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

*Winter 98/99 Vol.9 No.2*

# Statistical COMPUTING & GRAPHICS

***A WORD FROM OUR CHAIRS***

## Statistical Computing



*Karen Kafadar is the 1998 Chair of the Statistical Computing Section. In her column she discusses several actions that have increased the impact of the Section on the field of Statistics.*

It is a pleasure to report on the activities of the Section and the decisions of the Executive Board following the very successful Joint Statistical Meetings (JSM) in Dallas last August. A summary of the Section's program appears on page 37. We are grateful to Russ Wolfinger for the outstanding program that he organized with many useful and significant presentations to the statistics community at this annual meeting. The ASA Board of Directors presented a strategic initiative to carry the

CONTINUED ON PAGE 2

## Statistical Graphics



*Michael Meyer is the 1998 Chair of the Statistical Graphics Section. He encourages those who are not members of the Section to join, making possible the continuation of many Section initiatives.*

Congratulations to Ed Wegman for putting together a interesting and well-attended graphics program for the 1998 Annual meetings, and even more congratulations to former Section officer David Scott who had overall responsibility for the JSM program.

Debby Swayne has already put together an impressive set of sessions for the next annual meetings. Together with the computing Section program chair, she outlines the invited program in an article on page 38. Perhaps

CONTINUED ON PAGE 3

***SPECIAL FEATURE ARTICLE***

## Coding and Model Selection: A Brief Tutorial on the Principle of Minimum Description Length

By Bin Yu

Model selection is one of the most active research areas in statistics nowadays. Aside from its tremendous practical value, model selection is interesting because it addresses a different kind of inference problem than we ordinarily encounter. Criteria like AIC and BIC have entered even elementary statistics textbooks. Bayesian model selection has been rejuvenated in part because of new computational schemes like Markov chain Monte Carlo. Here we introduce an information theoretic approach, the Principle of Minimum Description Length (MDL) originally proposed by Rissanen (1978, 1989).

We believe that MDL provides a natural solution to the model selection problem, enjoying rich connections with both frequentist and Bayesian statistics. As its name suggests MDL recommends

*choosing the model that gives the minimum description length of data.*

Many statisticians may well question the need for yet another model selection criterion. More to the point, how are concepts from information theory relevant?

### *Why description length in statistics?*

First let us make it clear that in this article description length is exchangeable with code length so the question really is: *why code length in statistics?* To answer that, we need to answer first *What is a code?* Given a finite

CONTINUED ON PAGE 27

*EDITORIAL*

# An E-mail List?

In this edition of the Newsletter, we have again assembled a collection of interesting articles and news tid bits relevant to our membership. Unfortunately, once again we are slightly late in getting this publication out to you. Of course, the on-line version has been up for a couple weeks, so if you are ever eager to see what your dues are paying for, check out the Newsletter Web site `cm.bell-labs.com/who/cocteau/newsletter`.

The special feature article this issue is by Bin Yu. It is on a class of model selection criteria that have strong connections with information theory. As you will see, the so-called principle of Minimum Description Length (MDL) brings together many familiar notions in both frequentist and Bayesian statistics.

This issue also features three articles of interest to the graphics members. The first is by Dan Carr, Tony Olsen, Suzanne Pierson and Pip Courbois on variations of the boxplot for use in a spatial context. Dan and Company explore linked micromap boxplots, linked micromap bivariate boxplots and angular boxplot glyphs. Next, Newsletter Co-Editor Antony Unwin and Graham Wills discuss Diamond Fast, a tool for comparing time series. Lastly, Heike Hofmann takes us through interactive methods for multivariate, categorical data.

Duncan Temple Lang and Steve Golowich introduce us to the capabilities of an application called *make*. Duncan and Steve present a number of situations in which *make* can help organize tasks, including text preparation with LaTeX (a topic dear to our hearts as this Newsletter is set with LaTeX).

Andreas Buja, Editor of the *Journal of Computational and Graphical Statistics* (JCGS), presents the first in (a hopefully long) series of columns. We hope that Adreas' contributions will be a useful link between our two publications. JCGS is an outstanding journal and we encourage you to check it out if you haven't.

We end this issue with Section News. Deborah Swayne brings us up to date on the Graphics Section's Video Library. The extensive collection is now available over the Web! We also get a report on the Computing Section's awards for the best contributed paper and the best presentation at the JSM. Our Program Chairs for 1998 give us a wrap-up of the JSM in Dallas, and our Chairs for 1999 give us a peak at what's planned for Baltimore!

Finally, to better serve our members, we were hoping to create an e-mail list by which we could directly communicate with each of you. This year one of us (Mark) is Program Chair for the Computing Section. In performing the necessary duties, it would have been helpful to poll the Section for suggestions on sessions, etc. Let us know what you think of this idea.

Mark Hansen
*Editor, Statistical Computing Section*
*Bell Laboratories*
`cocteau@bell-labs.com`

Antony Unwin
*Editor, Statistical Graphics Section*
*Universität Augsburg*
`unwin@uni-augsburg.de`

# Statistical Computing

CONTINUED FROM PAGE 1

association into the next decade; matters that relate to Section activities were presented at the meeting of the Executive Board in Dallas, and members are encouraged to review these initiatives on the ASA Web site (`www.amstat.org`) and communicate their ideas to our Council of Section representatives.

Thanks to increasing participation and suggestions from our members, our Section has continued and increased its involvement in statistical computing and education during the past year. Due to their previous successes, the Executive Board voted to continue support for the Undergraduate Data Analysis Contest and the Poster Competition, both of which are designed to introduce students in high school and college to statistics. Brian Marx from Louisiana State University expressed thanks to our Section for our support of the International Workshop on Statistical Modeling, which was used to support travel for Section members overseas, as did Snehalata Huzurbazar from University of Wyoming for supporting the travel of Section members to the New Researchers Conference. Lionel Galway, our awards chairman, initiated and implemented an awards process at the JSM this year for students in Statistical Computing. The Section sponsored their travel to present their papers in a special contributed session, all of which were excellent. If you missed that session in Dallas, be sure to make a note to attend it in Baltimore next year. It is exciting to see new members become involved in these important issues facing statistics as we enter the 21st century.

An exciting cooperative effort was initiated last year, through Daryl Pregibon (1997 Section Chair), by the chairman, Won Kim, of the Knowledge, Discover, and Data Mining (KDD) special interest group of the Association for Computing Machinery (ACM). The mission of this Section is to contribute to the development of research that overlaps data base technology, artificial intelligence, and statistics, and the Executive Board agreed to cooperate in the announcement of each other's conferences and workshops.

To increase the impact of our Section on the field of statistics, we need to continue to identify opportunities such as this cooperation with KDD. A justifiable concern with the survival of our profession is insulation from other fields working on fundamentally statistical issues. The past year's JSM in Dallas presented several sessions where statisticians collaborated with scientists and professionals from other fields. Many more opportunities exist, and our members are providing a great service not only to our profession by way of enhancing its reputation among other disciplines, but also to the advancements in these fields. This newsletter will announce conferences with a significant statistical computing component sponsored by other disciplines to assist in this identification of opportunities for our members; if you are aware of such meetings, please notify the Newsletter Editors so they can be included in subsequent issues. For meetings such as next year's JSM in Baltimore or the Interface Symposium in New Orleans, it is too late to propose invited sessions, but you might consider a contributed submission (see the announcement on page 32). These presentations stimulate others in our profession to work on problems in diverse areas that use increasingly important computational resources and hence influence the direction of our field as a whole.

These accomplishments would not be possible without the support and participation of our members and our Executive Board. I would like to take this opportunity to express my sincere thanks to my predecessors, Sallie Keller-McNulty (1996) and Daryl Pregibon (1997) and to the members of the Executive Board (see back page) for their dedicated participation and advice on matters related to our Section. Next year, Jim Rosenberger will provide outstanding leadership to this section, and I look forward to working with him and seeing all of you at future meetings and mixers.

Karen Kafadar
*University of Colorado at Denver*
kk@math.cudenver.edu

# Statistical Graphics

CONTINUED FROM PAGE 1

more importantly it is not too early to start thinking about the program for the year 2000 (Is this the millennium or not?—who cares so long as the computers keep working). The graphics program chair for those meetings is Dan Carr, dcarr@galaxy.gmu.edu

I am looking forward to handing over the reins of the Graphics Section to Di Cook dicook@iastate.edu in January. I am sure she will do a better job, and be a more active Section Chair, than I was able to be.

Many of you have had some interactions with Bob Newcomb rnewcomb@uci.edu who has been our Section Secretary/Treasurer since at least 1995. What you may not realize is that Bob does a great deal to keep the Section running. Not only does he manage our finances and ensure that we stay coordinated with the main ASA office, he is also responsible for keeping the Section web site (follow the link from www.amstat.org/sections/) up to date. On a more personal note Bob has been wonderful at periodically nudging me to do the things that need to be done.

Another of the quiet Section heroes is Debby Swayne. I've already mentioned that she is Program Chair for next year. She also manages the Section graphics library, which is a wonderful resource. Information about the graphics library is available on the Section home page (see her article on page 34).

Finally, a plea for membership. According to the official accounting our membership has fallen. We have not yet confirmed if the Section membership has actually fallen, or if this is just an accounting artifact. But, it doesn't really matter. Section membership is important. Section membership supports the newsletter—and ours is surely the best Section newsletter—and determines the number of sessions at the annual meetings. If you are reading this newsletter and are not a Section member, please consider joining. If you are already a member, consider passing a copy of the newsletter to your colleagues and encouraging them to join.

Mike Meyer
*The Boeing Company*
michael.m.meyer@boeing.com

# Boxplot Variations in a Spatial Context: An Omernik Ecoregion and Weather Example

By Daniel B. Carr, Anthony R. Olsen,
Suzanne M. Pierson, Jean-Yves P. Courbois

## 1. Introduction

This article presents three variations on boxplots for use in a spatial context. The three variations are linked micromap boxplots, linked micromap bivariate boxplots, and angular boxplot glyphs. The specific examples show Omernik Level II Ecoregions. The variables chosen for illustration purposes are growing degree days and precipitation.

This article is closely related to papers by Carr et al (1998a, 1998b). The first paper provides a general description of linked micromap (LM) plots such as that in Figure 1b. The second paper puts LM plots to work in describing Omernik Level II ecoregions. It also promotes LM plots as useful methodology in the KDD pattern discover process and as overviews that provide an orienting context for drilling down to finer detail in extensive hierarchically organized summaries. This relatively brief article adapts graphics and text from the second paper while focusing attention on boxplots.

In this paper Section 2 provides context for the graphics with a description of Omernik ecoregions and the data sets. Section 3 provides background on boxplots and then presents both univariate and bivariate LM boxplots. Section 4 motivates two angular boxplot glyphs and presents the one that is more striking, the Portuguese man of war. Section 5 closes with connections to other work and challenges for the future.

## 2. Ecoregions and Datasets

While this paper focuses on graphics design, a scientific context exists behind the examples: the presentation of summary statistics for ecoregions and the attempt to learn about the intellectual structure that underlies ecoregion definition. In this section we provide background on ecoregions and associated datasets.

### 2.1 Ecoregions

Ecoregions are a way of codifying the recognizable regions within which we observe particular patterns or mosaics in ecosystems. The general consensus is that such ecological regions, or ecoregions exist. However, disagreement in how the regions should be constructed continues to be a scientific issue (Omernik, 1995). In describing ecoregionalization in Canada, Wiken (1986) stated: "Ecological land classification is a process of delineating and classifying ecologically distinctive areas of the earth's surface. Each area can be viewed as a discrete system which has resulted from the mesh and interplay of the geologic, landform, soil, vegetative, climatic, wildlife, water and human factors which may be present. The dominance of any one or a number of these factors varies with the given ecological land unit. This holistic approach to land classification can be applied incrementally on a scale-related basis from very site-specific ecosystems to very broad ecosystems."

Within the United States two alternative approaches in the construction of ecoregions are those developed by Omernik (1987, 1995) and Bailey (1995a, 1995b, 1998). Each constructs a hierarchy of ecoregions that corresponds to viewing the United States at different scales. Omernik's approach is conceptually similar to that described by Wiken, where the ecological regions gain their identity through spatial differences in the combinations of the defining factors and which factors are important vary from one place to another and at all scales. Bailey (1998) develops ecological regions hierarchically. First, he identifies ecological regions of continental scale based on macroclimate, where macroclimates influence soil formation, help shape surface topography, and affect the suitability of human habitation. The continent is subdivided with three levels of detail into domains, within domain divisions, and within division provinces. Domains and divisions are based largely on broad ecological climatic zones while provinces further sub-divide the divisions on the basis of macro features of the vegetation. Hence Bailey uses macroclimate as the controlling factor in the formation of ecoregions while Omernik uses all available factors where the importance of the factors varies among ecoregions. Some scientists question whether enough is known to delineate ecoregions. While our knowledge is limited, others proceed on the basis that our approximations of the "true" ecoregions will continue to improve as more information is gathered.

Our interest is not to define ecoregions or even to validate them. We simply believe that it is important to describe quantitatively some of the key characteristics associated with ecoregions to gain a better understanding of how ecoregions have partitioned these characteristics. In view of the newsletter page size, we show Omernik's level II ecoregions (Figure 1a) for the

Figure 1. a=Omernik Ecoregions b=Linked Micromap Boxplots

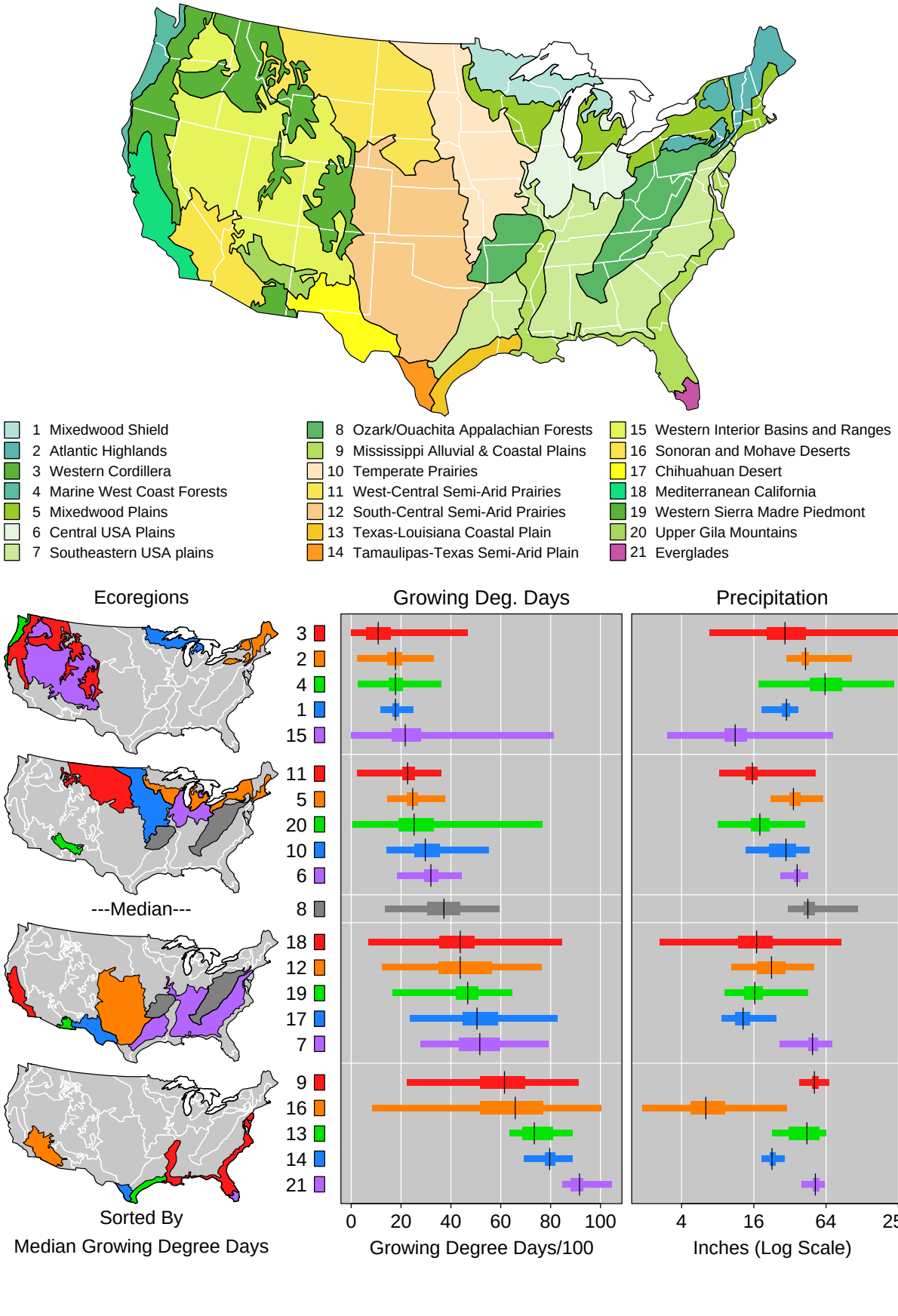
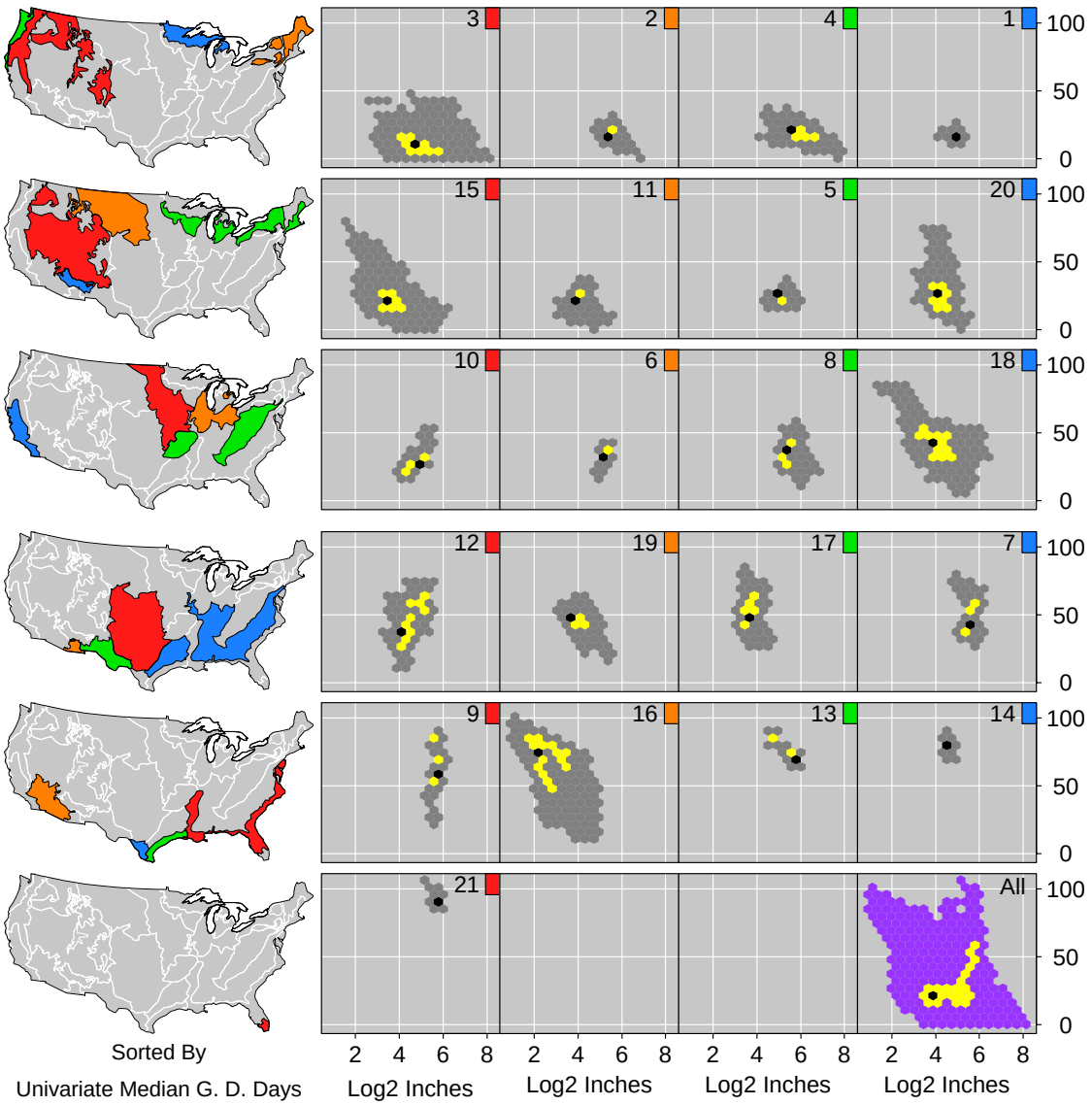


Figure 2: LM Bivariate Boxplots

1961-1990 Precipitation (x) versus Growing Degree Days/100 (y)




continental U.S. examples here. This coarse level of aggregation bothers some people, for example when they see the Mississippi Valley in the same ecoregion with the Southeastern seacoast and Cape Cod. Level III and level IV ecoregions are progressively finer partitions of North America and are much more homogeneous in terms of familiar variables.

### *2.2 Data Sets*

The data sets used in our graphics are readily available. Substantial thought and processing was involved in producing the data sets. Thus this paper builds upon the work of others.

Ecoregion delineation depends on climate as a major forcing function. We use nationally consistent climate data sets developed using PRISM (Parameter-elevation Regressions on Independent Slopes Model). PRISM, described by Daly et al (1994), is an analytical model that uses point data and a digital elevation model (DEM) to generate gridded estimates of monthly and annual climatic parameters. PRISM models data from individual climate stations to a regular grid through locally weighted precipitation/elevation regression functions. Orographic effects on precipitation and other climate parameters are well known and are used to define the local regions. PRISM has been shown to out perform other common spatial interpolation procedures such as

kriging. The regular grid used is 2.5 minute by 2.5 minute latitude/longitude grid, i.e., a nominal 5 km by 5 km grid. For our purposes having a regular grid of climate data ensures that the entire area of an ecoregion is represented in any statistical summary we construct. PRISM data we use are based on 30-year normal climate data for 1961-1990. Key annual parameters investigated are mean temperature, temperature range, growing degree days, and total precipitation, and monthly precipitation. Further information on PRISM and the data sets can be found at `www.ocs.orst.edu/prism/`.

Figures 1 and 2 focus attention on just two PRISM variables, growing degree days and precipitation. The delineation of ecoregions depends not only on the annual characteristics of climate but also on seasonal characteristics. Thus monthly precipitation is important to study. As indicated above many variables such as elevation, land cover, and soils also relate to ecoregions. A dated visual summary of level II ecoregion land cover classes is available in Carr and Olsen 1996. (A few boundaries have changed.)

Figure 3 (page 10) in this paper concerns precipitation trends. The figure makes direct use of annual weather station precipitation data for the period 1961-1996. The precipitation data (72 Mbytes) and weather station location descriptors (10.2Mbytes) are available from the National Climatic Data Center Web site `www.ncdc.noaa.gov/ol/climate/onlye/coop-precip.html#Files`.

Our data processing classified weather stations into ecoregions. We used each station's most recent latitude and longitude (some change over time) to locate it in one of 672 polygons that define the Level II ecoregions in the continental U.S. We omitted stations that had fewer than 24 annual values over the 36 years. Some 4075 weather stations satisfied the criteria of being in an ecoregion and having sufficient annual values. We then calculated trend using Sen's slope, which is the median of slopes computed using all pairs of available years for each station. This provides an estimate that is little influenced by an occasional extreme value. The angular boxplot glyphs in Figure 3 represent variation in 36-year trends at different weather stations with the 21 Level II ecoregions.

## 3. LM Boxplots

Figure 1b shows linked micromap univariate boxplots. We briefly discuss the basic LM plots elements: micromaps, linking labels, boxplots, and sorting. For more LM plot details see the previously mentioned papers.

The micromaps on the left are caricatures of level II ecoregions. The caricatures simplify the polygon boundaries and reduce the 672 polygons for the continental U.S. to 44 polygons. Arguments can be more for going even further and, for example, removing Cape Cod and the island in Lake Superior.

Omernik's Level II definition of ecoregions partitions the continental U.S. into 21 ecoregions. The long labels for the ecoregions appear in Figure 1a. Linking ecoregion name to location is difficult in Figure 1a. There are many ecoregions and some involve disjoint areas. The colors may be pleasant to view, but are not easy to discriminate. For most readers the colors do not have names, and this can make it much harder to remember the exact color that is to be found on the map. Plotting the ecoregion numbers on the map typically solves the memory problem, but serial search is still involved. With disjoint areas for the same ecoregion, finding one number does not mean the search task has been completed. The symbol congestion and search problem gets worse with Level III ecoregions.

Micromaps with color links provides a solution to the name and region linking problem. To find the polygons for Ecoregion 3 one simply looks for the red regions in the top micromap. The only difficulty concerns micromap caricatures with polygons so small that they do not reveal their color. The state map in Carr and Pierson 1996 enlarge small states to make sure their color is visible. In the current map caricature, Cape Cod (Ecoregion 9) is too small to reveal its color. The development of micromap caricatures involves compromise.

The use of integers as ecoregion labels is another compromise. Descriptive labels always have merit. However, in this and other examples involving more statistical summary columns, we choose to reserve space for the statistical graphics panels. The integer labels take up little space and become familiar with repeated usage.

The color rectangle as a part of the label can be dropped in Figure 1b because the boxplots provide the color. For those experienced with LM plots, dropping the line of colored rectangles simplifies appearance and complicates color linking only slightly. We retain the rectangles in the current context.

The boxplot or box and whiskers plot is a well-known distributional caricature that has now appeared in some grade school curriculum. However, the choice of summary statistics and the graphical representation are not universal (see McGill, Tukey and Larsen 1978; Frigge, Hoaglin, and Iglewicz 1989; Tukey 1993; and Carr 1994). Figure 1b shows a five number summary: the extrema, the 1st and 3rd quartiles and the median. An-

other common choice represents adjacent values and outliers rather than just extrema. We note that the outlier portion of this choice does not scale well for large samples from thick tailed distributions. For example if estimates followed a t-distribution with 3 degrees of freedom roughly 2.75 percent of the estimates would be flagged as outliers on each end of the distribution. Outlier overplotting was problematic for precipitation in an alternative version of Figure 1b. When outlier overplotting is problematic a compromise caricature that we do not show uses adjacent values and adds an outlier dot for a maximum or a minimum that is not an adjacent value. The compromise uses a filled dot to indicate multiple outliers thus hiding the details of outlier multiplicity and location.

Our graphical representation follows the design of Carr (1994). That is, the thick rectangle extends from the 1st to the 3rd quartile, the thin rectangle extends to the extrema, and the vertical median line extends outside the thick rectangle. Comparison of adjacent medians can typically be based on the judgement of horizontal distance between line endpoints. Using dots for the median (Becker and Cleveland 1993) is a viable choice since the area symbol is easy to spot even though enclosed in a rectangle. However, the comparison of neighboring medians may not be as accurate since the direct distance between points is not the correct distance to judge.

The boxplot is suitable for describing a batch of data. The batch of data here reflects variation due to changing spatial location. However, one would like the items in a batch to be as comparable as possible. This is not the case here because the cells are defined using equal angle grids and not equal area grids. Carr et al (1997) question the use of equal angle grids in environmental sciences and in NASA's level 3 satellite products. They promote the use of discrete global equal area grids. While transformations can be made so the North Pole is not as wide as the equator, changing to equal area grids for reasons of comparability or polar modeling adds complexity and uncertainty. We choose to avoid the complexity of regridding the data for this article on boxplots. A consequence is that in large north to south ecoregions, such as Ecoregion 9, the northern portion is over represented in the boxplot due to smaller area grid cells. If the North is cooler than the South the growing degree day percentiles will be shifted a bit toward cooler values.

The notion of growing degree days many not be familiar, but the calculation is straight forward. If the average daily temperature is over 50 degrees Fahrenheit, the day counts. The degrees accumulated for each day that counts is the average daily temperature minus 50 degrees. As a calibration case suppose a grid cell in Florida had a daily average value of 75 degrees, each day for the whole year. The growing degrees days would then be 365 * (75-50) = 9125. To provide easier to remember two digit numbers, the scale for Figure 1b shows growing degree days divided by 100.

The precipitation data for Figure 1b has a thick right tail. A logarithmic transformation helps to pull in the tail, and provides better resolution for small precipitation values. The choice here was to use log base 2, since powers of 2 are familiar to many scientists (see Cleveland 1985).

Sorting helps to bring out patterns in LM plots. Carr and Olsen (1996) discuss sorting methods and show ecoregion examples. In Figure 1b sorting by increasing growing degree days arranges the ecoregions in the micromap panel sequence so that northern ecoregions tend to appear in the top panels and southern ecoregions appear in the bottom panels. This matches common expectation. Some of the north to south anomalies, such as Ecoregion 20, are easier to understand when elevation boxplots are included. While pattern details may be well-known to experts, those new to the field may delight in seeing patterns that confirm their limited knowledge and in finding anomalies. As an interpretation reminder, the patterns in growing degree days refers to the thirty year average and not to a particular day or year.

Sorting Figure 1b by increasing precipitation reveals a strong West to East pattern. With the exception of Ecoregion 4 that contains the Pacific Northwest rain forest, the high precipitation is in the East. A plot is available on our Web site.

The juxtaposed plots make it tempting to look at the growing degree day and precipitation medians to see if there is a relationship. The first impression is that if there is a relationship, it is weak. Of course juxtaposed univariate plots, such as bar plots, dot plots or boxplots, provide a poor way to look for a functional relationship even if one variable is sorted. While perhaps unknown to popular press, the scatterplot provides the standard for assessing a functional relation between two variables. We omit this scatterplot and proceed to represent the bivariate data for all grid cells and not just the univariate medians.

The current application with 481,475 grid cells warrants the use of density estimation as part of the graphic representation process. Scott (1992) provides methods for both density estimation and graphical representions that emerged from years of research with binning and other density estimation methods. Proponents of binning for dealing with large data sets also include Carr et

al (1987). Their space-time example included point in polygon selection on a map to defined two subsets, representation of density differences in a scatterplot matrix, and a representation of a temporal mismatch. Bivariate binning methods scale well in the number of cases. Carr (1998) shows a scatterplot matrix representing over billion point pairs. Such density estimates can vary over many orders of magnitude. Appreciating the density surface details is not necessarily a trivial task. For an overview we seek something far simpler and turn to a bivariate boxplot.

Chances are that many people have proposed bivariate boxplots. So far bivariate boxplots have not seemed to catch on, likely due to lack of promotion. Possibly disagreement on details is behind this lack of promotion. Certainly there are issues concerning what generalization is most appropriate for the 1st, 2nd, and 3rd quartiles. (See Small [1989] concerning bivariate median options that were available years ago.) Perhaps it is better to show density modes and two alpha contours. Our attitude is to get on with the graphics. We illustrate the approach of Carr (1991) but are quite willing to substitute other methods.

Carr (1991) binned the data using hexagon grids to speed the processing. He directly used the high density cells containing 50 percent of the data in place of the interquartile range. To obtain a median he smoothed the high density cells (an optional step), and then eroded the cells using gray level erosion. The gray level erosion process removes counts from the cells proportional to exposed surface area. The last cell eroded provides location of the median in the binned density representation. (S-PLUS™ now provides the needed tools for hexagon grids.) This simple procedure generalizes to other regular grids in two and higher dimensions. With translucence or the see through contour representations in Scott (1992), showing 3-D boxplots is not a problem.

If one were to use the 50% high density region in univariate boxplots rather than the interquartile range, comparison becomes harder because the high density region can be disjoint. Carr (1991) addresses the increased difficulty of comparison in the bivariate context by developing a difference plot. Cells in two gray levels distinguish the two mismatched portions of the high density region. An arrow shows the change in the median. The example interleaves difference plots between bivariate boxplots in an age and sex conditioned two-way layout of bivariate boxplots. With more space we could interleave the bivariate difference plots in the linear sequence shown in Figure 2.

Figure 2 shows a bivariate boxplot of the combined data in the lower right corner. The purple shows the support of the bivariate pairs, (30-year average precipitation on a log scale, 30-year average growing degree days). The yellow cells are the 50% high density cells and the black cell is the median cell. The positive trend of the high density cells contrasts strongly to the negative slope suggested by the region of support. The latter is all one would see in an overplotted view. How useful it is to study this aggregate plot can be argued. Scientists tend to be interested in detail. When scientists see a plot splitting into non-homogeneous pieces as in Figure 2 they naturally want to study the pieces. Nonetheless overviews are often valuable for providing a frame of reference as one approaches the pieces.

We mention several patterns in Figure 2, but the existence of Level IV ecoregions suggests that scientists would prefer to focus attention on more highly stratified ecoregion views. Already suggested was the fact the variation of individual ecoregions is generally much smaller than that of the composite. The small amount of yellow in the Ecoregion plots indicates a high concentration of bivariate values within tight domains. Several boxplots suggest that they are mixtures of tight patterns. Few bear much resemblance to a bivariate normal distribution. A moderately elliptical exception, Ecoregion 10, shows no yellow so the cell with yellow overplotting black median dot contains over 50 percent of the observations. Panels 15, 18, and 16 (top to bottom order) catch the eye with relatively big bivariate support regions (gray) that have negative slopes. The high density yellow cells reflect the negative slopes as well. The high density cells in Region 16 show a bifurcation that motivates further investigation and suggests possible subdivision of the ecoregion. (Carr et al 1998b note that Region 16 is homogeneous in terms of land cover). Regions 10 and 9 show high positive slopes. This suggests that most of the variation is in terms of growing degree days and that growing degree days is not major factor in their definition. Note that both regions cover a large range of latitude. In general the growing degree day variation in the bivariate boxplot appears associated with latitude variation in the micromaps. This motivates putting growing degree days on the y axis.

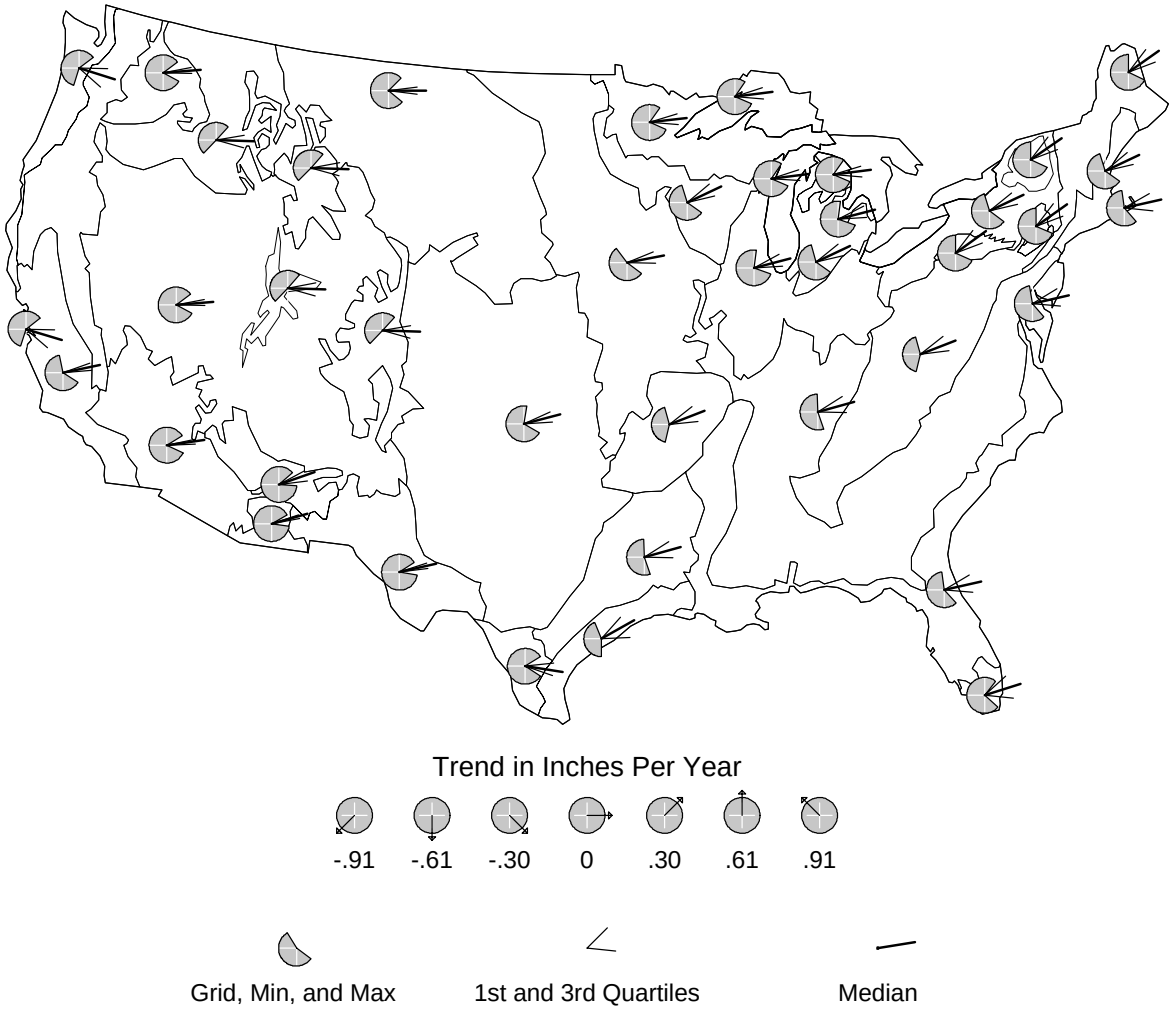
Figure 2 uses the univariate growing degree days as the basis for sorting, but the micromap layout is different than in Figure 1. Due to the size of bivariate boxplots, Figure 2 shows four or fewer ecoregions per micromap. The vertical list of names and color links is absent since they are mostly redundant with the names and color tags in each bivariate boxplot panel. The list would clarify the bivariate boxplot order that is left to right, top to bottom. Careful comparison of bivariate median values

for growing degree day values against the reference grid indicates the bivariate and univariate ranking are in reasonable agreement. Other sorting can bring out other patterns.

## *4. Angular Glyph Boxplots and a Precipitation Trend Example*

Carr and Pierson (1996) favored LM plots over choropleth maps in terms of representing statistical summaries more faithfully. Their reasons include better perceptual accuracy of extraction and easy visual representation of uncertainty. However, classed choropleth maps are not the only way to represent spatially indexed statistical summaries. Glyph plots provide a viable alternative for representing summaries so they deserve consideration. Carr et al (1998b) propose two angular glyph boxplots but illustrate only one. Figure 3 shows the more controversial glyph that we call the Portuguese man of war. The glyph extends the trend glyph design of Carr, Olsen, and White (1992). In their trend glyph design, ray angle represents the sulfate deposition trend and the straight edges of a circle sector represent 90 percent confidence bounds. A local scale, a circle with lines at regular angles, provides the basis for judging ray angles. We continually attempt to put Cleveland and McGills (1984) guidance concerning perceptual accuracy of extraction into practice. They promote representations that use position along a scale as being the best. Our encoding uses position along a scale, albeit an angular scale.

Figure 3: 1961-1996 Precipitation Trends
Angular Boxplots For Stations Within Ecoregions



Then boxplot glyph shown in Carr et al (1998b) uses the circle sector to represent the interval from first to third quartile. Two shorter length rays (but longer than the reference circle) encode the extrema. The Portuguese man of war glyph shortens the extrema rays to the length of the reference circle and cuts away the circle sector between extrema rays. As can be seen in Figure 3, the glyph also removes the arc between lines showing the first and third quartiles. The gestalt is striking. Some symbols are aligned and some twist. The departures from symmetry about the horizontal zero trend line and about the median line are clear. The gestalt is so striking that it may impede people looking at the detail. Also cutting away part of the reference circle makes it a bit harder to judge angles. Thus this Portuguese man of war glyph may lose in careful testing for perceptual accuracy of extraction.

The angular limits for the boxplot glyph in Figure 3 are from 135 degrees to 135 degrees. There is some advantage in restricting the limits to 90 degrees so, for example, the largest local y coordinate indicates the largest increase. The extension here is to increase angular resolution. The legend scale has limits of -.91 and .91 inches per year. This implies a change of roughly 32 inches over the 36 years. The scale is defined symmetrically so both extremes are not necessarily achieved. However, a study of the glyphs indicate that this is nearly so. Time series plots for individual stations confirm such substantial change.

The existence of multiple polygons for the same ecoregion complicates examination of Figure 3. The same glyph appears for each polygon that comprises a given ecoregion. This gives more visual weight to ecoregions with multiple polygons. If unequal visual weight is given, a more reasonable choice is to favor large area ecoregions. The glyph design itself could be modified to indicate the number of weather stations with each ecoregion.

Glyph plots have some problems. We have slightly revised the GIS-provided representative point for each polygon to eliminate glyph overplotting in Figure 3. Symbol congestion control becomes increasingly problematic as the number of polygons increase, for example with level III ecoregions. (The trend glyph plot in Carr, Olsen and White [1992] was restricted to a regular hexagon grid to control symbol congestion.) The glyphs provide a spatial overview, but in this gray level example do not connect strongly to ecoregions and their names. LM plots provide a strong summary to name connection and avoid multiple glyphs, placement and symbol congestion problems. Also the straight line scale boxplot is a bit easier to assess than an angular scale boxplot. Still, the gestalt spatial impression of glyph plots has value.

## *5. Connections and Future Challenges*

The article builds upon previous work by many people. Carr et al (1998b) cite many that inspired our development of LM plots. Our shortened list here includes Cleveland and McGill (1984), Cleveland (1985), Monmonier (1988, 1993), Tufte (1990, 1993) and Kosslyn (1994). Our scholarship in regard to bivariate boxplots is at best dated. We welcome references to the work of others and constructive comments about better designs.

The Splus script files we used to produce the graphics are available by anonymous ftp to `galaxy.gmu.edu`. The subdirectory is `pub/dcarr/newsletter/boxplots`. The contexts include the statistical summaries used in the graphics but not the prior data. For more information on the data or on algorithms such as gray level erosion in 3-D and 4-D please contact us directly. There is also a Web site, `www.galaxy.gmu.edu/~dcarr/graphgall/ecoregions/ecoregions.html` that contains a growing number of Omernik level II and level III ecoregion examples.

We want to make a point relative to our first presentation (Olsen et al 1996) of Omernik ecoregions. The 4 x 8 foot poster example provided a beginning description of Omernik's Level III ecoregions. This poster included over one hundred micromaps, boxplots of ecoregion digital elevations and even Loveland's 8 million pixel map of 159 land cover classes for the continental U.S. (See Loveland et al 1995). While large high-resolution graphics that integrate different views would seem to provide a powerful tool in the attempt to understanding large data sets, such graphics fall outside the mainstream. Today's emphasis is on Web graphics. Tufte (`www.clbooks.com/nbb/tufte.html`) comments that, "the problem with the Web is that it is low resolution in both space and low in time." As one way to appreciate this statement, note that a typical workstation monitor shows only 1024 * 1280 pixels (roughly 1.3 million pixels). Thus without reducing resolution, Loveland's map requires over six complete screens. Something is lost in comparison to high quality printed maps. There is no denying the power of human computer interface methodology such as logical pan and zoom, but the challenge of large data sets and complex conceptual structures motivates use of all powerful tools including good old human pan and zoom of large high quality graphics.

The task of developing and presenting statistical overviews for massive data sets is a big challenge. The environmental science community has worked on the task longer than many disciplines. How does one represent environmental facts and knowledge? Two recent book length overviews, Stone et al (1997) and Wahlstrom, Hllanaro, and Maninen (1996), involved the work of many people, including in one case layout specialists and graphics designers. The integration of photographs, text, and statistical graphics serves as an inspiration. Our challenge is to learn when others have blazed the trail, to make the improvements when we see the opportunity and to adapt the methods to other applications. Perhaps an even harder challenge is to see the patterns in important data not collected and take action.

## 6. Acknowledgements

S-PLUS is a registered trademark.

EPA funded the majority of the work behind this paper under cooperative agreements No. CR8280820-01-0 and No. CR825564-01-0. Additional federal agencies, BLS and NCHS, supported some facets of this work. The article has not been subject to review by BLS or NCHS so does not necessarily reflect the view of the agencies, and no official endorsement should be inferred. It has been subjected to EPA's peer and administrative review and approved for publication. The conclusions and opinions are solely those of the authors and are not necessarily the views of the Agencies.

## 7. References

Bailey, R.G. (1995a) "Description of the ecoregions of the United States," Misc. Publ. No. 1391 (rev.), USDA Forest Service.

Bailey, R.G. (1995b) *Ecosystem Geography*, Springer-Verlag, New York.

Bailey, R.G. (1998) "Ecoregions map of North America: Explanatory Note," Misc. Publ. 1548, USDA Forest Service.

Becker , R.A. and Cleveland, W.S. (1993) "Discussion of Grapical Comparison of Several Linked Aspects: Alternative and Suggested Principles," JCGS, Vol. 2, No. 1, 41-48.

Carr, D.B. (1991) "Looking at Large Data Sets Using Binned Data Plots," in *Computing and Graphics in Statistics*, eds. Buja, A. and Tukey, P., 7-39.

Carr, D.B. (1994) "A Colorful Variation on Boxplots," *SC& G Newsletter*, Vol. 5, No. 3, 19-23.

Carr, D.B. (1998) "Multivariate Graphics," *Encyclo. of Biostat.*, eds. Armitage, P. and Colton, T., Vol. 4, 2864-2886.

Carr, D.B. and Olsen, A.R. (1996) "Simplifying Visual Appearance By Sorting: An Example Using 159 AVHRR Classes," *SC& G Newsletter*, Vol. 7, No. 1, 10-16.

Carr, D.B., Olsen, A.R. and White, D. (1992) "Hexagon Mosaic Maps for Display of Univariate and Bivariate Geographical Data," Cartography and Geographic Information Systems, Vol. 19, No. 4, 228-236,271.

Carr, D.B., Olsen, A.R., Courbois, J.P., Pierson, S.M., and Carr, D.A. (1998a) "Linked Micromap Plots: Named and Described," *SC& G Newsletter*, Vol. 9, No. 1, 24-32.

Carr, D. B., A. R. Olsen, S. M. Pierson, and J. P. Courbois (1998b) "Using Linked Micromap Plots To Characterize Omernik Ecoregions," *Data Mining and Knowledge Discovery*, submitted.

Carr, D.B., Littlefield, R.J., Nicholson, W.L., and Littlefield, J.S. (1987) "Scatterplot Matrix Techniques For Large N," *JASA*, Vol. 82, 424-436.

Carr, D.B., Kahn, R., Sahr, K., and Olsen, A.R. (1998a) "ISEA Discrete Global Grids," *SC& G Newsletter*, Vol. 8, No. 2/3, 31-39.

Carr, D.B. and Pierson, S. (1996) "Emphasizing Statistical Summaries and Showing Spatial Context with Micromaps," *SC& G Newsletter*, Vol. 7, No. 3, 16- 23.

Cleveland, W.S. (1985) *The Elements of Graphing Data*, Wadsworth, Monterey, CA.

Cleveland, W.S. and McGill, R. (1984) "Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods," *JASA*, Vol. 79, 531-554.

Daly, C., Neilson, R.P., and Phillips, D.L. (1994) "A statistical-topographic model for mapping climatological precipitation over mountainous terrain," *Jour of Appl Meteorology*, 33, 140- 158.

Frigge, M., Hoaglin, D.C. and Iglewicz, B. (1989) "Some Implementations of the Box Plot," *The American Statistician*, 43,50-54.

Kosslyn, S. M. 1994. *Elements of Graph Design*, W.H. Freeman and Company, New York, NY.

Loveland, T.R., Merchant, J.W., Reed, B.C., Brown, J.F., Ohlen, D.O., Olson, P., and Hutchinson, J. (1995) "Seasonal land Cover Regions of the United States," *Ann of the Assoc of American Geographers*, Vol. 85, 339-355.

McGill, R., Tukey, J.W. and Larsen, W.A. (1978) "Variation of Boxplots," *The American Statistician*, Vol. 32, 12-16.

Monmonier, M. (1988) "Geographical Representations in Statistical Graphics: A Conceptual Framework" in *1998 ASA Proc of the Section on Stat Graphics*, 1-10.

Monmonier, M. (1993) *Mapping It Out*, University of Chicago Press, Chicago, IL.

Olsen, A. R., Carr, D.B., Courbois, J.P, and Pierson, S.M. (1996) "Presentation of Data in Linked Attribute and Geographic Space," Poster presentation, ASA Annual Meeting.

Omernik, J.M. (1987) "Ecoregions of the conterminous United States," *Ann Assoc Am Geographers*, Vol.77, 118-25.

Omernik, J.M. (1995) "Ecoregions: a spatial framework for environmental management," in *Biological Assessment and Criteria: Tools for Water Resource Planning and Decision Making*, eds. Davis, W.S. and Simon, T.P. Lewis Publishers, Boca Raton, FL, 49-62.

Omernik, J.M. and Bailey, R.G. (1997) "Distinguishing between watersheds and ecoregions," *J of the Am Water Resources Assoc*, Vol. 33, 935-49.

Scott, D.W. (1992) *Multivariate Density Estimation, Theory, Practice and Visualization*, John Wiley and Sons, New York, NY.

Small , C.G. (1989) "A Survey of Multidimensional Medians," Tech Report Stat-89-08, Dept of Statistics and Actuarial Science, University of Waterloo.

Tufte, E.R. (1983) *The Visual Display of Quantitative Information*, Graphics Press, Cheshire, CT.

Tufte, E.R. (1990) *Envisioning Information*, Graphics Press, Cheshire, CT.

Tukey, J.W. (1993) "Graphic Comparisons of Several Linked Aspects: Alternatives and Suggested Principles," *JCGS*, Vol. 2. No. 1, 1- 33.

Dan Carr
*Institute for Computational Statistics and Informatics*
*George Mason University*
dcarr@galaxy.gmu.edu

Tony Olsen
*EPA National Health and Environmental Effects Research Laboratory*
tolsen@mail.cor.epa.gov

Suzanne M. Pierson
*OAO Corporation*
spierson@mail.cor.epa.gov

Pip Courbois
*Oregon State University*
courbois@stat.orst.edu

# Exploring Time Series Graphically

By Antony Unwin and Graham Wills

## *Introduction*

Interactive graphical tools are available in a number of packages (Data Desk, JMP, SAS Insight and to a lesser extent LispStat, S+ and others), but none provide interactive tools for working with time series. This is surprising, as graphical displays have always been considered important for time series analysis. The natural time ordering of the data makes graphs very informative and the lack of independence of the data makes simple summary statistics less useful than in other contexts.

Diamond Fast was designed to provide some basic tools for exploring multivariate, irregular, short time series where analytic methods could not be applied, but it proves to be valuable for working with univariate, regular, long series too [Unwin and Wills]. The main idea is to display series in a common window, to query them, to move them around, and to compare them. Its a bit like working with icons in a window. As the package has not been widely publicized but is still in use, we thought it would be helpful to summarize its main features here.

## *Scaling*

Most books on statistical graphics explain at length how scales should be chosen and drawn. There are some specific recommendations for time series in Cleveland [1993], where he points out that different aspect ratios may be necessary for studying local and global structure. This is not so relevant for interactive graphics where a series can be interactively rescaled both vertically and in time and where not only the scales, but also individual points can be directly queried. Figure 1 shows weekly values of the Dow Jones Industrial average between 1929 and 1932. The series has been queried at its maximum before the crash. The same mouse click can call up information on a point, on an axis or on the graph itself, depending on where you click.

The main method of changing the vertical scaling in Diamond Fast is to grab the series graph and pull it upwards to magnify or downwards to shrink. This is very effective as you can see how the series changes and judge the effect you want to obtain. Technically this is achieved by distorting the current screen image and when the mouse button is released an accurate picture of the graph is drawn at the new resolution.

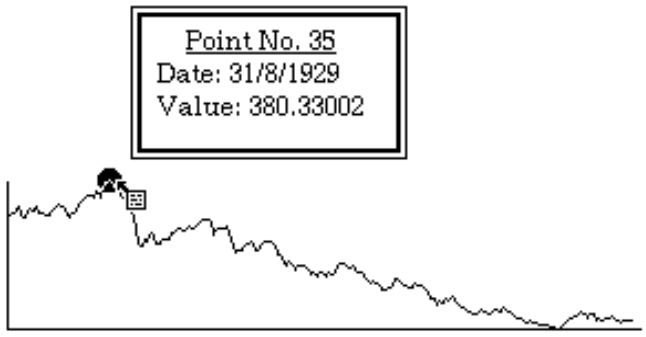



**Figure 1.** Querying time series (Dow Jones Industrial Average 1929-1932)

Changing time scaling is carried out for all series on the same screen simultaneously. This can be done dramatically by a factor of 10 with a single click or more precisely by dragging a scaling bar. Here are two views of monthly Canadian unemployment data. The graph on the left of Figure 2 shows all 20 years of the series (240 points) but has been crushed into a very short time-scale. The overall trend and range of seasonality are clearly shown. The graph on the right of Figure 2 shows only some of the series, just under two years, at a scale 100 times bigger. The points have been marked to emphasize the effect.

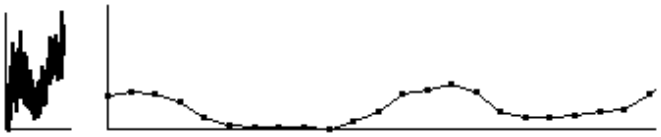


**Figure 2.** Differing time scales (Canadian Unemployment 1956-1975)

Another issue is whether to mark the points at which data are recorded or to show a smooth curve or both (as in the graph on the right above). Obviously, any recommendation must depend on the series being plotted. Dense, regular series (e.g. daily currency rates over several years) are better as a smooth curve. Irregular short series (e.g. temperature measurements of a patient at various times) should be displayed with points (and possibly lines joining them). In Diamond Fast you can switch from one alternative to another with a simple menu selection.

## *Overlaying time series*

Many time series have a strong seasonal component and it is necessary to incorporate this in a model to interpret short-term changes effectively, for instance in interpreting sales data or unemployment data. In Figure 3 you can see the monthly unemployment series for Canada overlaid on itself with a lag of one year. A copy of the series has been made and has then been picked up and overlaid on the original series. Changes in level, changes in seasonality and other features are visible in this static display with careful study, but interactive exploration (realigning the graphs, rescaling, zooming and interrogating, all carried out with direct mouse control) highlights them much more effectively.

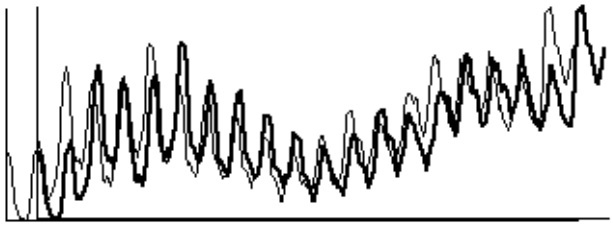


**Figure 3.** Overlaying time series (Canadian Unemployment 1956-1975)

## *Multiple time series*

Much of time series literature is concerned with analysis of single series and there are many different approaches for modeling the data [see Chatfield amongst others]. The literature for multidimensional series is much sparser and the models require strong assumptions about relationships between series. Some data sets can readily be modeled in this way but many cannot. Any assumption of stable relationships between series over lengthy periods can be difficult to justify. Sales data of sets of products are regularly recorded, but the relationships between the products may change over short periods of time due to pricing, distribution or advertising campaigns. Financial series may show strong relationships of one kind over one period and different relationships over another, groups of short-term and longer-term interest rates series being one example.

The tools needed to explore multidimensional series include being able to properly align series recorded with different frequencies or over differing time series, being able to overlay one series on another to explore lagged effects, (similar to what was done in Figure 3 for studying seasonality), and being able to commonly scale related series. It is also important to be able to move around and rescale individual series directly on screen.

For instance, in the following example displayed in Figure 4 we have three different displays of waterlevels at four French locations, correctly aligned in time. The leftmost panel shows that measurements started and finished at different times. Three of the series all have a step increase at about the same time. Displaying the series by points (the middle panel) shows that one

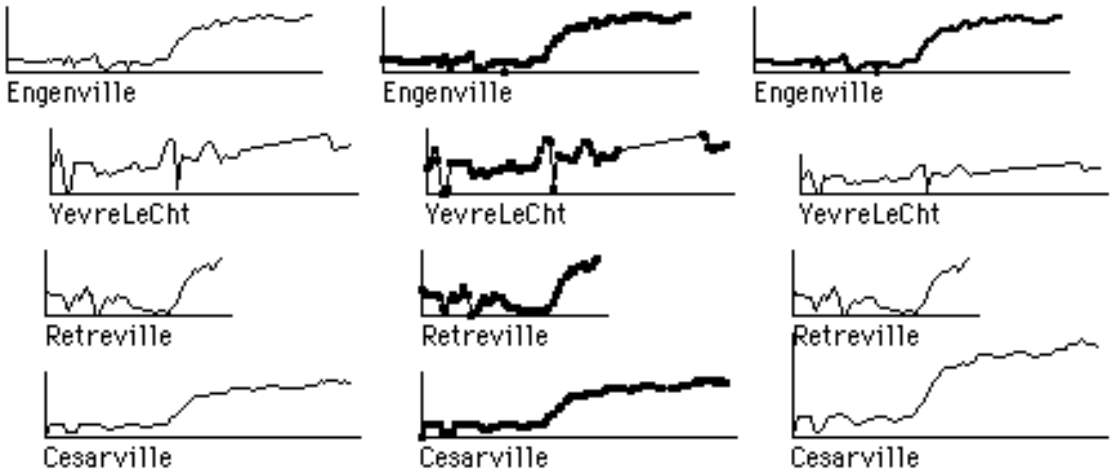



**Figure 4.** Viewing multiple time series in various ways (French water levels): (a) Default display, leftmost panel (b) Showing points, middle panel (c) Common scaling, rightmost panel.

(YevreLeCht) has two possible outliers and a gap in data recording. Scaling all series equally to the same scale as Engenville (the selected graph, the rightmost panel) shows that YevreLeCht has a smaller range and Cesarville has the largest. (Querying reveals that Retrevilles values are all higher than the others, though this could also have been seen by setting a standard origin for all four graphs.)

In practice it would be unusual to examine these displays in parallel. It is more effective to switch between a number of displays, examining possibly interesting features in more detail, choosing appropriate bases for comparions, zooming in and out, querying particular features and so on. Static displays can convey some of the results of exploratory analyses but none of the power of the process.

## Summary

This short note has illustrated some of the features of the software Diamond Fast for the interactive graphical analysis of time series data. Although the software is almost ten years old, most of its features are still not available in mainstream packages. Given the wide range of potential applicability of these ideas and their obvious value in practice, we can only conclude that this is another excellent example of the lack of use of exploratory graphical tools in statistics. Perhaps someone can explain to us why this is so.

## Technical details and availability

Diamond Fast runs only on Macintosh computers, but it can run on anything from a 10 year-old Mac Plus to a modern Power Mac (on which it will run rather faster). Enquiries about availability should be made to the first author.

## References

Chatfield, C. (1995) *The Analysis of Time Series: an Introduction,* (6th ed.) Chapman and Hall, London.

Cleveland, W. S. (1993) *Visualizing Data,* Hobart Press, Summit, NJ.

Unwin, A.R., and Wills, G. (1988) "Eyeballing Time Series," in *Proceedings of the 1988 ASA Statistical Computing Section*, 263-268.

Antony Unwin
University of Augsburg
unwin@uni-augsburg.de

Graham Wills
Bell Laboratories
gwills@lucent.com

# Simpson on Board the Titanic? Interactive Methods for Dealing with Multivariate Categorical Data

By Heike Hofmann

## *You've read the papers? Seen the film? Now look at the data!*

Using passenger data of the Titanic disaster as given in Table 1, the following article illustrates some fundamental aspects when working with multivariate categorical data.

For this purpose we show a commented session of the software package MANET – a tool for exploring data interactively and, as we think, intuitively.

### *A First Look at the Data*

Even a static display of the marginal distributions in conventional and well-known barcharts reveals a lot of information on *Class*, *Age*, *Sex* and Survival which is not so easy to detect in the counts of Table 1.

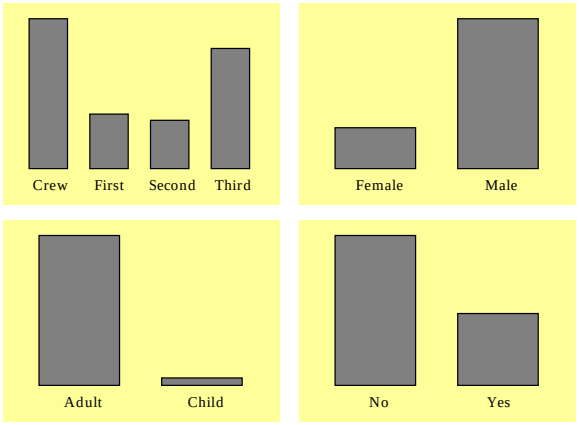



**Figure 1.** Four bar charts showing the marginal distributions according to class, age, sex and, of course, survival of those on board the Titanic.

### *Selection & Highlighting*

The most fundamental technique of interactive graphics is the possibility to **select** a part of a graphic, **mark** it (use e.g. color, shading or a different symbol) and at the same time also **mark** all corresponding parts of data in any other linked graphic. Figure 2 again shows the four barcharts from Figure 1. But now, survivors are selected – together with the conditional views of *Survival* dependent on the other variables.

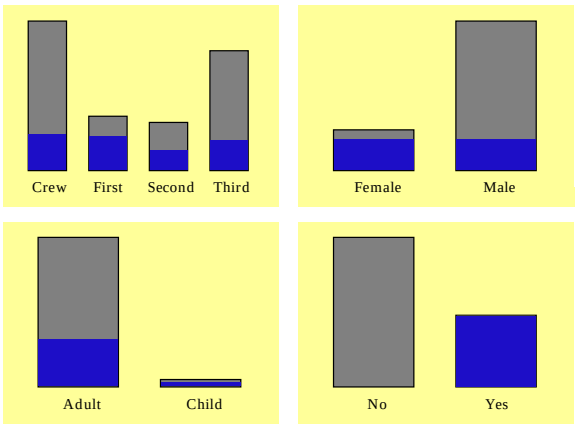



**Figure 2.** Four barcharts with marginal distribution of the properties. Highlighted (dark) are survivors.

Notice that about as many females as males survived the shipwreck. To get exact values another feature is necessary.

### *Interactive Querying*

Results of an interactive query (by clicking with a question mark on the point of interest) in a graphic are dependent on the type of the graphic as well as on the specific part queried. Clicking on a bin in a barchart (see Fig. 3) shows the category name, the number highlighted and the total count.

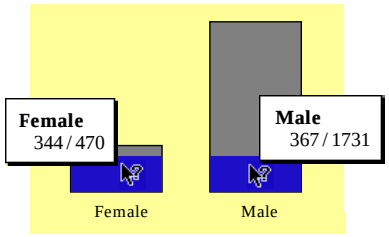



**Figure 3.** Interactive query in a barchart gives the name of the bin, counts and number of selected.

### *Spineplot*

When relative survival rates of women and men are of interest rather than absolute counts, one would have to calculate the survival rates: $344/470 \approx 73.2\%$ (women), $367/1731 \approx 21.2\%$ (men). Another representation is more useful to compare them.

In contrast to a barchart, counts in a category are represented by the **width** of the bin in a **spine plot** (Hummel, 1996). Highlighting proportions are still drawn from

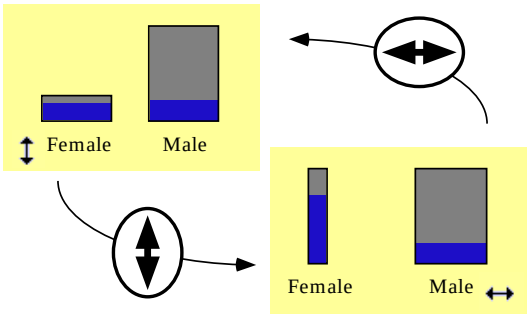



**Figure 4.** Switching between bar and spine representation – the arrow-symbols within the circles show whether to concentrate on width (spine plot) or height (barchart) of a bin.

bottom to top. Therefore heights of highlighting can be interpreted as rates within categories and can be compared directly (see Fig. 5).

In practice, as spines and bars are not essentially different from a data point of view, there is only one kind of graphic necessary and a switch to change representation (see Fig. 4). The switch is realized by so-called *cues* – 'hot' areas in a graphic where the cursor changes its shape to indicate a possible action via clicking.

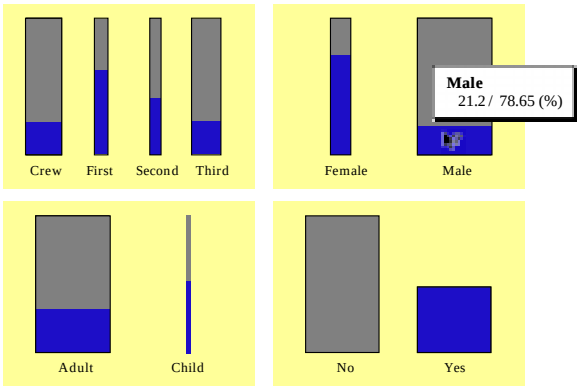



**Figure 5.** Spineplots of Class, Age and Sex. Survivors are still selected. Differences in survival rates within each variable are easy to read off by comparing heights.

Together with the representation the kind of information changes. The result of a bin-query as in Figure 5 is the name of the bin as before but the numbers of highlighting and counts are given as proportions; proportion of highlighting within the category as well as the proportion of the category within the total data set.

Highlighting in spineplots is represented fundamentally differently from the total numbers and therefore can be seen as adding a further dimension to the graphic.

### *Mosaic plots or 'women and children first'*

Mosaicplots are a graphic for visualizing multivariate categorical data. The basic idea of spine plots is generalized.

Each combination of the variables is represented by one bin, the area of each bin is proportional to the number of cases within the combination (for a more detailed description of the construction see Hartigan & Kleiner 1981 and Theus 1996). Mosaicplots therefore are the graphical analog of a multidimensional contingency table – Figure 6 for example contains all the information of Table 1.

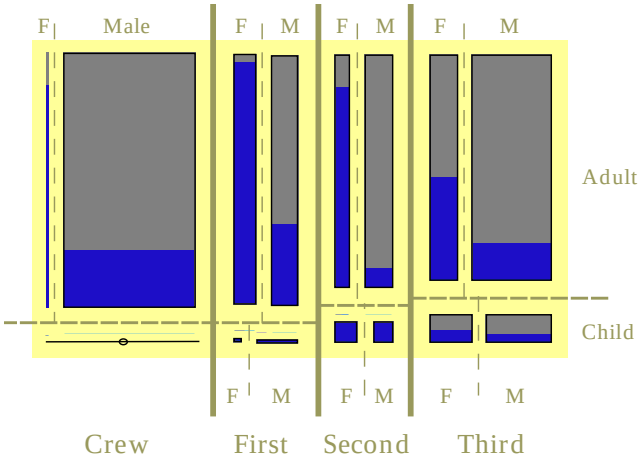



**Figure 6.** Mosaicplot of persons according to Class, Age and Sex. Highlighted are survivors. The motto 'women and children first' is easy to recognize as the pattern of survival.

The feature emphasized in Figure 6 is the difference of survival rates between men and women within each combination of class and age. Besides one can see that the proportion of children on board rises from first to third class while there were not any children in the crew (indicated by the printed zero on the line). All children in first and second class survived independently of sex while with children in the third class the same effect occurs as with the adults: females have by far higher survival rates than males.

There are several points to note:

- Labelling the bins would be helpful for interpretation, yet this leads to excessive space requirement in higher dimensions.
- Due to the strictly hierarchical construction of mosaics the order of variables has a crucial influence on the visual effect in the graphic.

- It is not easy to compare areas even if they are rectangular.

These problems, however, are more of a static nature and easy to solve using interactive methods.

Both the second and the third point originate from the same problem: the order of variables. While it is not easy to compare different areas, comparing heights or widths is. By reordering variables most comparisons concerning different areas can be re-expressed in terms of height (or width) of bins. Effective and feasible features for interactive reordering are therefore required.

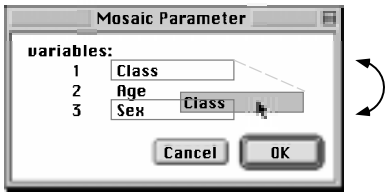



**Figure 7.** Changing the order of variables in a mosaicplot via drag & drop.

Figure 7 shows a mosaic's dialogbox for changing the order of variables by drag & drop. What cannot be shown in Figure 7 is, that while the parameter box is open the associated mosaic plot is changed in the background corresponding to the user's action. The result of an action is instantly visible which makes comparisons between different orders of variables easy.




**Figure 8.** Mosaicplot of the data in Table 1. The ordering of variables is Sex, Age and Class. Survivors are highlighted. Stress is laid on survival rates by class within sex.

As for the problem of labelling the bins, Figure 8 also provides a solution: interactive querying not only reveals the number of cases but also the current combination of categories.

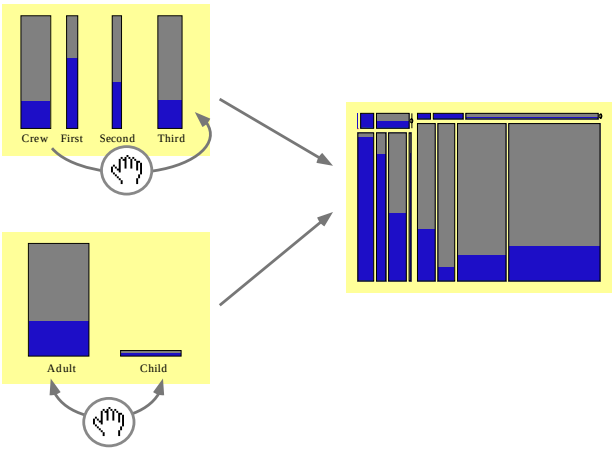



**Figure 9.** Reordering bins in barcharts forces reordering in mosaics.

### *Ordering of Bars*

Categories in discrete data have generally no default order and therefore most standard packages use lexicographical ordering.

Instead of having to rename categories *'00-january'*, *'01-february'*,..., it is much more helpful and transparent to have the possibility of re-ordering bins interactively, especially considering that during the process of exploration quite different orders can become of interest and one would most certainly not consider continually renaming categories by hand.

Figure 9 shows the re-ordering of two MANET-barcharts. Any kind of change within the barcharts is recorded and if necessary forces changes in other graphics, too. This is an example of the difference between mere **data linked** and a more general **variable linked** interactivity.

### *Grouping of Bars*

Another very useful feature in barcharts is **regrouping** of bars. In Figure 10 the three different classes of passengers are combined to one class.

Of course this change in a variable is also transferred to the representation in a mosaicplot. Figure 11 shows a mosaic plot of persons (now only distinguished between passengers and crew members) vs. sex. Highlighted are survivors. By comparing the rates one can see that both female and male crew members had better chances of surviving than female or male passengers. When collapsing over sex as in the spine plot this result is inverted though. The reason for this case of Simpsons paradox (Simpson, 1951) is of course the strong interaction between sex and passenger/crew, which can also be seen in the mosaic.

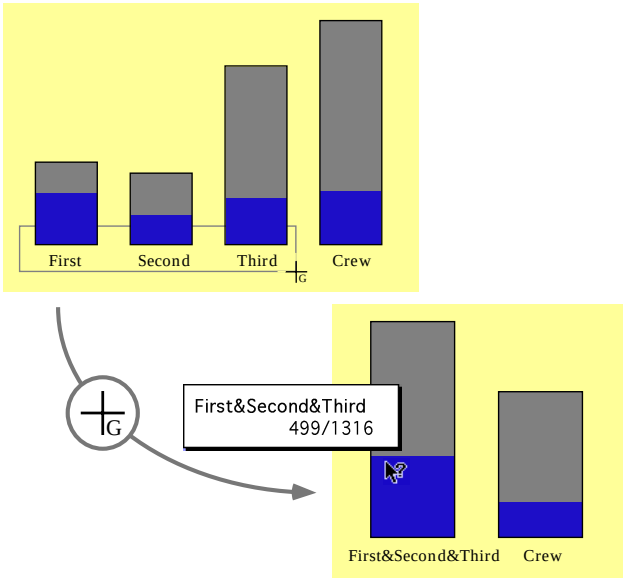



**Figure 10.** Grouping in a barchart. Several bins are collected into one bin. A bin-query now provides summary information on this bin.

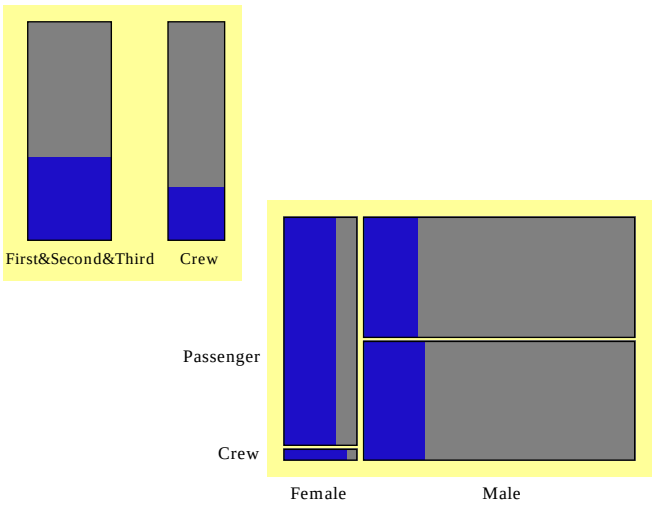



**Figure 11.** Example for the Simpson-paradox. While both male and female crew members have higher rates of survival than passengers (see mosaic), collapsing over sex as in the spine plot reverses this result.

### *Look at the results interactively*

The results of this paper can be looked at interactively in a video on our homepage `http://www1.math.uni-augsburg.de/`, where there is also more information available on the software MANET (as well as download possibilities) and on the work of our research group at the university of Augsburg.

| **Persons on board the Titanic** | | Survived | | Died | |
|---|---|---|---|---|---|
| | | F | M | F | M |
| Crew | | 20 | 192 | 3 | 670 |
| First | Adult | 140 | 57 | 4 | 118 |
| | Children | 1 | 5 | 0 | 0 |
| Second | Adult | 80 | 14 | 13 | 154 |
| | Children | 13 | 11 | 0 | 0 |
| Third | Adult | 76 | 75 | 89 | 387 |
| | Children | 14 | 13 | 17 | 25 |

**Table 1.** Overview of persons on board the Titanic according to class, age, sex and surviving.

Source: MacG. Dawson Robert J. (1995) "The 'Unusual Episode' Data Revisited," *Journal of Statistics Education*, Vol. 3, No. 3.

### *References*

Hartigan J.A. & Kleiner B. (1981) *Mosaic for Contingency Tables*, Computer Science and Statistics, Proceedings of the 13th Symposium on the Interface, pages 268-273.

Simpson, E.H. (1951) "The interpretation of interaction in contigency tables," *J. Royal Statist. Soc. (B)*, Vol. 13, pages 238-241.

Theus, Martin (1996) *Theorie und Anwendung Interaktiver Statistischer Graphik*, Wißner Verlag, Augsburg.

Heike Hofmann
University of Augsburg
`hofmann@math.uni-augsburg.de`

# Just Make It!

By Duncan Temple Lang and Steve Golowich

Except for all the others, computer files can be divided into three categories: those we create directly; those created by the computer itself as output from commands; and those created through interactive sessions (e.g. *S*). The first group consists of things like *C* and *Java*$^{TM}$ source code; LaTeX files; shell, *S* and *Matlab* scripts, etc. The second group includes compiled code; executables; libraries; DVI, Postscript, and PDF files; tar and jar files; and the list goes on. Undoubtedly, the most intellectual effort goes into creating elements of the first group. However, a surprising amount of manual effort goes into telling a computer how to generate elements of the second group based on the first group. This article discusses how to reduce this effort and specifically focuses on using the application *make*.

Many readers will be somewhat familiar with the general ideas of *make* and may use it for different projects. Typically, these involve compiling *C* or *Fortran* code. *make* takes care of managing the relationships between the different files and replacing the manual commands with an automated way of building the system. Basically, it removes the tedium from involved deterministic procedures. This not only makes it more reproducible so that others can use code, etc., but is also much less error-prone. It exploits the talents of the computer—tedious, rule-based, repetitive tasks.

An often overlooked fact about *make* is that it can be used for more modest and less complicated applications and projects. For example, when creating documents that include figures generated by S scripts based on datasets stored in other files, it is important to ensure that the figures are up-to-date with respect to both the data and the S code. *make* can determine when it is necessary to update such figures, as well as execute the commands to do so. Another common scenario in our work is the periodic arrival of data that we analyze locally. We use *make* to both create transformed versions of the data for use in different applications and also ensure that these derived versions are synchronized with the original source. As a final example, *make* can be used to maintain files and directories in a Web site by creating HTML files from TeX documents, etc., and copying only the relevant files to the public area.

The examples above illustrate the flexibility of *make*. The good news is that all of these can be done quite simply. This article's message is that *make* is a good thing with which to be familiar. It is the most widely used tool for installing applications but is also useful for personal projects, be they multi-directory or single file endeavors. *make* can save you time by condensing multiple commands (using arbitrary languages and applications) that you repeat frequently into single commands. It is capable of minimizing the amount of work the computer does. Finally, it can reduce the time tracking subtle bugs caused by simple but common mistakes.

The focus of this article is on a general strategy for reusing the inputs to make. To describe this we have to discuss the basics of *make* and what it does, which we will do with a brief example. Then we will describe the general configuration. All of the material discussed here, along with more details, is available from the Web site [1]. Feel free to use, enhance and comment on it.

Before we get into the details, we will just mention a few general things about *make*. It grew up in the UNIX world but there are several implementations that run on all sorts of different operating systems, including Windows 95 and NT, etc. Indeed, most of the project management facilities in the integrated development environments (IDEs) such as Visual *C++*, Café, etc. are simpler front-ends to *make*. In several places in this article, we use some features that are particular to the GNU dialect of *make* and recommend that you use it. There are several other tools that are related to *make* and installation of files. These include *autoconf* and *imake* and are worth investigating when considering portability of make files across machines and operating systems.

## Basic make Concepts

So what is *make*? It is a program which reads and obeys *rules*. Each rule defines relationships between files and consists of two parts.

1. A target file is considered out-of-date relative to a list of prerequisite files if any of these prerequisites have been modified more recently than the target (if the target does not exist, it is considered out-of-date). The first part of the rule specifies the target(s) and the list of prerequisite files.
2. The second part is the body of the rule and consists of one or more commands to be executed if the target is determined to be out-of-date. This is used to create a new version of the target.

*make* consists of both a language for defining these rules and variables with which one parameterizes them, and an engine or solver that links them together and satisfies them. Rather than specifying a "program" for bringing files up-to-date (e.g. a shell script), with *make* one specifies constraints between files which *make* con-

verts into dependency graphs connecting the targets. Then it traverses these graphs to bring one or more of the targets up-to-date as necessary using the bodies of the rules. From one perspective, the input to *make* is a small archive of simple commands that are indexed by relationships between files. You may just want to collect several independent commands in a single location rather than using multiple shell scripts or shell aliases. Then the targets are just simple names and are unrelated to files. Alternatively, you can specify a complicated web of dependencies as individual elements that *make* will connect and resolve. From *make*'s perspective, the input is a blueprint for what to do and when to do it.

That is the abstract notion of *make*. An example will help clarify things as well as allow us to look at the syntax of the input. Rather than the usual example of compiling *C* code, we will use this document as an example of using *make*. Basically, our approach for LaTeX documents is to have each section in a separate file. The overall document is described in another file (call it **doc.tex**) and is a simple collection of LaTeX `input` directives for each of the sections. The preamble (macros, document class and `\begin{document}` declarations) are in a file called **header.tex** and the postamble in **footer.tex**. This approach has several advantages, one of which is that it allows us to LaTeX individual sections.

So what are the things that we want to be able to do when writing this document? Obviously we need to create the DVI and Postscript files for the entire document. We also want to be able to create these files for the individual sections. We need to spell-check only the section files. We want to remove all of the intermediate files – the **.dvi** and **.ps** files for the sections. Each of these operations will be a rule for *make*. But to start, we will create a simple way of referring to all of the different section files. *make* supports variables to store strings or list of strings. So we tell *make* to store the list of sections filenames in the variable `${SRCS}` with the line

```
SECTIONS= intro makeIntro GenericDesign \
 AutoDepends MultiDir Debugging Summary
```

We don't bother specifying the extension (**.tex**) for each file because *make* has several functions that allow us to operate on strings and lists using regular expressions. We can create a new variable from `${SRCS}` by appending **.dvi** to each of the elements with

```
$(SECTIONS:%=%.dvi)
```

Note that we refer to the value of a variable using the $ symbol and enclose the name of the variable in () or {}. Variable names consisting of a single letter need not be enclosed in parentheses.

The first rule we create specifies how to make the DVI file for the complete document. This file should be recreated if any of the sections or **doc.tex**, **header.tex** or **footer.tex** files have been changed more recently than the **doc.dvi** itself. To (re)create this file, we run **`latex`** twice, then **`bibtex`** and then **`latex`** again to get all the cross-references and citations correctly calculated. The following rule specifies both the dependencies and the commands to generate the file.

```
doc.dvi: doc.tex $(SECTIONS:%=%.tex) \
              header.tex footer.tex

        $(LATEX) $*
        $(LATEX) $*
        bibtex $*
        $(LATEX) $*
```

Because the command for invoking LaTeX differs from machine to machine, we refer to it with a variable so we only have to change it in one place. This isn't as necessary for **`bibtex`**. The only mysterious symbol is `$*`. This is a variable that is provided by *make* for use within the body of a rule. There are numerous of these *automatic variables* referring to the target name, the prerequisites that are more recent than the target, the most recent prerequisite, etc. `$*` is the name of the target with the extensions removed. These help to generalize the body of the rule rather than hard code the target (**doc**).

One important point to remember about *make* is that each line in the body of a rule must start with a TAB character. This can be the cause of frustrating bugs and makes `makefile mode` in emacs a useful editing tool. Each line in the body is executed in its own shell. However, commands separated by a ';' on a single line are run consecutively in a single shell.

The next rule relates to Postscript files. The command to create a **.ps** file from a DVI file a simple application of **`dvips`**. It is the same for all DVI files and the dependencies are also simple—just the DVI file. Rather than enumerating the rule for each Postscript target, we can write a general rule that specifies how to make any **.ps** file from a **.dvi** file:

```
%.ps: %.dvi
        $(DVIPS) -o $@ $<
```

The % says to match any target ending with **.ps** and create it from the corresponding **.dvi** file using this command. This rule suffices for **doc.ps** and all the different individual sections. The variables `$@` and `$<` are provided by *make* and give the target name (e.g. **doc.ps**) and the most recent prerequisite (**doc.dvi**), respectively.

Note that with these two rules specified, *make* can infer that to create **doc.ps**, it must first make **doc.dvi**. At this point, you can give the command '`make doc.ps`' at the shell prompt and *make* will do the right thing!

Now, to create the DVI file for a given section, we can use a general rule again that relates any **.dvi** file to its **.tex** file. The rule is

```
%.dvi: %.tex
        $(LATEX) "\\input{header} \
                  \\input{$*}     \
                  \\input{footer}"
```

The body tells LaTeX to read its input not from a file, but from the command line. This takes care of inputting the relevant files with the section in the middle referred to by `$*`. The double slashes (`\\`) are needed to cause *make* to escape these from the shell and itself correctly. Notice that there are now two rules for creating **doc.dvi**—a generic one and a specific on. *make* knows to use the more specific rule.

At this point, the hard part is done. The remaining rules are

```
spell:
        ispell $(SECTIONS:%=%.tex)

.PHONY: clean
clean:
        -rm $(SECTIONS:%=%.dvi) doc.dvi
        -rm $(SECTIONS:%=%.ps)
```

There are two curiosities here. The clean rule is declared as a "phony" target, meaning that it is not associated with an actual file and is always considered out of date. Hence, the action of the command '`make clean`' will be always to execute the body of the rule. Also, The '-'s in the body of the clean rule indicate that *make* should ignore errors from these commands and continue no matter what. This may be necessary since some of the DVI files may not exist but we want to ensure that the next line of the body is executed.

All these commands go into a file, typically called **GNUmakefile** or **makefile**. Then you can build one or more of the targets with the command '`make target`'. As an aside, I rarely type these commands as most of the modes in emacs (for *C*, *Java*™, LaTeX, etc.) can be easily configured to call make with the appropriate target name.

One last thing to note is that you can and should put comments into your makefiles. Anything on a line after a '#' character is ignored. Some rules can get complicated or subtle. When you revisit them they can be obscure!

## Reusing Rules & Variables

We have covered enough about *make* to understand how to specify quite general things. There is a lot more functionality that you can read about in [3]. But now we can turn to making it easier to use *make*. Suppose that we have a few different documents or several unrelated *C* applications, each in their own directory. In each one, we can use a makefile to specify the rules for creating the different targets. But rules such as the one for creating a Postscript file from DVI file are the same for all documents, and compiling a *C* file is the same for each application. If they do differ, it is most likely in arguments which we can parameterize and override with *make*'s variables rather than copying the entire rule. What we would like to do is to identify the rules and variables that can be shared across different projects and store these in different makefiles that each project's makefile would read. Well, GNU *make* lets us do exactly that. Thus we need only create these centralized files once and use them everywhere. If we find a mistake in them or a better way to do something, we just change it in this one place and it is automatically available to all of the different projects. We still have the flexibility to override variables and rules to account for peculiarities of a particular project. We can also specify "macros" or pieces of text that can be used in bodies of rules that parameterize certain commands.

The command for instructing GNU *make* to read another makefile *in situ* is '*include*'. It takes a list of one or more names referring to the files to include and acts as if the contents of those files were actually located in this makefile. We can use this by having a standard, or template, **GNUmakefile** that is copied when starting each new directory/project. It *include*s two files—one containing the general rules (**.GNUmakeRules**) and the other containing variables that parameterize those rules (**.GNUmakeConfig**). These included files can also include other files themselves which add rules for more specific purposes and which provide values for variables particular to a machine, operating system or compiler. The basic layout of all these files is displayed in Figure 1 and explained below. The template makefile for a project might look something like

```
# variables to control the
# 2nd level includes
include $(HOME)/.GNUmakeConfig

# other variables (possibly overriding
# central ones)
..
include $(HOME)/.GNUmakeRules

# project-specific rules
```

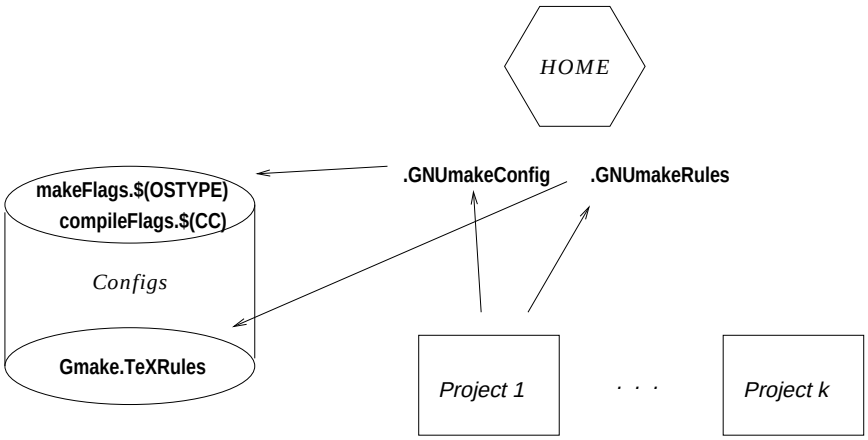



**Figure 1.** Included Makefiles Configuration.

It has three sections punctuated by the two general includes. Before **.GNUmakeConfig** is included, a project can preempt some of these variable definitions to customize how lower level include files are read. In the second stanza, we add new variables such as listing the source files, etc. Also, we can override some of the general variables just included as necessary. Then we include the general rules file in which we also compute some convenience variables (e.g. all of the **.dvi** or **.o** files) from the existing variables. Finally, we can specify our own project-specific rules. This framework provides the flexibility to ignore or override some of the general settings but without having to manually pick and choose which ones we want.

We put the two included files in our home directories so that it is easy to locate and refer to them. All the files that they include, however, are put into another directory which is specified in a *make* variable. This avoids clutter in our home directories.

Since the variables in the **.GNUmakeConfig** file are primarily used to parameterize the general rules, it makes sense to discuss the contents of the file **.GNUmakeRules** first. We start this file by defining variables that are computed from other variables according to some simple naming conventions. For example, we use the variable `${SRCS}` to list all the source files that need to be compiled in a given directory. We use `${OBJS}` to store the compiled filenames and compute it from `${SRCS}`. We can do this by removing the extensions **.c**, **.cc**, etc. from the different filenames and appending **.o** to each of the resulting elements. The following line does this

```
OBJS=$(addsuffix .o,$(basename  \
       $(filter %.c %.cc %.cxx, \
         $(SRCS))))
```

and ilustrates some of the useful functions of GNU *make*.

An important thing to consider when defining variables in this and the **.GNUmakeConfig** file is whether they have already been declared. Since the regular or toplevel makefile can declare these variables anywhere, we need to enclose these centralized definitions inside conditionals. Again, GNU *make* helps here by providing a collection of *if* operations. The following example can be applied to most variables.

```
ifndef LATEX
 LATEX=latex2e
endif
```

This tests to see if the variable is already defined, and if not, defines it.

The same caution applies to defining rules and we can enclose one or more rules inside a test for a particular variable being defined. For example,

```
ifndef NoTeXRules
include \
  $(MAKE_CONFIG_HOME)/GNUmake.TeXRules
endif
```

only reads the specified file if the variable `${NoTeXRules}` has not been defined. This approach allows different rules to be ignored or excluded when necessary by simply defining the appropriate variable in the project's makefile. The example also illustrates that it can be useful to keep unrelated rules in different files and include them rather than in-lining them. This *include* command reads the LaTeX rules. In addition to making the rules file more readable, separating the different groups into different files also allows different subsets of files to be more easily read rather than the whole collection.

There is a caveat to using *ifndef*. *make* defines some common variables, such as `${CC}`, when it starts. So they are defined, but not by the toplevel makefile. However, we can test *where* a variable was defined using the *origin* function (see [3],p 85) For example,

```
ifeq "$(origin CC)" "default"
 CC=gcc
endif
```

assigns the value gcc to the variable CC only if the latter was not explicitly defined before, but just implicitly by *make*. Other possible sources for a variable definition include the command line and a shell environment variable.

The remainder of the **.GNUmakeRules** file declares the different collections of rules that might apply to different projects. These include rules for compiling *C, C++*, *Fortran* and *Java*$^{TM}$ code; rules to clean directories of intermediate and installed files, etc. In addition to containing a particular collection of rules, this file is a useful location for adding general rules as you find a need for them.

The **.GNUmakeConfig** file should contain definitions for the different variables used to parameterize the generic rules. They should all be defined conditionally to ensure that you don't override the toplevel makefile's definitions. In addition to regular variables, GNU *make* allows us to define verbatim variables, or macros, for collections of commands. These aliases can be used in the bodies of rules to simplify their specification. Like simple assignments, they are not evaluated until needed so they can refer to other variables that are not yet defined (e.g. all of the automatic variables, and ones that the user may set later). Examples of useful aliases are for linking applications; the body of the rule in the previous section for running LaTeX on a complete document with **`bibtex`**; etc. The definitions of these macros look something like

```
ifndef c-link
define c-link
  $(CC) -o $@ $(LDFLAGS) $+ $(LIBS)
endef
endif
```

and can be used in rules like

```
app: $(OBJS)
        $(c-link)
```

to link object code to create an executable.

As in the rules file, it is useful to group collections of variables in other files and have these *include*d by the file **.GNUmakeConfig**. There is more of a need for it here than in the rules file as the value of one variable may dictate the default values of others. For example, different compilers use different flags. If the value of `${CC}` is gcc, one set of flags is appropriate. However, if the Solaris compiler is specified, a different set of flags should be set for compiling and linking. This can be achieved with a collection of *ifeq .. else .. else .. endif* clauses. However, this makes the file look complicated and on some machines certain clauses are redundant as the clause can never apply. A neater approach is to have separate files that specify the flags for each compiler. We give each of these a name based on the compiler name (e.g. **compilerFlags.gcc**) and then include the appropriate one with the directive

```
-include \
 $(MAKE_CONFIG_HOME)/compilerFlags.$(CC)
```

(Recall that the '-' tells make to go on if the file does not exist.) Similarly, we can include operating system specific variables by including a file whose name is a function of the variable `${OSTYPE}` and even machine specific commands using something like

```
-include                \
   $(MAKE_CONFIG_HOME)/\
   machineFlags.$(shell hostname)
```

## Automating Dependencies

What we have discussed so far moves the work of maintaining dependencies from us to *make*. However, we are charged with specifying the dependencies. For small projects, this is not excessively complicated and in the case of *Java*$^{TM}$ not even necessary. However, for compiling *C*/*C++* and Fortran code, the dependencies are more important. So, wouldn't it be nice if we could have these generated for us. Well, it turns out that a combination of *make* and a modern *C*/*C++* compiler can do this.

Most compilers have a flag which changes its task from producing compiled, or object, code to outputting a list of dependencies that *make* can read for the specified source files. For example,

```
  gcc -MM  nrutil.c
```

produces the output

```
nrutil.o: nrutil.c nrutil.h
```

because **nrutil.c** contains a line

```
#include "nrutil.h".
```

Now, this output can be inserted directly into the makefile so that whenever either of **nrutil.h** or **nrutil.c** is modified **nrutil.o** will be remade. Alternatively, we can filter the output from this command into a file, say **.nrutil.d**, and have *make* read it using the *include* directive. A simple rule to create the dependencies for a given *C* source file and filter them to the file ***.source-file-name*.d** is

```
.%.d: %.c GNUmakefile
        $(make-depends)
```

where the macro `${make-depends}` is defined as

```
$(CC) $(DEPENDENCIES_FLAG) \
      $(CFLAGS) $< > $@
```

(The value of `${DEPENDENCIES_FLAG}` is compiler-specific and can be defined in the general configuration files.)

Now, we arrange to have *make include* the **.d** files using the command

```
include $(SRCS:%.c=.%.d)
```

If any of these files does not exist or are out-of-date, GNU *make* will attempt to create it using the rule above and then read it.

This saves us a lot of work and can be put in the general rules file **.GNUmakeRules**. However, the very alert reader may notice something slightly wrong. If we modify any of the **.h** files, the corresponding **.o** file is recompiled, but the **.d** file is not rebuilt. So the dependencies may be incorrect. We must change the `${make-depends}` macro so that the dependency file itself depends on the same prerequisites as the **.o** file. The following does this

```
define make-depends
 $(SHELL) -ec "$(CC) $(DEPENDS_FLAG)      \
               $(CFLAGS) $(@:.%.d=%.c) | \
            perl -npe 's/($*\\.o[ :]*)/$@ \
                 \1/;' > $@"
endef
```

and can be digested at your leisure. If you understand this, you can pretty much get *make* to do anything! This is as bad as it gets.

## Multi-Directory Projects

If we have several related directories, we may be able to streamline the actions of *make* by consolidating common definitions and rules into a single configuration file. Such might be the case when LaTeX'ing the different chapters of a book, the lecture notes of a course, or when compiling the various components of a software project. We will use the example of a directory **Utilities/**, which contains subdirectories **LinAlg/** and **Opt/** which, in turn, contain the source code for object files that we use often.

Each of **LinAlg/** and **Opt/** should contain a makefile capable of recompiling just the files that it contains. In addition, the top level directory **Utilities/** should have a makefile capable of compiling all of the subdirectories at once, and perhaps even consolidating the resulting object files into an archive or shared library. With this setup, one *make* command suffices to do all of the compilations, which makes it very easy to regenerate the entire structure on a new machine. We can tie the various pieces together with a project-specific configuration file, e.g. **Utilities/UtilMake.config**, which we will include in all of the makefiles below **Utilities/**. This way, if something changes, we need only change it in one place and we can be sure the change will be seen everywhere.

The inclusion of **.GNUmakeConfig** should be done by **Utilities/UtilMake.config**, which should also contain any project-specific definitions. For instance, in our example we may wish it to contain

```
LINALG_DIR = $(UTILS_DIR)/LinAlg
OPT_DIR = $(UTILS_DIR)/Opt
include $(HOME)/.GNUmakeConfig
```

With these definitions, the subdirectories **LinAlg/** and **Opt/** can refer to each other. The variable `${UTILS_DIR}` is of global importance, since we will use it in other projects, so we put its definition in **Local.make** where all of our makefiles will see it.

The makefiles in **LinAlg/** and **Opt/** will start with

```
include ../UtilMake.config
```

and proceed with definitions and rules for compiling the sources into object files, most of which will employ the general rules in **.GNUmakeRules**. For convenience, we may wish to make the default target in each of these makefiles to be the object files derived from the sources in that directory.

The makefiles in **LinAlg/** and **Opt/** allow us to recompile the codes in those directories conveniently. How may we recompile both of them at once? The answer is to put a makefile directly in the parent directory **Utilities/** which recursively calls make on each of the subdirectories. For example, it might look like

```
include UtilsMake.config

ELEMENTS=$(LINALG_DIR) $(OPT_DIR)

.PHONY: all
all:    $(ELEMENTS)

include $(HOME)/.GNUmakeRules

.PHONY: $(ELEMENTS)
$(ELEMENTS):
        $(MAKE) -C $@
```

With this makefile, the command `make all` will result in *make* being called on each of the subdirectories

(the `-C $@` means that *make* will first change directories to `$@` before looking for a makefile on which to act). Because we decided on the object files as the default target in each subdirectory, the action of *make* at the top level will be to recompile all of the object files, but only if necessary. In actual use, we would also want to add an `install` target to the top-level makefile which would collect the object files into an archive or shared library, and also collect the header files from the subdirectories in a single place (e.g. **Utilities/**). The latter could be accomplished by adding an `install` target in each subdirectory, the rule for which would be to copy the header files to the appropriate place. The default `install` rules, in turn, could be centralized in **.GNUmakeRules**. The top-level makefile would then invoke `make install` on each subdirectory, as part of the rule of its own `install` target.

There are several benefits to the organization we have described. One is that each of the individual makefiles is relatively simple, since we have moved most of the complicated default definitions and rules to the central locations **.GNUmakeConfig** and **.GNUmakeRules**. It is easy to piece together many makefiles by recursively calling make, as we did above to recompile all of **Utilities/** with a single command. If some project-specific definitions must be made, these can be done in project-specific files like **UtilsMake.config**, which is included by every makefile associated with the project. The general rule we try to follow is, whenever possible, to condense related definitions into a single file and include it where needed.

## Debugging

Since *make* uses a language for the inputs, it may happen "ocassionally" that it does not understand what you meant. It may be your fault! So, we must have a way of determining why *make* thinks it should have done what it just did. Since it is a rule based language rather than a procedural one, it requires different styles of debugging. Other than reading the manual or thinking about the inputs from *make*'s point of view, we can ask *make* to give us some diagnostic information. There are a variety of different flags that can print *make*'s state, perform what-if type analysis and perform dry-runs. The common command-line flags for such explorations are `-n`, `-d`, `-q`, `-W`, `-p` and `--warn-undefined-variables`. You can read about them with the command

```
make --help
```

or in the manual [3] (page 95).

An alternative approach is to create new rules or add commands to existing rules to print out the values of different variables. Also, if your makefile includes a variety of other files, exclude some of them and see if that helps explain the problem. Essentially, a binary or split-search is a good tool in the absence of reading the manual!

## Summary

Hopefully this article has whetted some readers' interest in using *make*'s more unusual features or perhaps just to start using it for new projects. Using *make* can definitely get complicated, but usually when the situation would be too complicated to easily manage manually. Most importantly however, we hope that this article helps encourage people to spend a little time configuring tools which make their work easier and less error-prone in the longer term. Basically, *make the computer do the work it is good at*.

You can read more about *make* from a variety of different sources. The O'Reilly book [2] gives a description of the different dialects of make and the hows and whys of using *make*. The GNU *make* manual [3] provides the most complete information on GNU *make*. It can be read directly via the help system of many emacs installations. You can get GNU *make* from the usual GNU mirrors (see **w**www.gnu.org). The different makefiles that we use should also be available from the Web site [1].

## References


[1 ] `cm.bell-labs.com/stat/duncan/Make` Configuration files on Web site.

[2 ] Oram, A. (1991) *Managing Projects With Make*, O'Reilly & Associates.

[3 ] Stallman, R. M. and McGrath, R. (1998) "GNU Make, A Program for Directing Recompilation," GNU Documentation Project, `www.gnu.org/home.html`.



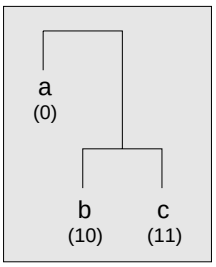
Duncan Temple Lang
Bell Laboratories
`duncan@research.bell-labs.com`

Steven Golowich,
Bell Laboratories
`golowich@research.bell-labs.com`

CONTINUED FROM PAGE 1

alphabet (or set) $\mathcal{A}$, a binary code $\mathcal{C}$ is a map from $\mathcal{A}$ to strings of 0's and 1's.

***Example*** (A binary, prefix code): Let $\mathcal{A} = \{a, b, c\}$. In Figure 1, we present a tree-based scheme for encoding the elements of $\mathcal{A}$. By associating codewords with the end-nodes of a binary tree, we form what is known as a *prefix* code, meaning that no codeword is the prefix of any other. This prefix property implies that we can encode messages without having to introduce separating characters between the codewords. For example, the string 0001110 can be uniquely decoded as $aaacb$.




$$\begin{aligned} \mathcal{C} : \mathcal{A} &\rightarrow \text{strings of 0's and 1's} \\ a &\rightarrow 0 \\ b &\rightarrow 10 \\ c &\rightarrow 11 \end{aligned}$$

**Figure 1.** Representing a code as a binary tree.

To discuss the code or description length of messages formed using the code in this example, we first need the length function $L : \mathcal{A} \rightarrow Z^+$. Here, $L(a) = 1$, $L(b) = 2$, and $L(c) = 2$. The function $L$ records length in units known as *bits* for *bi*nary dig*its*. In this particular example, we can re-express the code length function as follows:

$$L(\cdot) = \Leftrightarrow \log_2 P(\cdot)$$

with $P(a) = 1/2$, $P(b) = P(c) = 1/4$. So the code length function is just the negative logarithm of a probability function. This holds in general. That is, for any probability distribution $P(\cdot)$ on $\mathcal{A}$, the quantity $\Leftrightarrow \log_2 P(x)$ may be regarded as the code length of $x$ under some prefix code. This fact can be verified through an application of Kraft's inequality (Cover and Thomas, 1991):

***Kraft's inequality:*** A function $L(\cdot)$ is the code length of a binary, prefix code if and only if

$$\sum_{x \in \mathcal{A}} 2^{-L(x)} \leq 1.$$

Therefore, for any probability distribution $P$ on a finite set $\mathcal{A}$, the function $L(x) = \Leftrightarrow \log_2 P(x)$, $x \in \mathcal{A}$, (ignoring the issue of integer truncation) satisfies the above condition and hence is the code length of some prefix code on $\mathcal{A}$.

Given a probability distribution $P$, or equivalently the length function $\Leftrightarrow \log_2 P$, several algorithms exist that actually create the mapping $\mathcal{C}$ between elements of $\mathcal{A}$ and strings of 0's and 1's (see Cover and Thomas, 1991; or Hansen and Yu, 1998). For statistical applications of MDL, however, we don't actually have to exhibit a code, but instead focus on the coding interpretation of the quantity $\Leftrightarrow \log_2 P(\cdot)$.

So far, we have considered only codes for finite sets $\mathcal{A}$, but the extension to countable sets is straightforward. For most statistical applications, however, the set $\mathcal{A}$ will be perhaps an interval or even the entire real line. In such cases, a discretization or quantization has to be made. As we will see, this process adds only a constant term (reflecting the precision of the discretization) to the code length function. Therefore, for a fixed precision, we view $\Leftrightarrow \log_2 f(x)$ as an (idealized) code length for a density function $f$. We will return to discretization issues in the next section.

## 2. Forms of Description Length

We now introduce two methods for deriving valid description lengths of data. We focus on one example to illustrate the close ties between coding and statistical modeling. Consider two up-and-down indicators derived from the daily **return** and intra-day **volatility** of the Dow-Jones Industrial average. For each of the 6430 trading days between July, 1962 and June, 1988 we created a pair of binary variables. The first takes the value 1 if the return on a given day is higher than that for the previous day (an "up"), and 0 otherwise (a "down"). The second variable is defined similarly, but for the volatility series. For the daily returns there are 50.71% 1's or up's; while for the volatility data there are only 31.47% 1's or up's. We plot the last 1000 days of both series in Figure 2. The vertical bars in the panel beneath each plot mark an "up" day.

Now, imagine transmitting these indicators from one location to another, assuming that their length ($n = 6429$) is known to both sender and receiver. We'll first postulate an iid Bernoulli $(p)$ model for each up-and-down series. Take the daily returns, for example. The Bernoulli model implies that the up's and down's in this series were generated by a sequence of independent coin tosses, where the probability of a 1 (an "up") is $p$. Introducing such a probability model does not necessarily imply that we believe the true or data-generating mechanism to be this simple. In other words, the model is used solely to generate a code for our indicators, which by Kraft's inequality is equivalent to constructing a joint probability distribution on $\{0, 1\}^n$.

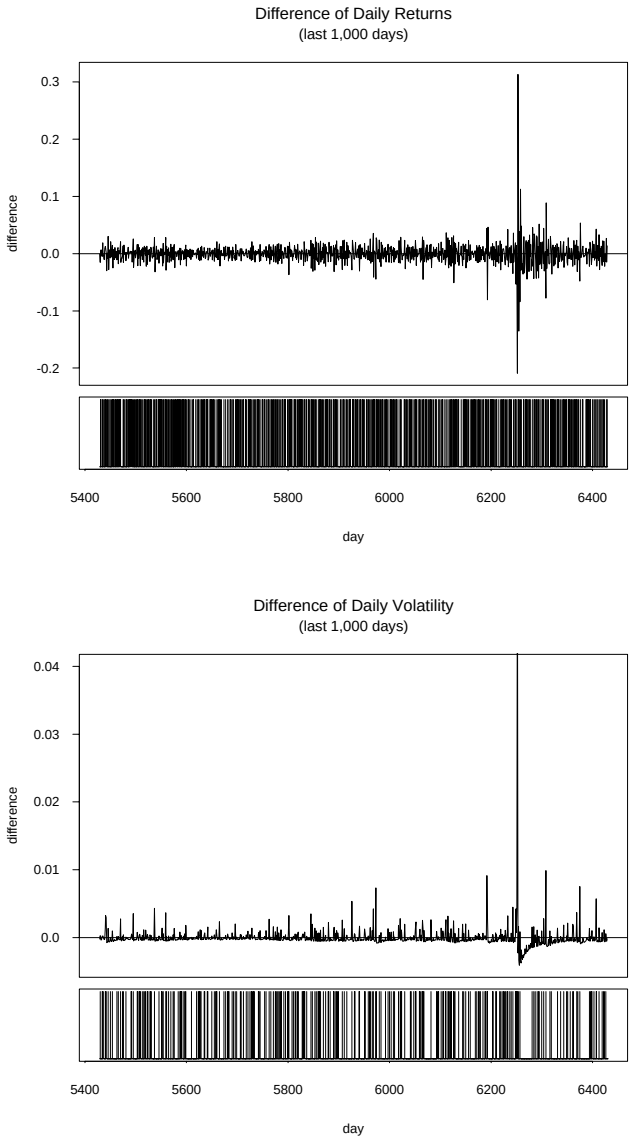



**Figure 2.** Difference of the volatility and return series. The horizontal line in each plot corresponds $y = 0$.

### *(i) Two-stage Description/Code Length*

Each choice of the parameter $p$ for the iid Bernoulli model generates a probability distribution suitable for encoding our sequences of up-and-down indicators. For a receiver to decode our message and retrieve the data, they have to know which code was used, or more precisely which value of $p$ was chosen. As its name would suggest, in a two-stage encoding scheme this information is sent first as overhead. The data are then transmitted in the second stage using the selected code.

For the iid Bernoulli model, we estimate $p$ by $\hat{p} = k/n$, where $k$ is the total count of 1's or up's. We communicate $\hat{p}$ by transmitting the value $k$. To do this, we build a code on the integers $\{0, \ldots, n\}$ using a uniform distribution. From the previous section, we know that it costs $\Leftrightarrow\log_2[1/(n+1)] = \log_2(n+1)$ bits to transmit any $k$ between 0 and $n$. Next, the data series is encoded using the codebook corresponding to the iid Bernoulli $(\hat{p})$ model. Having obtained $k$, the receiver can easily decode our message and recover the up-and-down data.

Combining the cost for each stage, we get a total code length of

$$L(\,\hat{p}\,) + L(\text{ data } |\,\hat{p}\,)$$
$$= \log_2(n+1) + \log_2[\,(k/n)^k(1 \Leftrightarrow k/n)^{n-k}\,]$$

to communicate our data using the simple iid Bernoulli model. Evaluating this expression for the return series, we find that this encoder requires 6441 bits to transmit the up-and-down string. For comparison, keep in mind that it only costs $n = 6429$ bits to send these indicators directly (as they are also binary strings). Therefore, the Bernoulli model is 0.2% more expensive. For the volatility series, on the other hand, we need only 5789 bits with the Bernoulli encoder, a 10% savings over direct transmission.

In general, given a model with a parameter (vector) $\theta$, the code length for a two-stage encoder takes the form

$$L(\,\hat{\theta}\,) + L(\text{ data } |\,\hat{\theta}\,).$$

This is no more than a penalized (log-) likelihood, where the code length for the estimated parameter $\hat{\theta}$ serves as the penalty. If we choose $\hat{\theta}$ to be the maximum likelihood estimate of $\theta$, then it can be shown that asymptotically this penalty is $\frac{d}{2}\log n$ where $d$ the dimension of the parameter. Essentially, one encodes each component of the parameter vector with precision $1/\sqrt{n}$ (the estimation rate), yielding a cost of $\Leftrightarrow\log_2(1/\sqrt{n})$ or $\frac{1}{2}\log_2 n$ for each.

### *(ii) Mixture Form of Description Length*

Rather than encode in two stages, suppose that we instead build a code using a mixture distribution. For the iid Beronoulli $(p)$ model, consider a uniform [0,1] prior on $p$ and integrate to form the probability distribution

$$m(\text{ data }) = \int_0^1 p^k(1 \Leftrightarrow p)^{n-k}dp = \frac{k!(n \Leftrightarrow k)!}{(n+1)!}.$$

Again, from Kraft's inequality, we know that this mixture can be used to generate a code with length function $\Leftrightarrow\log_2 m(\cdot)$. Under this scheme, we pay 6434 bits to encode the return indicators (roughly 0.1% more than direct transmission), while the volatility series costs us only 5782 bits (a 10% savings).

In a general model with parameter (vector) $\theta$, the mixture form of description length is given by

$$\Leftrightarrow\log_2 m(\text{ data }) = \Leftrightarrow\log_2 \int_\theta f(\text{ data } |\theta)w(\theta)d\theta,$$

where $w(\cdot)$ can be loosely thought of as a prior on the parameter space. Here, unlike in the Bayesian context, the role of a prior is purely functional, appearing only induce a joint distribution on the $n$-tuples. (And fortunately, to first order all smooth priors give the same code length in a regular parametric family.)

### 3. Which Form to Use in MDL?

Now let us come back to the MDL principle. Consider a simple parametric family of models (probability distributions) $f_\theta$ indexed by the $d$-dimensional parameter vector $\theta$

$$\mathcal{M} = \{f_\theta : \theta \in \Theta \subset R^d\}.$$

Given $n$ data points $x^n = (x_1, \ldots, x_n)$ suppose we want to choose a single member of the family to use for encoding $x^n$. Following the principle that we take the model providing the shortest description of the data, the MDL principle suggests that the best model is $f_{\hat{\theta}}$, where $\hat{\theta}$ is the maximum likelihood estimate of $\theta$.

To see this, we appeal to Shannon's coding theorem. Returning to our example from the previous section, assume that our data are taken from a finite set $\mathcal{A}$. (Also, without loss of generality, we will take natural $\log$'s from now on.)

***Shannon's coding theorem:*** For any prefix code $C$ with length function $L$, the average code length $L(P)$ satisfies

$$L(P) \geq H(P) \equiv \Leftrightarrow \sum_{x \in A} P(a) \log P(x),$$

for any probability distribution $P$. Equality holds if and only if

$$L = L_P = \Leftrightarrow \log P.$$

□

The expression $H(P)$ defined above is known as the *entropy* of $P$. It follows from Shannon's coding theorem that the best (on average) description length of a data string $x^n$ generated by $f_\theta$ is

$$\Leftrightarrow \log f_\theta(x^n).$$

Hence MDL recommends the maximum likelihood estimator since

$$\min\{\Leftrightarrow \log f_\theta(x^n)\} \Leftrightarrow \max\{f_\theta(x^n)\}.$$

While it is comforting that MDL produces a sensible answer for a single family, we are really interested in model selection. That is, we want to choose between competing model *classes* having different dimensions:

$$\mathcal{M}_d = \{f_{\theta_d} : \theta_d \in \Theta_d \subset R^d\}, \qquad d \in \mathcal{D},$$

for some index set $\mathcal{D}$. In the example with the Dow-Jones up-and-down strings, we saw two forms of description length. In one case, we summarized an entire class of models in two stages, while in the other, we formed a code from a mixture distribution derived from the members of the class. Which form of description length will produce a "valid" model selection criterion? More precisely, which form gives us a "fair" description length corresponding to a class of models? To decide the question, we need to extend Shannon's coding theorem to the problem of *universal* coding. While this sounds ominous, the basic idea can be explained through a special case, which is nothing more than a familiar hypothesis testing problem.

***Example:*** Assume $x^n$ are iid observations from $N(\theta, 1)$ ($\theta \in R^1$) and we want to test the hypothesis $H_0 : \theta = 0$ vs. $H_1 : \theta \neq 0$. This is the same as choosing between the models

$$\mathcal{M}_0 = \{N(0,1)\} \quad \text{and} \quad \mathcal{M}_1 = \{N(\theta, 1) : \theta \neq 0\}.$$

Based on $\mathcal{M}_0$, Shannon's theorem gives us the code length

$$\begin{aligned} L_0(x^n) &= -\log f_0(x^n) \\ &= -n \log(2\pi)/2 + \sum_{t=0}^{n-1} x_{t+1}^2/2. \end{aligned}$$

Using $\mathcal{M}_1$, we can follow our Dow-Jones example and apply the two-stage form

$$\begin{aligned} L_1(x^n) &= \frac{1}{2}\log n - \log f_{\bar{x}_n}(x^n) \\ &= \frac{1}{2}\log n + n\log(2\pi)/2 + \sum_{i=1}^{n}(x_i - \bar{x}_n)^2/2 \end{aligned}$$

If we are going to use MDL to pick between these two models, how do we know we can't do better than $L_1(x^n)$ based on $\mathcal{M}_1$? Why is $L_1(x^n)$ the "description length" based on the class $\mathcal{M}_1$? To settle the issue, a universal coding theorem over model classes is needed. This means finding a lower bound (in some sense) on the code length obtainable from a model class and a exhibiting a code that achieves it.

***Rissanen's pointwise lower bound (1986a):*** Suppose $x^n = (x_1, x_2, \ldots, x_n)$ are iid observations from a density $f_\theta$ belonging to the class $\mathcal{M}_d = \{\theta \in \Theta_d \subset R^d\}$. Assume that $\theta$ can be estimated at the $n^{-1/2}$ rate (by, say the maximum likelihood estimator). Then for any joint density $q_n$, the difference between its average expected code length and the entropy of $f_\theta$ is bounded below by $\frac{d}{2}\log n$ asymptotically. That is,

$$\frac{E_\theta[\Leftrightarrow q(x^n)] \Leftrightarrow nH(f)}{\frac{d}{2}\log n} \geq 1$$

for all $\theta$ values except for a set which depends on $q$ and has Lebesgue measure zero. □

Comparing this result to Shannon's coding theorem, we find that we must pay a price when comparing model classes. The penalty we incur, $\frac{d}{2}\log n$ measures the complexity of the model class $\mathcal{M}_d$, relative to the sample size $n$. Any code or distribution $q_n$ achieving this lower bound fairly represents the model class and qualifies for use in MDL. For example, the two-stage and mixture forms discussed above each achieve this lower bound and hence yield valid model selection criteria.

## 4. MDL in Normal Linear Regression

Now let us look at MDL more closely in normal linear regression. For a response variable $y$ and a covariate vector $x = (x_1, ..., x_M)$, we write the model as

$$y = \sum_{\gamma_m=1} \beta_m x_m + \epsilon,$$

where $\epsilon \equiv N(0, \sigma^2)$, and the binary vector $\gamma = (\gamma_1, \ldots, \gamma_M) \in \{0,1\}^M$ indexes the $2^M$ possible models. Our data consist of $n$ independent units of $(y, x)$.

Let's consider the two forms of MDL discussed previously. First, the two-stage encoding scheme produces an MDL criterion that is essentially BIC, the Bayesian Information Criterion (cf. Schwarz, 1978):

$$BIC(\gamma) = \frac{n}{2}\log RSS(\gamma) + \frac{k_\gamma}{2}\log n,$$

where $k_\gamma$ is the number of variables in the model $\gamma$ (the number of 1's). Similar in spirit is Akaike's Information Criterion, AIC (Akaike, 1974):

$$AIC(\gamma) = \frac{n}{2}\log RSS(\gamma) + \frac{k_\gamma}{2}2.$$

Under the assumption that the true model consists of a some collection of the covariates, all the MDL criteria we will discuss, including BIC, have been shown to be consistent and optimal in term of prediction (cf. Wei, 1991; Yu and Speed, 1994; Barron et al, 1998), while AIC is known to be inconsistent and sub-optimal in terms of prediction.

Next, Hansen and Yu (1998) studied in detail two mixture forms of MDL called gMDL and iMDL, in which the hyperparameters introduced by the "priors" are further optimized according to the MDL principle. The prior specifications used to derive a mixture distribution are given by:

- For any fixed $\gamma$, take an inverted gamma prior on $\tau = \sigma^2$,

$$p(\tau) = \sqrt{\frac{a}{2\pi}}\tau^{-3/2}\exp\left(\frac{\Leftrightarrow a}{2\tau}\right); \text{ and}$$

- given $\tau$, let $\beta$ have a multivariate Gaussian prior

$$p(\beta|\tau) \sim N(0, c\tau\Sigma).$$

In Hansen and Yu (1998), candidates for $\Sigma$ were chosen for computational convenience (although, if an application suggests an appropriate prior variance-covariance matrix, it should be used). The hyperparameters $a$ and $c$ are set to their "optimal" values in the sense of minimizing the overall code length. (While this strategy is consistent with the MDL principle, a proper coding story requires us to transmit these values as overhead. Fortunately, we can encode them with the same precision across different model classes, adding a global constant to our selection criterion. In short, we can safely ignore the cost of transmitting these hyperparameters.)

To derive the mixture code length under this set-up, define the quantity

$$S_c = y^t(I_{n\times n} + cX\Sigma X^t)^{-1}y,$$

so that the negative, log-marginal density of $y$ (the code length for transmitting $y$ under this mixture code) is given by

$$\begin{aligned}\Leftrightarrow\log m(y) &= \Leftrightarrow\frac{1}{2}\log a + \frac{n+1}{2}\log(a + S_c) \\ &\quad + \frac{1}{2}\log|I_{n\times n} + cX\Sigma X^t|.\end{aligned}$$

Minimizing over $a$ by taking $\hat{a} = S_c/n > 0$, yields

$$\Leftrightarrow\log m(y) = \frac{n}{2}\log S_c + \frac{1}{2}\log|I_{n\times n} + cX\Sigma X^t|.$$

The mixture form gMDL is obtained from this expression by substituting Zellner's g-prior (1986), $\Sigma = (X^tX)^{-1}$, and minimizing over $c$:

$$gMDL = \begin{cases} \frac{n}{2}\log S + \frac{k}{2}\log F, & R^2 \geq k/n, \\ \frac{n}{2}\log(y^ty) & \text{otherwise} \end{cases}$$

where $S = RSS/(n \Leftrightarrow k)$ and $F = \frac{(y^ty - RSS)}{kS}$. For ease of notation, we have suppressed the dependence of $RSS$, $S$, $F$ and $k$ on the model index $\gamma$.

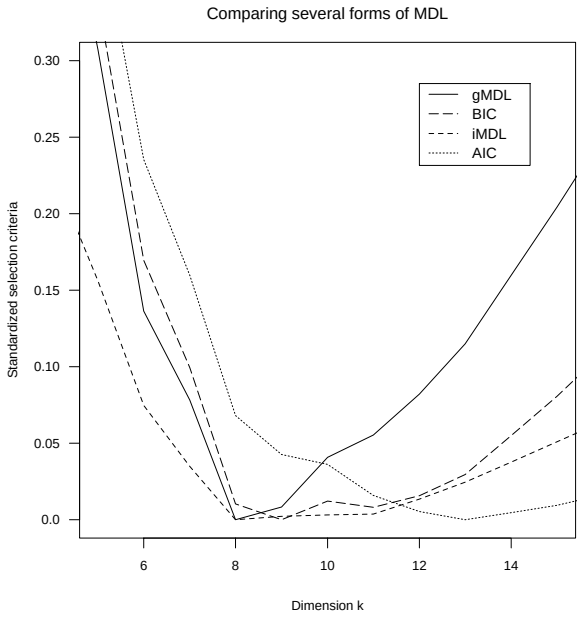
The mixture form iMDL takes $\Sigma = I_{k\times k}$ as in Rissanen (1989). In this case, the minimizing value of $c$ has no closed-form expression, but instead must be computed via an iterative formula (see Rissanen, 1989; Hansen and Yu, 1998).

***Example:*** (Number of bristles on a fruit fly): Data were collected by Long et al (1995) to identify genetic loci that influence the number of abdominal bristles on a type of fruit fly. Several researchers (Cowen, 1989; Doerge and Churchill, 1996; Broman, 1997) have cast this problem as one of model selection in simple linear regression. Let $y$ denote bristle count, and let $x$ be a vector of 39 covariates: one indicator for gender; on-and-off indicators at 19 genetic markers; and each of

their interaction terms with the gender variable. Broman (1997) modified the penalty term in BIC in order to select a "biologically meaningful" model for $y$. That is, for $\eta = 2, 2.5, 3$ he created the criteria

$$BIC_\eta = \frac{n}{2}\log RSS + \eta\frac{k}{2}\log n.$$

All three give rise to an 8-term model that includes an intercept, the main effect of gender, five marker main effect terms (markers 2, 5, 9, 13 and 17), and one gender $\times$ marker interaction term (at marker 5). When $gMDL$ and $iMDL$ are applied to the same data, they both automatically pick the same 8-term model. See Figure 2. Further evidence is provided in Hansen and Yu (1998) to show that mixture forms of MDL are adaptive in the sense that the penalty on dimension can be as small as AIC's, or larger than BIC's, depending on which is most desirable.




## 5. Concluding Remarks

This article covers only the very basic ideas of the MDL paradigm. We have seen only two forms of description length. Another important form, called PMDL, has been derived from a predictive coding scheme. It is very useful for on-line transmission of data and in the context of time series modeling. PMDL also coincides with P. Dawid's prequential approach to statistics. See Hannan and Rissanen (1982), Rissanen (1986b), Hemerly and Davis (1989), and Dawid (1984, 1991). Moreover, a closely related approach to MDL is Wallace's Minimum Message Length (MML) principle. Actually Wallace and Boulton (1968) proposed MML in a classification setting ten years before MDL (Rissanen, 1978), but stopped short of having a more general framework for other statistical problems. See Wallace and Freedman (1987) for more details on MML.

Instead of the pointwise approach to universal coding, the minimax framework is a vital alternative which has been used in the coding context before MDL. In the parametric case the minimax approach provides a natural framework in which connections can be made with Jeffreys' prior (Barron and Clarke, 1990). In the nonparametric case, it brings MDL results seamlessly together with the existing minimax estimation theory. Indeed, MDL research is moving beyond the $\log n$ term in the parametric case and is yielding adaptive estimation procedures in the nonparametric case (cf. Barron et al, 1998). More excitingly, new forms of MDL are emerging, such as gMDL and Normalized Maximum Likelihood (NML) (Rissanen, 1996).

For more extensive reviews on MDL, please see Hansen and Yu (1998) and Barron et al (1998). The former is aimed more towards a statistical audience and the latter towards an information theory audience.

## Acknowledgements

This article is based on a seminar given at the statistics department, University of California at Santa-Barbara, in June 1998. The hospitality of their faculty are greatly appreciated. Special thanks are due to M. Hansen for his permission to include here much material from Hansen and Yu (1998) and for his help on the presentation of this article.

## References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Trans. AC* **19** 716-723.

Barron, A., Rissanen, J. and Yu, B. (1998). The Minimum Description Length principle in coding and modeling. Invited paper for the 50th anniversary issue of *IEEE. Trans. Information Theory*, (in press).

Clarke, B. S. and Barron, A. R. (1990). Information-theoretic Asymptotics of Bayes Methods. *IEEE Trans. Informa. Theory* **36** 453-471.

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*, Wiley, New York.

Dawid, A. P. (1984). Present position and potential developments: some personal views, statistical theory, the prequential approach," *J.R. Stat. Soc.* Ser. B **147** 178-292.

Dawid, A.P. (1991). Prequential Analysis, Stochastic Complexity and Bayesian Inference. *Fourth Valencia International Meeting on Bayesian Statistics*, Peniscola, Spain.

Hannan, E. J. and Rissanen, J. (1982). Recursive estimation of mixed autoregressive-moving average order. *Biometrika* **69**, 81-94.

Hansen and Yu: Model Selection and the MDL Principle *J. Amer. Statist. Assoc.* (Submitted).

Hemerly, E. M. and Davis, M. H. A. (1989). Strong consistency of the predictive least squares criterion for order determination of autoregressive processes. *Ann. Statist.* **17** 941-946.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica* **14** 465-471.

Rissanen, J. (1986a). Stochastic complexity and modeling. *Ann. Statist.* **14** 1080-1100.

Rissanen, J. (1986b). A predictive least squares principle. *IMA Journal of Mathematical Control and Information*. **3** 211-222.

Rissanen, J. (1989). *Stochastic complexity and statistical inquiry.* World Scientific, Singapore.

Rissanen, J. (1996). Fisher information and stochastic complexity. *IEEE Trans. Informa. Theory* **42** 48-54.

Schwarz, G. (1978). Estimating the dimension of a model. *Ann. Statist.* **6** 4611-464.

Speed, T. P. and Yu, B. (1994). Model selection and prediction: normal regression. *Ann. Inst. Statist. Math.* **45** 35-54.

Wallace, C. S. and Boulton, D. M. (1968). An information measure for classification. *Computing Journal* **11** 185-195.

Wallace, C. S. and Freeman, P. R. (1987). Estimation and inference by compact coding (with discussions). *Journal of Royal Statist. Soc.* **B 49** 240-251.

Wei, C. Z. (1992). On the predictive least squares principle. *Ann. Statist.* **36** 581-588.

Zellner, A. (1986). On assessing prior distributions and Bayesian regression analysis with g-prior distributions. *Bayesian Inference and Decision Techniques: Essays in Honor of Bruno de Finetti*, Eds. P.K. Goel and A. Zellner, 233–243. Amsterdam: North-Holland.

Bin Yu
Bell Laboratories and
University of California, Berkeley
binyu@research.bell-labs.com



## *OF INTEREST TO OUR MEMBERS*

# 31st Symposium on the Interface

## *Models, Predictions, and Computing*

***June 9-12, 1999***

Hyatt Regency Woodfield
Schaumburg, Illinois

***Program Chairs***

Mohsen Pourahmadi
Division of Statistics
Northern Illinois University
pourahm@math.niu.edu

Kenneth Berk
Department of Mathematics
Illinois State University
kberk@math.ilstu.edu

***Keynote Address***

Leo Breiman, UC Berkeley
*The Mysteries of Prediction*

***Banquet Speaker***

Stephen Wolfram, Wolfram Research

## *Call for Abstracts*

The Interface Foundation of North America cordially invites you to participate in the upcoming Interface Symposium, the premier annual conference on the interface of computing and statistics. Participants are strongly encouraged to submit contributed papers. Contributed papers are included in the conference proceedings *Computing Science and Statistics*. Abstracts for contributed papers are due by April 1, 1999 and may be submitted from the Interface '99 Web page www.math.niu.edu/Interface99/.

The invited program includes sessions on robust visualization methods; bootstrapping; data mining; advances in tree-based methods; computational issues involved in mixture models; and recent computational and theoretical developments in McMC. In addition, there are three scheduled short courses covering data mining, repeated measures and graphical methods for categorical data.

# A Word from the Editor of JCGS

By Andreas Buja

This is the first Newsletter column for the "Journal of Computational and Graphical Statistics" (JCGS), and hopefully the first of an indefinite series. I thank Mark Hansen for the invitation and Merlise Clyde for the idea. We hope that this column will be a useful link between our two publications by letting, so to speak, the right hand know what the left hand is doing.

Let me start by introducing JCGS: As the name says, our journal publishes statistics articles with a computational or graphical bent. Although this is a young journal started in 1992 by Bill Eddy, its name is already outdated. The term "graphical statistics" has fallen by the wayside, and "us graphical statisticians" now sail under the broader banner of "data visualization." Just the same, the current name is probably here to stay.

JCGS is the child of three sponsoring societies: ASA, IMS and the Interface Foundation of North America (IFNA), which is nice because it gives us institutional stature. After Bill Eddy, the founder of JCGS, Bill Kennedy became the second editor, and I'm the third with a three year term from 1998 through 2000. (I understand that I was asked only after the search committee was unable to find another Bill for the job.)

JCGS is peer reviewed, and like other journals that aspire to be in the top ranks, we can describe our goals with the usual phrases: We want to publish outstanding articles of the highest quality that are theoretically sound and practically relevant. We'd like the work to be pathbreaking, but we'll settle for cutting edge. If you have a recent paper that deserves some of these labels, please, do send it to us. If you think you have something that's close, send it to us also and let us make the judgment. We have an excellent editorial board of associate editors who represent a broad range of expertise.

About 85% of the articles we publish are computational and 15% graphical (or "visual"?). Currently, the major computational topics are Bayesian computing and MCMC, EM, computations for hierarchical models, trees, wavelets and non-parametric regression. We see a few papers in computations for time series, spatial data and shape data. Next year we will publish a collection of articles on massive datasets. We hope to attract more papers in classification, an area which has been given new life by our computer science colleagues in machine learning. On a different level are papers on systems and languages for data analysis. Graphical papers tend to be in one of three categories: new ideas for visual rendering of data, methodology of data visualization, or software systems for data visualization. About once a year we produce a color issue for articles that make essential use of color.

In summary, we have an attractive and interesting journal. This opinion seems to be shared by ever more authors who submit manuscripts to JCGS: From 1997 to 1998 the submission rate shot up by 60%! In order to increase the attraction not only for authors but readers as well, I have taken on a personal crusade to improve the writing in our published articles. All articles should have reader-friendly abstracts and introductions that are informative for the general reader in statistics. If the writing of a submission has problems, chances are that the review will ask the authors to consult the article "The Science of Scientific Writing" by Gopen and Swan. I recommend it strongly and urge that everybody pick it up by visiting the Web page with instructions for JCGS authors, at `www.amstat.org/publications/jcgs/ifa.html`. Those who teach might even want to use the article in their graduate classes – it could spare them the pain of having to rewrite some of their students' theses.

One more word to authors of submissions: **JCGS' review is now fully electronic**, on each step, from authors to editor to associate editor to referees and back. E-mail, ftp or http are the media for submission; see the above Web page for details. Paper is accepted only in case of hardship. Sadly, those bundles of five paper copies that I'm still receiving by mail end up in the waste basket.

In future columns I hope to be able to report about new developments, such as a move towards electronic publishing. In the more immediate future, JCGS will acquire a Software Review Section. Stay tuned...

Andreas Buja
Managing Editor for JCGS, 1998-2000
AT&T Labs – Research
andreas@research.att.com

# Computing Section Awards Announcements

By Lionel Galway

### *1998 Best Contributed Paper*

This year the Statistical Computing Section made an award for best contributed paper in the Section's sessions at the Joint Statistical Meetings. The award selection was made by the session chairs and the Program Chair and Section Award Officer based on the papers submitted. The winner this year was Ranjan Maitra of the Department of Mathematics and Statistics at the University of Maryland Baltimore County for his paper on "Clustering Massive Datasets." Dr. Maitra was presented with a plaque at the Section business meeting. (See page 36 for a picture of Section Chair Karen Kafadar awarding the prize.)

At the 1999 JSM in Baltimore, we will also be sponsoring a best contributed paper competition. All papers presented in the contributed sessions sponsored by the Statistical Computing Section are eligible. The selection will be made from papers submitted over the summer to the chairs of the sessions. The award will be presented to the author(s) of the best paper at the joint business meeting/mixer of the Statistical Computing and Statistical Graphics Sections.

### *Contributed paper presentation in 1999*

Beginning in 1999, the Statistical Computing Section will also make an award to the best presentation in its contributed and topic (special) contributed sessions. The award will be based on audience feedback from forms distributed at each session and will be presented at the Section business meeting at the 2000 Joint Meetings. The presenter will receive a plaque and $100.

Lionel Galway
Awards Officer, Computing Section
The RAND Corporation
Lionel_Galway@rand.org



# The Video Library Heads for the Web

By Deborah F. Swayne

The Graphics Section maintains a library of videos on graphics, and the library's Web site has recently been graphically enriched with individual frames taken from each video. Prospective borrowers can now browse the library using a visual index, and then go to the video's home page for more images and more information. The next step may be to make the videos themselves available using the Web.

The library now includes nearly forty videos, spanning well over 30 years of history in the field of dynamic and interactive statistical graphics. The earliest film in the collection is *Multidimensional Scaling* by Joe Kruskal, made at Bell Labs in 1962. It shows the progress of an iterative multidimensional scaling algorithm, and it is quite short: Though it gives the illusion of a real-time process, I'm told that it was composed as an animation of successive screen shots, allowing ample time for computing each scatterplot.

A more modern version of that same strategy was used to create a more recent video: Bill Eddy and Audris Mockus in their 1994 video *Incidence of Disease Mumps.* They used a succession of computationally intensive smoothing algorithms to visualize the incidence of mumps in the U.S. over space and time, writing the results directly to video disk.

That's a sample of the kind of information I've been starting to collect from the video authors. I'd like each video's page to include, at a minimum, figure captions, an abstract, a list of published references, and perhaps a few words about the data used in the analysis.

More than that, I'd like to add descriptions of the hardware and software platforms used in the software development and video recording, and anecdotes about the history of the work and the people. The evolution of the environments in which this work was done is an important part of the history of the field, and many of the best stories remain outside the published record, and are then forgotten. The library can then be a unique archive as well as a lending library.

A few of the pages are nearly complete: usually pages whose authors have offices just down the hall from mine, so that it is easy to obtain their help. The *Focusing and Linking* video by Buja, Michalak, Stuetzle and Willis, for instance, now has informative captions, and so does the *Seenet* video, by Becker, Wilks and Eick.

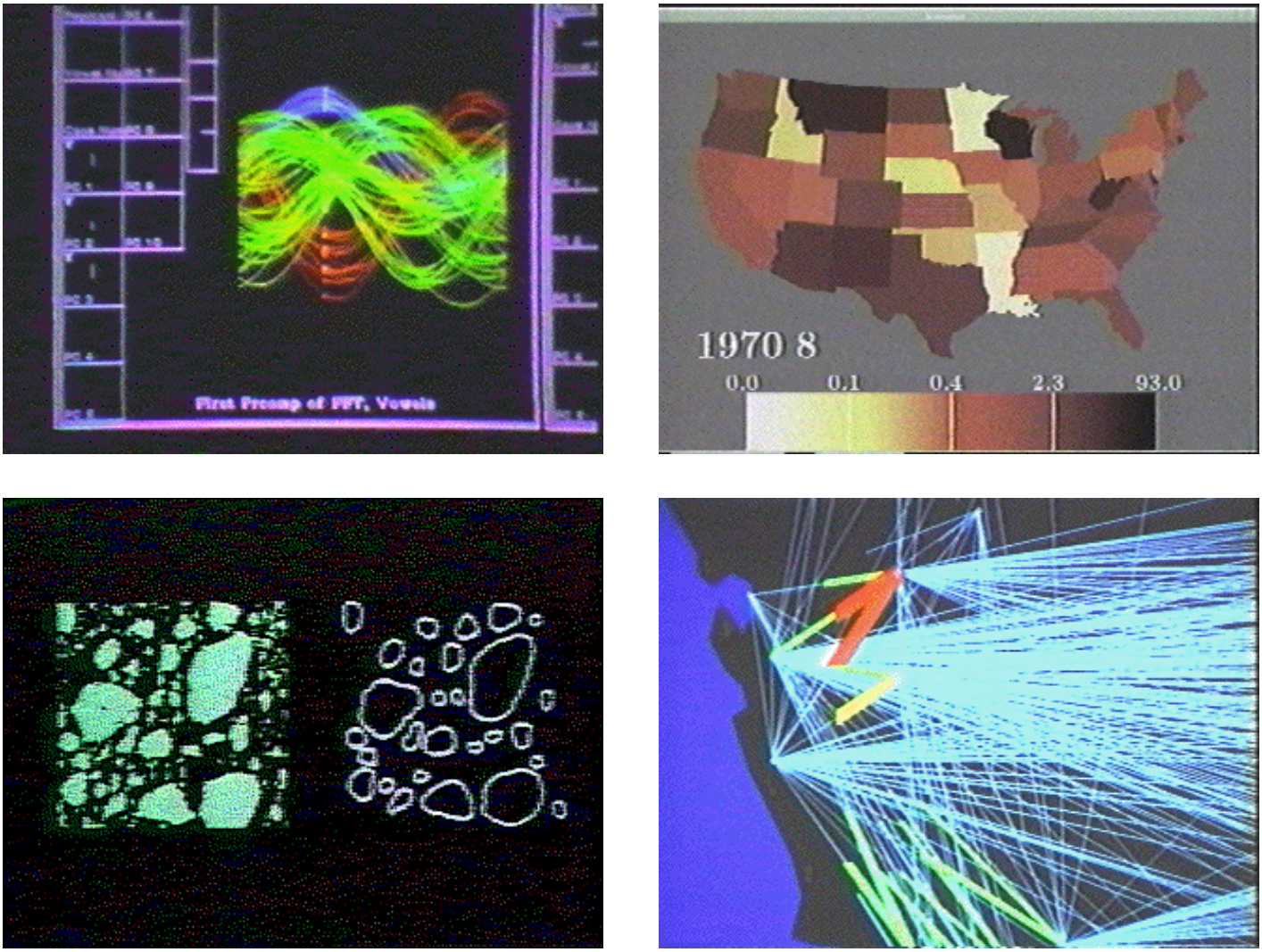



**Figure 1.** Four video stills taken from the library (from left to right and top to bottom): a frame from *Focusing and Linking as Paradigms for the Visualization of High-Dimensional Data* by Andreas Buja, John McDonald, John Michalak, Werner Stuetzle and Steve Willis, 1991; a frame from *Incidence of Disease Mumps* by William F. Eddy and Audris Mockus, 1994; a frame from *Automatic Tracing of Ice Floes on Satellite Images* by Jeff Banfield, 1987; and a frame from *Seenet* by Richard A. Becker, Allan Wilks, and Steven Eick, 1992.

## *Missing in action*

Sadly, a lot of interesting work is not represented in the collection at all. Some films have failed to find their way to us, and of course not all researchers have had the desire or the resources to produce films. Soon the ability to capture scenarios on the computer screen will replace the use of magnetic tape. Indeed, we should begin to prepare the library for that step by investigating the use of digital storage media and decide whether we want to use the internet for video distribution. If any readers of this article have knowledge or interest in the topic of video storage and distribution, I would like to hear from them.

The Web pages described above can be reached from the home page of the statistical graphics section, and here's a shortcut that will work until a new librarian steps forward:
`www.research.att.com/~dfs/videolibrary`

Deborah F. Swayne
Video Librarian, Graphics Section
AT&T Labs Research
`dfs@research.att.com`

Scenes from the Computing and Graphics Sections Mixer at the 1998 JSM. Top row: Karen Kafadar presents Ranjan Maitra with the award for the best contributed paper in the sessions sponsored by the Computing Section; and Mike Meyer addresses the Sections. Bottom row: Bob Newcomb presents prizes to winners of our famous raffle (pictured are Deborah Swayne, Tim Hesterberg, John Sall and Bob Newcomb).

# Computing and Graphics Activities at JSM 1998: A Success!

By Russ Wolfinger and Edward J. Wegman

## *Statistical Computing*

Statistical computing made another strong showing at the Dallas meetings. The six invited and six contributed sessions were well attended and offered a broad spectrum of computing topics. The posters and roundtables covered an array of interesting topics. Our joint mixer with the Graphics section was one of the best of the meetings, including its "3-digits-are-sufficient" raffle. Many thanks to all who participated and helped make all of these activities profitable to attendees!

## *Statistical Graphics*

The Statistical Graphics Section sponsored six session, one contributed, two special contributed and three invited. The sessions were: 1) Density Estimation and the Bootstrap (regular contributed co-sponsored with the Statistical Computing Section), 2) Application of Graphics (special contributed), 3) Visualization & Smoothing (special contributed), 4) Real Success Stories of Statistical Graphics (invited), 5) Statistical Graphics on the Web (invited), and 6) Graphics for Data Mining (invited). The Application of Graphics session was held on Sunday afternoon at 2:00 p.m. and featured some excellent contributed work in application areas including survey analysis, drug design and principal components and factor analysis. Immediately following this session was the Density Estimation and Bootstrap session at 4:00 p.m. This session was jointly sponsored with Statistical Computing and featured among others David Scott, who found time to give a contributed paper in addition to his duties as overall Program Chair for the JSM. Monday morning bright and early at 8:30 a.m. saw the Visualization and Smoothing session which was well attended. This session featured the Statistical Graphics Section student award winners, John Emerson and Elizabeth Brown who spoke respectively on Examining Patterns in Television Viewing with Mosaics in S-PLUS and A Bivariate Smoothing Spline Model of Changes in Plasma Viral Load and CD4+ Lymphocyte Count in HIV/AIDS Patients. Other excellent talks were given including XGobi and XploRe meet VirGIS by Juergen Symanzik.

Later on Monday, the first invited session on Real Success Stories featured Jeffrey Solka from the Naval Surface Warfare Center speaking about the use of graphics in Naval problems and Martin Theus of AT&T Laboratories speaking on the use of graphics in categorical data analysis. The session entitled Statistical Graphics on the Web was extremely well attended and featured Steve Eick, Dan Rope, and Mike Minnotte. Steve and Dan took opposite tacks in developing interactive web-based tools, Steve espousing a more modular ActivX approach while Dan espoused a more integrated JAVA Beans approach. Mike reported on joint work with Webster West in a graphics technique called the Data Image. The final Statistical Graphics session reflected a fairly pervasive interest by many sections of ASA in data mining. Scheduled the in last time slot of the JSM, attendance dropped off somewhat from the other session. However, this session had perhaps some of the most evocative paper titles of the meeting including Mining the Sands of Time and Crystal Vision. The focus was of course the use of graphical tools in the data mining process.

Perhaps the most disappointing aspect of the program was the lack of contributed papers from the membership of the Statistical Graphics section. Indeed, only seven papers were contributed initially and the three contributed paper sessions were constructed by extended negotiations with the program chairs from other sections. Opportunities for presentations abound, so I strongly encourage the members of ASA to submit papers with a graphics or visualization theme. And, of course, if you are not already a member of the Section, please join.

Russ Wolfinger
1998 Computing Program Chair
*SAS*
sasrdw@sas.com

Edward J. Wegman
1998 Graphics Program Chair
*George Mason University*
ewegman@galaxy.gmu.edu

# Looking ahead: JSM 1999, Baltimore

By Mark Hansen and Deborah F. Swayne

It is hard to believe, but the Invited and Topic contributed portions of program for the 1999 JSM in Baltimore are already (essentially) completed. We have worked quite hard to put together a set of talks that explore the latest challenges in statistical computing and graphics. The Computing Section has five Invited Sessions:

***Statistics of the Internet*** This session explores the statistical challenges posed by modeling network traffic data. The speakers are Vern Paxson, Kim Claffy, and Anna Gilbert.

***Several Views of Model Selection*** A fundamental statistical principle is that of parsimonious modeling. The three talks in this session tackle this problem from very different viewpoints. The speakers are Ed George, Bin Yu, and Martin Wells.

***Practical Smoothing: Four applications of nonparametric function estimation*** The idea behind this session is simple. While hundreds of papers are written each year on smoothing, very few start with a realistic application. The talks in this session demonstrate the practical impact that modern, computationally intensive smoothing methods are having in other disciplines. The speakers are David W. Scott, Matt Wand, Jean Opsomer, and Robert Kohn.

***Open Statistical Computing on the Internet*** The Internet represents a tremendous opportunity to assemble applications in statistical computing using resources distributed via the network. This session will give an overview on the technologies available and report on projects utilizing these for open statistical computing on the Internet. The speakers are Erich Neuwirth, Rudolf Müller, and Duncan Temple Lang.

***Scalable Nonparametric Methods for Multivariate Analysis*** Massive, high-dimensional data sets are common currency in the quantitative world. While there exist many disparate efforts at analyzing such data most do not perform well as the number of dimensions and or the number of data points increase. This session explores the possibility of a unified conceptual framework for the rigorous analysis of multivariate data that emphasizes computational feasibility. The speakers are Regina Liu, Peter Rousseeuw, and Theodore Johnson.

The Graphics Section has 3 Invited Sessions. The section also organized a fourth (the last in the list below) that is to be sponsored by ENAR (and co-sponsored by the Graphics Section).

***Using Results from Perception and Cognition to Design Statistical Graphics***, organized by Dan Carr (GMU). The three speakers are all researchers in human perception and cognition: Cynthia Brewer, Gordon Osbourn, Stephen Ellis. They will describe current work in the areas of human perception and cognition and the resulting implications for visualizing data and summaries.

***Network Visualization***, organized by Linton Freeman (UC-Irvine). The speakers (Vincent Duquenne, Lothar Krempel, Linton Freeman, David Krackhardt) work in the area of the exploration and display of network data as graphs. One purpose of this work is the analysis of social networks, which suggests that it may have much in common with statistical data visualization.

***Interactive statistical graphics – mathematical concepts and implementations***, organized by Adalbert Wilhelm (U of Augsburg). In this session, researchers in statistical graphics (Adi Wilhelm, Lee Wilkinson, Dan Rope) will present mathematical approaches to formally describe the construction and interpretation of graphics, and then describe a system incorporating such concepts.

***Recent Advances in Medical Imaging***, organized by Bill Eddy (CMU). It was originally conceived as a graphics session, but it is now being sponsored by biometrics folks since Graphics was running out of slots. Ranjan Maitra, Nicole Lazar and Jonathan Raz will discuss the role or statistical methodologies in recent technological advances in medical imaging.

The complete Invited program is now available through the ASA Web site (`www.amstat.org`), and links to the Computing and Graphics portions of the program are available through the Sections' home pages. We hope that you enjoy the sessions. If you have any comments, please contact us at the e-mail addresses below.

Mark Hansen
1999 Computing Program Chair
*Bell Laboratories*
`cocteau@bell-labs.com`

Deborah F. Swayne
1999 Graphics Program Chair
*AT&T Labs – Research*
`dfs@research.att.com`

## SECTION OFFICERS

### Statistical Graphics Section - 1998

**Michael M. Meyer,** Chair
412-268-3108
Carnegie Mellon University
MikeM@stat.cmu.edu

**Dianne H. Cook,** Chair–Elect
515-294-8865
Iowa State University
dicook@iastate.edu

**Sally C. Morton,** Past–Chair
310-393-0411 ext 7360
The Rand Corporation
Sally_Morton@rand.org

**Edward J. Wegman,** Program Chair
703-993-1680
George Mason University
ewegman@gmu.edu

**Deborah Swayne,** Program Chair–Elect
973-360-8423
AT&T Labs – Research
dfs@research.att.com

**Antony Unwin,** Newsletter Editor (98-00)
49-821-598-2218
Universität Augsburg
unwin@uni-augsburg.de

**Robert L. Newcomb,** Secretary/Treasurer (97-98)
714-824-5366
University of California, Irvine
rnewcomb@uci.edu

**Michael C. Minnotte,** Publications Liaison Officer
801-797-1844
Utah State University
minnotte@math.usu.edu

**Lorraine Denby,** Rep.(96-98) to Council of Sections
908-582-3292
Bell Laboratories
ld@bell-labs.com

**David W. Scott,** Rep.(98-00) to Council of Sections
713-527-6037
Rice University
scottdw@rice.edu

**Roy E. Welsch,** Rep.(97-99) to Council of Sections
617-253-6601
MIT, Sloan School of Management
rwelsch@mit.edu

### Statistical Computing Section - 1998

**Karen Kafadar,** Chair
303-556-2547
University of Colorado-Denver
kk@tiger.cudenver.edu

**James L. Rosenberger,** Chair–Elect
814-865-1348
The Pennsylvania State University
JLR@stat.psu.edu

**Daryl Pregibon,** Past–Chair
908-582-3193
AT&T Laboratories
daryl@research.att.com

**Russel D. Wolfinger,** Program Chair
919-677-8000
SAS
sasrdw@sas.com

**Mark Hansen,** Program Chair–Elect
908-582-3869
Bell Laboratories
cocteau@bell-labs.com

**Mark Hansen,** Newsletter Editor (96-98)
908-582-3869
Bell Laboratories
cocteau@bell-labs.com

**Merlise Clyde,** Secretary/Treasurer (97-98)
919-681-8440
Duke University
clyde@isds.duke.edu

**James S. Marron,** Publications Liaison Officer
919-962-5604
University of North Carolina, Chapel Hill
marron@stat.unc.edu

**Janis P. Hardwick,** Rep.(96-98) Council of Sections
313-769-3211
University of Michigan
jphard@umich.edu

**Terry M. Therneau,** Rep.(97-99) Council of Sections
507-284-1817
Mayo Clinic
therneau@mayo.edu

**Naomi S. Altman,** Rep.(97-99) to Council of Sections
607-255-1638
Cornell University
naomi_altman@cornell.edu

## INSIDE


A WORD FROM OUR CHAIRS . . . . . 1

EDITORIAL . . . . . 2

SPECIAL FEATURE ARTICLE

Coding and Model Selection: A Brief Tutorial on the Principle of Minimum Description Length 1

TOPICS IN INFORMATION VISUALIZATION

Boxplot Variations in a Spatial Context: An Omernik Ecoregion and Weather Example 4

Exploring Time Series Graphically . . . . . 13

Simpson on Board the Titanic? Interactive Methods for Dealing with Multivariate Categorical Data . . . . . 16

MAKING YOUR LIFE EASIER

Just Make It! . . . . . 20

OF INTEREST TO OUR MEMBERS

31$st$ Symposium on the Interface . . . . . 32

A Word from the Editor of JCGS . . . . . 33

NEWS CLIPPINGS AND SECTION NOTICES

Computing Section Awards Announcements . . . . 34

The Video Library Heads for the Web . . . . . 34

Computing and Graphics Activities at JSM 1998: A Success! . . . . . 37

Looking ahead: JSM 1999, Baltimore . . . . . 38

SECTION OFFICERS

Statistical Graphics Section . . . . . 39

Statistical Computing Section . . . . . 39



# Statistical COMPUTING & GRAPHICS

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



Mark Hansen
*Editor, Statistical Computing Section*
Statistics Research
Bell Laboratories
Murray Hill, NJ 07974
(908) 582-3869 • FAX: 582-3340
cocteau@bell-labs.com
cm.bell-labs.com/who/cocteau

Antony Unwin
*Editor, Statistical Graphics Section*
Mathematics Institute
University of Augsburg
86135 Augsburg, Germany
+49-821-5982218 • FAX: +49-821-5982280
unwin@uni-augsburg.de
www1.math.uni-augsburg.de/∼unwin/



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

Nonprofit Organization
U. S. POSTAGE
**P A I D**
Permit No. 50
Summit, NJ 07901

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

This publication is available in alternative media on request.