A joint newsletter of the Statistical Computing & Statistical Graphics Sections of the American Statistical Association.

August 1993 Vol.4 No.2

Statistical COMPUTING & GRAPHICS

A WORD FROM OUR CHAIRS

Statistical Graphics

In this issue, we'd like to share an exciting new service with all of you: the video lending library of the Statistical Graphics Section.

Work on the library grew out of the video theater that the Section has been successfully running at the annual meetings. The theater was started by Lorraine Denby at the 1988 meetings in New Orleans and has been carried out annually ever since. Forrest Young founded a video committee of the graphics section, and Sally Morton has taken over that work.

Sally Morton has organized the Statistical Graphics Video Library

Each year many people who attend the annual meetings get a chance to see the video theater—perhaps you have been in the audience and have experienced the wide variety of statistical graphics presented. If you have never seen it or if you want to see what's new this year, be sure to look for it all day Wednesday, August 11, in San Francisco at the Joint Meetings.

One of the results of the theater was a flurry of requests to obtain one or more of the tapes, for study, or to show to colleagues or students. In response to those requests, Sally Morton has organized the Statistical Graphics Video Library. It is currently composed of the material that has appeared in the video show, but over time will pick up new contributions. The library also serves as an archive where the material will be preserved in a central and accessible place. (Even the authors can lose track of material over the years, as I'm sure many of you know.) The library contains 1/2" VHS video tapes and, for selected tapes, an abstract, related publications and technical reports—even comments from previous viewers.

CONTINUED ON PAGE 7

FEATURE ARTICLE

So, You Want to Make A Video!

William F. Eddy
Kensuke Shirakawa *Carnegie Mellon University*

Introduction

The basic reason for making a computer-based video recording is to create a (semi-)permanent record of a dynamic graphical display. The video recording can be studied at leisure, displayed at a seminar or a meeting, or sent to friends.

There are two fundamental approaches:

- one can make a continuous recording of what takes place on a monitor (a real-time recording);
- one can prepare an animation by recording one frame at a time (an animation or surreal-time recording).

Most of you will make continuous recordings rather than frame-at-a-time recordings simply because you have neither the need nor the equipment for making a frame-at-a-time recording.

In either case there are several basic pieces of equipment that you will need:

- a source of video;
- a video recorder; and
- a video monitor.

By a source of video, we mean some piece (or pieces) of equipment which produce a video signal compatible with the video recorder and monitor.

Before we get into a detailed discussion of the basic pieces of equipment it will be useful to have some minimal understanding of the video signal and how it differs from, say, the signal that drives your computer monitor.

CONTINUED ON PAGE 4

EDITORIAL

Surprisingly (at least for us) it is time for another newsletter. We again have a bumper issue with a new feature and great columns. With just a bit of cooperation from the postal service, this issue should be in your hands before the August Joint Meetings. We again encourage you to use the newsletter to communicate with the membership of the two largest sections of the ASA. Our deadline for the remaining issue of 1993 is the last day of October.

In this issue we have one very provocative letter to the editor. Our reader has suggested several challenges for the newsletter and the profession in general. In future newsletters we hope to provide some guidance in improving electronic communications amongst statisticians and between statisticians and their colleagues and collaborators. If you have any solutions to these problems, or suggestions for other issues which are important to the profession, please write to us.

The section chairmen did receive another, less flattering, letter expressing disappointment about the theoretical discussions of topics from related fields such as linear algebra and pattern recognition. That reader expressed the strong preference for articles helpful for a practicing statistician about statistical graphics, software reviews and notices. We too, would like more articles like that, but we need people to write them. Please volunteer! All the same we will not shy away from more theoretical articles that we believe will be of interest to the members of the sections.

In this issue our chairs present quite different columns. Rick Becker and Sally Morton announce the Graphics Video Library as a service to Graphics Section members and the statistical community in general. Sandy Weisberg muses on both the quality of some simple statistical analyses performed in non-traditional ways and the obfuscating power of a "pretty" document. (We hope our Newsletter is pretty **and** useful.)

Our feature article describes some methods for producing a computer generated video. The article talks about the equipment needed to create a video, sources of video input, and problems with creating images tuned to the U.S. video standard. Next issue we will have a followup feature concentrating on the differences between real-time and frame-by-frame video production.

Our regular columns continue. This issue we welcome Dan Carr as a columnist. Dan's column, *Topics in Scientific Visualization* will build upon his previous feature article. We also welcome Phil Spector as a regular columnist with *Unix Computing*.

This month we also give the definitive (though not necessarily final) answer to sending e-mail to the U.S. President Clinton. We have reprinted a short selection of press releases showing that the White House finally has an internet address.

One of the editors (JLR) just returned from Eastern Europe, and faced the challenge of logging into his computer from Lithuania. Though a daunting task—name services on the local machine did not recognize his internet address—the mission was accomplished by first logging into a borrowed account in Oslo, Norway, where the usual telnet command immediately produced his familiar login screen. In seconds, communication from behind the former iron curtain could proceed without any interference. E-mail and internet connections are creating dramatic opportunities for closing the communication gap which existed between the east and west for decades. The efficiency of this new mode of communication should help to bypass and eliminate the frustration caused by bureaucratic delays in earlier modes of communication.

Submissions should be sent by e-mail to either of the editors. If you can prepare your article in TEXor LATEXthat will make our lives just a little easier. Otherwise plain old ASCII format is fine.

We look forward to your letters and comments.

James L. Rosenberger
Editor Statistical Computing Section
JLR@stat.psu.edu

Mike Meyer
Editor, Statistical Graphics Section
mikem@stat.cmu.edu

LETTERS TO THE EDITORS

We are publishing the following interesting letter without the direct permission of the author. We hope you find the letter as thought provoking as we did. If you are interested in the ideas outlined in the letter, feel free to contact us or any of the columnists. We particularly encourage follow-up letters. We have changed a few words to protect the identity of the author. MMM and JLR, eds.

Now that most people are, or will be, on the internet, and many departmental computing facilities have the capability of setting up public files for perusal by ftp, I think it would be a good idea to think about how to set up, organize, and use ftp areas, and perhaps think

of some common conventions to make them easier to use. They have the potential for being a **very** powerful means of communication. After all, I can browse many ftp areas all over the world.

It would be a good idea to think about how to set up, organize, and use ftp areas, and perhaps think of some common conventions to make them easier to use.

I didn't want to write a letter to the editors of the Computing and Graphics Newsletter, I wanted to persuade YOU to write an article about this!! *But we are publishing the letter anyway, Eds.* Let me just list some topics that might be covered, and a few ideas.

1. Common naming of the ftp area. Our department's ftp 'virtual machine' (or whatever you call it) is `ftp.stat.fred.edu`. I am told that the reason for this is to allow the aliasing of `ftp.stat.fred.edu` to whatever machine happens to be handy. On the other hand, I notice that many departments just use their usual 'machine' name—this makes it easy to remember. If you know someone's e-mail address, then you know the ftp address for their ftp area. Perhaps you could explain all this. Most ftp logins are as 'anonymous'. I would strongly urge other logins to be avoided, unless you seriously want to discourage browsers.
2. Organization of files and directories in the public area. After logging in by ftp and doing `ls` you usually see a list of directories, including `pub`, and it seems that pub is commonly accepted as where the searching or browsing user wants to go. Some ftp pub directories are models of mnemonic file and directory names and others are filled with nothing but non-informative names like `ox.qprz` or other stuff which looks like junk, and you get turned off. Others have long lists of technical report files listed by some 8 digit TR code which is completely non-informative. If you don't know which TR you want, or, if like me, you can't remember and type in the 8 digit code you want without a mistake, you get very frustrated. Let me suggest a few obvious names for sub-directories of pub: papers, announcements, software, admissions information. The approach taken by our department with respect to papers and software might be useful for large departments. Under `pub` there are several directories labeled by the names of faculty members, i.e., `pub/wilma`. Any regular faculty member can create such a directory if they so choose. (There are also some other directories with non-informative names!) In `pub/wilma`, Wilma gets to put up whatever junk she wants (subject to whatever space limitations the system manager might impose). I have entered several compressed postscript files of manuscripts with the following naming convention: If I am an author, then the name of the file is `keyword.ps.Z`. If I am not an author, or, if the file is more appropriately identified by someone else's name, for example a student or visitor, the name of the file is `person.keyword.ps.Z`. It's usually not all that hard to choose a keyword which will clue the browser in to what they might want to look for. There is also a file `person.keyword.Z` in `pub/wilma` which happens to be a very large compressed shar file containing several bundled `.ps` files. Since it might not be evident what to do with such a file after uncompressing it, there is also an ASCII file called `person.keyword.README`, which will appear in the file listing just ahead of `person.keyword.Z`, and which contains instructions for handling `person.keyword.Z`. I think this or a similar convention would be handy if it is not otherwise obvious what to do with the file of interest.

   I also put up an ascii file called `bkinfo` which has some information concerning a book I wrote, and another ASCII files with the self explanatory names such as `stat840.annct`. These files give information about courses I am, or have, taught. I put the 840 announcement up for the convenience of students in other departments who might be interested—I was surprised to get an e-mail from England expressing interest in obtaining further information about the course. Let me suggest that people offering advanced graduate level or non-standard courses consider putting up the course descriptions in their ftp areas—we can learn something from what other people are teaching! Book announcements would also be of interest.
3. Saving paper. We have the ghostview/ghostscript combintation up on our system. This wonderful code allows fairly easy viewing of postscript files at an X-terminal. I think that ghostview (which was written by Tim Theisen at Madison, tim@cs.wisc.edu) and ghostscript are available through the Free Software Foundation. Perhaps you know of this or other public software which makes life easier for browsers in cyberspace, that

you could tell the readers about.

4. Information for novice system managers. Perhaps people just setting up their internet connections need some information on how to get started, which you could provide (?)

...best regards!

Making a Video . . .

CONTINUED FROM PAGE 1

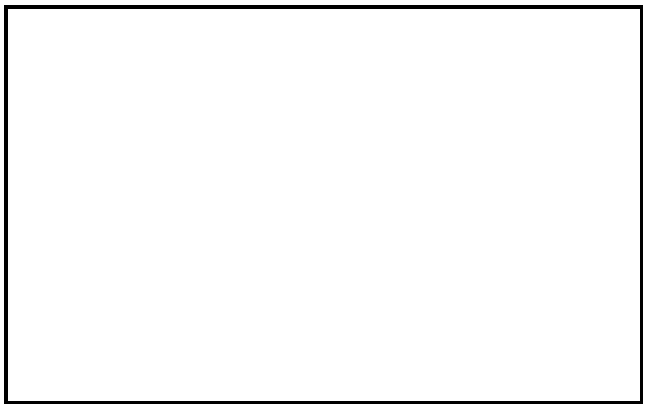
The NTSC Signal

Spatial and Temporal Resolution

In 1941, the National Television Standards Committee, NTSC, established a standard for black-and-white broadcast television in the United States. The standard was modified in 1953 to incorporate color and the revised standard remains in use today throughout North America and Japan. Most of the rest of the world uses the PAL video standard; the two standards are incompatible. Here we will only discuss the NTSC standard but roughly equivalent ideas apply to the PAL standard.

The spatial and temporal resolution of a television monitor is typically much lower than the resolution of a workstation monitor.

The spatial and temporal resolution of a television monitor is typically much lower than the resolution of a workstation monitor. This is not surprising given that the NTSC specification was determined in 1941. The signal specified by the NTSC consists of 525 horizontal scan lines repeated 29.97 times per second. Thus, the horizontal frequency is $525 \times 29.97 = 15,734.25$; that is, the rate at which lines are scanned is 15.73425KHz. In order to increase the apparent scan rate, NTSC scan lines are *interlaced*. The 525 lines are divided into two *fields* each having 262.5 lines. If the lines on the screen were numbered consecutively from top to bottom, the odd-numbered lines would be in one field and the even-numbered lines in the other, *but* the lines are numbered as they appear in the signal, *not* as they appear on the screen. In the signal, the first twenty lines in each field contain information which is not part of the visible picture; the first visible (half-)line is the second half of line 283 and the last visible (half-)line is the first half of line 263. Thus the ordering of line numbers on the screen from top to bottom (omitting the first half-line and the last half-line) is 21, 284, 22, 285, . . . 261, 524, 262, 525. Therefore, each pair of consecutive lines is refreshed about 60 times per second, giving the illusion of an increased scan rate.



The interlacing of visible scan lines in the NTSC standard as it appears on the screen

For comparison, *our* computer monitor has 1024 horizontal scan lines repeated 72 times per second and the scan lines are not interlaced. Our computer monitor has 1280 pixels (picture elements) on each scan line. Of course, the actual signal driving the monitor is analog and so we should really talk about the bandwidth of the signal rather than the number of pixels. The bandwidth for our computer monitor is 135MHz. This number should be compared to 72 (frames per second) $\times$ 1024 (lines per frame) $\times$ 1280 (pixels per line) = 94.371840 MHz. The difference between the two numbers is due to horizontal and vertical synchronization and other non-visible portions of the signal.

When the NTSC signal was defined it was entirely analog so there is no notion of pixel. However, we can estimate the number of "pixels" by solving the following equation for the unknown number P (pixels per line):

$$\begin{aligned} 30(\text{frames per second}) &\times 525(\text{lines per frame}) \times P \\ &= 4.5\text{MHz}. \end{aligned}$$

The number 4.5MHz is the bandwidth of the original black and white NTSC signal. Assuming there is the same fraction (94/135) of visible signal (and ignoring the fact that not all of the 525 lines are visible) this yields an estimated value of $P = 201$ pixels per line.

The bottom line is simply this. NTSC television has an effective spatial resolution on the visible screen that is of the order of 400 $\times$ 200 pixels. The number 400 (rather than 525 - 40 = 485) is explained more fully in Section 5. The number 200 we just estimated above. For comparison, remember the 1024 $\times$ 1280 of our workstation monitor.

Color Encoding

NTSC video cannot provide the variety of colors of a workstation monitor nor can it provide them at the high spatial resolution of a workstation monitor. This section explains the details and will be easier to read if you have some basic knowledge of color systems. See, for example, "An Introduction to Color Systems" in Volume 1 Number 1 of this newsletter.

Assuming now that you understand the RGB additive color system it is fairly straightforward to specify the NTSC color system. We begin by thinking of a three-wire version of the NTSC signal and then describe the one-wire signal. In the descriptions below we are thinking of the video source as beginning with RGB inputs; it is simplest to imagine the RGB values in the unit cube $[0,1]^3$.

NTSC Component Video - YC_RC_B The three-wire version is often called component video; it is also known as YC_RC_B or YUV (also called Betacam). (Actually, there are some minor differences between YC_RC_B and YUV but they don't matter for our purposes here). The brightness (also known as the *luminance*) is derived from the (NTSC) RGB primaries by the equation:

$$Y = .299R + .587G + .114B.$$

In 1931 the Commission International de l'Eclairage (CIE) determined three additive primary colors and named them X, Y, and Z. The CIE Y primary is a mixture of the spectral colors which closely mirrors the response of the typical human eye to those spectral colors at constant intensity. The letter Y used in the equation above is not precisely the same as the letter Y used by the CIE; the distinction is that the NTSC Y implements *gamma-correction* through a power transformation. Gamma correction accounts for the non-linear intensity change generated by a linear voltage change in the television tube. The luminance is the portion of the video signal which is used by a black-and-white television set.

All of the color can then be encoded in two additional signals. These two are collectively known as the *chrominance* and are given by the equations:

$$C_R = R - Y;$$

$$C_B = B - Y.$$

Thus component video is a simple linear transformation of the RGB representation (although it is not an orthogonal transformation). Notice that whenever $R = G = B$ the two chrominance signals will be zero; the color will be gray.

NTSC Composite Video - YIQ For transmission and recording purposes, the three signals need to be combined and compressed into a single (or one-wire) signal. The need for compression is an historical artifact generated by the Committee in its 1953 revision to incorporate color; they wished to make the new signal acceptable to old television sets and there wasn't enough bandwidth available in the broadcast spectrum to do it without compression. There was a total of 6MHz of available bandwidth for each broadcast channel. The compressed signal is called NTSC *composite* video and it is the standard signal used by consumer television sets.

The composite signal is obtained by first converting the color (by a further linear transformation) into two signals known as I and Q. The YIQ form of color encoding can be derived from the YC_RC_B or directly from the RGB. We give the transformations directly from RGB. The *Intermodulation* (Orange - Blue) axis is given in terms of RGB by:

$$I = .596R - .275G - .321B.$$

The *Quadrature* (Purple - Green) axis is given in terms of RGB by:

$$Q = .212R - .523G + .311B.$$

Note that I and Q are both orthogonal to the vector

$$L = .333R + .333G + .333B$$

but are not orthogonal to Y or to each other. They form an oblique coordinate system in a plane orthogonal to L.

After extensive tests of the color vision of many individuals the bandwidth of the the two color signals necessary for the best pictures (given the total bandwidth constraint of 6MHz) was established at approximately 1.2MHz for I and 0.4MHz for Q. Note that this implies an effective spatial resolution of 20 (!) pixels per line for the Q portion of the signal.

By a complex process which doesn't really concern us here the I and Q signals are modulated onto a sine wave with frequency 3.58MHz below the luminance portion of the signal.

The bottom line is simply this. NTSC video does not have the variety of colors not the high spatial resolution of a workstation monitor.

Video Sources

There are two basic methods for generating video from a computer. The cheapest and most common approach is to get an NTSC *video frame buffer.* An NTSC video frame buffer is a video card similar to the one which drives your computer monitor but its output signal is

NTSC composite or NTSC RGB component video. The price of a frame buffer varies from about $500 to $5000, mainly depending on the quality of the signal produced and the price of the computer that will use it. If the frame buffer output is NTSC RGB component video you may also need a *color encoder* (a device for performing the transformations described above). Some professional-level recorders have built-in color encoders so that a separate encoder becomes unnecessary.

The alternative to a frame buffer is a *scan converter*, sometimes called a down converter. A scan converter takes any sort of video as input (for example, the output to your computer monitor) and samples it in (space and) time to reconstruct a new signal that matches the NTSC standard. The price of a scan converter ranges from $10,000 to $25,000 depending mainly on the quality of the signal produced and the number of features you get in addition to scan conversion.

A third alternative (which we don't particularly recommend unless you are desperate) is to use a video camera to record the image on the screen. Direct recording with a camera often leads to artifacts in the picture because of the differing scan rates between the workstation monitor and the camera and/or the lack of synchronization between them.

Video Recorders

Assuming now that you have a reasonable source of video, we turn to the problem of recording the video signal. There are two basic media: tape and disk, and various subcategories within these two.

Video Tape Recorders

Video tape recorders (VTR) are available in different recording formats, with each format differing in picture quality, price, popularity, and ease of use. The following is an incomplete list of formats which are currently popular and available on the market, in order of increasing picture quality (and price). (Note: some of the formats have improved versions listed in parentheses. These are fully compatible with the original format **except** for a very noticeable loss in picture quality when playing improved version tapes in original version VTRs.)

- VHS (S-VHS)
  VHS stands for *Video Home System*. VHS is recorded on 1/2 inch tape and is the standard format used in consumer VTRs. S-VHS has a larger bandwidth available for recording the chrominance portion (the combination of the I and Q) of the video signal, leading to improved picture quality. Consumer VTRs do not typically allow frame-by-frame editing, so a VHS VTR is useful only for continuous recording of computer output and for dubbing from other formats. There are professional S-VHS VTRs which can do frame-by-frame recording (Price: $5000).
- 8mm (Hi8)
  This is a newer format available for consumer VTRs, using a much smaller cassette with 8mm (Surprise!) tape. This format is not yet in wide spread use except in camcorders. Some 8mm VTRs allow frame-by-frame editing.
- Umatic (Umatic-SP)
  This professional format is the one of the oldest. It uses 3/4" tapes. Some Umatic VTRs allow frame-by-frame editing.
- Betacam (Betacam-SP)
  One of the main features of this format is that the Y, C_R and C_B signals are recorded on three separate tracks. Composite NTSC signals are separated into the three signals by the VTR for recording.

Video Disk Recorders

A video disk recorder (VDR) is an alternative to a VTR for recording video. There are two kinds of VDR. First, there are magnetic (e.g., Abekas A60) VDRs which are rewritable (or write many). In other words, it is possible to erase and record over a previously recorded portion of the disk the same as one might rerecord on a videotape. Second, there are laser VDR (e.g. Sony LVR-5000A) which are write once. Once a signal is recorded on the disk, it can not be erased.

Compared to most VTRs, VDRs can have higher picture quality by allowing a larger bandwidth for the recorded signal and are especially useful for frame-by-frame editing purposes; because access to frames is random (unlike the sequential access to frames on tape), any frame on the disk can be accessed very quickly. The advantages of the VDRs are offset by the high price of the machines (between $10,000 and $70,000) and the high price of the disks (for the write-once versions).

Video Monitors

In this section, we will describe the basic workings of an NTSC video monitor.

An NTSC video monitor is simply a cathode ray tube (CRT). Inside the monitor, there are three cathodes, one for each of the primary colors: Red, Green and Blue (RGB). The cathodes (or electron guns) inside the tube produce an electron beam which excite the red, green or blue phosphors on the face of the tube (the screen). The input voltage to each cathode (which is controlled by

the information in the television signal) determines the intensity of the electron flux and hence the brightness of the phosphor. The three cathodes scan horizontally (left to right from viewer's point of view) across the screen under control of a magnetic field at the $\sim$15KHz rate described earlier. When the beams reach the right-hand side of the screen the streams of electrons are turned off while the guns return to the left hand side of the screen. This time period is called horizontal retrace and and the corresponding portion of the signal is called horizontal blanking. Then another line is scanned in the same way. There are a total of 262.5 lines contained in each of the two fields (remember the signal is interlaced) and then the streams are turned off while the guns return to the top of the screen. (This time period is called the vertical retrace and the corresponding portion of the signal is called vertical blanking.)

The maximum number of horizontal lines visible on the screen is limited to 485; the remainder are used to transmit other information. Consumer television monitors are adjusted so that not even this many can be seen. Early in the history of television it was believed that people would accept television more readily if they didn't see the corners of the picture. Consequently, most sets are adjusted to *overscan*; typically, about 400 lines are visible. In contrast, professional monitors are adjusted to *underscan* (so one can see the corners of the picture).

The standard input for most home television sets is the composite NTSC signal. Professional monitors can also accept some of the various types of component signals.

Partially supported by ONR Contract N00014-91-J-1024 and Bureau of the Census under Joint Statistical Agreement 91-25. Department of Statistics, Carnegie Mellon University, Pittsburgh, PA 15213-3890.

Bill Eddy
Kensuke Shirakawa
Carnegie Mellon University
bill@stat.cmu.edu
ken@stat.cmu.edu

FROM OUR CHAIRS . . .

CONTINUED FROM PAGE 1

The video tapes are great for educational institutions: several people have used them as the basis for a "stat day" production, for statistical computing classes, or even for a video lunch complete with popcorn.

Lending policies for the library are still in flux, but they are likely to include

- loans lasting up to 3 weeks
- a limit of 3 videos at one time (with exceptions for statistics day presentations where more videos will be available for a shorter period)
- The borrower will be responsible for including postage payment along with the request for videos.
- Borrowers who are not members of the graphics section will also be asked to pay a lending fee of $10; borrowing from the video library is one of the many services members receive in return for their section dues.

Authors of the video tapes that are distributed as part of the lending library have given permission for the library to lend the tapes but have NOT authorized extra copying. After viewing a tape, should you wish to acquire a personal copy, you should contact the author.

The video tapes are great for educational institutions, especially statistical computing classes

To find out more about the video library, contact Sally Morton via e-mail at `Sally_Morton@rand.org`, or call (310) 393-0411 x7360. Also, if you have your own video tape that you think should be part of the video library, please contact Sally.

The following list describes the initial library materials, giving title, author and institution, date, and running time (minutes:seconds) of the video.

Multidimensional Scaling. J.B. Kruskal, AT&T Bell Laboratories, (1964). 1:00

Image of a Thunderstorm. Anne Freeny and John Gabbe, AT&T Bell Laboratories, (1966). 3:00

Multidimensional Scaling. J.B. Kruskal, AT&T Bell Laboratories, (1970). 1:00

Real-time Rotation. Jih-Jie Chang, AT&T Bell Laboratories, (1970). 2:00

Prim-9. J.W. Tukey, J.H. Friedman and M.A. Fisherkeller, Stanford Linear Accelerator, (1973). 26:00

Ozone in the Northeast. Richard A. Becker, William S. Cleveland, Beat Kleiner, and Jack L. Warner, AT&T Bell Laboratories, (1978). 3:00

Dynamic Displays of Data. Richard A. Becker and Robert McGill, AT&T Bell Laboratories, (1985). 23:00

Brushing a Scatter Plot Matrix. Richard A. Becker and Robert McGill, AT&T Bell Laboratories, (1985). 15:00

Data Analysis Networks in DINDE. R.W. Oldford and S.C. Peters, University of Waterloo, (1986). 24:00

Brushing and Rotation on an Iris. Richard A. Becker, William S. Cleveland and Gerald Weil, AT&T Bell Laboratories, (1987). 20:00

Dataviewer: A Program for Looking at Data in Several Dimensions. Andreas Buja and Paul Tukey, Bell Communication Research, (1987). 21:00

Automatic Tracing of Ice Floes on Satellite Images. Jeff Banfield, University of Washington, (1987). 8:00

Antelope: Data Analysis with Object Oriented Programming and Constraints. John McDonald, University of Washington, (1987). 11:00

Use of the Grand Tour in Remote Sensing. John A. McDonald and Steve Willis, University of Washington, (1987). 15:00

Odds Plots: Finding Associations between Views of a Data Set. Werner Stuetzle, University of Washington, (1987). 21:00

Plot3d: Data Animation on a Sun Workstation. Paul Tukey and Vonn Marsch, Bell Communication Research, (1987). 6:00

Graphical Programming. G.D. DesVignes and R.W. Oldford, University of Waterloo, (1988). 26:00

Higher Hierarchical Views of Statistical Objects. C. Hurley and R.W. Oldford, University of Waterloo, (1988). 20:00

Treetools. Marilyn Becker, Linda A. Clark, and Daryl Pregibon AT&T Bell Laboratories, (1989). 24:00

Visualizing Multivariate Structure with VISUALS/Pxpl. Forrest W. Young & Penny Rheingans, University of North Carolina at Chapel Hill (1990). 9:25

Thin Plate Splines and the Analysis of Biological Shapes. Fred L. Bookstein and William Jaynes, University of Michigan Medical Center (1990). 20:58

Focusing and Linking as Paradigms for the Visualization of High-Dimensional Data. Andreas Buja, Bell Communications Research, and John McDonald, John Michalak, Werner Stuetzle and Steve Willis, University of Washington. (1991) 28:00

Xgobi: Dynamic Graphics for Data Analysis. Deborah F. Swayne, Dianne Cook, and Andreas Buja, Bell Communications Research. (1991) 12:23

Visualizing Panel Data. Martin Koschat and Deborah Swayne, Bellcore (1992). 20:00

A Simple Dynamic Graphical Diagnostic Method for Almost Any Model. George S. Easton, University of Chicago. (1991) 31:00

Smoother's Workbench. Lise Manchester, Dalhousie University (1992). 14:00

Tokyo Data Map. Planned by the Tokyo Government, Produced by Dentsu Inc. and Dentsu Prox Inc. (1992) 10:25

Edge Information at Landmarks in Medical Images. Fred L. Bookstein and William D. K. Green, University of Michigan Medical Center (1993). 26:15

Grand Tour and Projection Pursuit. Dianne Cook, Andreas Buja, Javier Cabrera, and Deborah F. Swayne, Bell Communications Research. (1993) 19:00

Rick Becker
Chair, Statistical Graphics Section
rab@research.att.com
Sally Morton
RAND
Sally_Morton@rand.org

Statistical Computing

For the applied statistician, the phrase *statistical computing* seems to be almost redundant: how can we do statistics without also doing computing? It is no wonder, then, that the Statistical Computing Section of ASA has always been very large. The section has provided a forum both for the creators of statistical software and for the users of that software. This is a tradition I hope will continue into the future.

As little as twenty years ago, statistical computing was a smaller, well defined, area. One could then realistically try to understand the differences and similarities of all available software packages; after all, very few were available. For standard problems, one could expect standard results; apart from computational problems, a number labeled as a mean or as a standard deviation always meant the same thing. Even today with the bewildering array of statistical packages we can still expect a certain core of information to be given correctly and uniformly for standard elementary problems.

Consider, for example, comparing the means of two populations based on a random sample from each population. For this problem, what might we expect of a computer program? It should produce sample sizes, averages and standard deviations for each sample; it should tell something about extremes in the data by providing minima and maxima or better yet boxplots or a similar graph of each sample, and it should provide reasonable summary statistics like the two sample t-statistic for the comparison, along with an appropriate p-value, given the usual normality and common variance assumptions. I would be surprised to find a statistical package that fails to do the two sample problem reasonably well.

Unfortunately, statistical packages are not the only computer programs used to do statistics. I recently had the following consulting problem. A company makes a very successful medical device that is used throughout

the world. Of roughly a half million uses, the device fails for unknown reasons that could possibly be attributed to the device about one time in ten thousand uses. A particular doctor had a patient for whom the device seemed to be failing, and so the doctor replaced the device. He then took the allegedly "bad" device, and one unused and therefore probably "good" device and did a number of physical measurements. He found that on one measure of surface roughness, the value obtained for the "bad" device was twice the value for the "good" device. He therefore concluded that surface roughness caused the failure. Were this claim true, the company would certainly need to change its manufacturing process to eliminate devices with high surface roughness.

The first response of the company was to measure more parts. Finding allegedly "bad" devices was difficult, but five had been returned to the company after being removed for symptoms like those the doctor had seen. Five other devices thought to be "good" were also measured. The measurements were done at the same lab the doctor used, by the same technician, using the same protocol. I have often found that the statistical methodology in software that supports high-tech measuring instruments is of very low quality, so the numbers given to the engineer for evaluation are often not good summaries of the large number of measurements taken. This was certainly the case here. Putting this important issue aside, the company had two samples of size five, along with the two measurements obtained by the doctor.

I received the data neatly displayed in a spreadsheet, with a line marked "Mean" for the group means and "Standard deviation" for the group standard deviations. These were computed by an engineer using the functions built into the spreadsheet program. The spreadsheet program did not have a two sample t-test as a built-in function, and consequently the engineer did not compute it. He perhaps would have gotten the formula wrong if he tried to write a function to compute it.

Even today with the bewildering array of statistical packages we can still expect a certain core of information to be given correctly and uniformly for standard elementary problems.

In fact, *all* the summary statistics displayed in the spreadsheet were somehow wrong. The calculation of the means was correct, but the numbers were given in a fixed format, and significant digits were lost. The standard deviations were computed by using a divisor of n, not $(n-1)$, and so these were biased and too small. The summaries did not tell a useful story. We expect a number labeled "Mean" or "Standard deviation" to be just what they say they are. However, this may not be so when the end user makes up the labels, does the computing, and supplies the format for the output.

After recomputing everything using a standard statistical package, it was evident that the standard deviation between devices was roughly the same as the difference between the "good" and "bad" device observed by the doctor. Of course the doctor's claim could still be correct, but given the large standard deviation and the very low rate of failure, the claim has no substantiation from the data.

The point of this story is that statistical computing is done in all kinds of programs, in the software that runs equipment, in data base systems, and most particularly in spreadsheet programs. People knowledgeable both about spreadsheets and statistics can use a spreadsheet to good advantage to do statistical computing; they are certainly useful in viewing and organizing data. Whereas the novice user of nearly any statistical package is almost certain to get a good summary of a two sample problem, the novice user of a spreadsheet is almost equally certain to get a bad summary.

Sanford Weisberg
Chair, Statistical Computing Section
sandy@umnstat.stat.umn.edu

DEPARTMENTAL COMPUTING

User Education

A most bedeviling and daunting task in any computing environment is attempting user education. There is so much to learn. The users are so variable—some are knowledgeable and need just the slightest hints about how to proceed while others show a resistance that would lead you to believe they must be pretending to be so uninformed. User perception of the process can be unrealistic—many seem to want to know how to do just one thing, without any thinking of the context of their request.

In this article, I'll try to portray useful approaches to user education. I assume at all times that people want to learn and that they deserve to learn. Training helps others do their jobs. I operate in a chronically understaffed environment, so many of the suggestions are intended to be time-savers. We want independent users who use computers efficiently to get their work done. We don't

want to burn out the system people getting there.

Who gets trained?

In any change of system environment (UNIX from DOS, Windows from DOS, PCs from mainframes), everyone will need some training. The most advanced users may only need to be pointed at appropriate documents. Others will need extensive hand-holding. Choosing the right level of training is important in designing a user education program. Down-time for retraining staff is typically unacceptable, so this should be done in parallel, while both systems are available. Care should be taken to make the staff comfortable with any big change.

When installing major new versions of software, the users of that software may need a refresher course. This may take the form of a seminar or a shorter, less formal approach to group training. For smaller changes (patches, minor releases), an e-mail notification may be all that is required.

At the the University of Florida Stat Department, we have over 100 major pieces of software installed on our system, including major statistical systems and custom statistical software. We are constantly updating our local software collection. We have noticed that some users can be testy about receiving e-mail regarding such updates. These users don't like receiving "junk" e-mail, that is, e-mail about software they don't use. I see little alternative to sending out such messages. They are easy to delete, and the alternative—trying to target messages to "users" of the software—is a nightmare to manage, and is poorly defined. Who is a user?

Who does the training?

Training must come in two forms—generic and specific. Generic training covers basic principles, procedures and techniques that apply across a broad range of environments and can be done by computing center staff, vendors, and textbooks. Specific training is required in addition to differentiate what is said in generic presentations from the conditions that exist in your local system. For example, a SAS manual might describe system options, but a specific local guide will describe the local defaults. The separation of generic and specific training can confuse users, but is a natural result of today's complex software environments. Integrated training, where all training is delivered from a single source familiar with local conditions is typically beyond the resources of most departments.

Both insiders and outsiders will do training—outside training is used for generic material and inside training is done by faculty, staff and students and supplies local details. Such a split will often need to be explained to users who expect to get all their information from one source.

What gets taught?

In the good old days, we tried to teach almost everything. Those days are long gone. Users need to understand that education is an on-going process and that software can be used at different levels. A good example is a graphical e-mail program. Most graphical e-mail programs can be used with only a few minutes of "training." This assumes, of course, that the user is familiar with basic mail and file concepts. Learning a new mailer is trivial if the concepts are in place. Learning a first mailer is a little tougher, and learning a mailer without some understanding of mail and files is difficult. With five minutes of show and tell, a user with mail and file background should be able to send and receive messages. More training can come later (forwarding, filing, inserting messages in replies). Editors and word processors can be introduced in a similar manner. Within a few minutes users should learn how to create and save documents and make simple insertions and deletions, assuming, of course, that they are familiar with basic editing and file concepts.

Major language-based systems such as S, SPSS and SAS require a different approach. The generic/specific distinction is present, but for these systems the emphasis is on the generic. Users must understand that these systems reward careful study—immediate productivity gains may occur and more study will result in more gains. Unlike push-button tools, statistical analysis requires attention to detail and careful thought. Statistical software is no different. Professional tools require an investment of time to acquire skills. The goal should be self-sufficiency, a fluency in the system that does not require constant reference to manuals and books. Users should have a basic understanding of procedural computing before attempting to learn about these systems.

We use university organizations for generic training in SAS, computing concepts, and programming languages. Our specific training focuses on the interfaces of these software systems with our local environment.

When?

Continuous training is a requirement. We are never done learning, and this is especially true when it comes to computing. With systems changing, new features being added, new languages and opportunities developing, we must stay focused on the task of learning about our tools. When people tell me they are too busy to learn about their tools, I am reminded of the carpenter who was too busy pounding nails with a hammer to

learn about a power nailer. Some learning results in dramatic short-term productivity increases. Other learning is more foundational, requiring considerable investment for an eventual return. We need to clearly differentiate this range when explaining user training.

Where?

The best situation is a room with many (10–15) systems, where an instructor can lead a group of users through tasks, and where users can try things with "over the shoulder" assistance. The next best solution is a projection system where the instructor shows what can be done using a live interactive demonstration. The University of Florida is networking our classrooms to increase the number of sites where such demonstrations can be performed. The traditional method of instruction using a board or overheads is a poor substitute for either of the above. Consider trying to learn how to play the piano by having someone talk about it and write descriptions on a blackboard. Using a computer is like that (especially using a mouse!) You have to do it to learn it.

Reduce, Reuse, Recycle

Training materials are costly to prepare. Before preparing any materials, give some thought to their lifecycle.

Reduce. The first question should be are they necessary at all? Can we get by with materials from other sources? Do we need to produce new materials? I find that new materials are most often needed for interface issues—describing how to move data from one system to another, and describing how commands operate at our site that are different from the way the commands might be described in vendor documentation. I try hard not to duplicate information in the vendor materials and to refer users to vendor documentation whenever possible.

One of the most frequent requests from users is for one page summaries of programs. Obviously a system like S can not be reduced to a single page, but a two-sided introduction to S-mode with references to S documentation is possible and useful. There's a trap here, of course. You want to provide brief introductory documentation so as not to scare off the new user with a 702 page S book, but at the same time, you do not want to cripple the user by implying that the one page document is sufficient for any particular set of tasks. I have seen many users hobbled with the notion that the one page document was intended to be sufficient. The user must know where to get additional and related information.

Reuse. Try to get mileage out of the work you have done to prepare materials. I have placed a PostScript version of my department's computing manual in the anonymous ftp area of `ftp.stat.ufl.edu`. We print this manual and make it available in our computing labs. We also put the manual at a local copy center where our users can purchase bound copies. In addition, our one-page summaries will eventually appear in our gopher server (`stat.ufl.edu`).

Last call: At the ASA meeting in San Francisco this August, I will be hosting a roundtable luncheon on the topic of Departmental Computing. Please join us!

Michael Conlon
University of Florida
`mconlon@stat.ufl.edu`

UNIX COMPUTING

Introducing the Author

We have invited Phil Spector, the Applications Manager of the Statistical Computing Facility in the Department of Statistics at the University of California, Berkeley to write a regular column on the use of the UNIX operating system. His facility supports a network of Sun Microsystems Sparc Stations running UNIX for use by the faculty, students and staff of the Statistics Department, as well as other users from around the campus and the community. Along with teaching a graduate course in Statistical Computing, he provides consulting and troubleshooting for a variety of statistical and other software packages, as well as general UNIX problems. He received a Ph.D. in Statistics from Texas A&M University, and spent four years as a Senior Research Statistician at SAS Institute before moving to Berkeley.

Beginnings

What I'd like to do in this first article is just talk very loosely about what UNIX is. If you're a long time user, most of this will be very familiar, but I think it's worthwhile to lay down a foundation before getting into the specifics of UNIX. In future articles, I'll try to present tricks and useful techniques for using various UNIX commands, based on my experience using UNIX to solve statistical problems.

What is UNIX?

UNIX is a computer operating system, that is, a collection of programs which allows users to manipulate information and manage resources on a computer. (UNIX is not an acronym for anything, but rather a little joke by the original developers. They were working on a system called "MULTICS", which stood for Multiplexed Information and Computing System, and were not very happy with its performance. So they created a two user

system which they referred to as "UNICS", i.e. Uniplexed Information and Computer System. Somewhere along the road "UNICS" became "UNIX", and the rest, as they say, is history.) UNIX is different from other operating systems since it was designed specifically to run on a variety of computers. Most operating systems will run only on a particular brand of hardware, so if you made a commitment to an operating system, you were making a commitment to one type of hardware, from one particular vendor. Since the source code for UNIX is written in C, instead of some computer-specific machine language, UNIX can run on any computer with a C compiler, which nowadays means any computer. If you learn UNIX, then, you'll have a tool which you can use on just about any computer you encounter in the course of your work. And if you also choose to learn the C programming language, you can extend the capabilities of the UNIX operating system in any direction you choose.

UNIX is different from other operating systems since it was designed specifically to run on a variety of computers.

The basic core routines which comprise UNIX are known as the kernel. In the early days of UNIX, users weren't considered to be serious unless they "hacked the kernel", that is, made their own personal changes to the heart of the operating system. (Of course, UNIX was the only operating system which would allow this kind of thing, since the others consisted of tightly guarded proprietary machine language.) However the vast majority of UNIX users now never have to deal with the kernel, and, in fact, might not even know of its existence. This is because of the existence of programs known as shells or command interpreters. These programs provide an easier way to access the power of the UNIX operating system than dealing with the kernel directly. Shells are discussed in more detail in a later section of this article, and will be the topic of future articles as well.

Files in UNIX

Along with portability, one of the major advances of the UNIX operating system is in the way it organizes and accesses files. Files are the basic units of information on a computer and can include text, graphics, programs, sounds, or just about anything else. Under UNIX, files are named in a hierarchical fashion, using a slash (`/`) to separate the different levels of organization. Each level in the hierarchy is called a directory. For example, many installations store locally developed information in a directory called `/usr/local`; within that directory might be sub-directories for source code (`src`), executable commands (`bin`), help files (`help`), and other information. (The term `bin` is often used for directories containing commands because they are usually binary files, as opposed to text files.) Therefore the full name of a local help directory would be `/usr/local/help`; a particular helpfile, say for text processing, might be found in `/usr/local/help/textprocess`.

Files are the basic units of information on a computer and can include text, graphics, programs, sounds, or just about anything else.

Another innovation in the UNIX file system is that all external devices, like disk drives, tapes, CD/ROMs, etc. are all treated like files. For example, a magnetic tape drive might be called `/dev/rmt0`. In this way, dealing with a wide variety of peripherals is simplified because programs which access them do so in the same fashion as they would access a regular file.

Shells in UNIX

There are a wide variety of shells available within UNIX. Some of the names which you may hear when people talk about shells include the Bourne shell, the C-shell and the Korn shell. The core functions which the shells provide are pretty similar, but each offers some special features of its own. One feature common to all the shells eliminates the need to specify or even know the full name of a file, which includes all the pieces of the hierarchy described above. (This full specification is often referred to as a file's pathname.) When working in UNIX you are said to be "in" a particular directory, known as your current or working directory, and it is the shell that keeps track of this and lets you change your working directory. When you refer to a filename which doesn't begin with a slash (`/`), it is interpreted relative to your current directory. So even if your files are stored in the directory `/disk0/users/files/me`, you could refer to a file called `my.file` (whose complete pathname is `/disk0/users/files/me/my.file`) as simply `my.file`, if your current directory is set to `/disk0/users/files/me`. By default, when you log in, your working directory is set to what is known as your home directory, where your personal files are stored. You can create as many levels of additional directories as you wish in your home directory.

One of the basic functions of the shell is to execute the commands you type. This means that the shell needs to know about where executable commands are found on your system (that is, in which directories commands may be found). As with most of the other shell services, this is generally transparent to the user. You just type

a command like `ls` (to list the names of files in a directory) or `date`, to tell you the time and date, and the shell searches the path to find the command.

Graphical User Interfaces

Recently, with the introduction of computer monitors capable of displaying a wide variety of text and graphics, graphical user interfaces (GUIs) have become increasingly popular. Instead of limiting your communication with the computer to just typing in commands in response to a prompt, these graphical interfaces allow you to perform many tasks by manipulating icons (a small object on the computer screen that represents a program or file on the computer), or making choices from menus. Two of the more common interfaces are Open Look and Motif. It's important to remember that these interfaces under UNIX serve the same function as the shell; they allow you to communicate more easily with the kernel, which is the heart of the operating system. In keeping with the UNIX tradition, the kernel itself doesn't need to be modified in order for a graphical interface to be used. Furthermore, one of the basic pieces of all the UNIX GUIs is a command shell, that is a window on the screen which allows you type UNIX commands into the shell of your choice, so you don't need to forego the command line to use a GUI.

Phil Spector
UC at Berkeley
spector@stat.Berkeley.EDU

BITS FROM THE PITS

Statistical Computing and Graphics in Science and Industry

This column features statistical computing and statistical graphics activities in science and industry. Your comments and suggestions for future columns are requested. Please send comments, inquiries, and suggestions to the editors or to Albert M. Liebetrau, Analytic Sciences Department, Battelle-Northwest, MS K7-34, P.O. Box 999, Richland, WA 99352, AM_Liebetrau@pnlg.pnl.gov, 509-375-2694.

Uncertainty Analysis for Computationally-Demanding System Codes

Part two of a two-part series

Introduction

In the first part of this series (Vol. 4, No. 1), I outlined a strategy for uncertainty analysis for computer codes whose computing demands render conventional methods impractical. The fundamental idea is to develop a computationally tractable analogue, which we call a performance assessment (PA) code, to the computationally demanding code of interest, and then use this analogue whenever possible. There are three basic steps in the implementation of this strategy:

- design: selecting an initial set of model inputs or realizations
- approximation: approximating the response surface of the underlying code
- updating: determining the locations or inputs for additional runs.

One can view this strategy as an attempt to develop a two-tiered system model which includes a PA model that "sits atop" an underlying model, the latter typically being built up from several detailed component models. Most computations for uncertainty analysis are done at the PA level using conventional methods. When it is necessary to drop down to the lower level, results of these new runs are systematically used to improve approximations at the upper level.

In this article, we describe some of our experiences in trying to develop suitable approximations. We also comment, in turn, on tools currently available for design and updating.

Design

The efficient selection of an initial set of model runs is a sampling design issue. Many efficient sampling methods exist. One of the most widely-used is Latin Hypercube Sampling (LHS), which has proved very effective for sensitivity and uncertainty analysis; see Liebetrau (1991) for a brief summary, and McKay et al. (1979) or Iman and Conover (1980) for details. Recent work by Owen (1992) gives not only a central limit theorem for LHS, but also some insight into why LHS works so well in practice. Except as noted, we have used LHS to select the initial runs for the examples discussed below.

Approximation

To date, we have used two approximation methods, TREES and MARS, to estimate the response surfaces of the output of two codes, EPIC and AREST. The two codes provide a good test because they are quite dissimilar.

The most basic estimator we used is the TREES estimator, which is a simple piecewise constant function;

see Breiman et al. (1984). For the results presented below, we used the Splus implementation of this estimator (Chambers and Hastie, 1992).

We also used the MARS estimator, which can be regarded as an extension of the TREES estimator. The MARS estimator is formed as a linear combination of a set of basis functions; see Friedman (1991a, 1991b) for details. We used the implementation of the MARS estimator available via `ftp` from StatLib.

The EPIC and The AREST Codes

The EPIC code was originally developed to estimate the long-term effects of erosion on agricultural productivity. The code was recently modified to study the consequences of potential climate change scenarios. One output of the code is annual yields for selected cultivars. The code has an extensive list of inputs. It contains components for simulating erosion, plant growth, and related processes as they would occur over hundreds of years. It also contains modules that represent hydrology, weather, nutrients, soil temperature, tillage operations and a plethora of variables for describing farming practices. Moreover, the EPIC code can be used to model the effects on productivity, via photosynthesis and evapotranspiration, of changes in concentrations of atmospheric CO^2.

The EPIC code contains a stochastic weather generator that generates daily realizations of weather variables from monthly summaries. Consequently, EPIC code output is inherently stochastic. On the other hand, the yield surface is a reasonably linear function over the ranges of key input variables.

As might be expected, the large number of zeros presented a challenge for both methods of approximation.

The AREST code models the containment and release performance of waste containers in a geologic repository for the storage of high- level radioactive wastes. Outputs of the code are the concentration and cumulative release of selected radionuclides, expressed as a function of time, at some specified distance from the waste container. Inputs describe the waste container, the repository and geologic properties of the repository environment. It is relevant to note that over ranges of interest, the output of the AREST code is a highly nonlinear function of its inputs.

The AREST code can be run in either stochastic or deterministic mode; the latter was used to generate the results described herein. A significant feature of the output is that at early times, all simulated concentrations are zero. As time increases, the percentage of non-zero concentrations increases; however, the percentage of zero releases remains large for all time periods of interest. As might be expected, the large number of zeros presented a challenge for both methods of approximation.

The EPIC code is described in detail by Sharpley and Williams (1990). Additional information about the AREST code is found in Liebetrau et al. (1987).

Results

In this study, we used the EPIC code to predict corn yield for a "typical" midwestern farm under a variety of weather scenarios, controlled primarily by temperature and precipitation variables. The MARS code was first applied to 20 EPIC runs obtained by LHS, and then to 676 EPIC runs obtained from a factorial design. The input variables and their respective ranges were the same in both cases. As one might expect, the first fit was better than the second (R^2 = 0.96 vs. R^2 = 0.78). Because of the intrinsic randomness of EPIC code outputs, the MARS approximator did not interpolate the output: indeed, it would be undesirable to do so. The variability in MARS predictions was not significantly greater than that in EPIC outputs.

We next fit a MARS surface to the output of the AREST code, which can be run in a deterministic mode. The selected output variable was release rate at 106 years. We simulated 5000 observations of release rate from the model, of which 568 runs produced a release rate of zero at 106 years. (Run times for the AREST code are small enough that approximation is not required for most simulations.) All 5,000 observations were used to fit a MARS model containing third- and lower-order interactions. The R^2 for this fit was 0.84, which suggests that the approximation was not completely satisfactory. An examination of results revealed that the zero releases presented a major problem.

To deal with the difficulty of many zeros, we applied the TREES procedure to the two codes. When applied to the 676 EPIC code runs, the resulting R^2 was 0.80, which supports the earlier observation that the main source of lack-of-fit is the stochastic nature of the EPIC code output. The TREES predictor was next applied to 4,000 of the 5,000 release rates generated by the AREST code. The procedure was forced to interpolate the data. The remaining 1,000 outputs were then predicted from the fitted TREES estimator. Results of this exercise revealed that although this estimator captured much of the AREST code structure, it did not predict the zeros very well. The problem of predicting the zeros was mitigated somewhat when we tried to predict cumulative releases

instead of release rates. However, in this code and in many others, a boundary limitation is a salient feature and a suitable approximation algorithm must be able to deal with such constraints.

In summary, we were unable to completely approximate either model with either method. We did, however, capture much of the behavior of these models and obtained useful results. Moreover, the experience we gained will help to guide future work. A more extensive summary of results is available (Liebetrau et al. 1993).

Updating

The ideal strategy for developing PA codes should include an efficient updating algorithm. The algorithm should use an initial set of runs to assess the need for additional runs of the underlying code, and then facilitate the selection of optimal (most informative) input locations for these runs. The algorithm should indicate where the current approximation is unsatisfactory, and it should be adaptive. One approach to the sequential design of computer experiments is presented by Sacks and colleagues (1989a, 1989b), who treat the model output as the realization of a stochastic process. However, their methods rely heavily on the correlation structure among model inputs and are oriented more toward design than updating. We know of no updating methods that use model output directly. Additional work is required to develop widely applicable and truly efficient updating algorithms.

Methods for reducing computation will always be required if we hope to use models for uncertainty analyses

Summary

New computer models will continue to tax existing computing resources. Therefore, methods for reducing the computational requirements will always be required, especially if we hope to use the models for uncertainty analysis. The results described here represent one modest attempt to deal with this problem, but it is clear that much work remains.

References

Breiman, L. J.H. Friedman, R.A. Olshen, and C.J. Stone. (1984). *Classification and Regression Trees*. Wadsworth Statistics/Probability Series. Wadsworth International Group, Belmont, California.

Chambers, J.M. and T.J. Hastie, editors. (1992). *Statistical Models in S.* Wadsworth and Brooks/Cole, Pacific Grove, California.

Friedman, J.H. (1991a). "Multivariate Adaptive Regression Splines. (with discussion)" *Annals of Statistics*, **19**(1), 1–141.

Friedman, J.H. (1991b). "Adaptive Spline Networks." Technical Report No. 107, Laboratory for Computational Statistics, Department of Statistics, Stanford University, Stanford, California.

Iman, R.L., and W.J. Conover. (1980). "Small Sample Sensitivity Analysis Techniques for Computer Models, With an Application to Risk Assessment. (with discussion)" *Communications in Statistics, Part A–Theory and Methods*, **9**, 1749–1874.

Liebetrau, A.M. (1991). "Statistical Computing and Graphics in Science and Industry." *Statistical Computing and Statistical Graphics Newsletter*, **2**(2), 28–29.

Liebetrau, A.M., M.J. Apted, D.W. Engel, M.K. Altenhofen, D.M. Strachan, C.R. Reid, C.F. Windisch, R.L.Erickson, and K.I. Johnson. (1987). "The Analytical Repository Source-Term (AREST) Model: Description and Documentation." PNL-6346, Battelle Pacific Northwest Laboratory, Richland Washington.

Liebetrau, A. M., P. D. Whitney, D. W. Engel and C. A. LoPresti. (1993). "Computational Analogues to Complex Computer-Based Codes." Unpublished Technical Report.

McKay, M.D., W.J. Conover and R.J. Beckman. (1979). "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code." *Technometrics*, **21**(2), 239–245.

Owen, A.B. (1992). "A Central Limit Theorem for Latin Hypercube Sampling." *Journal of the Royal Statistical Society, Series B*, **54**(2), 541–555.

Sacks, J., S.B Schiller and W.J. Welch. (1989). "Designs for Computer Experiments." *Technometrics*, **31**(1), 41–47.

Sacks, J., W.J. Welch, T.J. Mitchell and H.P. Wynn. (1989). "Design and Analysis of Computer Experiments (with comments)." *Statistical Science*, **4**(4), 409–435.

Sharpley, A.N. and J.R. Williams, editors. (1990). *EPIC-Erosion/Productivity Impact Calculator.* Two Volumes. Technical Bulletin No. 1768, U.S. Department of Agriculture. Available from the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.

Albert M. Liebetrau
Battelle Pacific Northwest Laboratories
AM_Liebetrau@pnlg.pnl.gov

Plot Production Issues and Details

Background on the Column

In the last issue this column was initiated with an article about the production of stereo plots. We will continue to focus on plot production issues and techniques. Some topics will arise from my *exploratory data analysis* and *scientific visualization* classes. Hopefully the readership will suggest additional topics for development from these two general areas.

Scientific Visualization

Developments in graphics-workstation technology have had a major impact on the field of scientific visualization and should be exploited in the production of statistical graphics. Thus graphics-workstation topics will be fair game for this column. For example, last issue I promised a column on alpha-blending as part of the natural extension to translucent-stereo density plots. While that column is postponed pending scheduling of newsletter color production issues, the topic exemplifies my interest in utilizing advanced technology to facilitate understanding through visual representation. One goal of this column is to encourage cross fertilization between the areas of statistical graphics and scientific visualization.

Statistical graphics involves statistical modeling and the visual representation of central structure, residuals and uncertainty. At least in one interpretation statistical graphics are visual aids for the human endeavor of statistical visualization. Concern about uncertainty as understood through distributional summaries helps distinguish statistical visualization from other areas of scientific visualization.

> ***Statistical graphics involves statistical modeling and the visual representation of central structure, residuals and uncertainty.***

Statistical models define what is meant by central structure and residuals and provide a basis for obtaining distributional summaries. Statistical modeling is an art form that is beyond the scope of this column. This column focuses on plot production details that facilitate the representation of statistical modeling results. Statistical modeling alternatives and issues are raised only in passing. However the importance of modeling software often ties plot production to statistical packages despite the attractiveness of other scientific visualization software.

I have chosen to provide implementations in Splus since it provides a wealth of modeling tools and a programming language sufficient for the production of most static graphics. The Splus code for the examples in the column will be available by anonymous `ftp` from `galaxy.gmu.edu` in the directory `/submissions/eda`. Making the code available allows the procedures to be described in outline form. Hopefully the outline and comments will be of interest to devoted users of other packages. In some cases it may be relatively easy to adapt the provided code to other environments. Please note that the code has been developed in problem solving mode rather than as a polished product. Gentle notes about improvements will be appreciated.

While plot production details are the primary emphasis, the column will occasionally touch on educational materials useful in exploratory data analysis and scientific visualization classes. For example I suspect my list of favorite videos will be of interest. I would like to learn about materials that others are using so I will encourage others to collaborate on education materials columns.

Smoothed Cancer Rates and Hexagon Mosaic Maps

Recently Linda Pickle of the National Center for Health Statistics (NCHS) asked me to contribute a hexagon mosaic map of smoothed colon-cancer mortality rates for use in a NCHS project in evaluating the merits of different map styles. I have asked Linda to provide background about the NCHS project for this column. I then outline the steps followed in producing the hexagon mosaic map and raise some issues that warrant further attention.

Background On Smoothed Mortality Maps

Maps of cancer mortality rates published by the National Cancer Institute in the past (Mason 1975, 1976; Pickle 1987, 1990) turned out to be very successful "visualization tools" for public health researchers. These maps identified cancer "hot spots" and geographic time trends in the U.S. data that had not been noticed before in tabular publications of the mortality statistics. Follow up studies designed to determine the reasons for high rates in particular regions led to such important discoveries as the link between mouth cancer and snuff dipping and lung cancer and exposure to asbestos through shipyard work during World War II.

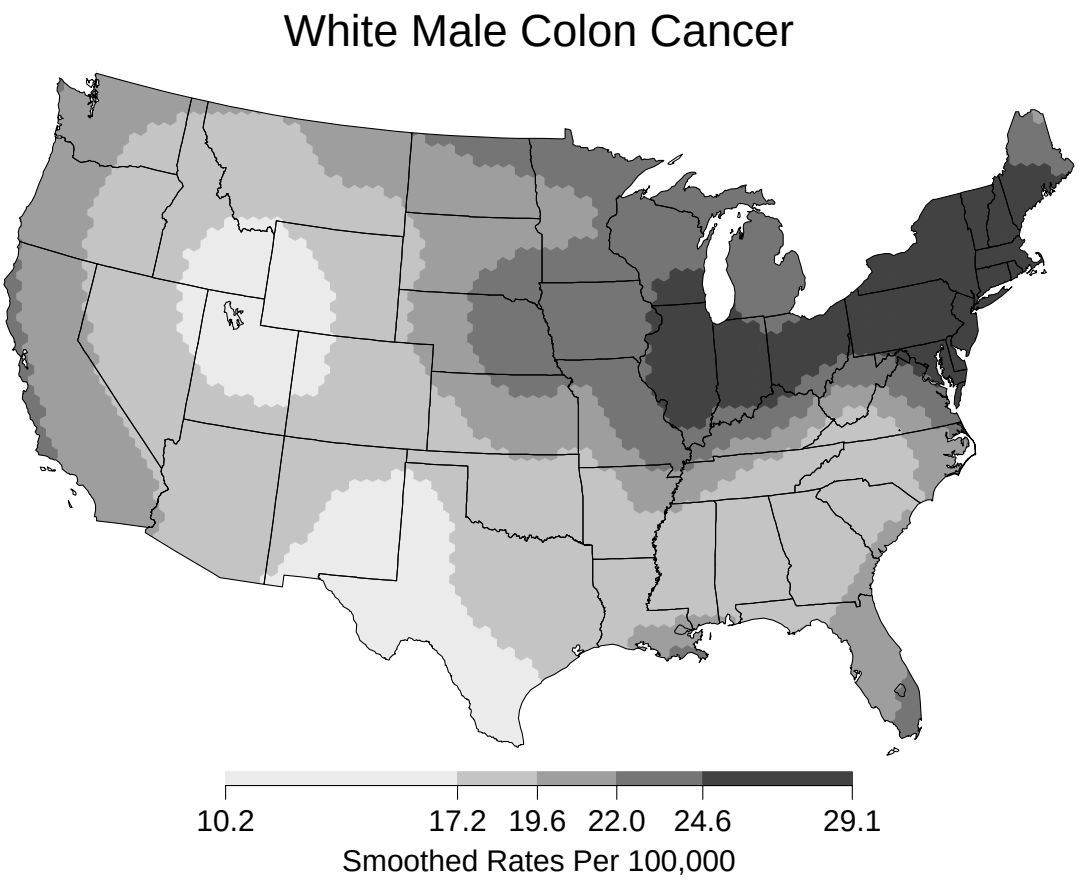


Figure 1

Hexagon Mosaic Map of Smoothed Mortality Rates. The map was produced using Splus, a product of Statistical Sciences, Inc. While Splus comes with a U.S. boundary and Becker and Wilks (1992) have provided map projection and other functions, the particular boundary files used here were obtained from Atlas Pro and are used with the permission of Strategic Mapping, Inc.

Maps of cancer mortality rates published by the National Cancer Institute in the past turned out to be very successful "visualization tools" for public health researchers.

The second atlas series showed the mortality data by time as well as place. Although regional differences in mortality rates seemed to be diminishing over time for many types of cancer, new "hot spots" appeared during the 1970s for several of the major cancers. Because of the success of these atlases, NCHS is planning a mortality atlas of leading causes of death, not limited to cancer. NCHS is the federal agency responsible for collecting and publishing information from all U.S. death certificates.

The problem for earlier atlas designers was how to produce a reasonable looking map overall. For example, it took the combined effort of two federal agencies to produce the hardcopy output for Mason's atlases. Now that mapping software is widely available and a standard desktop PC is powerful enough to process the large mortality data files, the problem we face today is how to choose from the many mapping options available. NCHS has funded several cognitive studies to date to test how geographic patterns are perceived when the underlying data are presented using various map styles. Because NCHS must map the entire U.S. at a small area level (at least 500 geographic units), some maps styles, for example those that use large area symbols such as framed-rectangles, are not feasible.

The goal is to design a map that will present the geographic patterns in the underlying data with the least amount of distortion or perceptual bias. The map should also be able to answer several types of questions typ-

ically asked by the viewer: (1) what are the general geographic patterns of rates (e.g. where are the rates high?); and (2) approximately how high are the rates in a certain area? It may be necessary to use a different map style to answer each of these questions.

The goal is to design a map that will present the geographic patterns in the underlying data with the least amount of distortion or perceptual bias.

NCHS has been experimenting with different methods of smoothing the mortality data. That is, if at least some of the random variation in the data can be removed, the broad geographic patterns should be more apparent in the map. Assigning the mortality rate to a single point within an area allows the data to be smoothed and represented by symbols at lattice points thus ignoring the original (usually irrelevant) political boundaries. Because the task is to map events among people a more appealing approach is to assign the rate to the population centroid rather than the geographic centroid, but populations centroids were not available when NCHS created test files for distribution.

For the latest experiment, NCHS provided directly age-adjusted mortality rates for colorectal cancer among white men during 1980-89. The basic geographic units were Health Service Areas (HSAs). The 802 HSAs are groups of counties defined according to where residents obtained their hospital care (Makuc 1991). NCHS is preparing a poster for this year's ASA annual meeting that will summarize the reaction of study participants to a number of map styles, including several smoothed maps.

Production of a Hexagon Mosaic Map

Figure 1 shows a hexagon mosaic map of smoothed colon-cancer mortality rates. The basic hexagon mosaic map is a choropleth map composed of hexagons. Carr, Olsen, and White (1992) used the hexagon mosaic map to represent sulfate deposition and trends and discuss possible merits of this type of map relative to closely related square mosaic maps and pseudo- color contour maps. Ultimately the representational form needs to be evaluated in perceptual studies and the NCHS study provides an all too rare opportunity for evaluation of map styles.

The production and interpretation of maps cannot be divorced from the application. Two differences between the sulfate map in the original application and the current mortality rate maps warrant mention. In terms of production, the sulfate observations were essentially point data. The supplied mortality rates were area-based estimates. Assigning an area-based rate to a point fails to capture finer scale variation within the area when such exits. The detail obtainable is limited to county level statistics for cause of death information. Aggregation to HSAs attempts to reduces variance by pooling information from "similar" counties. Step three below was not necessary in producing a smoothed sulfate deposition map because the transition from area-based estimates to point estimates was not needed.

In terms of interpretation, a smooth is better accepted when there is reason to believe that the underlying surface is smooth and something other than a flat plane. The sulfate deposition process, when viewed over a long period of time, makes a smooth deposition surface over the U.S. seem quite plausible. In fact a strong west to east gradient can be anticipated based on recorded SO^2 emissions and wind currents. The connection between spatial location and mortality rates is less obvious to those outside the epidemiology community. However spatial location often serves as a surrogate for variables that are "causally" related to mortality rates. In fact most major cancer types exhibit spatial clustering. While some spatial clustering might be happenstance, some patterns such as those in Figure 1 have remained relatively stable over three decades. The spatial deposition surface tends to be interpreted as a summary result while the mortality-rate surface is used primarily to generate hypotheses for further investigation.

The following annotated steps indicate decision points and tasks involved in the production of a hexagon mosaic map of smoothed mortality rates.

Step 1 is to pick a map projection. In the original application, sulfate deposition is conveniently described as an amount per unit area. This strongly favors use of an area preserving map projection. In the cancer mortality rate context, the choice of map projection is not so clear. I chose to stay with an Albers equal area conic projection and this choice is consistent with NCHS practice.

Step 2 is to select a grid resolution and generate a set of hexagon centers so the hexagons will cover the map of the continental United States. Selecting a grid resolution has received little discussion in the literature. Since the hexagons in a hexagon mosaic map have direct visual impact, the choice of resolution is likely more important than in applications in which the grid is hidden by subsequent contouring.

Defining a rectangle that encloses the U.S. and generating a hexagon grid that covers the rectangle is straight forward. The problem is eliminating hexagon center

points (centroids) for hexagons that fall completely outside the U.S. boundary. Since the U.S. boundary is a polygon each candidate centroid is tested using a point in polygon algorithm (see Littlefield 1984). Centroids inside the U.S. boundary are accepted. Each centroid outside the U.S. boundary is further tested and accepted if any of its surrounding hexagon edges intersects any of the U.S. polygon edges (see Sproelder and Ulling 1990 for an intersecting segments algorithm) .

Step 3 is to obtain point data for use in traditional model software. The supplied values were HSA centroids in Albers coordinates. Algorithms often available in GIS packages can produce centroids from the polygon boundaries of each area. As indicated above, obtaining centroids based on populations appears preferable for this type of data.

Step 4 is to model the point data. Common modeling approaches for point data include kriging, splines, and polynomial regression. The choice here was to model the mortality rates using local regression (loess). Cleveland, Grosse and Shyu (1990) discuss the modeling options available. The particular options used for Figure 1 include local quadratic modeling, the Euclidean distance option for the independent variables, inverse variance weights and direct modeling of data. The independent variables were the Albers coordinates representing the HSA centroids. Using the actual surface of the earth interpoint distances may be technically more desirable but the potential for improvement seems small compared to the complications involved. The inverse variance rate approximation was based on a Poisson death rate model and a rough estimate of the white male population size. Having population data available facilitates more sophisticated modeling. Cleveland and Devlin (1988) discuss the modeling procedure to assist in picking a smoothing fraction for the local modeling. Figure 1 represents the second plot produced and oversmoothes the rates.

Step 5 is to obtain estimates at the hexagon grid locations obtained in Step 2. In Splus the predict function combines the modeling results with new values of the independent variables to obtain estimates. (An alternative is to obtain the average surface value for each hexagon.) Those experienced in the use of 2-D smoothers are typically concerned about the behavior of the smoother near the boundaries of the modeled data (and near sharp discontinuities). In this application some of the hexagon centroids near the edges of the map lie outside the convex hull of the data so even "extrapolation" is involved. The irregularity of the map border also results in smoothing between regions separated by water. The appropriate measure of distance in such cases depends on the application. While values near the fringes of the map should be interpreted cautiously, the extrapolation problem is a mild one numerically since no point on the map boundary lies very far from some HSA centroid and mild philosophically since the modeled data includes people that live near the boundaries.

Splus supports the notion of a rectangular grid. A more general notion would provide for other lattices bounded by polygons. Algorithms could be developed to correspond to these lattices.

Step 6 is to determine the class intervals. My preference would be to determine the class intervals by the percent of people affected. For example a class boundary might be determined so the hexagons in the highest rate class includes five percent of the population. The boundaries in Figure 1 were supplied to be consistent with the HSA quintile boundaries in other plots.

Step 7 is to pick the color for each class and plot filled hexagons using the colors. Plotting hexagons is straight forward using the polygon plotting function.

Step 8 is optional and represents low population regions. Some analysts find it distressing to see values represented for deserts, lakes and rugged terrain where few people live. One possibility would be to overplot the hexagon cells that have populations below a certain threshold to indicate low populations regions. For example a reduced-size background-colored hexagon might be used. This step requires hexagon binning of detailed population data and was not used in Figure 1.

Step 9 is to clip the hexagons back to the U.S. boundary. Clipping involves overplotting undesired regions with polygons in background color. A clipping trick that seems to work (but may not be formally supported) in postscript is to construct a more complex polygon appending a closed surrounding rectangle to a closed U.S. boundary polygon. At least on one system, plotting this complex polygon in background color overplotted the region between the U.S. boundary and the bounding rectangle. A safer approach is to construct two simple clipping polygons by splitting up the U.S. boundary polygon into two pieces and adding a few points.

Note that postscript processing software differs especially in regard to the number of polygon vertices that can be processed. In Splus version 3.1, I had to invoke an option to handle over 2300 vertices in the U.S. boundary. (Another solution is to use a generalized boundary with fewer vertices.) The postscript previewer on my workstation would not handle that many vertices and

stopped. Two different postscript printers had no problems with the postscript file. (I have also observed grey-level texture patterns to vary from device to device.)

Step 10 is to add detail such as state boundaries, islands and lakes. Given the boundaries, this simply involves plotting polygons.

Step 11 adds a legend. The current legend involves plotting polygons, lines and text so was straight forward to produce albeit tedious the first time. Legends can provide additional information but that is a candidate topic for another column.

In summary, the production of a hexagon mosaic map is straight-forward, given boundary files and estimates on a hexagon lattice. Software that provides for filling polygons and plotting text should allow the production of such maps. Care needs to be given to the tasks of modeling the data and obtaining good estimates near the map boundaries. Several map enhancements are possible including marking regions with low populations and adding information to the legend. The hexagon mosaic maps fared well in the NCHS initial evaluation. At the very least the hexagon mosaic map provides one more alternative in the arsenal of tools for representing smoothed data on maps.

Research related to this article was supported by NSF under grant no. DMS-9107188 and EPA by EPA under cooperative agreement no. CR820820-01-0. The article has not been subjected to the review of the EPA and thus does not necessarily reflect the view of the agency, and no official endorsement should be inferred.

References

Becker, R. A. and A. R. Wilks. (1993), "Maps in S." AT&T Bell Laboratories Statistics Technical Report 93-2. Murray Hill, New Jersey

Carr, D. B., A. R. Olsen and D. White. (1992), "Hexagon Mosaic Maps for Display of Univariate and Bivariate Geographical Data." *Cartography and Geographic Information Systems*, **10**(4), 228–236, 271.

Cleveland, W. S. and Devlin, S. J. (1988), "Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting." *Journal of the American Statistical Association*, **83**, 596–610.

Cleveland, W. S., E. Grosse, and W. Shyu. (1990), "Local Regression Models," in *Statistical Models in S*, Eds. J. Chambers and T. Hastie. Wadsworth & Brook/Cole. Pacific Grove CA. pp 309–376.

Littlefield R. J. (1984), "Basic Geometric Algorithms for Graphic Input", In *Computer Graphics '84: Proceedings of the 5th Annual Conference and Exposition of the National Computer Graphic Association, Inc. (Vol 2)*, Fairfax, VA: National Computer Graphics Association, pp. 767-776.

Makuc, D. M., B. Haglund, D. D. Ingram, J. C. Kleinman and J. J. Feldman. (1991), "Health Service Areas for the United States", *Vital Health Statistics*, **2**, 112.

Mason, T. J., F. W. McKay, R. Hoover, W. J. Blot and J. F. Fraumeni, Jr. (1975), *Atlas of Cancer Mortality for U.S. Counties: 1950–1969*. Washington, D.C.: USGPO, DHEW Publ. No.(NIH) 75-780.

Mason, T. J., F. W. McKay, R. Hoover, W. J. Blot and J. F. Fraumeni, Jr. (1976), *Atlas of Cancer Mortality Among U.S. Nonwhites 1950–1969*. Washington, D.C.: USGPO, DHEW Publ. No.(NIH) 76-1204.

Pickle, L. W., T. J. Mason,, N. Howard, R. Hoover, and J. F. Fraumeni, Jr, (1987), *Atlas of U.S. Cancer Mortality Among Whites: 1950–1980*. Washington, D.C.: USGPO, DHHS Publ. No. (NIH) 87-2900.

Pickle, L. W., T. J. Mason,, N. Howard, R. Hoover, and J. F. Fraumeni, Jr, (1990), *Atlas of U.S. Cancer Mortality Among Nonwhites: 1950–1980*. Washington, D.C.: USGPO, DHHS Publ. No. (NIH) 90-1582.

Sproelder, H.J. W. and F. H. Ulling. (1990), "Two-Dimensional Clipping: A Vector-Based Approach", in *Graphics Gems*, Ed. A. Glassner. Academic Press, Inc. New York, pp 121–128.

Daniel B. Carr
George Mason University
dcarr@galaxy.gmu.edu

Linda W. Pickle
National Center for Health Statistics
lwp0@nch09a.em.cdc.gov

NET SNOOPING

Alex and Anonymous FTP

Most internet users are aware of at least one or two sites that offer an anonymous FTP service—that is a repository of information that any user can access via FTP. The service usually works by allowing the user to login as "anonymous" and generally sending their e-mail address as a password. Very few users are aware of all of the different FTP servers available to them. There are several ways of navigating through the wealth of available information.

One common tool for searching through anonymous FTPable material is the `archie` program. Archie is available from many good FTP archives, including my favorite one, `gatekeeper.dec.com`, in the `pub/net/infosys/archie` directory. Once one

has the source to the archie program it is very simple to ask archie (which actually asks one of several archie servers around the world) where to find a particular file. That is very handy, but one has to know what one is looking for.

Another way of navigating through the internet and through the world of anonymous FTP is by using a gopher client. We'll leave the rich world of gopher to another issue.

A third very useful invention is an experimental system call Alex. Alex is a global filesystem that provides users and applications transparent read access to files on anonymous FTP sites. The name Alex comes from the ancient Library of Alexandria. Alexandria gathered information from around the world into one easy to access location. Alex does an analogous thing in a very modern way. Alex was written as an experiment in networking by Vincent Cate `vac@cs.cmu.edu` in the School of Computer Science at Carnegie Mellon.

In order to use Alex you need to be able to use the Network File System (NFS) on your computer. Alex will presumably work on any computer that supports NFS but my only experience is on Unix systems, so that is what I will describe here. If you are willing to ignore the technical details of how Alex works, using it is really simple. The easiest way to describe Alex is via a quick demonstration. I'll use Alex to track down the sources for the Alex system.

A Quick Tour of Alex

I started out by `cd`ing to `/alex` and listed the files there. Here is what I saw:

```
alex> ls
ar      be      co      dk    . . .
arpa    br      com     ec    . . .
at      ca      cs      edu   . . .
au      ch      de      es    . . .
```

The "files" are the names of various top-level domains. That is `edu` contains the names of anonymous FTP sites with machine names like `fred.wilma.edu`. Similarly, `au` contains the hosts in Australia and `ch` those in Switzerland. If I were just exploring I would `cd` to one of those directories and see what was available. However I know that the Alex sources are on some machine at Carnegie Mellon, so I'll try to find that.

```
/alex> cd edu
/alex/edu> ls
acu         fit         ncsu    . . .
acusd       fiu         nd      . . .
alaska      fsu         nevada  . .
albany      fullerton   niu     . . .
andrews     gac         njit    . . .
. . . . . . . . . . .
clu         kent        psu     . . .
cmu         kenyon      purdue  . .
colby       kestrel     reed    . . .
. . . . . . . . . . .
```

There are over 300 sites with names that end in `something.edu`. One of them is cmu, so:

```
/alex/edu/> cd cmu
/alex/edu/cmu> ls
andrew          itc
cc              psy
cs              ri
ece             sei
hss             stat
```

By now the pattern is clear. We can see all of the anonymous FTP sites whose names end in `.cmu.edu`. Continuing a little further we find that the `alex` directory is in `/alex/edu/cmu/cs/sp`. An `ls` in that directory yields

```
/alex/edu/cmu/cs/sp/alex> ls
README              ls-lR
alex                readmes
cs-techreports      src
doc                 tools
links
```

To see that I'm really in the correct place I should look at the README file, e.g.,

```
/alex/edu/cmu/cs/sp/alex> cat README

Alex is a filesystem that lets
users access files in FTP sites
around the world just like they
access local files.  The source
for Alex is in the src
directory.  You can try Alex
out by doing the following as
. . . . . .
```

Thus it is very simple to navigate the world of anonymous FTP just as though there were one (enormous) global filesystem organized just like the regular file system. The only slightly odd (albeit very natural thing) is that one has to read the names of the directories in reverse order, just like e-mail addresses in the UK.

How does it work?

The implementation of Alex is really quite clever. To the user it looks like the `/alex` area is just a file sys-

tem, however that is an illusion. The filesystem is being mimicked by a program. The framework of the filesystem (up to the names of the various FTP servers) is periodically updated by the Alex server, but for most purposes this is static information. When one `cd`s to one of the server directories the Alex server silently opens an FTP connection to the named machine and performs the action you have requested—either listing the files in the directory or fetching a file. Of course Alex tries to be quite smart. It caches as much information as it can, to avoid repeatedly fetching the same file, and it tries to keep FTP connections open for as long as it can. But all of this is hidden from the user, and unless there is a problem, it just looks like one homogeneous file system. Quite a slick idea! In some ways Alex is similar to the Andrew File System (AFS) and the son of AFS, the OSF's Distributed File System (DFS). However, instead of inventing a new protocol (as AFS and DFS do), Alex re-packages anonymous FTP and NFS file servers into a very useful package.

How can I try it?

The full sources to Alex are available via anonymous FTP at `alex.sp.cs.cmu.edu` in the directory `src`.

Mike Meyer
Carnegie Mellon University
`mikem@stat.cmu.edu`

NEWS CLIPPINGS

Election Results

Statistical Graphics Section

David W. Scott, Chair Elect
Rice University

Sallie Keller-McNulty, Program Chair Elect
Kansas State University

Jane F. Gentleman, Section Representative 94-96
Statistics Canada

Sally C. Morton, Section Representative 94-95
RAND

Statistical Computing Section

Mary Ellen Bock, Chair Elect
Purdue University

John A. Rice, Program Chair Elect
University of California, Berkeley

Deborah F. Swayne, Secretary/Treasurer
Bellcore

Karen Kafadar, Publications Liaison Officer 94-96
National Cancer Institute

Ronald A. Thisted, Section Representative 94-96
University of Chicago

JCGS September Contents

The September 1993 issue of the *Journal of Computational and Graphical Statistics* will feature the following articles:

Projection Pursuit Indices Based on Orthonormal Function Expansions by Diane Cook, Andreas Buja, and Javier Cabrera.

Performance of the Gibbs, Hit-and-Run, and Metropolis Samplers by Ming-Hui Chen and Bruce Schmeiser

Empirical Likelihood Confidence Bands in Density Estimation by Peter Hall and Art B. Owen

On Generating Random Intervals and Hyperrectangles by Luc Devroye, Peter Epstein, and Jörg-Rüdiger Sack

Higher Order Asymptotic Corrections Applied in an EM Algorithm for Estimating Educational Proficiencies by Neal Thomas

ASA Poster Competition

Jerry Moreno *Chair of the Fourth Annual American Statistics Poster Competition—1993*

The American Statistics Poster Competition is a joint project of the Section on Statistical Graphics and the Center for Statistical Education. The basic purpose of the competition is to encourage our school children from kindergarten through the twelfth grade to use their creative skills to analyze data graphically. The contest is open to all public or private schools in the United States and Canada.

There are four categories: grades K–3, 4–6, 7–9, and 10–12. Prizes totaling $200 are given in each category. Plaques are given to the schools of the winning entrants. Honorable Mention certificates are awarded as well. In 1990, the first year of the competition, there were 90 entries from across the country. The number of entries grew in the next two years to 150 and 530, but the total this year dropped to 287. The committee will be looking into finding reasons for the decline.

The number of entries in the geographical partitioning of the United States (as defined in the Exploring Data

volume of the Quantitative Literacy series) are given in the following table.

	K–3	4–6	7–9	10–12	Total
Northeast	12	49	9	0	70
Central	8	17	18	4	47
South	3	55	33	28	119
West	0	0	11	40	51
Total	23	121	71	72	287

It should be noted that the counts are a little misleading in that one school in the South submitted 47 entries and one in the West submitted 39. Twenty-eight states were represented (eight in the Northeast, nine in the Central, nine in the South, and two in the West).

The $200 prize winners for 1993 are:

K–3 The Seeds of Life: Comparisons of Plant Growth Raymond Guido, III Saint Lawrence School Huntington, Connecticut Advisor: Naomi Macari

4–6 Hot, Cold, - Drip, Drop Erin Black and Aaron Wolk Schuyler Colfax School Wayne, New Jersey Advisor: Jill Gunderman

7–9 Clean Up Time Jeffrey Green and M. Thomas Marks Rancho San Joaquin Middle School Irvine, California Advisor: Pauline Embree

10–12 Is There Association Between the Length of the Foot and Forearm? Diane Marrapese Magnificat High School Rocky River, Ohio Advisor: Lois Andressakis

Honorable Mention Awards were given to:

K–3 What is the Favorite Planet of Boys and Girls? Kira Greco, Chris Gronkowski, and Gina Sobotka Heim Elementary Williamsville, New York Advisor: Eleanor Yuhasz

K–3 What is the Favorite Holiday? Eric Barber, Lyndsay Hall, Gina Sobotka, Lyndsay Trosterud Heim Elementary Williamsville, New York Advisor: Eleanor Yuhasz

K–3 What is the Most Popular Pet? Eric Barber, Adam Durfee, Kira Greco, Gina Sobotka Heim Elementary Williamsville, New York Advisor: Eleanor Yuhasz

4–6 These States Generated 27,650,000 tons of Trash in 1991 Tyler Tracy and Chad Parlin Cascade Brook School Farmington, Maine Advisor: Cynthia Herrick

4–6 How is the Weather in Israel and Virginia the Same? Yael Or and Katie Hayes Jack Jouett Middle School Charlottesville, Virginia Advisor: Sheila Porter

7–9 The Cost of Raising a Child Sara Alonso and Amy Looney Churchill High School San Antonio, Texas Advisor: Roland Rios

7–9 How Do Average 7th Grade Students Spend Their Allowance? Ryan Winters Jefferson Middle School Albuquerque, New Mexico Advisor: Paul Mitschler

10–12 Birds of a Feather Sheila Ross, Dave Russell, Rachael Davida Whitnall High School Advisor: Gail Burrill

Certification

Richard M. Heiberger *Temple University*

The Board of Directors of the American Statistical Association is seriously considering the issue of certification of consulting statisticians. A Committee on Certification was set up in the Summer of 1992 to recommend to the Board an approach to certifying individuals as statisticians.

The Committee presented a report in April 1993. The full text of the report appears in the July Amstat News, and was circulated by e-mail in May. *(The proposal is also available in StatLib. Eds.)* Letters to the Editor of Amstat News on the certification issue—both for and against certification—also appear in the July Amstat News along with the Committee's response.

Members of the Statistical Computing and Graphics Sections may wish to discuss the importance of issues such as the "consultant's" familiarity with and use of the common statistical software.

An Open Meeting on Certification was held at the March 1993 ENAR meeting and one is scheduled for Tuesday August 10 at 2 PM at the San Francisco Joint Statistical Meetings. We expect several members of the ASA Board of Directors to attend. All conference attendees are invited to this special session. Any implementation of the certification process has strong implications for the perception of statisticians by the general public and for the nature of the ASA itself. It is very important that these issues be identified and thoroughly discussed in advance of any decision.

The Committee on Certification began an e-mail distribution list for discussion of certification in October, 1992. We welcome subscribers—whether for or against the certification of statisticians—to subscribe to this list. If you wish to join the distribution list, send an e-mail message to `asacert@ctrvax.vanderbilt.edu`

Requests for addition to the mailing list are manually processed, so that you might not receive a response for several days. The discussion of the mailing list are archived in StatLib. Send the message `send index from asacert` to see what is available.

Electronic White House

In our last issue we reported on the state of e-mail communications to the White House and the U.S. government in general. There has been dramatic progress in the last few months and there are now direct internet addresses for various members of government—from the president on down. The following material, excerpted from three official press releases, gives some of the details. If anyone is aware of any other countries that allow e-mail access to their leaders, this *Newsletter* would enjoy hearing about it.

Letter from the President and Vice President in announcement of White House electronic mail access

Dear Friends:

Part of our commitment to change is to keep the White House in step with today's changing technology. As we move ahead into the twenty-first century, we must have a government that can show the way and lead by example. Today, we are pleased to announce that for the first time in history, the White House will be connected to you via electronic mail. Electronic mail will bring the Presidency and this Administration closer and make it more accessible to the people.

The White House will be connected to the Internet as well as several on-line commercial vendors, thus making us more accessible and more in touch with people across this country. We will not be alone in this venture. Congress is also getting involved, and an exciting announcement regarding electronic mail is expected to come from the House of Representatives tomorrow.

Various government agencies also will be taking part in the near future. Americans Communicating Electronically is a project developed by several government agencies to coordinate and improve access to the nation's educational and information assets and resources. This will be done through interactive communications such as electronic mail, and brought to people who do not have ready access to a computer.

However, we must be realistic about the limitations and expectations of the White House electronic mail system. This experiment is the first-ever e-mail project done on such a large scale. As we work to reinvent government and streamline our processes, the e-mail project can help to put us on the leading edge of progress.

Initially, your e-mail message will be read and receipt immediately acknowledged. A careful count will be taken on the number received as well as the subject of each message. However, the White House is not yet capable of sending back a tailored response via electronic mail. We are hoping this will happen by the end of the year.

A number of response-based programs which allow technology to help us read your message more effectively, and, eventually respond to you electronically in a timely fashion will be tried out as well. These programs will change periodically as we experiment with the best way to handle electronic mail from the public. Since this has never been tried before, it is important to allow for some flexibility in the system in these first stages. We welcome your suggestions.

This is an historic moment in the White House and we look forward to your participation and enthusiasm for this milestone event. We eagerly anticipate the day when electronic mail from the public is an integral and normal part of the White House communications system.

President Clinton
`president@whitehouse.gov`
Vice President Gore
`vice.president@whitehouse.gov`

Announcement of electronic mail system by House of Representatives

Chairman Charlie Rose and Ranking Minority Member Bill Thomas of the Committee on House Administration announced today the pilot program of the Constituent Electronic Mail System.

This groundbreaking new service will allow citizens to communicate directly with their Member of Congress by electronic mail. The House of Representatives has established an electronic gateway to the Internet, the vast computer network that is used currently by over 12 million people worldwide. Participating Members of the House have been assigned public mailboxes which may be accessed by their constituents from their home computers. In addition, many libraries, schools and other public institutions now provide, or soon will provide, public access to the Internet.

The Members of the House of Representatives who have agreed to participate in this pilot program are: Rep. Jay Dickey (AR-07), Rep. Sam Gejdenson (CT-02), Rep.

Newt Gingrich (GA-06), Rep. George Miller (CA-07), Rep. Charlie Rose (NC-07), Rep. Fortney Pete Stark (CA-13), and Rep. Melvin Watt (NC-12). These Members will be making announcements in their congressional districts within the next few weeks to make their constituents aware of the new service.

The Constituent Electronic Mail System represents a significant effort by the House of Representatives to expand communication with constituents. With the tremendous growth of electronic mail over the past several years, and the increasingly inter-connected nature of computer networks, the new service is a natural addition to the current methods of communication available to constituents. At the present time, House Members involved in the pilot program will largely respond to electronic mail messages from their constituents by postal mail, to ensure confidentiality.

Constituents of House Members participating in the pilot program who wish to communicate with those Members will be asked to send a letter or postcard stating their interest to the Member's office. The request will include the constituent's Internet "address," as well as that constituent's name and postal address. This process will allow Members to identify an electronic mail user as his or her constituent.

The pilot e-mail program will continue until sufficient feedback from participating offices has been collected to allow improvements and modifications to the system. When House Information Systems and the Committee on House Administration are satisfied that the system is sufficiently error-free, other Members of the House will be allowed to add this new service as technical, budgetary and staffing concerns allow.

For more information, Internet users are encouraged to contact the House of Representative's new on-line information service. Please send a request for information to `congress@hr.house.gov`

JCGS Session at Annual Meetings

The *Journal of Computational and Graphical Statistics* will hold an invited paper session at the Joint Statistical Meetings in San Francisco this summer. The session will be held at 2pm on Wednesday, August 11. The speaker is William Cleveland, AT&T Laboratories, and the title of his paper is "A Model for Studying Display Methods of Statistical Graphics."

Invited discussants include Leland Wilkinson, Systat, Inc., and Lothar Tremmel, Sandoz Pharmaceuticals Corp.

Session Organizer & Chair: William F. Eddy, Department of Statistics, Carnegie Mellon University, Pittsburgh, PA 15213-3890; 412-268-2725

Interface '94 Announcement

26th Symposium on the Interface: Computing Science and Statistics

Preliminary Announcement

Date: June 15 (Wed evening) to 18 (Sat noon), 1994 [this is earlier than the date previously announced]

Place: Research Triangle Park, NC Sheraton Imperial

Sponsor: Interface Foundation of North America

Host: SAS Institute

Cooperating Institutions: ASA, IASC, SIAM, ORSA

Contact: John Sall 919-677-8000, `sall@sas.com` SAS Campus Drive, Cary NC 27513 USA

Theme: Computationally Intensive Statistical Methods

Adjoining: This meeting is immediately before the Third World Congress on Statistics, with the International Statistical Institute's Bernoulli Society and the Institute of Mathematical Statistics, meeting at Univ of North Carolina, Chapel Hill.

Soliciting Proposals for the Program

The Theme will be Computationally Intensive Statistical Methods.

Conference topics will include MCMC/Gibbs samplers, bootstrap, wavelets, randomization tests, nonparametric regression and smoothing, mixed model computations, graphics and visualization, software development, statistical image processing, space-filling experimental designs.

The meeting is conveniently timed before the Third World Congress on Statistics, with the International Statistical Institute's Bernoulli Society and the Institute of Mathematical Statistics, meeting at Univ of North Carolina, Chapel Hill.

SAS Institute is host institution for the meeting. Please send session proposals to John Sall, SAS Campus Drive, Cary NC 27513 USA, 919-677-8000, Fax 919-677-8123, email `sall@sas.com`

SECTION OFFICERS

Statistical Graphics Section

Richard A. Becker, Chair
908-582-5512
AT&T Bell Laboratories
rab@research.att.com

Roy E. Welsch, Chair Elect
617-253-6601
Massachusetts Institute of Technology
rwelsch@sloan.mit.edu

Werner Stuetzle, Past Chair
206-543-0711
University of Washington
wxs@stat.washington.edu

David W. Scott, Program Chair
713-527-6037
Rice University
scottdw@rice.edu

William DuMouchel, Program Chair Elect
617-489-2631
dumouche@jimmy.harvard.edu

Michael M. Meyer, Newsletter Editor
412-268-3108
Carnegie Mellon University
mikem@stat.cmu.edu

Linda A. Clark, Secretary/Treasurer
908-582-4807
AT&T Bell Laboratories
lac@research.att.com

Andrew F. Siegel, Publications Officer
206-543-4476
University of Washington

Kinley Larntz, Rep. to Council of Sections
612-625-1953
University of Minnesota
kinley@umnstat.stat.umn.edu

James M. Landwehr, Rep. to Council of Sections
908-582-7405
AT&T Bell Laboratories
jml@research.att.com

Statistical Computing Section

Sanford Weisberg, Chair
612-625-8777
University of Minnesota
sandy@stat.umn.edu

Trevor H. Hastie, Chair-Elect
908-582-5647
AT&T Bell Labs
trevor@research.att.com

Paul Tukey, Past-Chair
201-829-4285
Bell Communications Research
paul@bellcore.com

Mary Ellen Bock, Program Chair
317-494-6053
Purdue University
mbock@l.cc.purdue.edu

Sallie Keller-McNulty, Program Chair Elect
913-532-6883
Kansas State University
sallie@cecil.stat.ksu.edu

James L. Rosenberger, Newsletter Editor
814-865-1348
The Pennsylvania State University
jlr@stat.psu.edu

John Sall, Secretary-Treasurer
Sas Institute
sall@sas.com

James E. Gentle, Publications Liaison Officer
703-993-1994
George Mason University
gentle@imsl.com

Daryl Pregibon, Rep. to Council of Sections
908-582-3193
AT&T Bell Labs
daryl@research.att.com

Paul F. Velleman, Rep. to Council of Sections
607-255-4411
Cornell University
qp2@cornella.bitnet

Russell Lenth, C. of Sections Rep.-elect
University of Iowa
rlenth@stat.uiowa.edu

Reader's Notes

INSIDE

Statistical COMPUTING & GRAPHICS

The *Statistical Computing and Statistical Graphics Newsletter* is a publication of the Statistical Computing and Statistical Graphics Sections of the ASA. All communications regarding this publication should be addressed to:

James L. Rosenberger
Editor, Statistical Computing Section
Department of Statistics
The Pennsylvania State University
University Park, PA 16802-2111
(814) 865-1348
JLR@stat.psu.edu

Michael M. Meyer
Editor, Statistical Graphics Section
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213-1380
(412) 268-3108
mikem@stat.cmu.edu

All communications regarding membership in the ASA and the Statistical Computing or Statistical Graphics Sections, including change of address, should be sent to:

American Statistical Association
1429 Duke Street
Alexandria, VA 22314-3402
(703) 684-1221

PENNSTATE

1855

Department of Statistics
University Park, PA 16802-2111

Nonprofit Organization
U. S. POSTAGE
P A I D
Permit No. 1
University Park, PA 16802

Published by the Penn State Department of Statistics
326 Classroom Building, University Park, PA 16802-2111