learning Tree International. Ian wrote the O'Reilly book, *Checking C Programs with Lint*, and can sometimes be reached at `ian@sq.com` or `uunet!sq!ian`.

S-PLUS Release 3

Release 3 of S-PLUS from Statistical Sciences Inc. is scheduled to become available as this issue of *SunExpert* goes to the post office. The following information, obtained from 3.0 beta documentation provided by the vendor, is included to show the extent of work that has been done to the product since Release 2.3. Also, a new textbook entitled *Statistical Modelling in S* is being produced by the S developers at AT&T, and will be included with the new version.

Release 3.0 incorporates new graphics features. You can have several graphics devices active at one time, instead of just one. There is a generic function to copy a plot from an active device to a hardcopy device. Other graphics highlights include full support for Open Look and Motif, similar to the generic X11 device in 2.3. Generic X11 is still available for those who don't care. Open Look and Motif allow dynamic colormap changes (editing of colors). The persp function has been extended and the contour function can save coordinates of contours for later plotting. The legend function lets you scale the legend. StatSci has also embedded a window-based data editor, a spreadsheet-like window for X11 (Open Look and Motif shortly). This release edits (fixed-sized) vectors, matrices and a new thing called a Data Frame object (see below). It also supports arrow key use, tabbing and entry/changing of values. In subsequent releases, StatSci will be handling other object types (lists, multidimensional arrays), and have more capabilities, such as adding/deleting rows/columns, etc. The new version also features a SAS interface. You can now import data from SAS via ASCII files.

New statistical and mathematical functions (these terms will make sense if you are a statistician; if not, they don't count) appear in Release 3.0. The binomial, hypergeometric and Wilcoxon rank sum distributions now have density, cumulative probability, quantile and random-number generators.

A variety of classical statistical inference functions are added. These include binomial, Person's chi-square, correlation, Fisher's exact test for count data, Friedman and Kruskal-Wallis rank sum tests, Pearson's, Mantel-Haenszel and McNemar's chi-square tests for count data, Proportions test, Student's t-tests, F-teset to compare two variances, Wilcoxon rank sum and signed rank tests.

Also added is a wide range of new statistical modeling capabilities drawn from AT&T's latest release of S, including enhancements to ANOVA, General Linear and Additive Models, Loess modeling, Tree-based models and non-linear models.

S-PLUS now supports some features of object-oriented programming, including classes, inheritance and methods; these are described in the additional textbook mentioned above.

S-PLUS also sports a new object type, Data Frame, which is like a matrix but with each column having an arbitrary type, meant for use with new statistical modeling features. Other enhancements include a paginator for large data objects (default is the *less* paginator), a way to dump and restore your S data between different machines, changes to IEEE arithmetic, minor changes to .C and .FORTRAN and numerous bug fixes.