



Statistical COMPUTING & GRAPHICS

A WORD FROM OUR CHAIRS

Statistical Graphics



Deborah F. Swayne is the 2001 Chair of the Statistical Graphics Section. Here is the news that she has from the section.

State of the section

We had a lively discussion on the section officers' mailing list this winter about the health of the section, which doesn't seem to be as active or innovative as it might be. We need to continue to think about how to engage the membership and how to productively focus our efforts. Contributions to this discussion from other section members would be most welcome.

We have some notable successes, such as the continuing popularity of our invited sessions and short courses, but we also have some notable weaknesses. I'll consider some of them here and describe the responses we're working on, and then tell you about some new ideas that have come up this year.

Student paper competition: a merger

We have not been very successful at attracting enough papers for a meaningful competition – we don't know how much this is due to our own inefficiency and how much to the fact that the number of papers on graphics produced each year is quite small, whether by students or not.

CONTINUED ON PAGE 4

Statistical Computing



Mark Hansen is the 2001 Chair of the Statistical Computing Section. Here are some of his news items.

Winter weather has finally settled over the northeast, bringing with it a kind of normalcy that I've been longing for. I have nothing to do on this chilly afternoon except revisit some of the initiatives I've been able to accomplish as Chair of the Section. Perhaps a simple list is the best format:

Section Web site

The domain www.statcomputing.org is now the official home of the Computing Section. On the site you will find a detailed history of the Section, news items, awards updates, conference announcements, bulletin boards and links to the Newsletter. Have a look and let us know what you think!

Section CD-ROM

Tony Rossini has recently compiled a collection of *essential* statistical software. We plan to distribute this collection in CD form to all active members of the Section. We are hoping that this will become an annual affair, offering a regular snapshot of the latest Computing tools.

CONTINUED ON PAGE 3

Report from the Workshop on Modern Statistical Computing and Graphics in Academia

The Statistical Computing and Graphics sections jointly sponsored a workshop prior to the Interface 2001 to brainstorm about how to recognize and promote statistical computing and graphics in academia. Participants discussed some of the problems facing young academics in our field, and sought actions that individuals and organizations might take to respond to these problems.

The workshop was attended by about two dozen people, most of them young academics and the rest senior academics and researchers in industry and the military.

The program for the workshop broke the day into four parts. Each of the first three sessions began with a short presentation and continued with discussion involving the whole group.

- Rewarding research in statistical computing (with Di Cook and Thomas Lumley as presenters and discussion leaders)
- Collaboration with other disciplines (Chris Genovese, Robert Gentleman)
- Curriculum content and development (Ross Ihaka, David Madigan, Michael Minnotte)

In the final session, workshop participants summarized the discussions of the day, and came up with some suggestions for action, and these are some of the more concrete proposals that emerged:

- **Problem:** Statistics Departments have a hard time evaluating work in statistical computing when it appears in tenure and promotion packages.
Response: Strengthen the outlets for publishing work in Statistics computing, especially software. We agreed that we would begin this process by adding our support to JSS rather than trying to spawn any parallel effort.
Actions: Formulate a document providing guidelines for evaluating software that might be used by Statistics Departments in promotion and tenure evaluations, and for review of journal articles on software.

- **Problem:** Statistical computing faculty have a hard time convincing their departments to adopt innovative practices.
Response: Build a web site which gathers data on policies, programs, practices that are friendly to stat computing.
Actions: We can begin with a survey of section members, and supplement what we learn with results of other ongoing investigations of statistics departments.
- **Problem:** Computational literacy is low in the Statistics profession.
Response: Offer introductory classes at Statistics meetings. Topics that would be appropriate include databases, scripting languages, and an introduction to R. Appropriate meetings include ENAR, WNAR, and the JSM. We would offer the classes as cheaply as possible, and offer them free to section members.
Actions: The sections should work with their Continuing Education representatives and Program Chairs.
- **Problem:** Statistics Department curricula offer few courses in statistical computing.
Response: Many departments are now experimenting, but there's little sharing of experience.
Actions: Collect syllabi, organize and make them available on the web.
- **Problem:** Junior faculty have to learn a lot in a short time.
Response: Offer experience and advice in some tricky areas, such as grant applications, preparing tenure and promotion packages
Action: Approach statisticians who have had success writing grant proposals, and those who have had experience in evaluating them, and ask them to write an article sharing their expertise.

If you wish to contribute to any of these activities, please talk to Debby Swayne or Mark Hansen.

The workshop organizers are grateful to Pfizer, Inc and Stata, whose generous contributions supplemented – and greatly exceeded – the seed money provided by the two sections.

Deborah F. Swayne, Di Cook, Mark Hansen, Guy Nason, Duncan Temple Lang and Adalbert Wilhelm
<http://www.statcomputing.org/workshops/scga/>



Editorial

We have finally managed to get newsletter issue out in 2001. Maybe we can think of it as the El Niño issue since its coming out just in time for Christmas and New Year's Eve. This is our first all electronic issue, we hope you enjoy it, and especially remember to use it as a navigation guide. In particular, all live links are surrounded by blue frames, so clicking on them will either awake your browser and take you to the site or if your browser is already working, it should just open up the relevant page in it. Here are some of the highlights of this issue.

In conjunction with Interface '01 the sections held a workshop on Modern Statistical Computing and Graphics in Academia. There is a report from this workshop in this issue, and the aim is to have several more substantial reports from the workshop in 2002 issues of the newsletter.

Perhaps as a consequence of the workshop this newsletter issue has a healthy selection of articles on software. Three commercial software production companies show their wares: Spotfire, Visual Insights, and S-Plus. Antony Unwin describes the software research activity at the University of Augsburg and Lee Wilkinson describes a new concept "streaming graphics" underlying his software Dancer. The next issue in 2002 will have several more articles describing software.

In addition we have two statistical computing papers on the topic of singular value decomposition (SVD). SVD is popular in the analysis of large multivariate data that is currently swamping many computers.

We have also included an account of an experimental class for teaching statistical computing, in the future, we would like to hear about our members' experience with various other software in the classroom.

Finally we wish to thank the contributors of articles in this issue for some thought-provoking reading. We had more than enough for this issue, which is not surprising since it took so long to be compiled. These additional articles will appear next year. We would like to encourage submissions for future newsletter issues. The newsletter has been traditionally an avenue for beautifully rendered color figures, and as we push further into the electronic domain we hope to become an avenue for dynamic, interactive figures. This year has been a *sloooooo* year for getting the newsletter out, but we promise to be more diligent in the new year.

Dianne Cook, Susan Holmes *Editors, Statistical Graphics and Computing Sections*



FROM OUR CHAIRS (Cont.) . . .

Statistical Computing

CONTINUED FROM PAGE 1

Refereed Papers.

This year, the Section on Statistical Computing is introducing an experimental refereeing process for contributed papers submitted to the Joint Statistical Meetings. We see this service as helping our members in two ways. First, by introducing some form of refereeing we can highlight outstanding contributed papers on statistical computing. Secondly, we hope to improve the overall quality of contributed papers and presentations at the JSM. Details of the process can be found on the Section Web site or a recent Amstat News article.

The Joint Statistical Meetings.

Doug Nychka put together an outstanding program for the JSM in Atlanta this year. The 2002 meetings will be in New York City, and Tim Hesterberg already has an impressive slate of sessions prepared. To welcome you all to New York, the Section will be raffling off a pair

of tickets to the Broadway show *The Producers*. They are excellent seats for the Tuesday night performance during the JSM.

Elections.

Please join me in congratulating our new Executive Committee Members Leland Wilkinson (Chair Elect), Mary Lindstrom (Program Chair-Elect), Charles Kooperberg (Secretary/Treasurer) and Robert McCulloch (Council of Sections Representative). I also want to thank our outgoing Secretary/Treasurer, Merlise Clyde. She has done a great job managing our finances for the last three years, and will be greatly missed.

Section Awards.

Lionel Galway has organized another very successful set of Section-sponsored competitions. Both the Student Paper Competition and the J. M. Chambers Award continue to draw outstanding entries. The winners for 2001 can be found on the Section Web site. The 2002 competition for these prizes is just starting, so please encourage eligible students to enter. (For 2002, the Computing and Graphics Sections have merged re-

sources and will be offering a single, joint Student Paper Competition.) Lionel also arranges our Best Contributed Paper Presentation Award, which this year went to Jay Servidea, a graduate student at the University of Chicago.

Workshop on Modern Statistical Computing and Graphics in Academia.

Just before the Interface meeting this year, our Sections sponsored a workshop that brought together some of the most influential members of the Computing community. In this edition of the Newsletter you will find a report on the steps that we will be taking to help encourage the research and education of Statistical Computing and Graphics.

We've accomplished a lot this year, and I expect that our

incoming Chair, Susan Holmes, will see that our Section continues to be one of the most active in the ASA. Finally, I want to thank our "new" Newsletter editors, Di Cook and Susan Holmes. Producing this volume is a time-consuming process. Extracting contributions from busy authors (and Section Chairs!) is a thankless job, but the end result is an important resource for our members. Please join me in congratulating them on another outstanding issue.

Peace.

Mark Hansen *Chair, Statistical Computing Section*



FROM OUR CHAIRS (Cont.) . . .

Statistical Graphics

CONTINUED FROM PAGE 1

As a result, the two sections will merge their student paper competitions into one, to be referred to as the "Statistical Computing and Statistical Graphics student paper competition." The graphics section is grateful to the computing section for supporting this idea, because their annual competition runs like a well-oiled machine and regularly attracts very good papers. The graphics section will help to disseminate publicity, will contribute judges who go to work every January, and will of course contribute prize money.

Data visualization exposition: planned for 2003

We haven't held our popular data visualization exposition or contest in several years, and we were a little bit too late to sponsor one at the 2002 JSM. That means we have plenty of time to plan one for 2003, and David James (chair-elect in 2002) has volunteered to lead the effort. Our target is to have web sites and flyers prepared to announce the contest prominently at the 2002 JSM, and there's a lot of work to be done before then. If you'd like to help plan it, or you have interesting data to suggest, please talk to David (dj@research.bell-labs.com).

New initiatives

The graphics section's major new activity this year was its cosponsorship of the workshop on the future of statistical computing and graphics in academia, held in conjunction with Interface 2001. You can read a re-

port on the workshop elsewhere in the newsletter. This was a very interesting experiment for our section, and I hope we continue to participate in the discussion reflected there, and that we contribute to implementing some of the suggestions that emerged.

Two other ideas have been put forth, and again, I'd love to hear from section members who find them interesting:

- Form an awards committee to recognize influential and innovative work in statistical graphics. This could involve honoring past work or new work.
- Design a set of posters that could adorn stat department hallways, to increase awareness of our field.

These two ideas might even have some synergy, since we could design posters based on the work of honorees.

Section communication

Mark Hansen and I began to use email this year to communicate with section members, and we hope you approve. We think both sections will continue to use email (sparingly!) to tell you what's going on and to ask for your reactions and ideas. If you would like to receive section email in the future, make sure that your email address in the ASA's records is up to date.

Deborah Swayne

Chair, Statistical Graphics Section



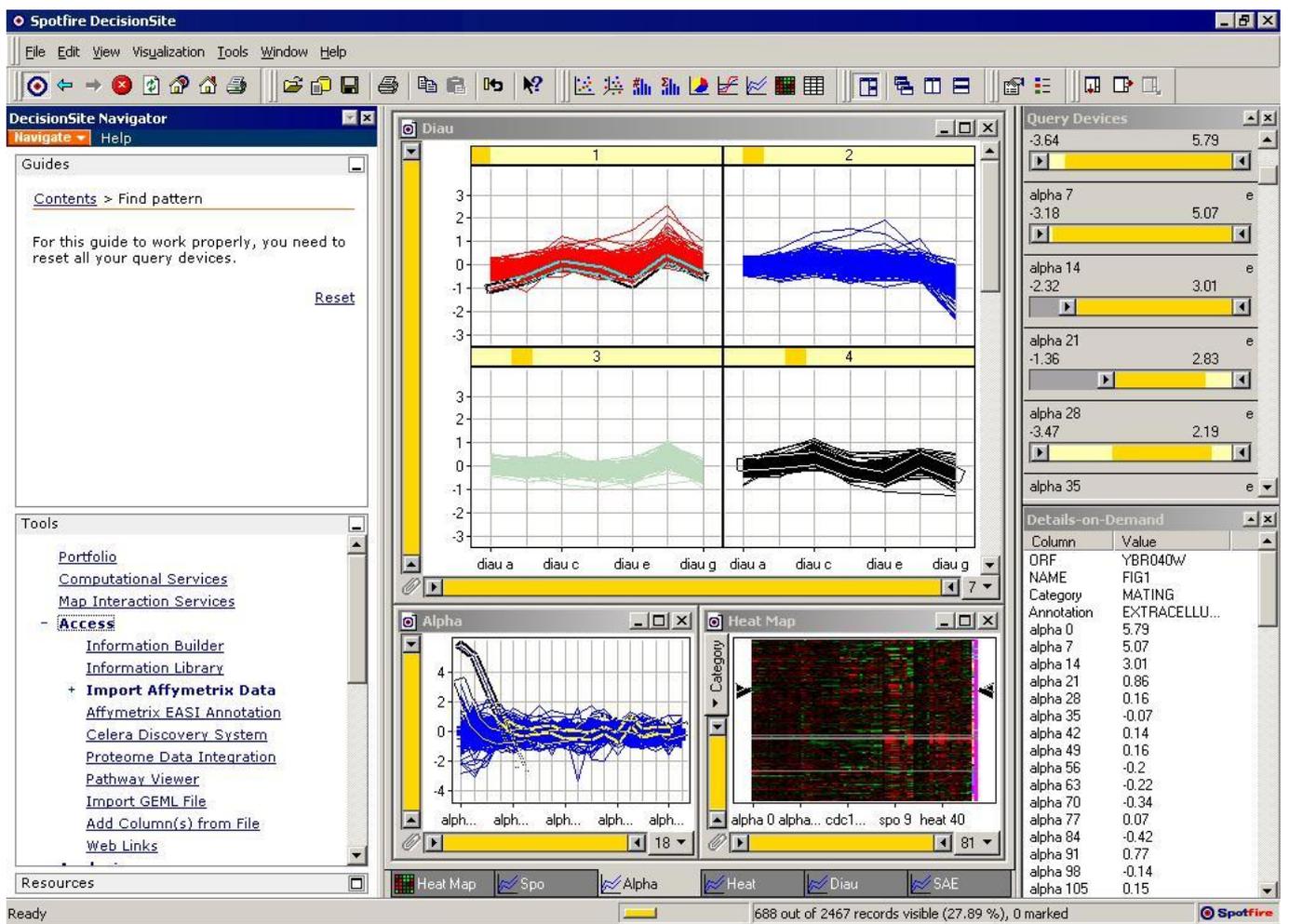


Figure 1: Spotfire DecisionSite - clustering and profiling genes.

SOFTWARE PACKAGES

The Next Stage For Analytics: Bringing Statistics to the Masses

Dr. Bill Ladd, Director Bioinformatics, Spotfire, Inc.
Applying Algorithms Across the Enterprise

Analytical applications are in high demand these days. For most organizations, that means better managing and analyzing the data generated by processes throughout the enterprise value chain. The volumes of data produced by today's high-throughput processes have made researchers increasingly turning to statisticians, computer scientists, and applied mathematicians for assistance with interpreting and analyzing research data. Statisticians need to find ways of making end-

users comfortable with the basic methodologies and tools so they can run their own analysis and not have to rely overworked statistical specialists.

Often, the development of a "correct" mathematical approach to a specific problem is the first in a series of steps that will make a new algorithm usable to end-user scientists. Once developed, the algorithm must be integrated into existing analysis processes. It must be delivered within an analytics framework that encourages data exploration and interpretation. And this framework must be deployed across broad, dispersed user communities- the globally distributed project teams that drive research decision-making.

Once an algorithm has been developed, the issue of deploying it to other researchers and other areas of the enterprise is critical. Algorithms deployment can be deemed inadequate because they fail to mirror the analytical process preferred by researchers. A framework

supported by a robust, adaptable platform and able to be configured to the established processes is needed to share such algorithms and “best practices” throughout an enterprise.

Analyzing the data and communicating the results to the appropriate places in the organization to facilitate decision-making is, ultimately, the reason why researchers generate data in the first place. And the analysis tools- be they the latest killer algorithm, a desktop analysis application, or a slide rule—are all means to this important end. Because the questions researchers ask often refer to the data they have moving through their research process, their needs can seem isolated and specific. Analytical specialists, in turn, deliver tools suited to these specific needs- solutions that maximize the use of the data at a specific point in the larger data flow that begins with laboratory information management systems or basic research experiments and ends with a finished product. But to be useful over time and add value to the data produced during research projects, analytics should focus on providing tools that can be added to or replaced within an ongoing and ever-changing analysis process.

Delivering an Analytic Framework: Spotfire DecisionSite

The Spotfire DecisionSite framework addresses these issues and provides a usable, eAnalytical application that has a number of key elements for statisticians to distribute algorithms and analysis tools.

Data Access

Crucial decisions, those that drive research projects, result when data can be compared across experiments, labs, and fields. Yet too often simply acquiring data from the various sources where it resides is difficult. A successful analytics framework must support truly integrated data access- data from any source, in any format, anywhere, at any time, without any end-user programming.

Data Visualization

Modern computing tools can do more than simply display results-they can provide interactive, fast, and flexible data visualizations that facilitate and even amplify human thought processes. An analytic framework must include data visualization tools that can present complex datasets in unique ways that engage these innate perceptive abilities. These visualizations must be made interactive, encouraging researchers to explore unexpected behavior in their data. Interactive visualizations let machines focus on what they do best-data processing-while humans focus on their talent-making

decisions.

Usability

An analytics framework may connect to all the right data sources and incorporate impressive visualizations and analysis options. But if it is not easy to use, it will not get used. A modern analytics framework must cater first to the needs of a wide variety of scientists, offering more sophisticated toolsets as an option for those scientists who want greater control.

“Guides” have emerged as a popular way of simplifying complex tasks by stepping users through common tasks, rather than searching through menus to find the option needed to complete a task. Guides are a useful usability tools because they can serve both novice users and power users. And when applied to research, guides can greatly speed many aspects of data analysis, from transacting with or entering experimental information to setting the parameters of a complex clustering method, statistical analysis, or data reduction algorithm.

Extensibility

Software solutions supporting analytics must be adaptable and readily configurable to support access to new, emerging data sources, new analytical tools, and new analysis processes. The architecture of the platform should be able to incorporate methods developed in many different architectures and runtime environments.

Many analytical applications claim extensibility, but place such strong restrictions on the approaches used to add functionality; in the end, they are quite limited in scope. When an analysis framework is well planned, such as Spotfire DecisionSite, the application infrastructure itself becomes an important part of the application’s functionality.

Deployability

Global project teams encompassing researchers from a variety of disciplines must be able to share data and come to consensus on research directions. Web infrastructures are ideal for these purposes and are the logical choice to underpin an analytics framework. Users gain access to the most up-to-date statistical and analysis methods that apply to their research. With one change to one part of the system, IT staff can deliver new features and functionality to researchers anywhere in the world.

Consistency

Well-planned analytics frameworks do more than meet the access, analysis and usability demands of the individual researcher- they provide consistency across

the broader organization. Analytic tools and statistical methods geared towards one application area can be adapted by researchers in another application area for their own purposes. As different user communities configure the environment for access to data and analysis methods relevant to their specific research, new classes of scientific investigation will be enabled that bring these previously disparate classes of data together.

Looking forward

If analytical specialists such as statisticians expect their methods to be adopted and used productively throughout the enterprise, they need to embrace the analysis processes and frameworks used by their customers- the end-user researchers and decision makers. Research is not slowing down and the generation of experimental data stands only to increase. In evolutionary terms, the fittest research organizations in the future will likely be those that can efficiently analyze data to come to decisions about what to do next. Statistics and analytics must itself evolve. Industrial quality research requires industrial quality software that lets analytical researchers focus on their specialty with access to the latest statistical methods, rather than on data management, reporting or storage.

For more information on the Spotfire DecisionSite eAnalytic application, please visit <http://www.spotfire.com> or send a request to info@spotfire.com.

Visualization for Information Consumers

Stephen G. Eick, Visual Insights

<http://www.visualinsights.com>

eick@visualinsights.com

Abstract

Within an organization there are different types of information consumers with varying needs and levels of sophistication. Our focus within the statistical graphics community traditionally has been on building sophisticated tools for highly-skilled analysts. By focusing broadly on the information needs and understanding different user tasks, we can create visual tools that appeal to all information consumers within an organization and extend our influence outside of our core area of statistical graphics.

Information Consumers

Within the statistical graphics community we delight in producing sophisticated tools for visual data analysis. Our tools are powerful and incorporate the latest thinking on statistical analysis and data mining. The guiding philosophy, it seems to me, is that we build the tools that we need to perform our analyses. The result is wonderful demonstrations at the JSM and other leading scientific conferences. Sometimes, however, it appears that some magical combination keys, selections, and mousing was needed to produce the result.

This lack of usability may have to do with our background. In the development of tools we simultaneously wear multiple hats. Frequently we are the creator, developer, user, and evaluator simultaneously. By combining these roles, we can rapidly iterate through various ideas and techniques. There is, however, a negative aspect of combined roles. Our perspective generally is that of a highly trained, sophisticated data analyst and our tools embody this perspective.

In an organization there are three different types of information consumers. At the highest level, the Executive wants quick summary information so that problems are apparent at a glance and there is enough detail for intuitive decision-making. The Manager, who reports to an executive, has time to explore data and look for trends. He needs the ability to customize the reports without being overwhelmed by complexity. The Analyst or Doer wants full access to all available information at a fine-grain detailed level. He or she has the time and interest to perform detailed studies, deep analyses, and needs powerful tools.

To meet the varying needs of information consumer I have defined four different types of visualizations. For concreteness, I will use examples from website analytics. Website analytics is the study of how users are accessing websites. It involves the collection and analysis of clickstream data, understanding how website effectiveness, investigating visitor demographics, studying of on-line transactions, and influencing behavior through on-line promotions.

Four Types of Visualizations

This section highlight four types of visualizations organized in increasing of complexity.

Digital Dashboards (Figure 1) provide a “at a glance” presentation for quick intuitive decision-making. The key intellectual property is the choice of statistics, appropriate time scales, pleasing presentation, and simple graphics. Users may manipulate the time ranges and select other tabs in the display for more detailed informa-

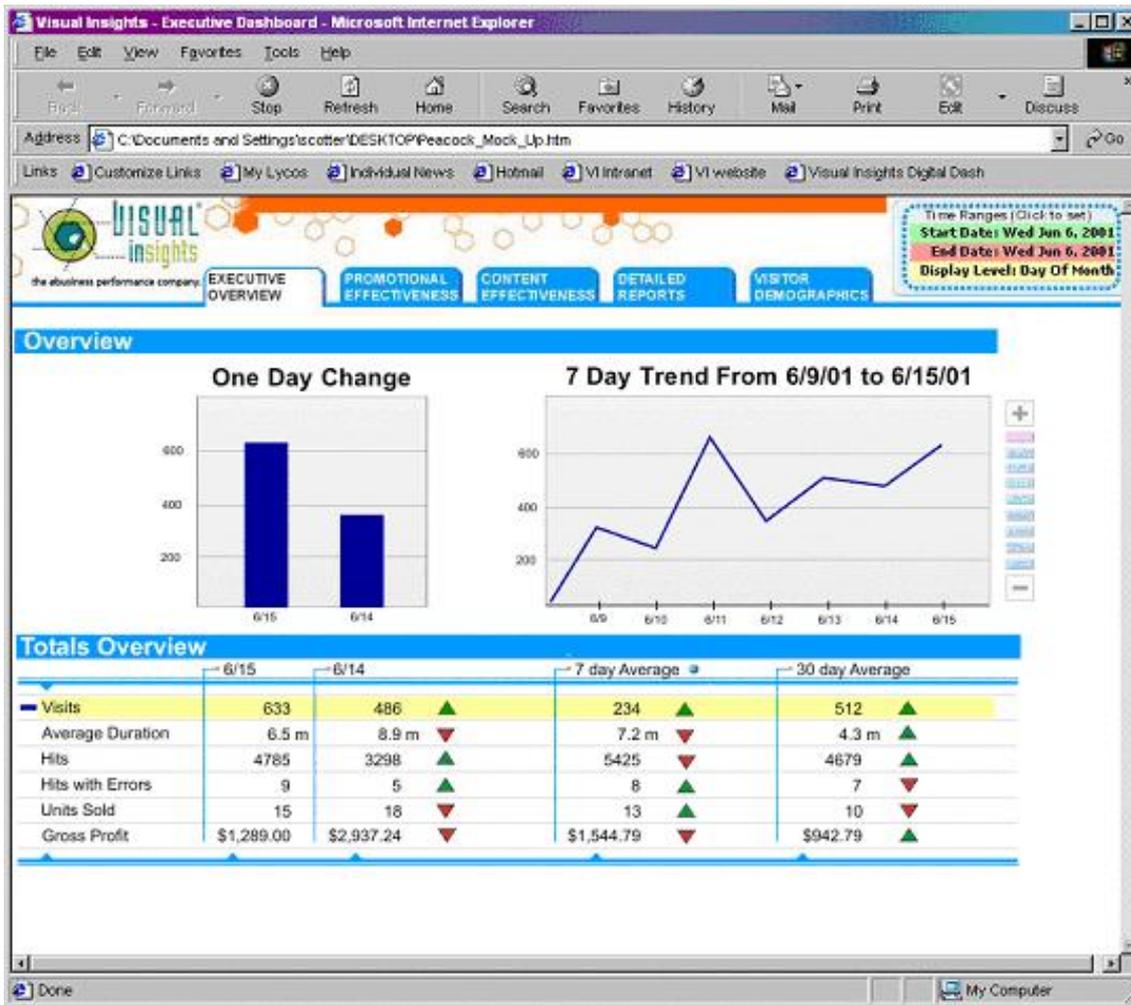


Figure 2: Browser-based Executive Dashboard showing website activity. Green and Red triangles highlight changes (positive and negative) from the previous time period.

tion. The interesting research issues involve the graphical presentation, work flow, and implementation.

Active Reports (Figure 2) are customizable graphical interfaces that are used for flexible reporting, trending and publishing. In contrast to browser-based digital dashboards, an active report is more complicated and more flexible. Our implementation of active reports is a client-side windows application where users may manipulate data, change thresholds, publish html snapshots, re-aggregate, export results sets. Active reports are frequently used by business analysts. The key issues involved with active reports include the flexibility of data access, ease of creating reports, and ability to integrate with other systems in an organization.

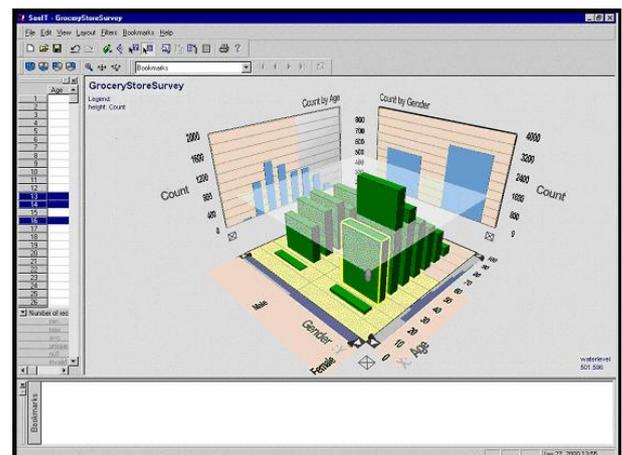
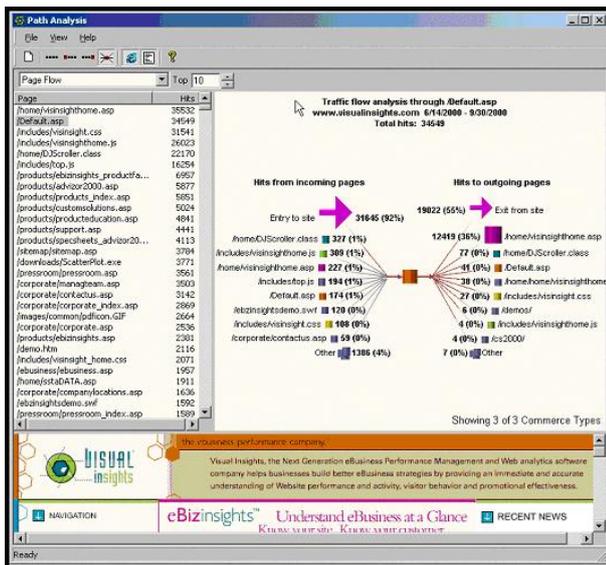


Figure 2: Active 3D Report. The transparent cutting plan helps users to visually compare bar heights.

Visual Discovery and Analysis (VDA) tools, such as XGobi, are powerful environments that combine state-of-the-art statistical techniques with visual data analy-

sis. They combine both statistical and graphical tools into a common framework.

Task-specific visualizations (Figure 3) capture domain knowledge to address domain-specific issues. For website analytics, a critical issue involves understanding that paths that browsers take to navigate the site - see Figure 3. The intellectual property includes novel visual metaphors and interaction techniques. Task-specific visualizations will be widely used by all information consumers in an organization.



Path visualization for www.visualinsights.com.

Summary and Conclusion

Within an organization there are different types of information consumers that have different needs. This essay highlights three different types of users:

- Executive,
- Manager, and
- Analyst

and four types of visualizations:

- Digital Dashboards,
- Active Reports,
- Visual Discovery Analysis Tools, and
- Task-specific visualizations.

Digital Dashboards, while engineered for an Executive, are useful for all members of an organization. Active Reports and VDA tools address Manager's and Analyst's needs. Task-specific visualization are role-neutral may be used by anybody with domain knowledge.

The important take away message for us in the statistical graphics community is that our background, training, and inclinations encourage us to build VDA tools

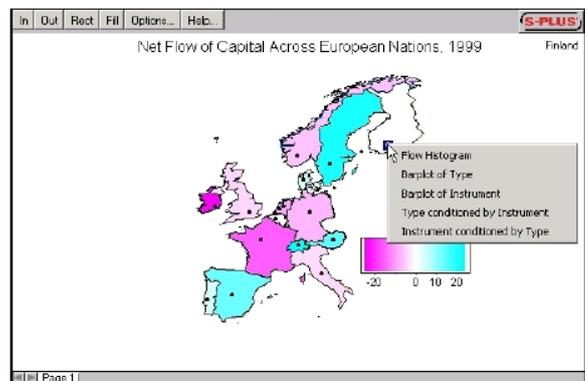
for analysts. These tools, while highly sophisticated, do not necessarily meet the needs of the other information consumers. To meet these broader needs, we must design tools that target tasks beyond our traditional focus on statistical graphics.

S-PLUS Graphlets

Charles Roosen, PhD, Senior Statistician, Insightful Corporation

S-PLUS 6 for UNIX and S-PLUS 6 for Windows introduce an exciting new technology for displaying graphics over the web. The Graphlet is a Java applet that supports displaying and interacting with S-PLUS graphics that have been saved in a file. The Graphlet can be embedded within a Web page, and viewed by a web browser.

Creating an S-PLUS Graphlet file is as easy as creating a graph in S-PLUS. Use the award-winning S programming language to create your graphic, and then simply save it as a Graphlet. Because the S language was designed explicitly for graphical display and statistical analysis, hundreds of high-level functions are at your disposal for creating an informative and attractive graph with just a few lines of code.



Graphlets allow you to add interactivity to your graphs. Graphlets are live objects, and the viewer can interact with the Graphlet by moving and clicking the mouse within the Web browser. With a simple click on a data point or label the viewer can be provided with further information through on-screen information, another Graphlet, or another page anywhere on the World Wide Web. The applications of this interactivity are endless. Here are just a few examples:

- Click on a financial time series chart to show all the news stories relevant to the date where the mouse was clicked.

- Make the bars in a bar-chart active, so that clicking on a bar will drill down to display another chart containing only the data in that bar.
- Link cities on a map to further detailed charts about that city.
- Add interactive labels to a scatter plot to provide more information about your data.
- Provide multiple graphs as labeled pages in a single Graphlet, allowing the viewer to browse among the pages at will.
- Display latitude and longitude coordinates of the mouse position within an image map, and allow the viewer to zoom in and display a specific region.

The Graphlet window provides further tools for the user to zoom in on selected regions of your graphic to view details. And because Graphlets—unlike GIF and JPEG graphics—are rendered on-the-fly using vector graphics, there is no loss of resolution or pixelation when looking at your graphic "up close and personal".

Since Graphlets are implemented using standard Java Applet technology, they can be embedded into any Web page using standard Web authoring tools such as Microsoft FrontPage. Graphlet files are very small (typically 20-60 Kb) so they download quickly across the Web. S-PLUS Graphlets can be viewed on any Java-capable browser, including Internet Explorer 5 and Netscape 4. The S-PLUS Graphlet applet is freely redistributable, so you can give S-PLUS Graphlets to friends and colleagues for use on their Web pages, even if they don't have S-PLUS.

For more information on Graphlets and a variety of live examples, see the Graphlet web page at <http://www.insightful.com/graphlets>.

COSADA Software Projects in Augsburg

Antony R. Unwin, Department of Computeroriented Statistics and Data Analysis, Augsburg University

antony.unwin@math.uni-augsburg.de

Introduction

MANET, TURNER, CASSATT, KLIMT, MARC — there has been a succession of statistical software developments at the department of ComputerOriented Statistics and Data Analysis (COSADA) at Augsburg University in Germany. (Martin Theus's MONDRIAN might also be counted, but he wrote it after he left Augsburg and was at AT&T.) All the software is for interactive

analysis. Users should be able to work directly with objects on screen and not have to type in commands. The software is research software, in that the main aim is to develop and test new interactive ideas, but great emphasis is also put on achieving an intuitive, fast and flexible interface. Ideas can best be tested if they are implemented in an effective way. The programmes have primarily been written by students. While all members of the department have an input into the design and testing of the software, the programming is always the responsibility of just one person.

Each programme is named after an impressionist painter, because software can only help the user to get impressions of their data. MANET and TURNER were developed for Macs but more recent programmes have been written in Java, so that they can potentially run on any computer system.

CASSATT

CASSATT implements interactive parallel coordinates. Consider, for instance, the 1999 reprise of the famous California-Bordeaux wine tasting data set (www.nerc.com/liquida/report20.html). There were 46 wines which were all scored by 32 judges. Did the judges agree on their rankings? Did at least some of the judges agree? Were the French and the Californian wines ranked differently? Were some wines always ranked higher than others? Which judge(s) were closest to the average ranking of all judges? Can patterns of rankings be determined? There are many interesting questions that may be pursued and a combination of analytic and graphical methods would be best. The figure shows the rankings by wine in a parallel coordinate plot. There is an axis for each of the 46 wines and each broken line represents the views of a single judge of the 32 judges. The axes have all been scaled equally (a menu command), as the default is to scale each axis individually (a natural default when, unlike here, different variables have non-comparable scales). The bottom of each axis represents the minimum data value (best rank) and the top is the highest data value (worst rank). From the large amount of lines crossing from the top of one axis to the bottom of the next axis, it is immediately apparent that the judges were not in close accord! To assist interpretation the wines and hence the axes have been sorted by average ranking (one of many sorting possibilities and again a menu command) with the best to the left. The judge (highlighted trace) who gave the overall average best wine the worst ranking, gave some of the least popular wines their best ranking. To improve visibility the lines for the judges who are not selected have been lightened (this is done using a floating dialog box).

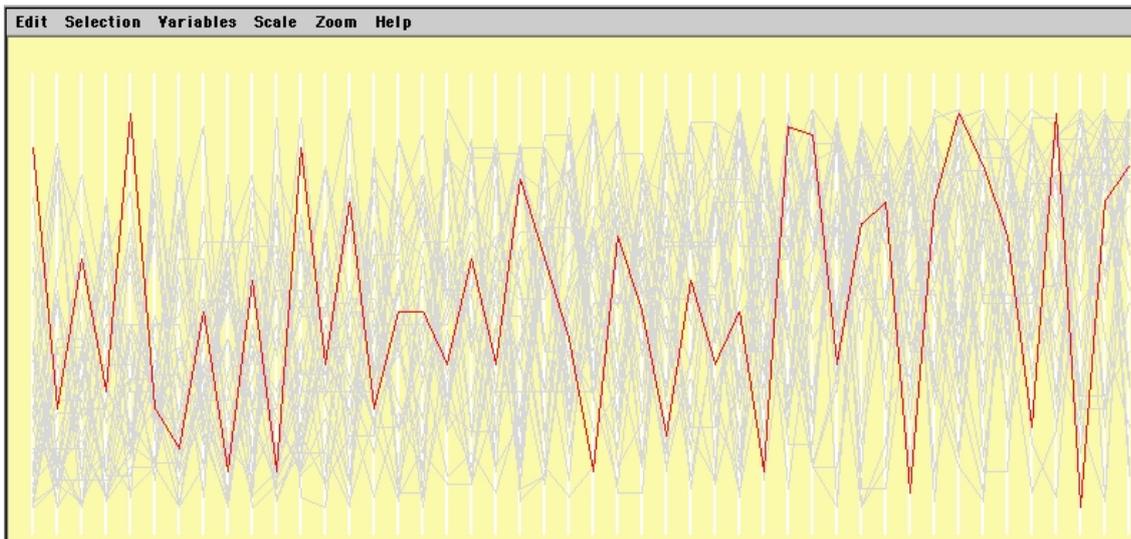


Figure 3: 32 judges rankings of 46 wines displayed in a CASSATT parallel coordinate plot.

No brief description can do justice to a sophisticated piece of software. CASSATT is a fully linked multi-window system providing not just parallel coordinate plots but also standard plots for continuous variables (dotplots, boxplots, scatterplots) as well. It offers the basic interactive tools (including querying, selection, linking, sorting and colouring) but also some specially relevant to parallel coordinates (inverting axes, selection methods for selecting not only points, but also lines and even angles). Selection sequences, which were introduced in MANET to provide an editable, graphical interface for selections, have been extended in CASSATT to include the selection history in a graphic form. The software has been used successfully (i.e. its interactive features are still fast enough) with datasets with over 300 variables and with a dataset with over 20000 cases.

Future directions

CASSATT assumes, like most statistical software, that the data set is supplied as a data matrix. It would be useful to be able to work with more complex structures. For instance, it is currently not possible in the tasting dataset to include which wines are French and which Californian, what vintages they were and what combinations of grapes they contained. If we had this information, we could interactively sort the wines by country of origin or by vintage.

Parallel coordinates are excellent for continuous variables but not so good for categorical variables. Too many lines simply overlap and it is hard to see structure. Missing values are another related problem. New approaches are needed. We intend to investigate the

linking of mosaic plots and parallel coordinates in the same package.

And, of course, we would like to interweave CASSATT with R, the way we are doing with KLIMIT.

CASSATT was written in Java by Sylvia Winkler, who shared first prize in the inaugural John Chambers software competition for her work.

Check CASSATT's homepage for further information: www1.math.uni-augsburg.de/Cassatt/

Streaming Graphics

Andrew A. Norton, Matthew A. Rubin, Leland Wilkinson, SPSS, Inc.

leland@spss.com.

Abstract

Streaming graphics, a cross between streaming video and dynamic graphics, is a new approach to visualizing data. The goal is to integrate and synchronize multiple streams of data in real time and display them at rates of up to 20 frames per second. This article describes the design of a system, called *Dancer*, that implements these ideas in Java.

Introduction

The term *streaming graphics* evokes the terms *streaming media* (e.g., Feamster, 2001), and *dynamic graphics* (e.g., Cleveland and McGill, 1988). While similar to both in its outward appearance, streaming graphics is basically different from both. Streaming media systems are generally concerned with delivering sound and

video information in real time. Dynamic graphics are concerned with using motion to reveal structure in static data. Streaming graphics systems are concerned with displaying analyses (summaries) of streaming data in real time. Applications of streaming graphics involve many different environments, including real-time monitoring of manufacturing processes, health indicators, financial statistics, and Web data.

Data

A streaming data source is fundamental to streaming graphics. In the simplest streaming data model, data arrive in a fixed-length buffer (window) at equally spaced time points. Upon the arrival of a new data packet, we lose one of the packets already in the buffer (usually the last). When a new data packet arrives, we refresh our display. We can also have a fixed-length buffer that receives data packets at irregular time intervals. In this environment, the arrival of new data can trigger a display refresh. There are many other prevalent streaming data environments, however, particularly ones involving multiple, asynchronous data events occurring simultaneously within multiple processing threads.

Streaming vs. Static Data

A popular metaphor for the difference between streaming data sources and static data warehouses is a *stream* versus a *pool*. In the massive data-mining environment, we often hear this metaphor upgraded to a *river* versus an *ocean*. The significance of this distinction, whatever the scale, is that our graphical and statistical algorithms for mining a stream (to stretch the metaphor) must adapt to its temporal nature. We must pay attention to speed of calculation, lest we be swamped with new data packets before we can compute a displayable result. We must allow calculations to persist wherever possible so that we do not waste time redoing subsets of previous calculations. We must learn to sample our data stream, or know how to bail-out of calculations when the results might be out-of-date. We must synchronize our results, lest we mistakenly display summaries of disparate events in the same time frame.

Many of these concerns do not apply to the traditional, static data mining environment. The premier consideration in the static environment is scalability with regard to the *size* of a dataset. We seek algorithms that scale comparatively well, in the sense of computing time being a constant, linear, or logarithmic function of the size of a dataset. In the streaming data environment, we need to worry about scalability with regard to the *momentum* of a stream. The *momentum* of a stream is its *mass* (the average size of data packets arriving in a fixed interval)

times *velocity* (the average number of packets arriving in a fixed interval). This Newtonian physics metaphor is an over-simplification, but it helps us to understand that attending to scalability with regard to the total bulk of the data we encounter will not be sufficient for building a streaming display system. Even a constant cpu-time calculation may still be too slow to handle the real-time data stream we must display.

Multiple vs. Single Data Source

In the static data environment, we merge different data sources into a single table before computing graphical and statistical summaries. We cannot employ this simplification in the streaming environment. Instead, *Dancer* is designed to handle multiple datasources. For example, we might want to attach a single display to a live feed from a stock exchange and another live feed from a commodities exchange. There is no time to index and merge these two feeds into a temporal database.

Multiple feeds imply a multi-threaded computing environment with event-notification through broadcasters and listeners. The data feeds run simultaneously, and each broadcasts the arrival of new data it receives. The statistical and graphical listeners attend to those messages and react as they choose. In our financial example, the result is a display that shows a moving time-series of a particular stock superimposed on a moving time-series of a related commodity.

Continuous vs. Discrete Time Scales

A consequence of this functional architecture is that *Dancer* operates on a continuous rather than discrete time scale. In theory, *Dancer* displays data at any instant. Its time scale moves (at least perceptually) continuously. This is in contrast to some time series displays, such as seen in traditional ARIMA (Box and Jenkins, 1976) packages, that contain measurements indexed on equally spaced time points.

Dancer has two time-scale modes. In the static-frame mode, geometric representations of data (points, lines, etc.) move through a motionless frame (the data area). We may clip the elements at the frame boundaries or let them pass beyond.

In the dynamic-frame mode, geometric elements are continuously repositioned within a frame that is continuously scrolling forward or backward in time. This mode requires careful programming to allow tick marks, grid lines, and scale labels to scroll smoothly and continuously, in 2D or 3D as required.

Geometry

Dancer is based on a geometric model of statistical

graphics described in Wilkinson (1999). That book contrasts the geometric model with the “chart-centric” model of statistical, business, and scientific graphics packages. In the geometric model, elements such as points, lines, and intervals are embedded in a frame defined by an algebra with three operators. These geometric elements are realized (made viewable, hearable, etc.) through a set of aesthetics such as size, shape, texture, and color. The *nViZn* system for rendering graphs on the Web is based on the same model (Wilkinson, et al., 2000).

Unlike *nViZn*, *Dancer* can bind each geometric element to a different data source. The behavior of every element in the frame is synchronized through the functional data model. Special rendering technology is required to enable each element to move up to 20 times a second based on incoming data.

Scene Tree vs. Geometric Primitives

A *scene tree* is a data structure that contains the information in a geometric scene. Scene trees are frequently used in 3D modeling to organize the arrangement and rendering of a collection of geometric objects. For example, if the root of a tree is a body, then the root’s children could be torso, head, arms, and legs. The children of arms could be upper arms, lower arms, and hands. And the children of hands could be fingers and palms. The advantages of this organization are several. First of all, children inherit attributes from parents, including aesthetics (e.g., color, texture) and localized coordinates. Second, adding objects to a scene (another body, for example) requires only appending a new subtree to an appropriate node. Third, a tree is a simple object that can be explored through depth-first or breadth-first search. This makes rendering rapid and efficient. And it makes interactions with elements (editing, brushing, linking) a simple matter of walking the tree to locate a selected element.

Wilkinson (1994) employed a *graph tree* model for constructing and rendering statistical graphics in the SYSTAT package. The hierarchical nodes of the tree were Window, Pane, Frame, Graph, and Element. With the Fisher-Anderson Iris data, for example, SYSTAT plots three scatterplot matrices (three species, four variables) in a window by assigning one Pane node to Window and three Frame nodes to Pane. Each Frame contains sixteen Graph nodes. Each Graph node contains two Element nodes (one for a cloud of points and one for a smoother). The most important consequence of this architecture is that SYSTAT has no scatterplot matrix routine. Instead, SYSTAT has a method for assembling graphical elements in a structure of one or more graphs

and frames. Anything it can do inside a single scatterplot it can do simultaneously inside multiple scatterplot matrices. For the same reasons, SYSTAT has no routine for drawing a Trellis (Becker et al., 1996). Instead, SYSTAT constructs a Trellis structure by assembling graphs in a hierarchy determined by values on categorizing variables. Details of look-and-feel can be handled in display modules that render this structure in different styles.

Scene trees and graph trees organize geometric primitives. They are essential for creating efficient real-time streaming graphics systems for a variety of reasons, some of which are evident in the SYSTAT architecture. Most important, perhaps, scene tree architecture enables rapid rendering.

Supervised vs. Immediate Rendering

When we render a scene that is rapidly changing, we may re-render the entire scene every time a change is detected or render only the parts that change. If our goal is to render complex scenes that may change up to 20 times a second, we choose the latter strategy.

In a graphics foundation like Java3D, the re-rendering housekeeping is taken care of by the Java scene tree. When the scene tree is updated, only parts of the screen that need updated are re-rendered. An added benefit is that foundations like Java3D are designed to accommodate 3D hardware accelerators.

Statistics

Statistical graphics depend on the calculation of statistics – point estimates, interval estimates, smoothers, summaries. In a static data environment we can precompute these statistics before rendering a graph. The most prevalent example of this strategy is in OLAP (On Line Analytic Processing) systems based on ETL (Extracting, Transforming, Loading) technology. These systems aggregate data from various sources, store them in a warehouse, and display graphics such as pie and bar charts on the aggregates. The Temple MVV visualization system of Milhalisin et al. (1995) works similarly.

Update/Downdate

Streaming graphics require a different type of statistical algorithm. In the simplest case, such as accumulation of sums and sums of squares and cross-products, we use an update/downdate strategy. When new data arrive in our computational buffer, we discard the oldest data item, downdate its contribution to the accumulated statistics, and update the statistics from the new contribution. This process can accumulate rounding error, so it is essential to recalculate occasionally all values from the data in a

window. This recalculation can be scheduled to occur at convenient times, similar to the way garbage collection is handled in an interpretive system.

Iterative calculations are not as simple. We can update and downdate Hessians and Jacobians, but handling convergence in a real-time environment is problematic. Other statistical algorithms present different problems. Guha et al. (2000), Datar et al. (2002), and others are concentrating on these issues.

Visualization

The simplest display for streaming data reflects the current state of the stream. Geometric elements such as points and lines move as the stream progresses. Real-time is only one aspect of what Dancer does with streaming data, however.

Instant replay vs. animation

Whatever the time interval (seconds, minutes, hours, ...) we cannot expect a viewer to observe a display continuously. A real-time system needs to incorporate alerts for out-of-bounds conditions. Visual and audio alerts are common in control applications in military, health care, and manufacturing.

When an alert occurs, it is not always clear what antecedent conditions led to its occurrence. Consequently, Dancer provides an instant replay feature to allow a viewer to rewind a scenario and replay it as an animation. This is an aspect of a more general Dancer feature allowing us to animate any data series. Although designed for real-time applications, Dancer's functional time model can be applied to any indexed (ordered) variable so as to allow animation of other variables. In this respect, Dancer can behave like exploratory systems such as XGobi (Swayne et al., 1998) or DataDesk (Velleman, 1988) that animate over a variable.

This type of *archival* animation needs to be distinguished from Dancer's normal real-time model, however. Archival animation offers the opportunity to pre-process data that we do not have in real-time. In the extreme, animation systems such as *Flash* (www.macromedia.com) can prepare bitmap frames that are played in a media player. These animations are fundamentally different from the architecture in Dancer.

Transformations of time

When we animate a sequence by buffering and replaying captured data, we may transform time in a variety of ways. Reversing time helps us to untangle sequential dependencies. Differencing time helps us to recognize rates. Double-differencing time reveals acceleration/deceleration. Polarizing time (a cyclical trans-

formation) helps us to compare cycles. Logging time helps us to view order-of-magnitude effects. Geologists and cosmologists often log time scales and, as Graham Wills has pointed out (personal communication), ordinary people often use a similar transformation when they look into the future on a day ... week ... month ... year time-scape.

Conclusion

Streaming data require streaming algorithms. Much of the technology for processing these data is relatively new and much is yet to be done. Because statistical graphics is more involved with processing data than with drawing pictures, there is considerable technology transfer than can occur from related fields in computer science.

An indication of the challenges involved in processing real-time data can be seen in the contrasts between *Dancer* and its sibling *nViZn*. Both are based on the graphics grammar model and both are programmed in Java to take advantage of a Web environment. The two programs share not a line of code.

References

- Becker, R.A., and Cleveland, W.S., and Shyu, M-J. (1996). The Design and control of Trellis display. *Journal of Computational and Statistical Graphics*, 5:123-155.
- Box, G.E.P., and Jenkins, G.M. (1976). *Time Series Analysis: Forecasting and Control*. San Francisco: Holden-Day.
- Chambers, J.M., Cleveland, W.S., Kleiner, B., and Tukey, P.A. (1983). *Graphical Methods for Data Analysis*. Monterey, CA: Wadsworth.
- Cleveland, W.S. and McGill, M.E., Eds. (1988). *Dynamic Graphics for Statistics*. Belmont, CA: Wadsworth Advanced Books.
- Datar, M., Gionis, A., Indyk, P., and Motwani, R. (2002) Maintaining stream statistics over sliding windows. Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco.
- Feamster, N.G. (2001). Adaptive delivery of real-time streaming video. Master's thesis, Department of Electrical Engineering and Computer Science, MIT. Online at <http://nms.lcs.mit.edu/papers/feamster-thesis.pdf>
- Guha, S., Mishra, N., Motwani, R., and O'Callaghan, L. (2000). Clustering data streams. In *Proceedings of the*

Annual Symposium on Foundations of Computer Science (FOCS 2000).

Mihalisin, T., Timlin, J., Gawlinski, E., and Mihalisin, J. (1995). Visual analysis of very large multivariate databases. *ASA Proceedings of the Section on Statistical Graphics*, 18-27.

Swayne, D.F., Cook, D., and Buja, A. (1998). XGobi: Interactive Dynamic Data Visualization in the X Window System. *Journal of Computational and Graphical Statistics*, 7:113-130.

Velleman, P.F. (1988). *Data Desk*. Ithaca, NY: Data Description Inc.

Wilkinson, L. (1994). *SYSTAT, Version 6*. Evanston: SYSTAT, Inc.

Wilkinson, L. (1999). *The Grammar of Graphics*. New York: Springer Verlag.

Wilkinson, L., Rope, D.J., Carr, D.B., and Rubin, M.A. (2000) The language of graphics. *Journal of Computational and Graphical Statistics*, 9:530-543.

TOPICS IN STATISTICAL COMPUTING

Rolling Your Own: Linear Model Hypothesis Testing and Power Calculations via the Singular Value Decomposition

Steve Verrill, Mathematical Statistician,
USDA Forest Service Forest, Madison
steve@ws13.fpl.fs.fed.us

Abstract

We outline the steps that would permit a statistician to produce special purpose linear model routines through the use of high quality public domain numerical analysis software.

Introduction

Good commercial linear model packages are readily available. It sometimes happens, however, that one would like linear model code that could be incorporated into a simulation. Verrill (1999) discusses such a simulation in the context of predictor sort sampling. See <http://ws13.fpl.fs.fed.us/ttconf.html> for a web interface to the simulation program.

Further, a sophisticated user can sometimes become frustrated with the inflexibility of a commercial package. This can be particularly true if the user is confronted with unbalanced data or complex hypotheses. In addition, some commercial linear models packages do not include the ability to perform power calculations.

In such cases the user can make use of public domain computer routines that yield flexible linear model capabilities. In this note we step potential users through the computations needed to perform hypothesis tests and power calculations. We follow the theoretical approach of Scheffé (1959). To do the numerical work we make use of the singular value decomposition (see, for example, Thisted (1988)). There are, of course, other numerical techniques that can be used to perform the necessary calculations (see, for example, Kennedy and

Gentle (1980), Gentle (1998)). We focus on the singular value decomposition because it yields an approach that is numerically stable, reasonably efficient, and simple to explain and implement. We also suggest the use of the DCDFLIB public domain package of distribution routines.

The relation between the singular value decomposition, least squares, generalized inverses, and estimability has been discussed in Good (1969) and Eubank and Webster (1985).

Hypothesis Testing

The standard linear model is

$$\mathbf{y} = \mathbf{X}\beta + \epsilon,$$

where \mathbf{y} is the $n \times 1$ vector of responses, \mathbf{X} is the $n \times p$ design matrix, β is the $p \times 1$ parameter vector, and ϵ is the $n \times 1$ vector of random errors. We assume that the ϵ_i are independent, identically distributed $N(0, \sigma^2)$ random variables.

We want to test a hypothesis of the form

$$\begin{aligned} \mathbf{c}_1^T \beta &= \eta_1 \\ \mathbf{c}_2^T \beta &= \eta_2 \\ &\vdots \\ \mathbf{c}_q^T \beta &= \eta_q. \end{aligned} \tag{1}$$

(It is often the case that $\mathbf{c}_i^T \mathbf{1} = 0$, in which case \mathbf{c}_i^T is referred to as a "contrast.")

For example, in a one-way ANOVA we are testing

$$\begin{aligned} (1 \ -1 \ 0 \ 0 \ \dots \ 0) \beta &= 0 \\ (1 \ 0 \ -1 \ 0 \ \dots \ 0) \beta &= 0 \\ &\vdots \\ (1 \ 0 \ 0 \ \dots \ 0 \ -1) \beta &= 0. \end{aligned}$$

To test hypothesis (1) we proceed in a series of steps:

Is the Hypothesis Overspecified?

We need to determine whether the \mathbf{c}_i are linearly independent. (If they are not, the hypothesis is overspecified. The user needs to think more clearly about the hypothesis, and arrive at a set of \mathbf{c}_i 's that are linearly independent.)

Let

$$\mathbf{C}_{p \times q} = (\mathbf{c}_1 \dots \mathbf{c}_q)$$

where $q \leq p$. The singular value decomposition of \mathbf{C} is

$$\begin{aligned} \mathbf{C} &= \mathbf{U}_{p \times q} \begin{pmatrix} \gamma_1 & 0 & \dots & 0 & 0 \\ 0 & \gamma_2 & 0 & \dots & 0 \\ & & \vdots & & \\ 0 & 0 & \dots & 0 & \gamma_q \end{pmatrix} \mathbf{V}_{q \times q}^T \\ &= \mathbf{U} \mathbf{D}_\gamma \mathbf{V}^T, \end{aligned}$$

where the columns of \mathbf{U} are orthonormal to each other, \mathbf{V} is an orthogonal matrix, and \mathbf{D}_γ is the diagonal matrix with $\gamma = \gamma_1 \geq \gamma_2 \geq \dots \geq \gamma_q \geq 0$, the singular values of \mathbf{C} , as its diagonal. Thus the rank of \mathbf{C} is just the number of nonzero singular values. Because of the limitations of computer arithmetic, the null γ_i 's will not in general be exactly equal to zero. We need to determine a threshold value. If a γ_i lies below that threshold, we take it to be equal to zero and conclude that the hypothesis is overspecified. A threshold value that is suggested in the numerical analysis literature (for example, Golub and Van Loan (1996)) is

$$\|\mathbf{C}\| \times (\text{the machine precision}).$$

Our experience suggests that

$$\text{threshold} = \|\mathbf{C}\| \times (\text{the machine precision}) \times 10$$

is a better rule of thumb. Recall that

$$\begin{aligned} \|\mathbf{C}\| &= \sqrt{\sum_{i=1}^p \sum_{j=1}^q c_{ij}^2} = \sqrt{\text{trace}(\mathbf{C}\mathbf{C}^T)} \\ &= \sqrt{\text{trace}(\mathbf{U}\mathbf{D}\mathbf{V}^T\mathbf{V}\mathbf{D}\mathbf{U}^T)} \\ &= \sqrt{\text{trace}(\mathbf{U}\mathbf{D}^2\mathbf{U}^T)} \\ &= \sqrt{\text{trace}(\mathbf{D}^2\mathbf{U}^T\mathbf{U})} = \sqrt{\gamma_1^2 + \dots + \gamma_q^2}. \end{aligned}$$

For double precision arithmetic on 32 bit computers one would use

$$\sqrt{\gamma_1^2 + \dots + \gamma_q^2} \times 10^{-15}$$

as the threshold value.

Are the \mathbf{c}_i 's Estimable?

We need to know whether there exists an \mathbf{a}_i that satisfies

$$E(\mathbf{a}_i^T \mathbf{y}) = \mathbf{c}_i^T \beta$$

for all β where E is the expectation operator. This holds true if and only if there exists an \mathbf{a}_i that satisfies

$$\mathbf{a}_i^T \mathbf{X} = \mathbf{c}_i^T \text{ or } \mathbf{c}_i = \mathbf{X}^T \mathbf{a}_i. \quad (2)$$

We determine whether equation (2) holds by first obtaining a singular value decomposition of the design matrix, \mathbf{X} .

$$\mathbf{X}_{n \times p} = \mathbf{U}_{n \times r} \mathbf{\Lambda} \mathbf{V}_{r \times p}^T \quad (3)$$

where the λ 's are nonzero ($r \leq p$ is the rank of \mathbf{X}), or

$$\mathbf{X}^T = \mathbf{V} \mathbf{\Lambda} \mathbf{U}^T.$$

Thus the projection operator onto the space spanned by the columns of \mathbf{X}^T , $L(\mathbf{X}^T)$, is $\mathbf{V}\mathbf{V}^T$, and $\mathbf{c}_i = \mathbf{X}^T \mathbf{a}_i$ for some \mathbf{a}_i if and only if $\mathbf{c}_i \in L(\mathbf{X}^T)$ or $\mathbf{V}\mathbf{V}^T \mathbf{c}_i = \mathbf{c}_i$. Using double precision arithmetic on 32 bit machines, one would take them to be equal if

$$\|\mathbf{V}\mathbf{V}^T \mathbf{c}_i - \mathbf{c}_i\| \leq \|\mathbf{c}_i\| \times 10^{-15}.$$

Find the \mathbf{a}_i that Satisfies $\mathbf{a}_i^T \mathbf{X} = \mathbf{c}_i^T$

Given that \mathbf{c}_i is estimable, there is a unique \mathbf{a}_i in the linear span of the columns of \mathbf{X} , $L(\mathbf{X})$, such that

$$\mathbf{a}_i^T \mathbf{X} = \mathbf{c}_i^T.$$

(This is one of Scheffé's lemmas.)

Using the singular value decomposition of \mathbf{X} , equation (3), we know that the solution of $\mathbf{c}_i = \mathbf{X}^T \mathbf{a}_i$ satisfies

$$\mathbf{c}_i = \mathbf{V} \mathbf{\Lambda} \mathbf{U}^T \mathbf{a}_i$$

or

$$\mathbf{U} \mathbf{\Lambda}^{-1} \mathbf{V}^T \mathbf{c}_i = \mathbf{U} \mathbf{U}^T \mathbf{a}_i. \quad (4)$$

But because $\mathbf{a}_i \in L(\mathbf{X})$ and $\mathbf{U}\mathbf{U}^T$ is the projection operator onto $L(\mathbf{X})$, from equation (4) we have

$$\mathbf{U} \mathbf{\Lambda}^{-1} \mathbf{V}^T \mathbf{c}_i = \mathbf{U} \mathbf{U}^T \mathbf{a}_i = \mathbf{a}_i.$$

Thus, we can use the left-hand side of the equation above to calculate the \mathbf{a}_i that satisfies $\mathbf{a}_i^T \mathbf{X} = \mathbf{c}_i^T$.

Find an Orthonormal Basis for the Linear Span of $\mathbf{a}_1, \dots, \mathbf{a}_q$ — $L(\mathbf{A})$

We simply perform a singular value decomposition of $\mathbf{A} = (\mathbf{a}_1 \dots \mathbf{a}_q)$ to obtain

$$\mathbf{A}_{n \times q} = \mathbf{U}_{n \times q} \mathbf{\Delta} \mathbf{V}_{q \times q}^T$$

and the columns of \mathbf{U} form an orthonormal basis for $L(\mathbf{A})$.

Find the Hypothesis Sum of Squares

Case 1 — $\eta_1 = \dots = \eta_q = 0$ Let $\mathbf{u}_{A,1}, \dots, \mathbf{u}_{A,q}$ be an orthonormal basis for $L(\mathbf{A})$ found as described in the preceding subsection. Scheffé's theory tells us that the hypothesis sum of squares is

$$HSS \equiv \sum_{i=1}^q (\mathbf{u}_{A,i}^T \mathbf{y})^2$$

with q degrees of freedom.

Case 2 — at least one of the η 's is nonzero

Scheffé's theory tells us that the hypothesis sum of squares is

$$HSS \equiv (\mathbf{A}^T \mathbf{y} - \boldsymbol{\eta})^T (\mathbf{A}^T \mathbf{A})^{-1} (\mathbf{A}^T \mathbf{y} - \boldsymbol{\eta}).$$

with q degrees of freedom.

We can make use of the singular value decomposition of \mathbf{A} given above to obtain

$$(\mathbf{A}^T \mathbf{A})^{-1} = \mathbf{V} \Delta^{-2} \mathbf{V}^T. \quad (5)$$

Then taking

$$\mathbf{w} \equiv \mathbf{V}^T (\mathbf{A}^T \mathbf{y} - \boldsymbol{\eta}),$$

we have

$$HSS = \sum_{i=1}^q w_i^2 / \delta_i^2.$$

Find the Residual Sum of Squares

Let $\mathbf{u}_{X,1}, \dots, \mathbf{u}_{X,r}$ be the columns of the \mathbf{U} matrix of the singular value decomposition, equation (3), of \mathbf{X} . Then the projection of \mathbf{y} onto the linear span of the columns of \mathbf{X} , $P_{L(\mathbf{X})}(\mathbf{y})$, equals

$$((\mathbf{u}_{X,1})^T \mathbf{y}) \mathbf{u}_{X,1} + \dots + ((\mathbf{u}_{X,r})^T \mathbf{y}) \mathbf{u}_{X,r}$$

and the residual sum of squares equals

$$\begin{aligned} RSS &\equiv (\mathbf{y} - P_{L(\mathbf{X})}(\mathbf{y}))^T (\mathbf{y} - P_{L(\mathbf{X})}(\mathbf{y})) \\ &= \mathbf{y}^T \mathbf{y} - ((\mathbf{u}_{X,1})^T \mathbf{y})^2 - \dots - ((\mathbf{u}_{X,r})^T \mathbf{y})^2 \end{aligned}$$

with $n - r$ degrees of freedom.

Form the F Statistic and Compare It to the Appropriate Critical Values

The F statistic equals

$$(HSS/q) / (RSS/(n - r)).$$

This should be compared to an $F_{q,n-r}(1 - \alpha)$ critical value where α is the significance level.

Confidence Intervals

Suppose that we are interested in confidence intervals on l estimable combinations of the parameters, $\mathbf{c}_1^T \boldsymbol{\beta}$, \dots , $\mathbf{c}_l^T \boldsymbol{\beta}$, and further suppose that the linear span of the \mathbf{c} 's has rank $q \leq l$. This rank can be determined as described in section 2.1. Let $s^2 \equiv RSS/(n - r)$. Then (see Scheffé (1959)) we know that an individual $(1 - \alpha) \times 100\%$ confidence interval for $\mathbf{c}_j^T \boldsymbol{\beta}$ is

$$\mathbf{a}_j^T \mathbf{y} \pm s \times t_{n-r}(\alpha/2) \times \sqrt{\mathbf{a}_j^T \mathbf{a}_j}$$

where $t_{n-r}(\alpha/2)$ is the appropriate t critical value, and \mathbf{a}_j satisfies $\mathbf{a}_j^T \mathbf{X} = \mathbf{c}_j^T$ (see section 2.3). Also, joint

$(1 - \alpha) \times 100\%$ confidence intervals for the $\mathbf{c}_j^T \boldsymbol{\beta}$'s, $j \in \{1, \dots, l\}$, are given by

$$\mathbf{a}_j^T \mathbf{y} \pm s \sqrt{q \times F_{q,n-r}(1 - \alpha) \times \mathbf{a}_j^T \mathbf{a}_j}$$

where $F_{q,n-r}(1 - \alpha)$ is the appropriate F critical value.

Power Calculations

The noncentrality parameter is given by

$$\begin{aligned} NCP &\equiv (\mathbf{A}^T \mathbf{X} \boldsymbol{\beta} - \boldsymbol{\eta})^T (\mathbf{A}^T \mathbf{A})^{-1} (\mathbf{A}^T \mathbf{X} \boldsymbol{\beta} - \boldsymbol{\eta}) / \sigma^2 \\ &= (\mathbf{C}^T \boldsymbol{\beta} - \boldsymbol{\eta})^T (\mathbf{A}^T \mathbf{A})^{-1} (\mathbf{C}^T \boldsymbol{\beta} - \boldsymbol{\eta}) / \sigma^2. \quad (6) \end{aligned}$$

Note that $(\mathbf{A}^T \mathbf{A})^{-1}$ can be calculated as in equation (5).

Scheffé's noncentrality parameter equals the square root of our noncentrality parameter. Our version corresponds with what the DCDFLIB library (see below) expects.

In a completely general power calculation program, possible values of $\boldsymbol{\beta}$, $\boldsymbol{\eta}$, and σ^2 would be specified by a user in the course of power calculations. In the common case in which $\boldsymbol{\eta} = \mathbf{0}$ and the $\boldsymbol{\beta}$'s represent treatment means, it might be more reasonable to expect a user to specify the components of the $\boldsymbol{\beta}$ vector as fractions of an overall mean and to specify a range of coefficients of variation. This would yield $\boldsymbol{\beta}/\sigma$ values that would enable one to calculate NCP .

Under the null hypothesis, $\mathbf{C}^T \boldsymbol{\beta} - \boldsymbol{\eta} = \mathbf{0}$ and the noncentrality parameter is 0, but under the alternative hypothesis $\mathbf{C}^T \boldsymbol{\beta} \neq \boldsymbol{\eta}$ and the noncentrality parameter is inflated above zero. If we operate at an α significance level, the power is given by

$$\text{Power} = \text{Prob}(F_{q,n-r,NCP} > F_{q,n-r}(1 - \alpha)), \quad (7)$$

the probability that a noncentral F distribution with q numerator degrees of freedom, $n - r$ denominator degrees of freedom, and noncentrality parameter NCP lies above the $100(1 - \alpha)$ th percentile of a central F distribution with q numerator degrees of freedom and $n - r$ denominator degrees of freedom.

Implementing the theory as code

Public domain FORTRAN or C code to perform the singular value decomposition is found in the LINPACK (Dongarra and others (1979)) or (C)LAPACK (Anderson and others (1995)) linear algebra libraries. These can be obtained over the internet at <http://www.netlib.org/liblist.html>. Public domain C++ code to perform the singular value decomposition can be found by searching on svd at

<http://www.netlib.org>. A Java translation of the LINPACK singular value decomposition can be found at http://www1.fpl.fs.fed.us/linear_algebra.html. This site also points to Java translations of the LAPACK linear algebra routines.

Public domain FORTRAN or C code to calculate the t distribution and the central and noncentral F distributions and their inverses can be found in the DCDFLIB library. DCDFLIB is a public domain library of “routines for cumulative distribution functions, their inverses, and their parameters.” It was produced by Barry Brown, James Lovato, and Kathy Russell of the Department of Biomathematics, M.D. Anderson Cancer Center, The University of Texas. DCDFLIB can be found at http://odin.mdacc.tmc.edu/anonftp/page_2.html.

A relatively raw example of the use of the LINPACK and DCDFLIB routines to create a linear models program can be obtained (or run over the Web) at <http://www1.fpl.fs.fed.us/glm.html>. This site includes sample input and output based on Table 9.1 in Milliken and Johnson (1992). We note that linear model routines produced by these methods would need extra work to become user friendly. In particular we have finessed the issue of the generation or input of the design, \mathbf{X} , and hypotheses, \mathbf{C}^T , matrices. We would expect a sophisticated user interested in simulations or special purpose analyses to be able to generate these matrices by hand. A person new to this approach might want to take a look at some of the examples in Milliken and Johnson (1992).

An algorithm that automatically generates design matrices for balanced factorial experiments is described in MacKenzie and O’Flaherty (1982).

Kennedy and Gentle (1980) (page 388) note:

[User friendly] computer software must include the ability to accept user specification of the model. Most programs in use today allow the user to provide some rather natural algebraic specification. The program then deciphers the specification and translates it into numeric coding for subsequent use. There are no established standards for doing this, but many techniques used in compiler construction can be applied to this problem.

An example of such an algebraic specification is discussed in Wilkinson and Rogers (1973).

References

- Anderson, E., Bai, Z., Bischof, C., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Ostrouchov, S., and Sorensen, D. (1995). *LAPACK Users’ Guide, Second Edition*. Society of Industrial and Applied Mathematics, Philadelphia.
- Dongarra, J., Moler, C., Bunch, J., and Stewart, G. (1979). *LINPACK Users’ Guide*. Society of Industrial and Applied Mathematics, Philadelphia.
- Eubank, R. L. and Webster, J. T. (1985). “The Singular-Value Decomposition as a Tool for Solving Estimability Problems,” *The American Statistician*, **39**, 64–66.
- Golub, Gene H. and Van Loan, Charles F. (1996). *Matrix Computations, Third Edition*. The Johns Hopkins University Press, Baltimore and London.
- Gentle, J. E. (1998). *Numerical Linear Algebra for Applications in Statistics*. Springer-Verlag, New York.
- Good, I. J. (1969). “Some Applications of the Singular Decomposition of a Matrix,” *Technometrics*, **11**, 823–831.
- Kennedy, W. J. and Gentle, J. E. (1980). *Statistical Computing*. Marcel Dekker, New York and Basel.
- MacKenzie, G. and O’Flaherty, M. (1982). “Direct Design Matrix Generation for Balanced Factorial Experiments,” *Applied Statistics*, **31**, 74–80.
- Milliken, G. A. and Johnson, D. E. (1992). *Analysis of Messy Data, Volume I: Designed Experiments*. Chapman and Hall, London.
- Scheffé, H. (1959). *The Analysis of Variance*. John Wiley and Sons, New York.
- Thisted, R. A. (1988). *Elements of Statistical Computing*. Chapman and Hall, London.
- Verrill, S. (1999). “When Good Confidence Intervals Go Bad: Predictor Sort Experiments and ANOVA,” *The American Statistician*, **53**, 38–42.
- Wilkinson, G. N. and Rogers, C. E. (1973). “Symbolic Description of Factorial Models for Analysis of Variance,” *Applied Statistics*, **22**, 392–399.

Correspondence Analysis with R

Brigitte Charnomordic, Biométrie, INRA, France and

Susan Holmes, Statistics Department, Stanford
susan@stat.stanford.edu

Abstract

We provide here a few didactic examples for understanding correspondence analysis using R to illustrate the use of matrix decompositions. We show an example of how correspondence analysis can help analyze difficult multidimensional data such as DNA micro-array data.

Introduction

In France, students in PhD or Masters programs in applied statistics have to take a course on correspondence analysis. In English speaking countries this method is only taught in some education departments¹, and there are very few books or articles in English on the subject, Michael Greenacre who studied in France under Benzecri[1] has written several very readable texts on the subject[5].

With today's explosion of data in the form of contingency tables both two-way and three-way, this method is very useful in applications such as gene expression microarray data, information retrieval, linguistics and sociology.

Correspondence analysis provides students at the masters level in statistics, applied mathematics, biology and engineering with an opportunity for performing multivariate data analysis without having to use prepackaged programs. Thus they also learn how to use graphical and numerical functions in high level languages such as R/Splus or matlab.

Aims and relevant data

We can make a dichotomy of data-mining and multivariate statistical methods into two groups, one series of methods confers a particular status to one variable or set of variables, these are to be predicted or explained, they are the response. Methods include regression, multiple response regression, discriminant analysis, analysis of variance depending on whether the explanatory variables are categorical or continuous. These are not the ones we are going to study here.

Correspondence analysis is useful when all the variables have the same status. This is sometimes called unsuper-

vised learning, and includes clustering, principal components as well. They have in common the creation of a new set of variables that simplify the arrays at hand. In the case of clustering, the new variable is a categorical, in correspondence analysis and principal components the new variables are continuous and enable the construction of useful new graphical representations of the data.

Correspondence analysis is an exploratory method because it does not presuppose any model for the data, as do Goodman's bilinear methods or factor analysis models for instance. Correspondence analysis and principal components can both be extended to three-way arrays, for instance for the analysis of bootstrap permutation tests or time series of matrices. Such data are often called data cubes.

Correspondence Analysis

Correspondence analysis (CA, also called homogeneity analysis and reciprocal averaging), can be used to analyze several types of multivariate data. All involve some categorical variables. Here are some examples of the type of data that can be decomposed using this method:

- Contingency Tables (cross between two categorical variables)
- Multiple Contingency Tables (cross between several categorical variables).
- Binary tables obtained by cutting continuous variables into classes and then recoding both these variables and any extra categorical variables into 0/1 tables, 1 indicating presence in that class. So for instance a continuous variable cut into three classes will provide three new binary variables of which only one can take the value 1 for any given observation.

To first approximation, correspondence analysis can be understood as an extension of principal components analysis (PCA) where the variance in PCA is replaced by an inertia proportional to the χ^2 distance of the table from independence. CA decomposes this measure of departure from independence along axes that are orthogonal according to the χ^2 inner product. If we are comparing two categorical variables, the simplest possible model is that of independence in which case the counts in the table would obey approximately the margin products identity for a $m \times p$ contingency table with a total sample size of $n = \sum_{i=1}^m \sum_{j=1}^p n_{ij} = n \dots$. Independence means

$$n_{ij} \doteq \frac{n_{i.} n_{.j}}{n}$$

¹ Exception made for a few statistics departments, for instance UCLA where Jan de Leeuw teaches multivariate statistics.

can also be written: $\mathbf{N} \doteq \mathbf{c}\mathbf{r}'\mathbf{n}$, where

$$\mathbf{c} = \frac{1}{n}\mathbf{N}\mathbf{1}_m \text{ and } \mathbf{r}' = \frac{1}{n}\mathbf{N}'\mathbf{1}_p$$

The departure from independence is measured by the χ^2 statistic

$$\chi^2 = \sum_{i,j} \left[\frac{(n_{ij} - \frac{n_{i.} \cdot n_{.j}}{n})^2}{\frac{n_{i.} \cdot n_{.j}}{n}} \right]$$

Example: Eye color -Hair color

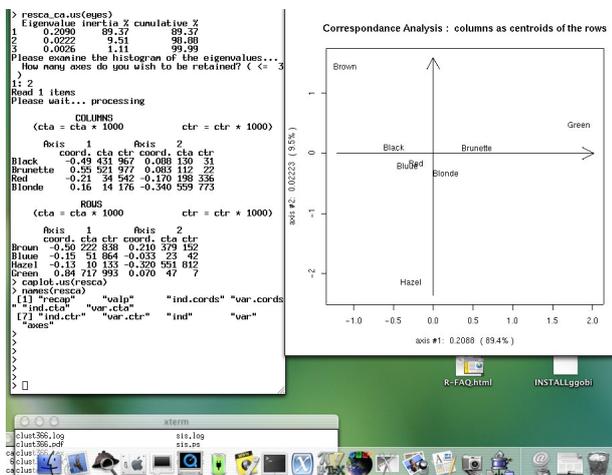
Here is a simple contingency table as our example from Snee (1974)[8].

eyes	Black	Brunette	Red	Blonde
Brown	68	20	15	5
Blue	119	84	54	29
Hazel	26	17	14	14
Green	7	94	10	16

```
> chisq.test(eyes)
Pearson's Chi-squared test
data:  eyes, X-squared = 138.2898,
df = 9, p-value = < 2.2e-16
```

This is a very extreme point in the χ^2 distribution. The inertia is the χ -squared statistic divided by the number of observations (sum(eyes)); Here, the inertia is $138.3/592 = 0.2336$. CA decomposes this inertia into the sum of eigenvalues of a symmetrized reweighted version of the original table, as will be explained below. The R command `resca<-ca.us(eyes)` provides first a scree plot of these eigenvalues, in this example:

	Values	Percent	
1	0.2088	89.	*****
2	0.0222	10.	**
3	0.0026	1.	.
Total	0.2336	100.	



R screenshot on Mac OS/X

Formulation as a generalized singular value decomposition

Given an $m \times p$ contingency table of counts \mathbf{N} of m levels for a row variable and p levels for a column variable. (This is equivalent to a binary matrix X with $n = \sum_{ij} n_{ij} = n_{..}$ observations on $m + p$ columns, a notion that is useful of the generalization later.)

The first transformation makes the contingency matrix \mathbf{N} into a frequency matrix $\mathbf{F} = \frac{1}{n}\mathbf{N}$. We will denote the row sums by $\mathbf{r} = \mathbf{F}\mathbf{1}_p$ and the column sums by the vector $\mathbf{c} = \mathbf{F}'\mathbf{1}_m$. These both sum to one

$$\mathbf{r}'\mathbf{1}_m = \mathbf{c}'\mathbf{1}_p = 1$$

In the case of independence

$$\mathbf{F} \doteq \mathbf{r}\mathbf{c}'$$

All the rows would be multiples of each other or as this is sometimes called, *homogeneous*. So, if all the rows were divided by the weight of that row, these so-called *row profiles* $\mathbf{F}\mathbf{D}_r^{-1}$ would be equal ($\mathbf{F}\mathbf{D}_r^{-1} = \mathbf{1}_m\mathbf{c}$), where \mathbf{D}_r^{-1} denotes the diagonal matrix with the vector \mathbf{r}^{-1} on its diagonal.

The average row in the case of homogeneity and independence is obtained by averaging the rows with the relevant weights for each column. The average of the row profiles is \mathbf{c} . The departure from independence and homogeneity is measured by some norm of $\mathbf{F}\mathbf{D}_r^{-1} - \mathbf{1}_m\mathbf{c}$ (or at the term by term level $\frac{f_{ij}}{r_i} - c_j$). With this notation we remark that

$$\begin{aligned} \chi^2 &= n \sum_{i,j} \frac{(f_{ij} - r_i c_j)^2}{r_i c_j} \\ &= n \sum_{i,j} r_i c_j \left(\frac{f_{ij}}{r_i c_j} - 1 \right)^2 \end{aligned}$$

Verification in R:

```
> F<-eyes/sum(eyes)
> r<-apply(F,1,sum)
> c<-apply(F,2,sum)
> E<-outer(r,c)
> sum((F-E)^2/E)*592
[1] 138.2898
```

To compute the distance between profiles, each column is reweighted by the inverse of its sum, this gives the χ^2 distance between row profiles.

$$\begin{aligned} \chi^2 &= n \text{ trace } ((\mathbf{F} - \mathbf{r}\mathbf{c}')'\mathbf{D}_r^{-1}(\mathbf{F} - \mathbf{r}\mathbf{c}')\mathbf{D}_c^{-1}) \\ &= \text{trace } (\mathbf{A}'\mathbf{A}) \text{ where } \mathbf{A} = \mathbf{D}_r^{-1/2}(\mathbf{F} - \mathbf{r}\mathbf{c}')\mathbf{D}_c^{-1/2} \end{aligned}$$

The latter decomposition shows a justification for choosing the matrix \mathbf{A} as a natural square root. $\mathbf{W} = \mathbf{A}'\mathbf{A}$ is in a sense the characteristic matrix-operator of

the analysis, in the same way the covariance or correlation matrices are those of principal components analysis. Generalizing principal component analysis to include metrics on the rows and the columns can lead to other multivariate techniques such as discriminant analysis (See Mardia, Kent and Bibby(1979) [7]).

Correspondence analysis decomposes the matrix \mathbf{W} : its eigenvectors give the axes that account for the largest part of the departure from independence, just as principal components provides the axes accounting for the largest variability. Computationally this is achieved by a generalized singular value decomposition

$$\mathbf{D}_r^{-1}\mathbf{F}\mathbf{D}_c^{-1} - \mathbf{1}'_m\mathbf{1}_p = \mathbf{U}\mathbf{S}\mathbf{V}'$$

with $\mathbf{V}'\mathbf{D}_c\mathbf{V} = \mathbf{I}_p, \mathbf{U}'\mathbf{D}_r\mathbf{U} = \mathbf{I}_m$

equivalent to the eigendecomposition $\mathbf{W} = \mathbf{A}'\mathbf{A} = \mathbf{V}'\mathbf{S}^2\mathbf{V}$ or the singular value decomposition

$$\mathbf{D}_r^{-\frac{1}{2}}\mathbf{F}\mathbf{D}_c^{-\frac{1}{2}} - \sqrt{\mathbf{r}}\sqrt{\mathbf{c}}' = (\mathbf{D}_r^{\frac{1}{2}}\mathbf{U})\mathbf{S}(\mathbf{D}_c^{\frac{1}{2}}\mathbf{V})'$$

where $(\mathbf{D}_c^{\frac{1}{2}}\mathbf{V})'(\mathbf{D}_c^{\frac{1}{2}}\mathbf{V}) = \mathbf{I}_p$, and $(\mathbf{D}_r^{\frac{1}{2}}\mathbf{U})'(\mathbf{D}_r^{\frac{1}{2}}\mathbf{U}) = \mathbf{I}_p$. The CA plots can be used to find out if there is an hidden ordination of the data, as for instance the chronological seriation studied below.

Finding an underlying ranking.

Example: Cox and Brandwood [3] tried to seriate Plato's works using discriminant analysis on the proportion of sentence endings in a given book, with a given stress pattern. Here we show how such an analysis can be done with correspondence analysis on the table of frequencies of sentence endings². The first 10 profiles (as percentages) look as follows:

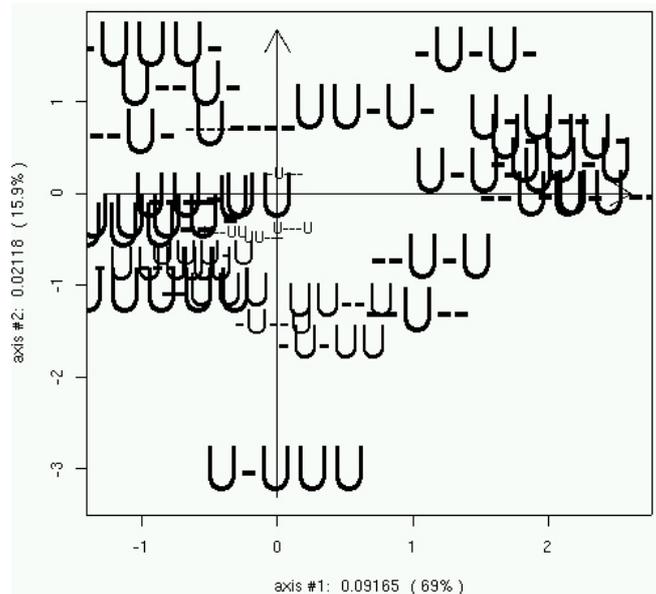
	Rep	Laws	Crit	Phil	Pol	Soph	Tim
UUUUU	1.1	2.4	3.3	2.5	1.7	2.8	2.4
-UUUU	1.6	3.8	2.0	2.8	2.5	3.6	3.9
U-UUU	1.7	1.9	2.0	2.1	3.1	3.4	6.0
UU-UU	1.9	2.6	1.3	2.6	2.6	2.6	1.8
UUU-U	2.1	3.0	6.7	4.0	3.3	2.4	3.4
UUUU-	2.0	3.8	4.0	4.8	2.9	2.5	3.5
--UUU	2.1	2.7	3.3	4.3	3.3	3.3	3.4
-U-UU	2.2	1.8	2.0	1.5	2.3	4.0	3.4
-UU-U	2.8	0.6	1.3	0.7	0.4	2.1	1.7
-UUU-	4.6	8.8	6.0	6.5	4.0	2.3	3.3

.....etc (there are 32 rows in all)

The eigenvalue decomposition (called the scree plot) shows that two axes will provide a summary of 85% of the departure from independence.

```
> res.plato_ca.us(platon)
Eigenvalue inertia % cumulative %
1 0.09170 68.96 68.96
2 0.02120 15.94 84.90
3 0.00911 6.86 91.76
4 0.00603 4.53 96.29
5 0.00276 2.07 98.36
6 0.00217 1.64 100.00
Please examine the eigenvalues...
How many axes do you wish to be retained?
```

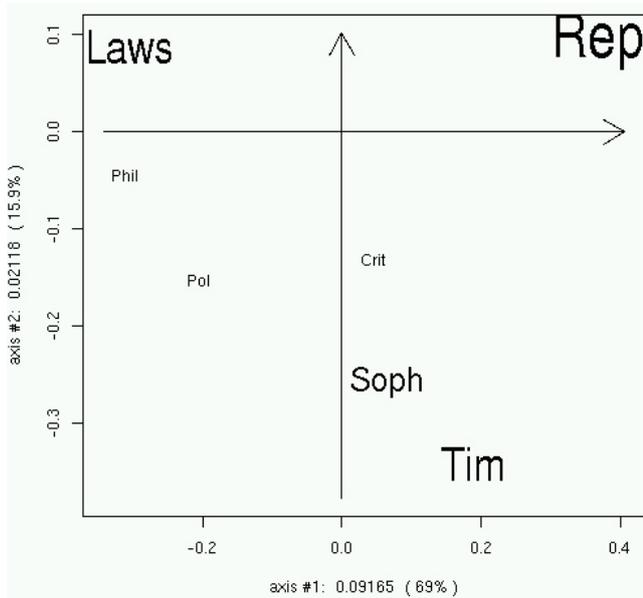
The function `ca.us` asks this question interactively, because only the screeplot visualisation can protect against the separation of 2 or more very close eigenvalues. The function then returns as the resulting data list the coordinates for plotting that can be visualized using `ggobi` if there are more than 3 relevant axes. Here is the correspondence analysis representation for the rows and columns taken separately. We have made the labels' sizes proportional to the quality of the representations in this first plane. These are obtained by typing `caplot.us(res.plato)`.



Plato sentence endings (rows)

The next plot shows very clearly a seriation of all the works, in fact, except for **Critias**, the seriation is determined by the first axis, the second axis helps to place **Critias** between **Politicus** and **Sophist**, a choice that has also been validated in Ledger(1989) and Cox and Brandwood (1959).

²A computerized analysis of Plato's work appears in Ledger (1989)[6]



Plato's works (columns)

Decomposition of Inertia

χ^2 distance between profiles

Here is a reason why such a weighted distance could be useful : Take a very simple contingency table :

$$X = \begin{bmatrix} 6 & 2 & 12 \\ 15 & 5 & 7 \\ 21 & 7 & 42 \end{bmatrix}$$

The row profiles are all

$$\begin{matrix} & [, 1] & [, 2] & [, 3] \\ [1 ,] & 0.3 & 0.1 & 0.6 \\ [2 ,] & 0.3 & 0.1 & 0.6 \\ [3 ,] & 0.3 & 0.1 & 0.6 \end{matrix}$$

We can see that the profiles are identical, the multinomials generating the rows are said to be homogeneous, and the contingency table is in fact only of rank 1:

$$X = \begin{bmatrix} 2 \\ 5 \\ 7 \end{bmatrix} \%* \% \begin{bmatrix} 3 & 1 & 6 \end{bmatrix}$$

This is exactly the problem we encounter in principal components analysis, we need the decomposition in singular values of X , but instead of centering the data with regards to the mean, the data will be centered at independence. Another difference lies with the choice of the metric for computing distances between rows or columns.

Decomposition of the difference from independence

A cloud, or scatter of weighted points :

These are points defined in an euclidean space, say \mathbf{R}^p for instance, so that distances between them are easy

to compute. However we associate to each multidimensional point a weight that changes the inertia of the scatter points. For instance if we have two points the one with a higher weight will 'pull' the center of gravity toward it. The same will happen for the 'minimum inertia' line, it will be pulled toward highly weighted points.

Distributional Equivalence :

If we add two rows that have the same profiles, this will not change the axes chosen to represent the data, (the column profiles' geometry remains unchanged). Thinking of the points as weighted points in a cloud, two points that would be at the same spot can be merged because we can add their weights.

Barycentric Representation :

Take the simplest case: row profiles of a 3-column contingency table. The profiles sum to one so are all representable in a triangle (called the 3-dimensional simplex). The vertices are the extreme profiles, say (1,0,0), (0,1,0) and (0,0,1). Although the row profiles are in a three dimensional space as they belong to this triangle they can be taken out and just looked at in these coordinates, called the barycentric co-ordinate system. Now an extra scale change will bring this representation to the correspondence analysis one : the dimensions will be weighted inversely by the relative weights of the columns , called column mean profiles, (which also add to 1). The distances we want to represent between points are to be the χ^2 distances relevant here, so the sides of the equilateral triangle are stretched to have sides inversely proportional to the square roots of their mean values.

The side of the triangle the most stretched corresponds to the least frequent column.

This representation is the one chosen by default by the function `caplot`, there is a delicate issue of choosing the scales in the two dimensions so that simultaneous representations of rows and columns are valid (the program warns the user when such a scaling has not been chosen). In the relevant choice of scaling, proximities between row and column points are hard to interpret, however it is easier to interpret the directions of the different rows and columns.

Reading the Output

Although the maps provided by doing both correspondence and principal components analysis look quite simple there are traps that lead to *misinterpretations* that must be avoided. Associated to the co-ordinates in the new spaces are what we call *loadings* or *contributions* and which are indicators of how true the proximities

in the image space are. To this end the object that the function `caplot` produce a listed output containing the eigenvalues, coordinates for the rows, for the columns, that are used for building the graphical representations and absolute contributions and squared cosines that are important diagnostic tools.

```
>names(resca)
[1]"recap" "valp" "ind.cords" "var.cords"
[5]"ind.cta" "var.cta" "ind.ctr" "var.ctr"
[9] "ind" "var" "axes"
```

- When trying to understand the most meaningful rows or columns for a given axis we look at the absolute contributions of rows or columns to given axis, this gives the amount of an axis's inertia explained by a single row or column.
- The relative contribution of an axis/ of two axes to the inertia of a row. This is the same as the cosine of the point with the axis that says how well a point is being projected onto the axis.

Contribution³ to the inertia from row i :

Distance from the i th row to the center of the row-points:

$$d_{\chi^2}^2(\text{profile}_i, \text{center}) = \sum_j \frac{1}{f_{\cdot j}} \left(\frac{f_{ij}}{f_{i\cdot}} - f_{\cdot j} \right)^2$$

$$= \sum_j f_{\cdot j} \left(\frac{f_{ij}}{f_{i\cdot} f_{\cdot j}} - 1 \right)^2 = \sum_k (s_k u_i^k)^2$$

This row will thus participate to the inertia by this amount weighted by the row's mass r_i . This can be decomposed into each of the axis separately thus giving an idea of the contribution of each row to the inertia of each axis, this is called the absolute contribution of row i to axis k , $r_i \sum_k (s_k u_i^k)^2$. The sum of all row's contributions to a given axis add to one. This translates the fact that $\mathbf{U}'\mathbf{D}_r\mathbf{U} = \mathbf{I}_m$, here are the absolute contributions for the eyes data:

```
%Output resca$indcta
      Axis 1   Axis 2   Axis 3
Brown   0.2225  0.3788  0.2163
Blue    0.0509  0.0232  0.4428
Hazel   0.0096  0.5513  0.3191
Green   0.7170  0.0467  0.0217
Total    1.      1.      1.
```

Thus we can see that the most important row category for explaining the first axis is Green eyes. The contributions to inertia as decomposed according to the column categories gives:

```
%Output resca$varcta
```

	Axis 1	Axis 2	Axis 3
Black	0.4312	0.1304	0.0668
Brunette	0.5213	0.1124	0.0031
Red	0.0340	0.1980	0.6109
Blonde	0.0135	0.5591	0.3192
Total	1.	1.	1.

Squared Cosine

For interpretation of the exactitude of the projection, it is important to consult the cosine of the angle between the point and it's projection onto the k th axis or the plane or space spanned by the relevant axes.

$$\cos^2(\text{row } i, \text{axe } k) = \frac{(s_k u_{ik})^2}{\sum_k (s_k u_{ik})^2}$$

```
%Output resca$indctr
      Axis 1   Axis 2   Axis 3   Total
Brown   0.838   0.152   0.010   1.
Blue    0.864   0.042   0.094   1.
Hazel   0.133   0.812   0.055   1.
Green   0.993   0.007   0.000   1.
```

In this case of course, taking 3 axes results in a complete reconstruction of the table, so the \cos^2 between the rows and this 3-space is 1.

On the other hand the column's cosine can also be provided, here the output gives:

```
%Output resca$varctr
      Axis 1   Axis 2   Axis 3   Total
Black   0.9670   0.031   0.002   1.
Brunette 0.9775   0.022   0.000   1.
Red     0.5424   0.336   0.121   1.
Blonde  0.1759   0.777   0.052   1.
```

This information can actually be incorporated into the graphic by making the size of the points label proportional to how close the point is from the plane. This is a 'perspective' type plot, points which are close, are well represented and have high cosines. Thus the large letters label points that are close to the plane, and are thus "well-represented".

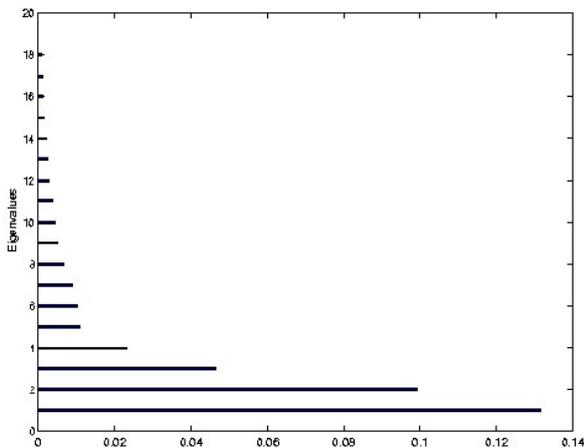
Summing these for the 2 for which the representation was made, and using these indices as the size of the font, enables one to see at a glance which rows and columns are important to interpret.

Complete Analysis of DNA-microarray data

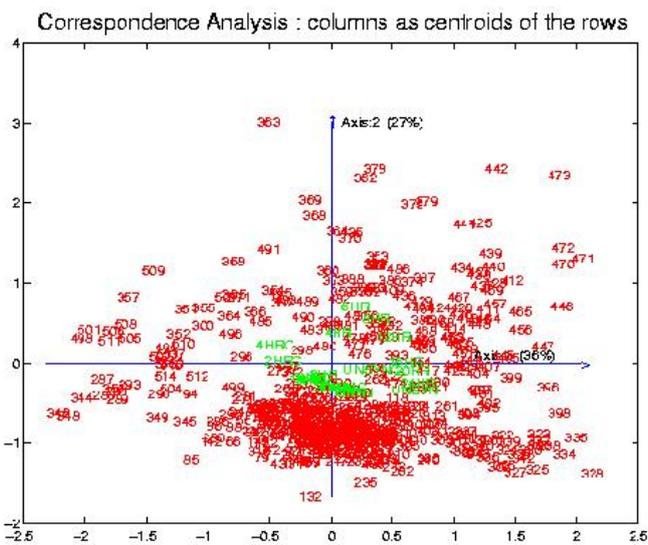
This data is freely available thanks to Iyer et al. (1999)[9], who analyzed this data in their Science magazine article. One of the basic methods they used was hierarchical clustering as very well explained by Carr et al. (1997)[2]. Here we present the results of the

³Sometimes called absolute contribution as different from the \cos^2 which are sometimes called relative contributions.

correspondence analysis alternative. Since this work was done, other implementations of correspondence analysis (and homogeneity analysis) have been published (see the software by Wong and Cui[10] and Feltenberg et al.[4]).

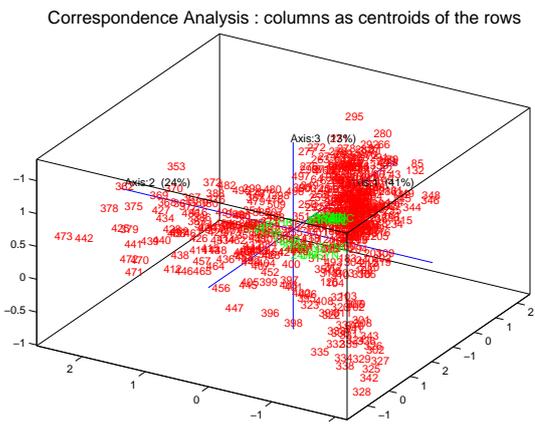


Eigenvalues for the microarray data

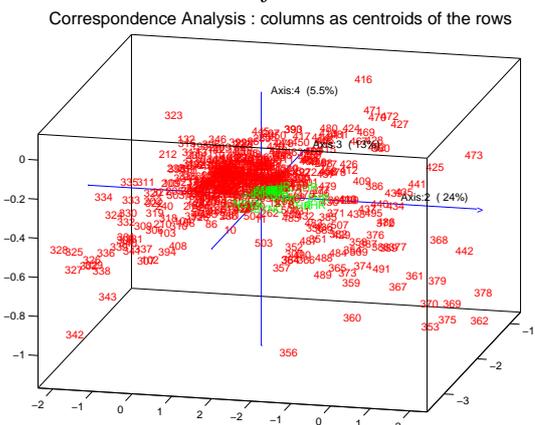


Columns as centroids of rows

The 3-dimensional space spanned by the first three axes explains 78% of the departure from independence of the table. Here are two 3 dimensional views.



Three first axes



Three next axes -rotated

References

- [1] BENZECRI, J. P. History and prehistory of data analysis. V: The analysis of correspondence, systematic index (french), *Cahiers de l'Analyse des Donnees* 2: 9–40, (1977)
- [2] D. CARR, R. SOMOGYI AND G. MICHAELS, *Templates for Looking at Gene Expression Clustering*, Statistical Computing and Graphics Newsletter, vol. 8, pp. 20-29, (1997).
- [3] D. R. COX AND L. BRANDWOOD, *On a discriminatory problem connected with the works of Plato*, J. Roy. Statist. Soc. Ser. B, 21 (1959), pp. 195–200.
- [4] K. FELLEBERG, N. C. HAUSER, B. BRORS, A. NEUTZNER, J. D. HOHEISEL, AND M. VINGRON, *Correspondence analysis applied to microarray data*, PNAS, (2001), pp. 10781-10786.
- [5] M. GREENACRE, *Theory and Applications of Correspondence Analysis*, Academic Press, (1984).
- [6] G.K.LEDGER, *Re-counting Plato*, Oxford University Press, Oxford, (1989).
- [7] K. MARDIA, J. KENT, AND J. BIBBY, *Multivariate Analysis*, Academic Press, (1979).

- [8] R. D. SNEE, *Graphical display of two-way contingency tables*, *The American Statistician*, 28 (1974), pp. 9–12.
- [9] V.R. IYER, M. EISEN, T. ROSS, T. DOUGLAS, G. SCHULER, T. MOORE, J. LEE, J. TRENT, L. STAUDT, J. HUDSON JR., M. BOGUSKI, D. LASHKARI, D. SHALON, D. BOTSTEIN AND P.

BROWN., *The Transcriptional Program in the Response of Human Fibroblasts to Serum*, *Science*, (1999) 283: 83-87.

- [10] W. WONG AND Y. CUI, *Gif Array Analyzer: analyzing microarray data with Homogeneity Analysis*, <http://biowww.dfci.harvard.edu/~ycui/Gif.html>

TEACHING STATISTICAL COMPUTING

Scenarios for Statistics

Susan Holmes, Statistics, Stanford

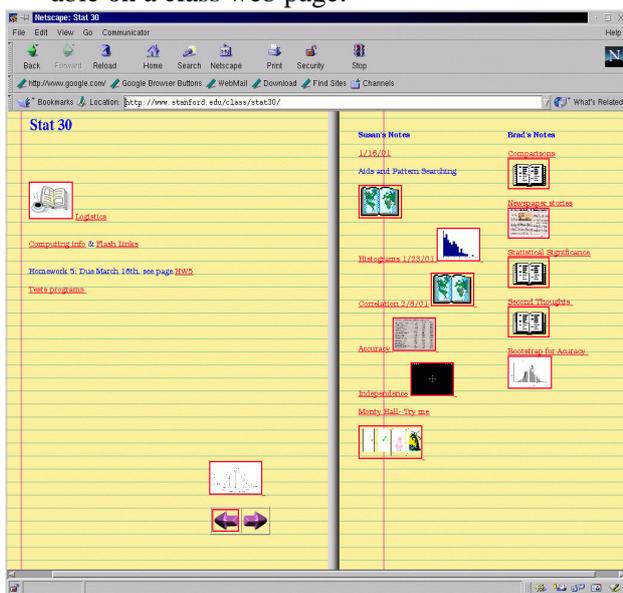
Statistical Computing for Artists

At Stanford this year Brad Efron and I decided to teach an introductory level statistics class aimed at *aesthetes*. The campus daily immediately misnamed the class: “Maths for fuzzies”. We wanted to draw esthetically inclined students to a class that would enable them to read and understand statistics as it appeared in the newspapers, then illustrate what they had learned in class with an artistic media.

Of course there was no calculus requirement, nor an inclination for maths expected. Everything was done with words, but we started the course curriculum backwards from the usual ordering. Starting with the computation of permutation and bootstrap distributions rather than the Normal distribution.

Familiarity with the computer was acquired in stages:

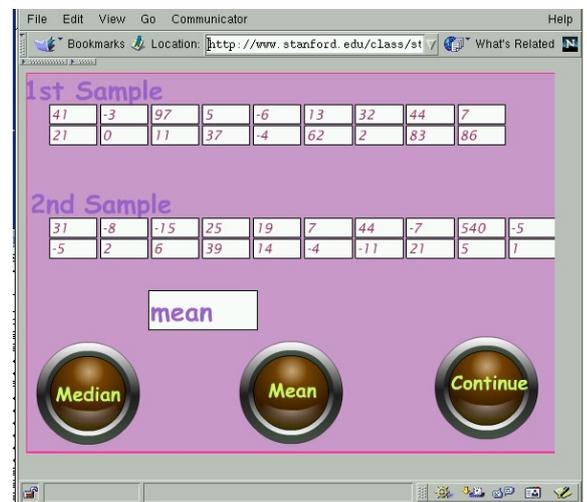
Using the browser The class notes were made available on a class web page:



Downloading and printing pdf files Some of our students did not know how to download or view PDF files.

Running applets There are many existing applets on the web for teaching probability, we used an array of ones from Buffon’s needle to Monty Hall

Using Flash to do some statistical analyses The first permutation tests were executed using a simple press button interface on data already entered into the program:



Interactive Flash Example

Making one simple Flash animation scene This was explained in a lab run by Carrie Grimes, a valuable Teaching Assistant by her prior experience with Flash. An animation used to explain paired comparisons can be accessed at:

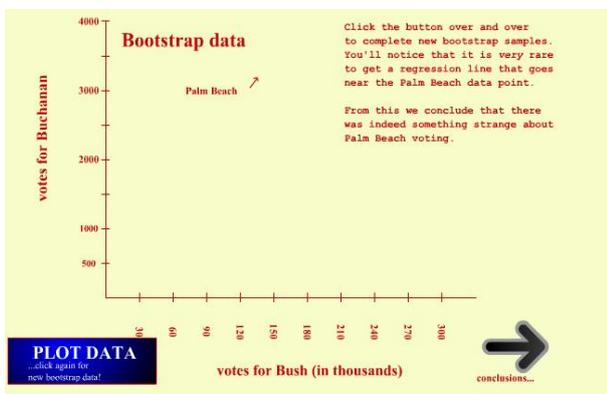


Publishing Flash animations on the web This involves publishing the html files that contains a pointer to the relevant swf files and uploading to the web page. This was much more difficult for our students than we had predicted.

Making an interactive Flash animation This was what the students wanted to learn most of all, the basis for being to make computed animated cartoons, this was their favorite lab!

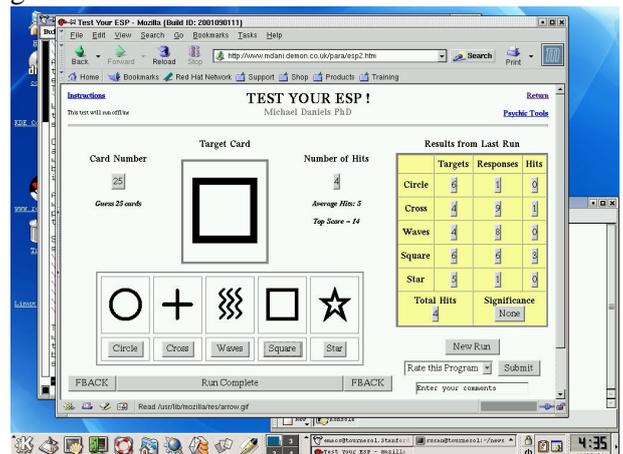
Making a whole Flash movie with several scenes This involved making complete storyboards containing the concepts and how they were illustrated as well as joining the scenes together.

Writing Flash actionscript programs to analyze data Some more ambitious students actually did quite a bit of programming, it is possible to allow users to enter their own data and analyze it, as this bootstrap flash animation, written by Anita Lillie shows:



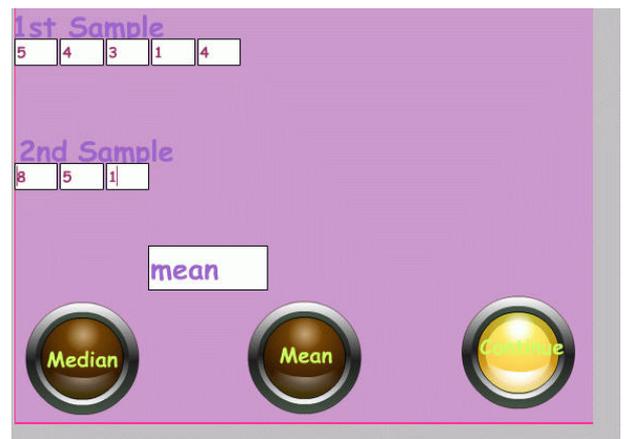
Bootstrap of a Regression(2000 Elections)

Many of the experimental data was collected by the students themselves, for instance by testing their ESP using Michael Daniels' ESP tester



This of course led to binomials, but we experimented with them using simulations more than formulas.

The first two sample testing examples were paired tests done with permutations, and the manipulations that involved the computer were done using scripts in flash actionscript.

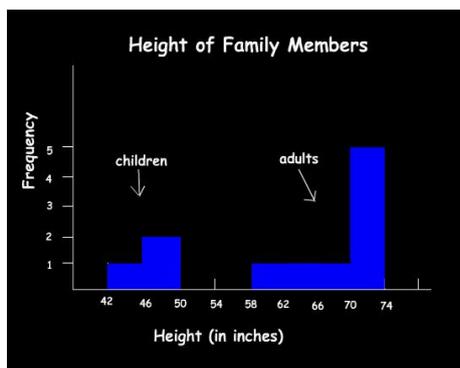


Entering the data

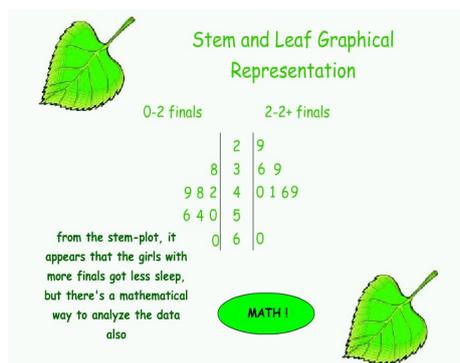
Instead of doing the analytical formula, we generated many samples under the null hypotheses. As we did also for two sample permutation tests, I called the permutations *shuffling* of the two samples.

There is a green shuffle button that just does a permutation. Then the difference has to be computed by clicking buttons, or as the students understand the step by step process, a special automatic button enables them to do blocks of permutations at the time.

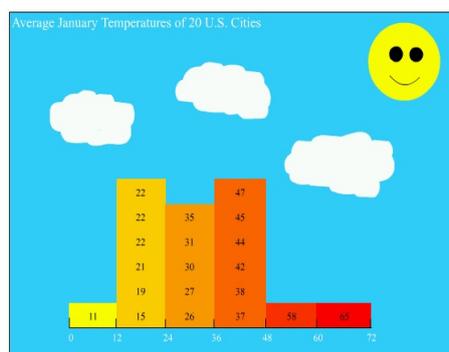
As homeworks and final projects, we asked the students to write their own animations to illustrate some of the concepts we did in class:



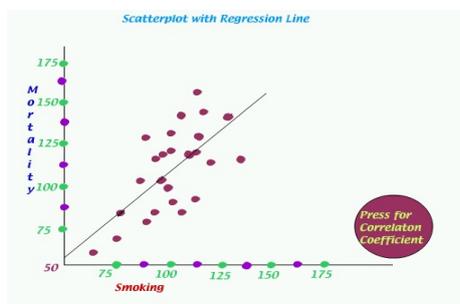
Anita Lillie's Histogram



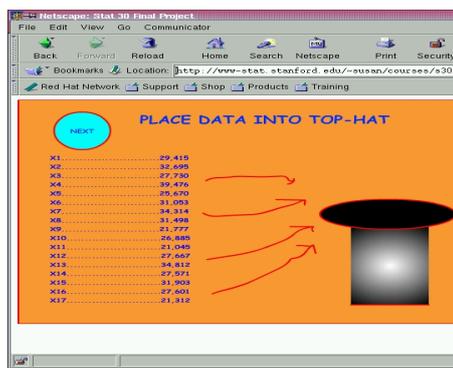
Sara Flores' Stem and leaf



Michael Areinoff's Histogram



Meredith Hoy's regression



Danny Jacobs' Shakespeare Bootstrap

Interactive Flash on the web

First the instructor can write very simple interactive programs to do sets of simulations. Our homework 3 consisted in the comparison of two samples of different sizes with a permutation test, the students had to simply run through the screens pushing a few buttons. This only required knowing about linking to a site on the web, and updating the browser so that the correct plug-in was available. At the time Flash 5 had only recently come out, and this was the version that made doing mathematics possible.

Flash Animations

The students spent 3 labs learning the basics of Flash animation, I can't say it was smooth for all of them there is a lot involved in going from only knowing email to being able to publish an animation. This then has to be uploaded to their own website. A lot of small steps that aren't so easy for an audience of technophobes. However the motivation of actually learning flash and seeing their artwork dance across the screen was very strong.

Flash Actionscripts

Here are some of the possibilities that were added with version 5 of flash and that has made mathematical functions possible:

Math.random, Math.floor, Math.round, Math.sin, Math.cos, Math.abs etc...

To see an example of what a script actually looks like, see the examples at Sample for building a Pie chart and Sample Actionscript for a 3d rotator.

A summary of the experiment

None of these students wanted to use spreadsheets or statistical software so Flash was an appealing alternative especially because no other course on campus catered to this arty crowd. The audience was coaxed into using the computer to program because Flash was written originally for designers, with no prior programming necessary at all.

To see for yourselves, an overview of the projects and homeworks, please visit our Student Sampler.

And from here?

Changing the students attitude in an elementary statistics class from passive consumers to active creators can definitely change their experience. Having to make an animation that explains a concept means that not only is the concept understood, but also that students acquire the focus to extract the few important features that need to be shown in an animation. The artistic and creative components kept them interested even when the statistics seemed arduous.

I have noted since working in the United States how education here is focused around personal work. Students thrive on giving their own take on problems and doing personal projects. The course was built to capitalize on this observation.

Maybe others will be interested by using Flash for teaching mathematics or statistics, I would be happy to provide information and advice.

References

- [1] S. BHANGAL,(2000), *Foundation Actionsript*. Friends of ED, Birmingham, UK.
- [2] M. WOOLDRIDGE, (2001), *Teach Yourself Visually Flash 5*, Marangraphics Inc/ Hungry Minds.
- [3] B. EFRON AND SUSAN HOLMES, (2000), *Scenarios for Statistics: Stat30*, Webpage:<http://www-stat.stanford.edu/~susan/courses/s30/>



NEWS CLIPPINGS AND SECTION NOTICES

Pictures from the Joint Mixer at JSM 2001



Thomas Lumley and Debby Swayne being helped choosing prize winners by an unidentified young statistician.



John Chambers awarding the John M. Chambers Statistical Software Award to Dr. Alessandra Brazzale.



SECTION OFFICERS

Statistical Graphics Section - 2001

- Deborah F. Swayne**, Chair
(973) 360-8423
AT&T Labs - Research
dfs@research.att.com
- Stephen G. Eick**, Chair-Elect
(708) 713 5169
Visual Insights,
steve.eick@visualinsights.com
- Edward J. Wegman**, (703) Past-Chair
(703) 993-1680
George Mason University
ewegman@gmu.edu
- David A. James**, Program Chair
(908) 582-3082
Lucent Bell Labs
dj@lucent.com
- Mario Peruggia**, Program-Chair Elect
(614) 292-0963
Ohio State University
peruggia@stat.ohio-state.edu
- Dianne Cook**, Newsletter Editor
(515) 294-8865
Iowa State University
dicook@iastate.edu
- Thomas Lumley**, Secretary/Treasurer
(206) 543-1044
University of Washington
thomas@biostat.washington.edu
- Graham Wills**, Publications/Electronic Commun.
(312) 651-3671
SPSS
gwills@spss.com
- Bradley A. Jones**, Rep.(99-01)Council of Sections
508-368-8458
Mathworks
brad@mathworks.com
- Charles B. Roosen**, Rep.(01-03)Council of Sections
MathSoft, Inc.
roosen@statsci.com

Statistical Computing Section - 2001

- Mark Hansen**, Chair
908-582-3869
Bell Laboratories
cocteau@bell-labs.com
- Susan Holmes**, Chair-Elect
Stanford University
susan@stat.stanford.edu

- Russell D. Wolfinger**, Past-Chair
(919) 677-8000
SAS
sasrdw@sas.com
- Douglas W. Nychka**, Program Chair
(919) 737-2534
NCAR, Geophysical Statistics
nychka@ucar.edu
- Tim Hesterberg**, Program Chair-Elect
(206) 283-8802
Insightful
timh@insightful.com
- Susan Holmes**, Newsletter Editor (00-02)
650-725-1925
Stanford University
susan@stat.stanford.edu
- Merlise Clyde**, Secretary/Treasurer (00-02)
919-681-8440
Duke University
clyde@isds.duke.edu
- John F. Monahan**, Publications Liaison Office
919-737-2541
North Carolina State University
monahan@stat.ncsu.edu
- Thomas F. Devlin**, Electronic Communication Liaison
(973) 655-7244
Montclair State University
devlin@mozart.montclair.edu
- Lionel Galway**, Awards Officer
(310) 393-0411, ext. 7957
RAND
galway@rand.org
- Ranjan Maitra**, Education Liaison
(410) 445-2436
University of Maryland
maitra@math.umbc.edu
- John J. Miller**, Education Liaison
(703) 993-1690
George Mason University
jmiller@gmu.edu
- Leland Wilkinson**, Rep.(99-01)Council of Sections
(312) 651-3270
SPSS
leland@spss.com
- David M. Allen**, Rep.(00-02)Council of Sections
(606) 257-6901
University of Kentucky
allen@ms.uky.edu
- Elizabeth Slate**, Rep.(00-02)Council of Sections
607-255-9148
Cornell University
ehs1@cornell.edu

INSIDE

A WORD FROM OUR CHAIRS	1
Statistical Computing	1
SPECIAL FEATURE ARTICLE	1
Editorial	2
FROM OUR CHAIRS (Cont.)...	3
Statistical Computing	3
FROM OUR CHAIRS (Cont.)...	4
Statistical Graphics	4
SOFTWARE PACKAGES	5
TOPICS IN STATISTICAL COMPUTING	15
TEACHING STATISTICAL COMPUTING	25
NEWS CLIPPINGS AND SECTION NOTICES	28
SECTION OFFICERS	29
INSIDE	30

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

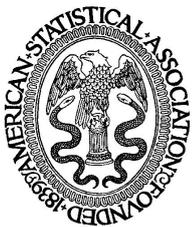
Susan Holmes
Editor, Statistical Computing Section
Stanford University
Stanford, CA 94305
(650) 725-1925 • FAX: (650) 725-8977
susan@stat.stanford.edu
<http://www-stat.stanford.edu/~susan>

Dianne Cook
Editor, Statistical Graphics Section
Department of Statistics
Iowa State University
Ames, IA 50011-1210
(515) 294 8865 • FAX: (515) 294 4040
dicook@iastate.edu
www.public.iastate.edu/~dicook

All communications regarding ASA membership 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 USA
(703) 684-1221 • FAX (703) 684-2036
asainfo@amstat.org

The *Statistical Computing & Statistical Graphics*



American Statistical Association
1429 Duke Street
Alexandria, VA 22314-3402
USA