

# MACSPIN: DYNAMIC GRAPHICS ON A DESKTOP COMPUTER

Andrew W. Donoho, D2 Software, Austin, Texas  
David L. Donoho, University of California, Berkeley  
Miriam Gasko, University of California, Berkeley

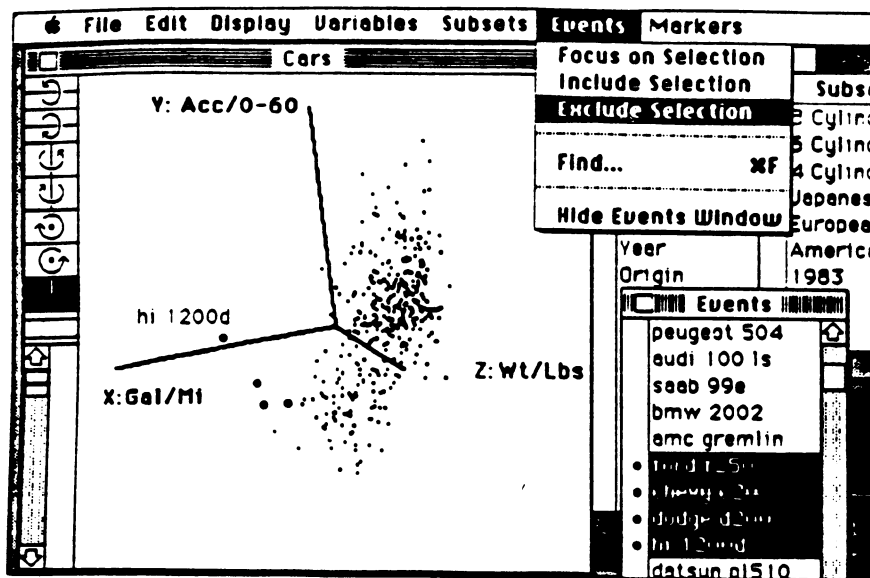


Figure 3: Highlighting and excluding trucks from the display.

Technical Report No. 104

August 1987

Department of Statistics  
University of California  
Berkeley, California

# MACSPIN: DYNAMIC GRAPHICS ON A DESKTOP COMPUTER

*Andrew W. Donoho, D2 Software, Austin, Texas  
David L. Donoho, University of California, Berkeley  
Miriam Gasko, University of California, Berkeley*

## 1. Introduction

MacSpin is a program for dynamic display of multivariate data. It uses rotation to display 3-dimensional scatterplots, and also offers dynamic graphics primitives such as animation, identification, and highlighting. Using these features, the user can visually find trends, clusters, outliers, and other patterns in multivariate data.

MacSpin has a broad range of data manipulation and calculation features. These allow the user to interactively transform, edit, split, and categorize his data as patterns in the display indicate. MacSpin also provides many display options and a number of statistical summaries. All these features are available to the user at the click of a mouse button. Thus the program speeds up the detective work required to understand the visual clues provided by the graphical display.

Dynamic graphics systems have been developed before at various research institutions; these systems ran on large, expensive, and/or exotic machines and had limited use. MacSpin runs on the Apple Macintosh, a personal computer. Like other Macintosh software, it places a great deal of emphasis on completeness of display options, on quality of the user interface and on ease of use. It can fairly be said that MacSpin offers a wider array of features than earlier systems; that where it overlaps with previous systems, MacSpin offers a considerably more complete implementation, and that all these features are packaged together intelligently, so that users who do not need certain features will not be bothered by them. MacSpin makes dynamic graphics perform up to the potential sketched out by earlier systems, and it makes dynamic graphics accessible to a broad audience of scientists and engineers.

In this paper, we begin by telling a few stories that show how the program is used on real data; we discuss some history of dynamic graphics; we discuss the environment and user interface of MacSpin; and finally, we discuss some system design issues.

## 2. Some Stories

In our experience, the best way to explain what dynamic graphics can do is to use it on real data and illustrate the features as one goes along. The MacSpin documentation does this for examples ranging from data on Galaxies and earthquakes to Urban Planning. Here we give brief examples for the CARS and DIABETES datasets.

### 2.1. Cars Data

This dataset consists of data on all the cars road-tested by *Consumer Reports* magazine between 1971 and 1983. The data will help us see how the auto industry changed over the last decade or so. The names of the 418 cars are listed in the events window on the MacSpin display (lower right): the portion we see includes the **Plymouth Barracuda** and the **Plymouth Fury III**, cars from the early 1970's. The **Variables** window (partially obscured by the **Events** window) shows the variables we have measured for each car; things involving performance (Gallons per Mile, Seconds to reach 60 MPH from a full stop), size (Horsepower, Weight, ...), and miscellaneous (Model Year, Continent of Origin).

*X-Y Plots.* The view in the plot window shows all the cars (American, European, and Japanese) in an x-y plot, with  $x = \text{Gal/Mi}$  (i.e. fuel usage per mile) versus  $y = \text{slowness (Secs. 0-60)}$ . The points

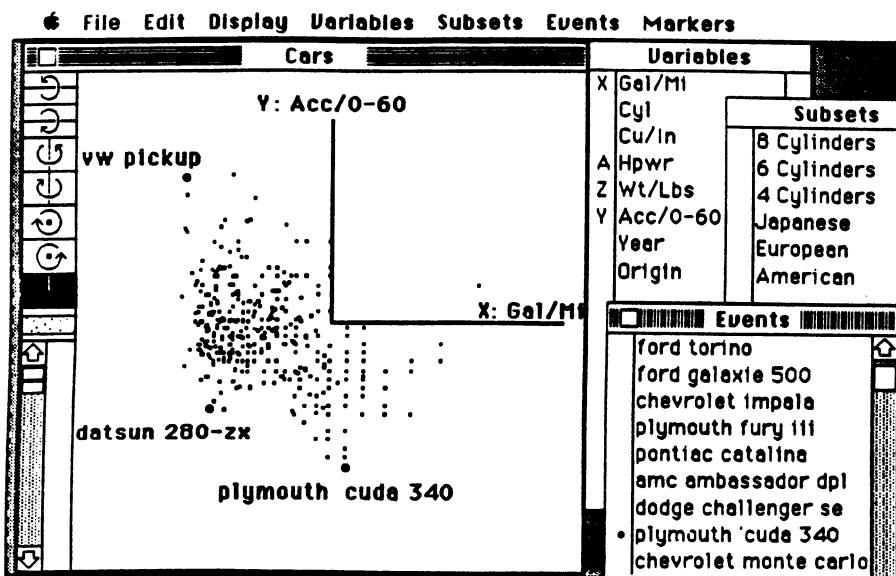


Figure 1. Gasoline consumption and acceleration time for 418 cars reviewed by *Consumer Reports* between 1971 and 1983.

represent individual cars. By moving the cursor to a point and clicking, we can find out its identity. The point at the upper left (slow but economical) is a VW pickup; the lowest point in the plot (the fastest in this dataset) is a Plymouth Barracuda. By holding down the control and option keys as we identify, the full data record pops up. This shows us that the fast, efficient car in the lower left is a 1981 Datsun 280 ZX.

datson 280-zx	
Gal/Mi	0.03058104
Cyl	6
Cu/In	168
Hpwr	132
Wt/Lbs	2910

Figure 2. Info Pop-up for Datsun 280 ZX.

*X-Y-Z Plots.* The x-y plot shows the general trend of the auto industry -- what combinations of speed and economy are available. By rotating the plot, we can get an extra dimension into the display. We point at a rotation icon on the far left, and hold down the mouse button. The 2-d plot becomes a rotating 3-d plot, the previously hidden z-variable coming into play. When we do this for the Cars data, and bring z=Weight into the display, we see a cloud of points rotating smoothly in space. The cloud is shaped like a sausage and shows the combinations of economy, speed, and weight being built during the 1971-1983 period.

As we rotate the plot, we notice a few interesting things. First, one point turns out to be an outlier. We stop the rotation and identify it; it is an International Harvester truck. Somehow a truck has slipped in to a database on cars! When we scroll to the truck's name in the events window, we see that *Consumer Reports* road-tested a few other trucks, too. By pointing at their names on the list, we can highlight them in the plot window. They are also outliers. By choosing Exclude from the events window, we can (temporarily) remove them from the display. The rotation has helped us identify and remove outliers.

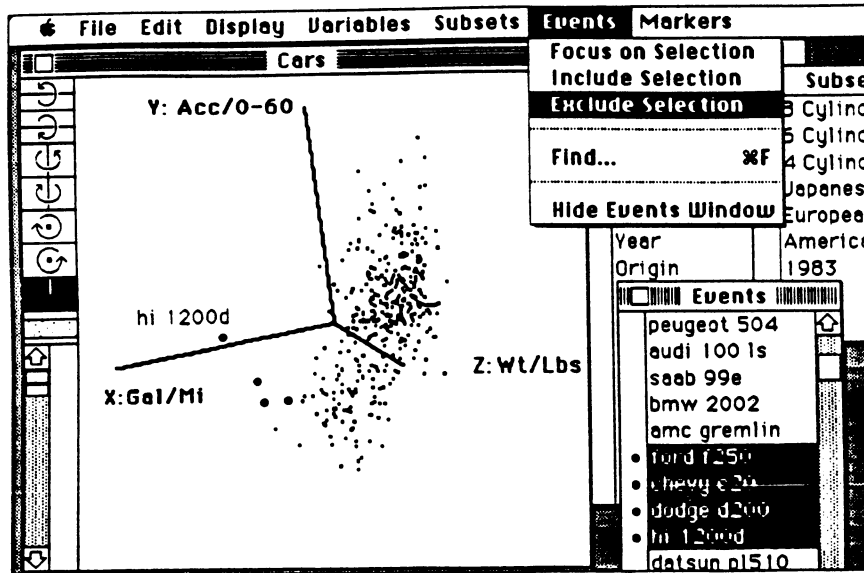


Figure 3: Highlighting and excluding trucks from the display.

*Highlighting Subsets.* Further rotation shows that the data consist of three clusters. Seeking for an explanation, we bring the Subsets window to the front. This shows some subsets of the data predefined (by us) as being interesting to look at. By pointing at the name of any subset, we can highlight its members on the display. When we do this, we see that the 3 clusters consist of 8, 6, and 4 cylinder cars, respectively. We could also highlight American, European, and Japanese subsets in turn, and find out where they are on the display.

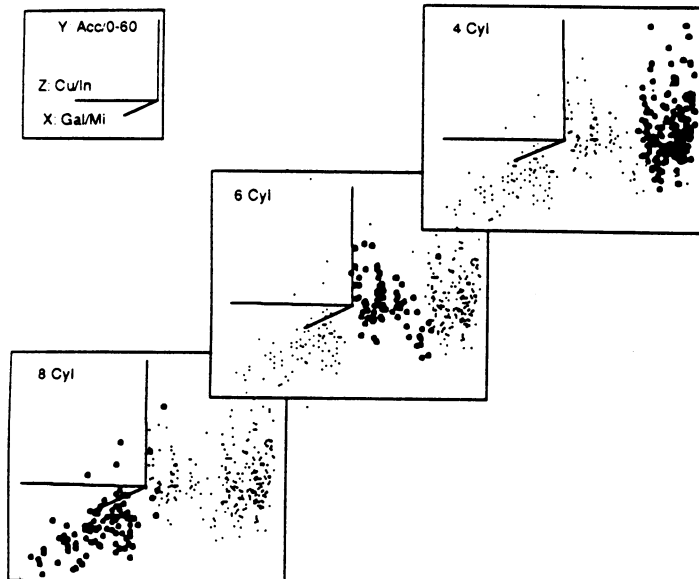


Figure 4: Highlighting subsets of 4, 6, and 8 cylinder cars.

*Animation* permits us to study the effect of a fourth variable on a display. Suppose we are interested in how the American auto industry has changed over time. We can select the American cars, and then select Focus from the events menu. MacSpin now temporarily excludes imported cars from the display. We then drag the Year variable to the scroll bar in the lower left. This will let us scroll through the data model-year by model-year. We begin at 1971. The cars made then are concentrated

in the lower left of the display: fast, heavy, gas-guzzling cars. As we scroll smoothly forward, we see that the data drift systematically towards the upper right-- toward slower, lighter, more economical machines.

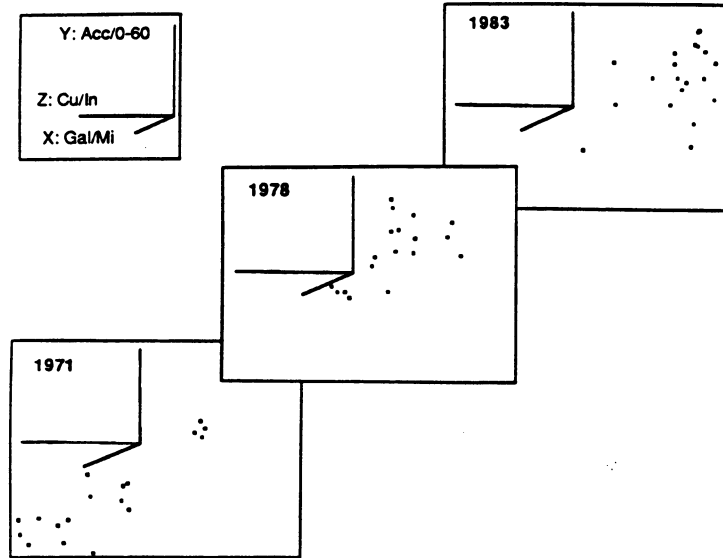


Figure 5: Animation showing changes in the performance of American cars over time: the years 1971, 1978, and 1983 are shown.

*Transformations.* The researcher can also transform existing variables to create new ones. Features like this make MacSpin useful not just for displaying data but also for manipulating it to get the right display. We just saw that cars became more economical over the period 1973-1981. Did they just become lighter and smaller, or was there an actual increase in mechanical efficiency? Dividing Gal/Mi by Weight gives us a standardized measure of fuel efficiency in which the effects of weight are taken out. Looking at plots with this new variable shows that American cars got more efficient and not just smaller over this period. Variable transformations are all included in a special transformations window, and executed by pointing and clicking with the mouse.

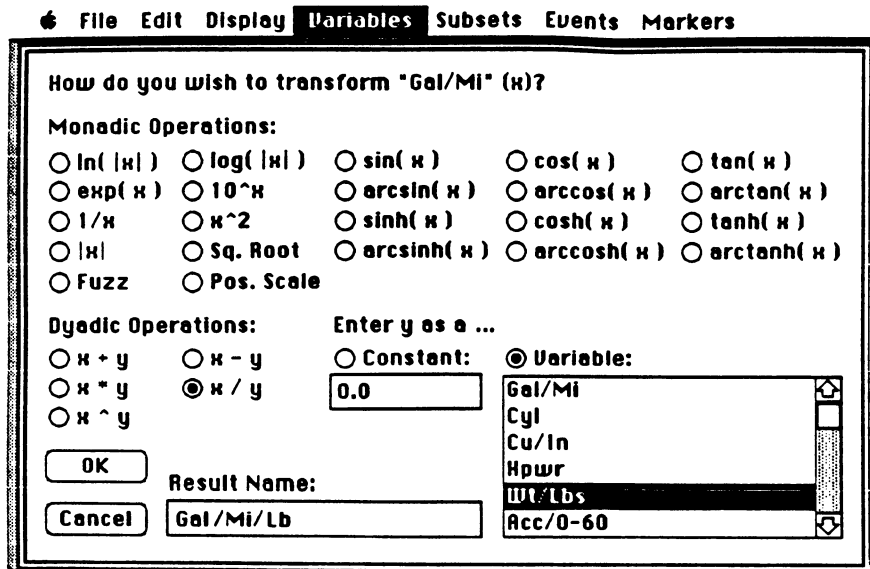


Figure 6: Transformations window.

*Markers.* MacSpin also makes it easy to get a hard copy of the data plots. (Since you can mark subsets with special symbols, you can use these to convey some of what the dynamic exploration showed you.) "Screen dumps" are generated using the Command-Shift-3 sequence. The figure below is derived from a screen dump. 1971 model cars are marked with a box, and 1983 model cars with an asterisk. The resulting image was cropped, and shadows and captions were drawn in, using MacPaint. Figure 7 conveys the interesting message that the *most* efficient 1971 model cars are less efficient than the *least* efficient 1983 models.

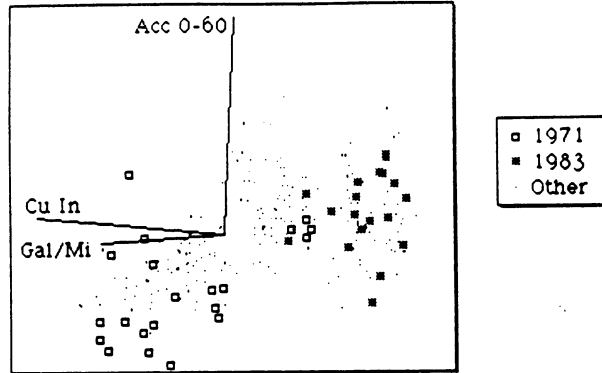


Figure 7: American cars, with special markers given to model years 1971 and 1983.

## 2.2. Diabetes Data

The **Diabetes** data were provided by Reaven and Miller of Stanford University. The original graphic analysis of it was done on the PRIM-9, and reported in *Diabetologica* in 1979. The data describe 145 nonobese and nonketotic patients who agreed to participate in a medical experiment. The purpose was to assess the relationships among various measures of plasma glucose and insulin in order to illuminate the etiology of "Chemical" and "Overt" diabetes. Each patient underwent a glucose tolerance test, and the following quantities were measured: **Age**, **Relative Weight**, **Fasting Plasma Glucose**, **Test Plasma Glucose** (a measure of insulin intolerance), **Steady State Plasma Glucose**, **Plasma Insulin** during the test. In addition we have the doctors' classification of these patients as overt diabetic, chemical diabetic, or normal. **Age** and **Relative Weight** turned out to be unimportant hence are excluded from our analysis (as they were by Reaven and Miller).

The opening view of our demonstration has the following variables assigned to the three axes: **Fasting Glucose** on the X-axis, **Test Glucose** on the Y-axis and **Test Insulin** on the Z-axis.

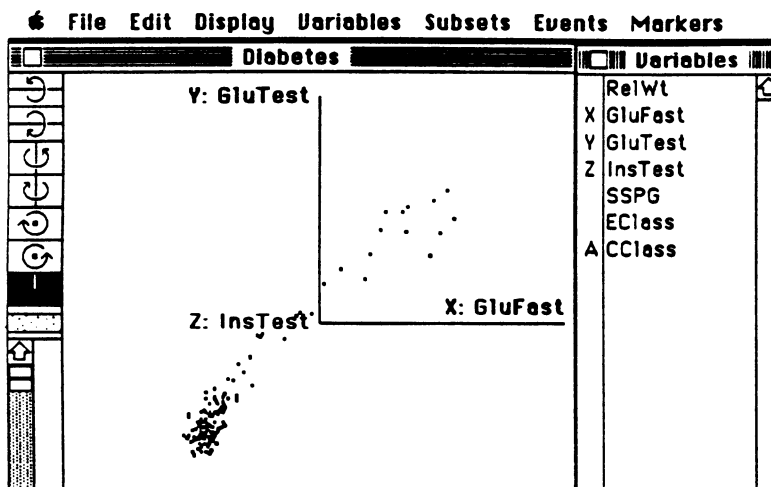


Figure 8: Data for 145 patients who underwent a glucose tolerance test.

This view, showing the data distributed in a sausage-shaped cloud, supports the interpretation that there is but one direction in which abnormality develops, as we progress from normal patients to Chemical to Overt diabetics. However, as soon as we start to rotate the data around the X-axis, and tilt it a bit to better show the third dimension, Z, we can see the pointcloud has, in fact, the shape of a boomerang. We can no longer accept that there is just one direction of disease development.

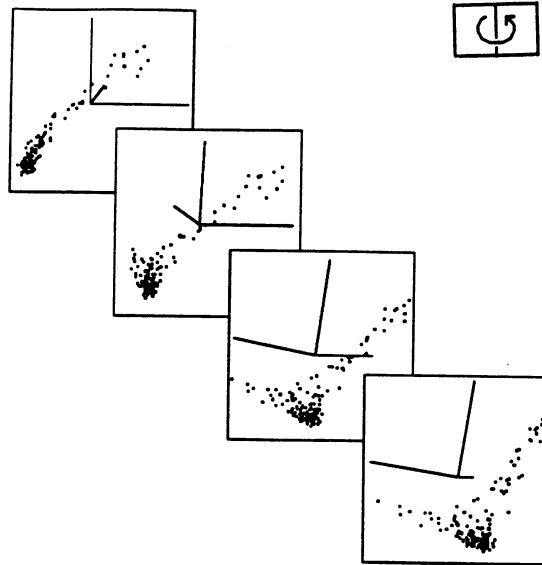


Figure 9: Rotation showing the boomerang aspect of the data.

The most natural question at this point is: What makes the two arms of the boomerang? As the doctors have classified the diabetic patients as either Chemical or Overt, we can highlight each subset separately. As Figure 10 shows, each arm corresponds to one of the groups.

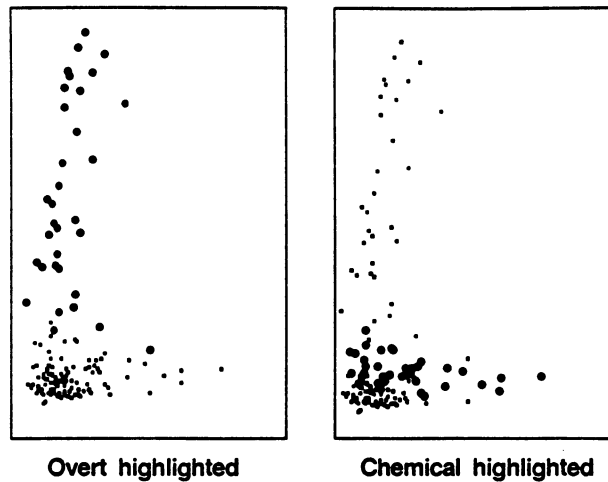


Figure 10: Chemical and overt diabetics shown occupying the two arms of the boomerang.

We can also mark the points corresponding to Overt diabetics with x's, and points of Chemical diabetics with diamonds. Whatever our notation, our conclusion is that Chemical and Overt diabetes are two different syndromes, not just one manifested at different levels of intensity.

### 3. Some History

MacSpin, of course, is not the first system for dynamic graphics. Over the last decade, dynamic graphics systems have been developed at a few research institutions. These were by and large *one-of-a-kind installations* and required expensive and/or exotic equipment. The first system of this kind, PRIM-9, developed by Friedman, Fisherkeller, and Tukey at the Stanford Linear Accelerator Center in the Early 1970's, ran on an IBM mainframe with a \$400,000 display unit, and cost several hundred dollars an hour to use. Other systems were built, for somewhat less cost, at Harvard (PRIM-H), at Stanford, (Orion) and at Bell Labs (Scatmat). These systems served to show that dynamic graphics could be used to analyze data in many different fields... so that a lot of potential was there. For example, PRIM-9 was used to analyze data from particle physics experiments, ... but it was also used to study the different diabetes syndromes. The PRIM-H system was used to inspect data on circadian rhythms of monkeys, on demographics of developing nations, and on clustering of remote galaxies.

Unfortunately, such systems were hidden away in labs and only a few people ever had access to them. Also, because they were used by so few people, issues like ease of use and capacity for working with real data on an everyday basis were secondary. The designers of these systems created them to make movies and videotapes illustrating the tremendous potential of dynamic graphics, but they did not use their systems for everyday work.

One difference between MacSpin and earlier systems makes this clear. Earlier systems were all demonstrated at conferences by projecting a videotape or movie that had been shot at the research institution and carefully edited. It would have been impractical to install a VAX and appropriate graphics devices on site for a live demonstration. At the 1986 Annual Meeting of the American Statistical Association, MacSpin was demonstrated live, using a Macintosh one of the authors carried into the lecture hall and plugged in, and a small projector which took the video signal direct from the back of the Macintosh. Dynamic graphics has become as portable as the 22-pound Macintosh.

(Of course, one can still use big computers in conjunction with dynamic graphics. For example, we like very much to use the S statistical system developed at ATT Bell Laboratories, running on a VAX at Berkeley. We use the Macintosh as a graphics terminal to the VAX, and can rapidly transfer to MacSpin any data we have in S which we would like to view with MacSpin. In this way can use the VAX from home over a phone line and do dynamic graphics at home, which the earlier systems could never have allowed us to do).

### 4. An Environment and a User Interface for Dynamic Graphics

A key feature of MacSpin, not present in earlier systems, is the supporting environment that MacSpin offers. This environment includes facilities for data input/output, data editing, graphical hard-copy, data transformation, and variations in the display options -- we will give details below.

It is hard to overstate the importance of this supporting environment. In our experience, successful use of dynamic graphics involves not just looking at a display of raw data, but a mixture of massaging the data (transforming and editing it), modifying the display in order to make the display more informative, and saving documentation of what has been seen. Indeed, exploration of data is inherently a process of discovering structure, and then removing it so that further structure becomes apparent. Thus, in principle, there ought to be a very tight loop in which the user alternately sees and then acts based on what he sees. The stories above give examples of the way this process happens in analyzing real data.

When the environment is not rich enough to support this tight loop, users become frustrated; they want to follow up on what they see, but cannot. In a few minutes of interacting with their data, many questions come to mind: "What happens if we look at gallons/mile instead of miles/gallon?", "Are all these outliers 4 cylinder cars?", "How well can we separate Chemical Diabetics from Overts using just two variables". The inability to follow up on such questions immediately imposes a burden on the user: he must remember the questions, and then improvise methods for answering them using, say, a statistical package or database management system. At the very least, this puts a burden on the user's memory; if the user has to use some other system or package to answer his questions it means that he must make constant context switches, and preserve his memory and sense of purpose across switches. The end result can easily be exhaustion, frustration, and loss of effectiveness.



So to encourage true high-interaction between user, data, and display requires a rich environment; but it also requires a clean *user interface* to that environment. It must be easy for the user to access the tools of the environment, or he will still get exhausted by the effort of using the system, and again lose the benefits of high interaction. Similarly, the visual design of the user interface must be clean and uncluttered so that the user's eye is not too much distracted by too much "screen junk". And the sequence of visual displays encountered in using the system must not be too varied or discontinuous, or else fatigue will arise from constant context switching.

We believe that MacSpin offers an excellent combination of richness of environment with quality of user interface. We now describe both elements of the combination in more detail.

#### 4.1. The Environment

The facilities that MacSpin offers can be classified as follows:

- > Display Options. Changing how data get displayed.
- > Information Pop-ups. Providing additional information about the data.
- > Subset operations. Defining, combining, removing subsets of data.
- > Transformations. Defining and transforming variables.
- > Input/Output. Data import/export, Data editing, Graphical Hardcopy.
- > Documentation. Manual with worked-out examples on real datasets.

We discuss these facilities in turn.

##### Display Options.

These range from very simple ones:

*Background color* -- the ability to display data as white points on a black background or as black ones on a white background;

*Axis visibility* -- the ability to have the tripod visible or invisible as rotation happens;

*Big Pixels* -- the ability to display points using either big or regular pixel widths;

*Save Orientation* -- the ability to "save" a certain 3-d viewpoint, and return to it later;

to more sophisticated ones:

*Depth Cueing* -- the ability to display points with size of marker indicating depth;

*Set Origin* -- the ability to set the origin of the rotation coordinate system to any point in the dataset, or to the centroid of any subset of points;

*Rotate N degrees* -- the ability to numerically "navigate" the orientation so that the plot is seen from any numerically specified viewpoint in 3-space;

*Spatial Scaling* -- the ability to force the scaling of the plot axes so that proportions seen in the display reflect the actual numerical proportions present in the dataset (that is, in stretching or shrinking the plot to fit on screen, the same stretch or shrink is applied to all three axes).

*Automatic Pilot* -- the ability to have the plot rotate continuously, without need for continual pressing of the mouse button. The rotation can be either in one direction, or else rocking back and forth a few degrees about a fixed viewpoint.

Other features control display of individual data points. Thus, when data points are highlighted because they have been selected, the user has available two different intensities. And, as we have seen, points can be displayed with markers -- articulated symbols rather than just dots -- to differentiate between subsets of data. The markers can also be temporarily suppressed to go back to a plain display.

In our experience, for each dataset, some subset of these display options will be essential. For example, spatial scaling is needed in looking at spatial data, markers are needed in data with categorical variables, and a white background is useful when preparing artwork. All these features, as discussed later, are available through simple point-and-click operations with the Macintosh mouse.

### **Information Options**

MacSpin can provide the user with additional information beyond what normally is showing in the display. Much of this is accomplished through "pop-up windows" which the user calls up by holding the option and or command keys and pressing the mouse button. The resulting information is about either MacSpin or about the user's data.

For information about MacSpin itself, there are pop-ups which: tell the user the current viewpoint (in latitude, longitude, etc.); tell him the current settings of the animation controls; and tell of the settings of the automatic rotation and rocking features.

For information about the dataset, we have already seen the pop-ups that give the full data record on an individual; there are also pop-ups which give a range of summary statistics on any variable in the dataset, pop-ups which give a list of the individuals in the dataset, sorted by a variable, and pop-ups which count and list the members of a subset.

These pop-ups allow the interested user to quickly get answers to many of the basic questions that come up during exploration, without cluttering the screen and forcing the uninterested user to see the information.

Other features which could be listed as information-providing include the ability to inspect a spreadsheet-like view of the entire dataset, which the user can scroll through and edit.

### **Subset Operations**

MacSpin allows the user to define and combine subsets of data. The definition can be based on selection from the plot window, using the mouse, or based on the label of a point matching a certain pattern (e.g. to make a subset "Ford" containing all individuals whose labels contain the string "Ford"), or based on a numerical criterion (e.g. to make a subset "Overt Diabetics" containing all individuals with GluFast > 120). The subsets can be combined via intersection, union, and difference. Once subsets are defined, they can be highlighted, specially marked, focused on or hidden from the display; and statistics of that subset alone can be computed.

As we have seen, the subset manipulations are the key to much exploratory work with real data. They allow one to focus attention on American cars, to isolate outliers in a subset and browse through them for clues as to where they came from, to compare different time periods in the dataset. We believe that MacSpin is unique in providing this broad range of subset operations. We have found these operations so frequently useful that the prospect of having to use a system without them would depress us greatly.

### **Transformations**

As we have seen, MacSpin allows the user to transform his data, by any of the typical operations (add two variables, take logs of a third, ...). The user may also create linear combinations of several variables; this is useful in interpreting linear regression and principal component analyses done elsewhere. MacSpin also allows the user to create new variables containing random numbers, and to create variables containing arithmetic progressions. These "artificial" variables are frequently useful. Thus, conditioning on a random variable allows one to select a random subset of the data; focusing on such a subset can be useful when the dataset consists of so many thousands of individuals, so that the display has gotten saturated with points. Progressions are useful, for example, so that one can select the last 418 points in the dataset, or if one wants to deal with time series.

### **Input-Output**

MacSpin reads in Ascii files with white space (blanks or tabs) separating the entries. It also accepts data from the Macintosh clipboard. This means that the process of transferring data from, say, a VAX, running S or from another Macintosh program is quick and painless. If there are problems reading the dataset, a box appears showing the offending line in an edit box. The user can at that point correct the error, discard the line, or abort the read. MacSpin also can save data in the form of text files or on the clipboard. Some users have reported that they use MacSpin as a data manipulation tool, because of its subset and transformation features, when they are dealing with spreadsheets. They take

the data from their spreadsheet program into MacSpin, manipulate it, and take it back into the spreadsheet program.

MacSpin allows three forms of graphical hardcopy. First, there is the "screen dump" which every Macintosh application offers. Second, there is a print command which allows the user to add captions and other embellishments. Third, the user can paste a figure onto the clipboard, and from there into MacDraw or another drawing program, where the figure can be edited, captions added, borders added, etc.

All of these abilities are easily used by anyone with a little Macintosh experience.

## Documentation

MacSpin comes with a 189-page manual and 32-page supplement. The documentation explains not only use of the features, but how the features fit into the process of data analysis. Seven datasets are analyzed in detail, and stories similar to the Cars and Diabetes stories are given for each. The datasets cover fields ranging from geophysics to botany. The program itself now comes with 15 datasets spanning an even wider range than is covered in the manual.

We believe that this documentation represents the first real attempt to explain what dynamic graphics is and what it can do outside of research articles. Judging from the response we have received, we believe that it does get some basic points about data analysis and graphics across to a large audience.

## 4.2. User Interface

The many features we have described and/or hinted at in the last subsection would be overwhelming to the user without a good user interface. That interface needs to make the easy things easy, to hide unnecessary details from the person who doesn't need them, but to provide lots of power so that users don't get the sense of frustration we talked about earlier.

Luckily, by following the Macintosh User Interface Standard, we have been ahead of the game from the start. As the Section 5 points out, the Interface Standard has taught us a lot about designing user interfaces using the mouse and windows, to get a visually and intellectually satisfying result.

As a crude example of how the Interface Standard helps one avoid clutter and yet offer both simplicity and power, consider the device common to all Macintosh programs, the device of having a menu bar at the top of the screen with pull-down menus. In MacSpin, this gives a pleasing classification of the tasks available to the user: *File*, *Edit*, *Display*, *Variables*, *Subsets*, *Events*, *Markers*. The user thus knows that commands associated with the display are on the *Display* menu; he can get them any time, but doesn't have to look at them when he doesn't need them.

As another example, consider the pop-up windows, that bring information to the sophisticated user, then vanish when the user is finished with them. The novice does not need to know they are available; the expert calls for them in an instant by holding down the option key as he clicks the mouse button. Functionality is provided without intellectual or visual clutter.

Nevertheless, it must be admitted that MacSpin offers an awful lot of features. To offer so many in another environment would perhaps be foolish; no one could remember them all, along with all the other things he has to remember, simply to work in that environment. Fortunately, all Macintosh applications work alike in certain ways, and this keeps all the complexity manageable. In learning a new program, the user only has to learn about the task unique to the program. All the other things about program operation -- starting up, opening files, editing data, moving and resizing windows, transferring data, printing -- work the same as in any other program. The user already knows them. In contrast, a user of Unix may have to remember the 8 different ways of quitting a program that his 8 most commonly used programs employ, and 8 different sets of option flags, and different file formats, and so on. In the Macintosh world, there is much less of the "removable complexity" so omnipresent in less unified systems such as Unix. Therefore individual applications can present a greater level of functionality without fear that users will be too confused or exhausted to make use of it.

### 4.3. Nonstandard applications

Of course, the proof of the pudding is in the eating. If MacSpin's user interface and full-featured environment are so worthwhile, then there should be things that users have come up with in the course of analyzing their data that illustrate this. We can give three examples.

#### Mercedes Benz Plot

The Fortune 250 data supplied with MacSpin give information about the top 250 companies in the Fortune 500 listing for 1985; the information includes the rankings of the company in 1980, 1975, and 1965. Suppose we make a plot in MacSpin with axes giving the 1985, 1980, and 1975 rankings. However, we look at it from an unconventional viewpoint: so that each pair of axes makes a 60-degree angle (the Mercedes-Benz symbol). We position the axes in this way by using the "navigation". When we do this, we get the plot shown in Figure 11 below. When viewed in this way, a point is in the upper right if it has been shrinking (rank in 1985 > rank in 1980 > rank in 1975; recall that a higher rank means smaller sales (Exxon's 1985 rank is 1)). A point is in the lower left if it has been growing (rank in 1985 < rank in 1980 < rank in 1975). By going through the subsets of this dataset (Chemicals, Drug, Oil, Computers, ...) and highlighting each in turn, one rapidly sees which industries have been growing and which have been shrinking.

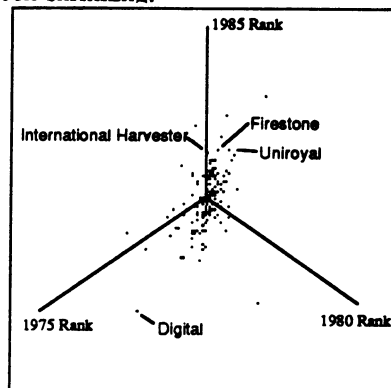


Figure 11. Mercedes-Benz Plot of the Fortune 250 data.

This example shows a nice, simple application of the navigation and subset highlighting features.

#### An Earthquake movie

The Quakes dataset supplied with MacSpin gives the location of 1000 earthquakes occurring in the South-West Pacific between 1964 and 1980. The events are labelled simply by their chronological order. If the user selects the high-intensity highlighting option, then when he points at a label (Earthquake #386) in the Events window, the position of that earthquake is highlighted on the display with a big, fat, dot -- practically, a 'ball'. Now suppose he brushes the mouse through the list of labels; then the quakes get highlighted one after the other, in a smooth fashion. The user sees the sequence of earthquakes "happenings" bam-bam-bam, all over the South-West Pacific. It looks as if a ball is bouncing all over the screen randomly. Occasionally, many earthquakes occur in a small region of space and time; such an aftershock sequence shows up vividly in the movie, as the "bouncing ball" stops hopping all over the screen and stays confined to a relatively small region. The aftershock sequences practically leap out at the viewer.

#### Flooding Oklahoma

The figure below shows a digital terrain map of an area in Oklahoma. The map consists of 4096 points, with X-Y coordinates in a regular 64 by 64 grid. The Z variable is altitude. By animating on a fourth variable, one can see easily how it correlates with topography. Even on an ordinary Macintosh, the speed of animation on this large dataset is fast enough that by animating back and forth, one gets a very good feel for anomalies and correlations. The example would not be possible except for the spatial scaling feature in MacSpin, which allows physically faithful scaling of plots.

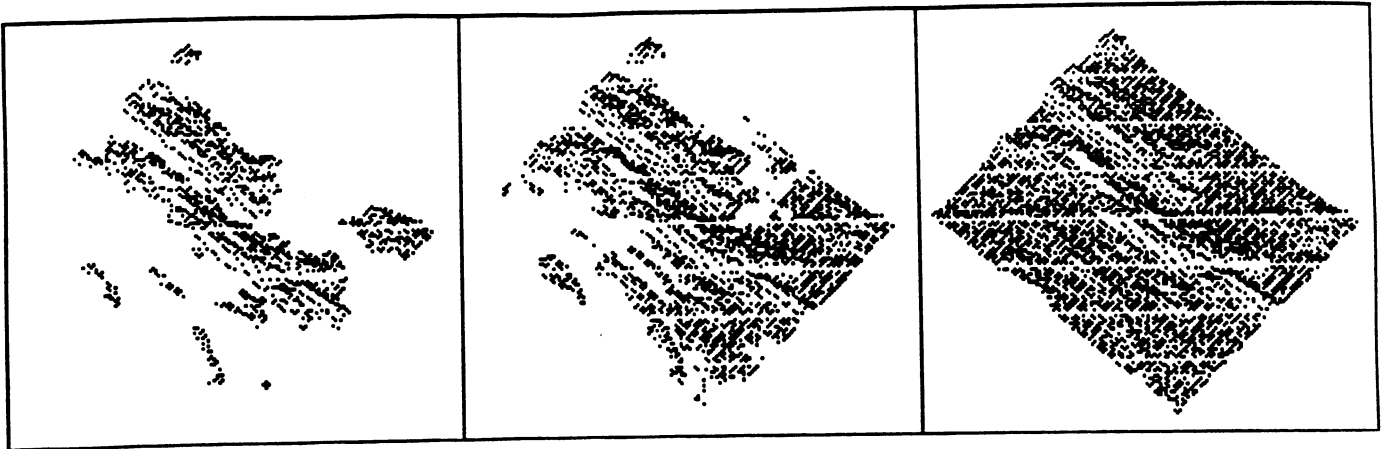


Figure 12. Stages in the Flooding of a digital terrain map.

These examples serve to illustrate the point that a rich environment allows one to see many different things, in many different types of data.

#### 5. Developing for the Macintosh.

Of course, the difference between MacSpin and earlier systems is far more than just the availability on a personal computer. As we have indicated, the choice of the Macintosh has had far-reaching effects on the functionality of the program.

As you probably know, the Apple Macintosh is a personal computer with a high-resolution bitmap display and a Motorola 68000 microprocessor. Like other personal computers, it is available inexpensively -- especially under some of the discount plans that exist for universities. There are now 1 million Macintoshes in use.

MacSpin follows the Macintosh User Interface Standard. This standard was created by Apple Computer to ensure that all software developed for the Macintosh follows a uniform protocol for issuing commands, displaying information, opening files, and so on. In our opinion, this standard is an important innovation in the computer world. Such a standard does not exist on any other computer -- whether it be an IBM PC, a SUN, a VAX, or an IBM mainframe. The effects of this standard are: Every Macintosh user already knows how to open files, edit them, print out results,... for *every* program that he might consider using on the Macintosh. It is no exaggeration to say that most Macintosh applications do not need user manuals. In a very real sense, most Macintosh users can profitably employ programs like music synthesizers, paint programs, page-layout programs -- even without being an expert in the fields addressed by those programs, and without being a computer programmer. Every Macintosh developer is taught, by the standard, how to produce an application with a clean, smooth user interface and a sharp appearance. Since most programmers, and most people, don't have any skills in graphical design or in user interfaces, this means that the ease of use and appearance of Macintosh software are an order of magnitude better than those of any other computer.

The Macintosh User Interface Standard is derived from years of experience with graphics user interfaces on the SMALLTALK-80 system developed at Xerox. For MacSpin in particular, following the the User Interface standard meant a prolonged period of apprenticeship for us, learning the standard and its rationale. For example, part of our task was to take standard User Interface notions which seemed designed for word processing and similar applications, and *discover* what they meant in a data analysis application. Eventually, we discovered things that were natural for data analysis that were not considered in the standard -- so we had to create new user interface notions in several places (our information and hyper information pop-ups are examples of this). In retrospect, we believe that only through such an apprenticeship can one develop the knowledge necessary to produce a polished user interface.

In particular, we don't think that a statistician could produce a very good user interface just using his subject matter knowledge and just experimenting with the computer on his own.

Once the design is done, implementing a good user interface is hard work. This can be judged from the fact that MacSpin is 36,000 lines of Pascal code (not counting comments). Thus MacSpin is as large as, or larger than, most statistics packages for microcomputers -- and MacSpin is a display program only. Again, we doubt that we would have invested this sort of effort in MacSpin, except for the social pressures brought to bear by the Macintosh community to do work meeting a certain standard of polish and ease of use.

## 6. Experiences

MacSpin has expanded by a few orders of magnitude the number of people who have a chance to use dynamic graphics in their work. Many users of MacSpin are, surprisingly, not scientists, or professional statisticians; we were somewhat skeptical of this trend at the beginning, but have found that the statistically uninitiated can find many uses for this program. In fact, some of the more interesting applications have been found in areas that only a novice would have the freshness of mind to think up. MacSpin has been used by amateur astronomers to look for patterns in local star clusters and by a homemaker to compare nutritional content of different recipes.

Our own experiences, and feedback from users, have indicated a few broad conclusions.

- [1] While rotation of point clouds is eye-catching, this is only a small part of what a good dynamic graphics program should do. Abilities such as subset highlighting and animation on a fourth variable play a large role in helping a user understand the data he is looking at. These allow the user to dig into the reasons that certain patterns appear on the point cloud display. In our experience, the MacSpin user spends about 20% of his time watching rotations, and the remainder of the time using other dynamic, interactive features of the program. Point cloud rotation without all the other features of MacSpin is just a gimmick -- it's eye catching, but it doesn't help answer any questions that come up during the analysis.
- [2] MacSpin involves the user fully in the analysis of his data. While using MacSpin, the user is constantly asking questions -- "Who is that outlier?", "Where are the American cars" -- and getting instant, graphical feedback. MacSpin establishes a very tight loop between the user and the display; this is only possible because of the graphics/mouse user interface. Moreover, using graphics and a mouse is not enough: very subtle defects in the user interface would be enough to sabotage the interaction loop entirely.
- [3] After building MacSpin, when we look at other people's work in dynamic graphics, we notice two common problems in their designs which would, we believe, sabotage the interaction process. One of these is constant *context switches*. By this we mean that the user is not presented with one basic display format and one uniform style of interaction, but instead, with frequent changes: a scatterplot is present; it goes away, and is replaced by a menu; the menu goes away, and is replaced by the scatterplot, etc. While the menu is present, the user cannot see the scatterplot, and vice versa. This means that the user constantly has to adjust to -- parse -- his changing visual environment, rather than focusing on his data. Also, the user is forced to remember things he saw in one view -- e.g. the list of variable names -- so that he can use the other view effectively. This means that the user's short term memory is occupied with incidentals rather than with the significant issues of analysis. In MacSpin we try, as much as possible, to leave the basic plot on the screen, and pop-up small windows and menus on the screen which leave everything else visible. Thus, to the user, it seems that there is one basic visual scene, with minor variations. Also, things like the list of existing variable names and subset names are still visible while a dialog box asking for the name of a new subset or variable is present -- so if the user forgets one of the names, he can refer back to the list.

A second common problem is *clutter* -- presenting too much information and too many options to the user at once. These command items -- often rarely used -- are a constant drag on the visual system. In contrast, the standard MacSpin display presents almost no command items; only things having to do with the dataset are visible -- the scatterplot window, the variables list, the

subsets list, and the event names list. Commands pull down or pop up and are conveniently organized for easy recovery. One could say that there is a lot of power hidden behind a chaste, attractive facade.

### 7. A Final Word

We should emphasize that the real value of dynamic graphics is in the *process* of human-data interaction. This interaction between the user and his data is very helpful in most investigations; it helps the user choose a strategy for further analysis and rapidly find out about the flaws and peculiarities of his dataset. Dynamic graphics does not replace other types of analysis, such as those one might want to do with S or SAS. However, it is an important addition to the analyst's toolkit.

#### Suggestions for further reading.

Becker, R.A. and Cleveland, W.S. (1984). *Brushing a Scatterplot Matrix: High Interaction Graphical Methods for Analyzing Multidimensional Data*. Manuscript.

Donoho, D.L., Huber, P.J., and Thoma, H.M. (1981). Kinematic Display of multivariate data. *Proceedings of the Interface between Computer Science and Statistics*, W.F. Eddy, Ed.

Fisherkeller, M.A., Friedman, J.H., and Tukey, J.W. (1974). *An Interactive Multidimensional Data Display and Analysis System*. SLAC PUB 1408. Stanford Linear Accelerator Center, Stanford, CA.

Friedman, J.H. and Tukey, J.W. (1974). A Projection Pursuit Algorithm for Exploratory Data Analysis. *IEEE Transactions on Computers IC-23*.

MacDonald, John A. (1982). *Interactive Graphics for Data Analysis*. Thesis, Stanford University.

Reaven, G. and Miller, R. (1979). An Attempt to Define the Nature of Chemical Diabetes Using a Multidimensional Analysis. *Diabetologica* 16, 17-24.

File Edit Display Variables Subsets Events Markers					
Cars					
	Gal/Mi	Cyl	Cu/In	Hpwr	Wt/Lbs
saab 99gle	0.0462963	4	121	115	2795
peugeot 604sl	0.0617284	6	163	133	3410
volkswagen	0.03174603	4	89	71	1990
honda accor	0.0338983	4	98	68	2135
pontiac lema	0.0463116	6	231	115	3245
mercury ze	0.05050506	6	200	85	2990
ford fairmo	0.04484305	4	140	88	2890
amc concord	0.04950495	6	232	90	3265
dodge aspen 6	0.0485437	6	225	110	3360
chevrolet ca	0.0588235	8	305	130	3840
ford ltd landau	0.0568182	8	302	129	3725
mercury gr	0.06060606	8	351	138	3955
dodge st. re	0.05494505	8	318	135	3830
buick estate	0.0591716	8	350	155	4360
ford countr	0.0645161	8	351	142	4054
chevrolet m	0.0520833	8	267	125	3605
chrysler leb	0.05405405	8	360	150	3940
vw rabbit c	0.03134796	4	89	71	1925
maxda glo d	0.0293255	4	86	65	1975
dodge colt h	0.0280112	4	98	80	1915
amc spirit d	0.03649635	4	121	80	2670

**TECHNICAL REPORTS**  
Statistics Department  
University of California, Berkeley

1. BREIMAN, L. and FREEDMAN, D. (Nov. 1981, revised Feb. 1982). How many variables should be entered in a regression equation? Jour. Amer. Statist. Assoc., March 1983, 78, No. 381, 131-136.
2. BRILLINGER, D. R. (Jan. 1982). Some contrasting examples of the time and frequency domain approaches to time series analysis. Time Series Methods in Hydrosciences, (A. H. El-Shaarawi and S. R. Esterby, eds.) Elsevier Scientific Publishing Co., Amsterdam, 1982, pp. 1-15.
3. DOKSUM, K. A. (Jan. 1982). On the performance of estimates in proportional hazard and log-linear models. Survival Analysis, (John Crowley and Richard A. Johnson, eds.) IMS Lecture Notes - Monograph Series, (Shanti S. Gupta, series ed.) 1982, 74-84.
4. BICKEL, P. J. and BREIMAN, L. (Feb. 1982). Sums of functions of nearest neighbor distances, moment bounds, limit theorems and a goodness of fit test. Ann. Prob., Feb. 1982, 11, No. 1, 185-214.
5. BRILLINGER, D. R. and TUKEY, J. W. (March 1982). Spectrum estimation and system identification relying on a Fourier transform. The Collected Works of J. W. Tukey, vol. 2, Wadsworth, 1985, 1001-1141.
6. BERAN, R. (May 1982). Jackknife approximation to bootstrap estimates. Ann. Statist., March 1984, 12 No. 1, 101-118.
7. BICKEL, P. J. and FREEDMAN, D. A. (June 1982). Bootstrapping regression models with many parameters. Lehmann Festschrift, (P. J. Bickel, K. Doksum and J. L. Hodges, Jr., eds.) Wadsworth Press, Belmont, 1983, 28-48.
8. BICKEL, P. J. and COLLINS, J. (March 1982). Minimizing Fisher information over mixtures of distributions. Sankhyā, 1983, 45, Series A, Pt. 1, 1-19.
9. BREIMAN, L. and FRIEDMAN, J. (July 1982). Estimating optimal transformations for multiple regression and correlation.
10. FREEDMAN, D. A. and PETERS, S. (July 1982, revised Aug. 1983). Bootstrapping a regression equation: some empirical results. JASA, 1984, 79, 97-106.
11. EATON, M. L. and FREEDMAN, D. A. (Sept. 1982). A remark on adjusting for covariates in multiple regression.
12. BICKEL, P. J. (April 1982). Minimax estimation of the mean of a mean of a normal distribution subject to doing well at a point. Recent Advances in Statistics, Academic Press, 1983.
14. FREEDMAN, D. A., ROTHENBERG, T. and SUTCH, R. (Oct. 1982). A review of a residential energy end use model.
15. BRILLINGER, D. and PREISLER, H. (Nov. 1982). Maximum likelihood estimation in a latent variable problem. Studies in Econometrics, Time Series, and Multivariate Statistics, (eds. S. Karlin, T. Amemiya, L. A. Goodman). Academic Press, New York, 1983, pp. 31-65.
16. BICKEL, P. J. (Nov. 1982). Robust regression based on infinitesimal neighborhoods. Ann. Statist., Dec. 1984, 12, 1349-1368.
17. DRAPER, D. C. (Feb. 1983). Rank-based robust analysis of linear models. I. Exposition and review.
18. DRAPER, D. C. (Feb 1983). Rank-based robust inference in regression models with several observations per cell.
19. FREEDMAN, D. A. and FIENBERG, S. (Feb. 1983, revised April 1983). Statistics and the scientific method, Comments on and reactions to Freedman, A rejoinder to Fienberg's comments. Springer New York 1985 Cohort Analysis in Social Research, (W. M. Mason and S. E. Fienberg, eds.).
20. FREEDMAN, D. A. and PETERS, S. C. (March 1983, revised Jan. 1984). Using the bootstrap to evaluate forecasting equations. J. of Forecasting, 1985, Vol. 4, 251-262.
21. FREEDMAN, D. A. and PETERS, S. C. (March 1983, revised Aug. 1983). Bootstrapping an econometric model: some empirical results. JBES, 1985, 2, 150-158.



22. FREEDMAN, D. A. (March 1983). Structural-equation models: a case study.
23. DAGGETT, R. S. and FREEDMAN, D. (April 1983, revised Sept. 1983). Econometrics and the law: a case study in the proof of antitrust damages. Proc. of the Berkeley Conference, in honor of Jerzy Neyman and Jack Kiefer. Vol I pp. 123-172. (L. Le Cam, R. Olshen eds.) Wadsworth, 1985.
24. DOKSUM, K. and YANDELL, B. (April 1983). Tests for exponentiality. Handbook of Statistics. (P. R. Krishnaiah and P. K. Sen, eds.) 4, 1984.
25. FREEDMAN, D. A. (May 1983). Comments on a paper by Markus.
26. FREEDMAN, D. (Oct. 1983, revised March 1984). On bootstrapping two-stage least-squares estimates in stationary linear models. Ann. Statist., 1984, 12, 827-842.
27. DOKSUM, K. A. (Dec. 1983). An extension of partial likelihood methods for proportional hazard models to general transformation models. Ann. Statist., 1987, 15, 325-345.
28. BICKEL, P. J., GOETZE, F. and VAN ZWET, W. R. (Jan. 1984). A simple analysis of third order efficiency of estimates. Proc. of the Neyman-Kiefer Conference, (L. Le Cam, ed.) Wadsworth, 1985.
29. BICKEL, P. J. and FREEDMAN, D. A. Asymptotic normality and the bootstrap in stratified sampling. Ann. Statist. 12 470-482.
30. FREEDMAN, D. A. (Jan. 1984). The mean vs. the median: a case study in 4-R Act litigation. JRES, 1985 Vol 3 pp. 1-13.
31. STONE, C. J. (Feb. 1984). An asymptotically optimal window selection rule for kernel density estimates. Ann. Statist., Dec. 1984, 12, 1285-1297.
32. BREIMAN, L. (May 1984). Nail finders, edifices, and Oz.
33. STONE, C. J. (Oct. 1984). Additive regression and other nonparametric models. Ann. Statist., 1985, 13, 689-705.
34. STONE, C. J. (June 1984). An asymptotically optimal histogram selection rule. Proc. of the Berkeley Conf. in Honor of Jerzy Neyman and Jack Kiefer (L. Le Cam and R. A. Olshen, eds.), II, 513-520.
35. FREEDMAN, D. A. and NAVIDI, W. C. (Sept. 1984, revised Jan. 1985). Regression models for adjusting the 1980 Census. Statistical Science, Feb 1986, Vol. 1, No. 1, 3-39.
36. FREEDMAN, D. A. (Sept. 1984, revised Nov. 1984). De Finetti's theorem in continuous time.
37. DIACONIS, P. and FREEDMAN, D. (Oct. 1984). An elementary proof of Stirling's formula. Amer. Math Monthly, Feb 1986, Vol. 93, No. 2, 123-125.
38. LE CAM, L. (Nov. 1984). Sur l'approximation de familles de mesures par des familles Gaussiennes. Ann. Inst. Henri Poincaré, 1985, 21, 225-287.
39. DIACONIS, P. and FREEDMAN, D. A. (Nov. 1984). A note on weak star uniformities.
40. BREIMAN, L. and IHAKA, R. (Dec. 1984). Nonlinear discriminant analysis via SCALING and ACE.
41. STONE, C. J. (Jan. 1985). The dimensionality reduction principle for generalized additive models.
42. LE CAM, L. (Jan. 1985). On the normal approximation for sums of independent variables.
43. BICKEL, P. J. and YAHAV, J. A. (1985). On estimating the number of unseen species: how many executions were there?
44. BRILLINGER, D. R. (1985). The natural variability of vital rates and associated statistics. Biometrics, to appear.
45. BRILLINGER, D. R. (1985). Fourier inference: some methods for the analysis of array and nonGaussian series data. Water Resources Bulletin, 1985, 21, 743-756.
46. BREIMAN, L. and STONE, C. J. (1985). Broad spectrum estimates and confidence intervals for tail quantiles.

47. DABROWSKA, D. M. and DOKSUM, K. A. (1985, revised March 1987). Partial likelihood in transformation models with censored data.
48. HAYCOCK, K. A. and BRILLINGER, D. R. (November 1985). LIBDRB: A subroutine library for elementary time series analysis.
49. BRILLINGER, D. R. (October 1985). Fitting cosines: some procedures and some physical examples. Joshi Festschrift, 1986. D. Reidel.
50. BRILLINGER, D. R. (November 1985). What do seismology and neurophysiology have in common? - Statistics! Comptes Rendus Math. Rep. Acad. Sci. Canada, January, 1986.
51. COX, D. D. and O'SULLIVAN, F. (October 1985). Analysis of penalized likelihood-type estimators with application to generalized smoothing in Sobolev Spaces.
52. O'SULLIVAN, F. (November 1985). A practical perspective on ill-posed inverse problems: A review with some new developments. To appear in Journal of Statistical Science.
53. LE CAM, L. and YANG, G. L. (November 1985, revised March 1987). On the preservation of local asymptotic normality under information loss.
54. BLACKWELL, D. (November 1985). Approximate normality of large products.
55. FREEDMAN, D. A. (December 1985, revised Dec. 1986). As others see us: A case study in path analysis. Prepared for the Journal of Educational Statistics.
56. LE CAM, L. and YANG, G. L. (January 1986). Distinguished Statistics, Loss of information and a theorem of Robert B. Davies.
57. LE CAM, L. (February 1986). On the Bernstein - von Mises theorem.
58. O'SULLIVAN, F. (January 1986). Estimation of Densities and Hazards by the Method of Penalized likelihood.
59. ALDOUS, D. and DIACONIS, P. (February 1986). Strong Uniform Times and Finite Random Walks.
60. ALDOUS, D. (March 1986). On the Markov Chain simulation Method for Uniform Combinatorial Distributions and Simulated Annealing.
61. CHENG, C-S. (April 1986). An Optimization Problem with Applications to Optimal Design Theory.
62. CHENG, C-S., MAJUMDAR, D., STUFKEN, J. & TURE, T. E. (May 1986, revised Jan 1987). Optimal step type design for comparing test treatments with a control.
63. CHENG, C-S. (May 1986, revised Jan. 1987). An Application of the Kiefer-Wolfowitz Equivalence Theorem.
64. O'SULLIVAN, F. (May 1986). Nonparametric Estimation in the Cox Proportional Hazards Model.
65. ALDOUS, D. (JUNE 1986). Finite-Time Implications of Relaxation Times for Stochastically Monotone Processes.
66. PITMAN, J. (JULY 1986, revised November 1986). Stationary Excursions.
67. DABROWSKA, D. and DOKSUM, K. (July 1986, revised November 1986). Estimates and confidence intervals for median and mean life in the proportional hazard model with censored data.
68. LE CAM, L. and YANG, G.L. (July 1986). Distinguished Statistics, Loss of information and a theorem of Robert B. Davies (Fourth edition).
69. STONE, C.J. (July 1986). Asymptotic properties of logspline density estimation.
71. BICKEL, P.J. and YAHAV, J.A. (July 1986). Richardson Extrapolation and the Bootstrap.
72. LEHMANN, E.L. (July 1986). Statistics - an overview.
73. STONE, C.J. (August 1986). A nonparametric framework for statistical modelling.

74. BIANE, PH. and YOR, M. (August 1986). A relation between Lévy's stochastic area formula, Legendre polynomials, and some continued fractions of Gauss.
75. LEHMANN, E.L. (August 1986, revised July 1987). Comparing Location Experiments.
76. O'SULLIVAN, F. (September 1986). Relative risk estimation.
77. O'SULLIVAN, F. (September 1986). Deconvolution of episodic hormone data.
78. PITMAN, J. & YOR, M. (September 1987). Further asymptotic laws of planar Brownian motion.
79. FREEDMAN, D.A. & ZEISEL, H. (November 1986). From mouse to man: The quantitative assessment of cancer risks.
80. BRILLINGER, D.R. (October 1986). Maximum likelihood analysis of spike trains of interacting nerve cells.
81. DABROWSKA, D.M. (November 1986). Nonparametric regression with censored survival time data.
82. DOKSUM, K.J. and LO, A.Y. (November 1986). Consistent and robust Bayes Procedures for Location based on Partial Information.
83. DABROWSKA, D.M., DOKSUM, K.A. and MIURA, R. (November 1986). Rank estimates in a class of semiparametric two-sample models.
84. BRILLINGER, D. (December 1986). Some statistical methods for random process data from seismology and neurophysiology.
85. DIACONIS, P. and FREEDMAN, D. (December 1986). A dozen de Finetti-style results in search of a theory.
86. DABROWSKA, D.M. (January 1987). Uniform consistency of nearest neighbour and kernel conditional Kaplan - Meier estimates.
87. FREEDMAN, D.A., NAVIDI, W. and PETERS, S.C. (February 1987). On the impact of variable selection in fitting regression equations.
88. ALDOUS, D. (February 1987, revised April 1987). Hashing with linear probing, under non-uniform probabilities.
89. DABROWSKA, D.M. and DOKSUM, K.A. (March 1987, revised November 1987). Estimating and testing in a two sample generalized odds rate model.
90. DABROWSKA, D.M. (March 1987). Rank tests for matched pair experiments with censored data.
91. DIACONIS, P and FREEDMAN, D.A. (March 1987). A finite version of de Finetti's theorem for exponential families, with uniform asymptotic estimates.
92. DABROWSKA, D.M. (April 1987, revised September 1987). Kaplan-Meier estimate on the plane.
- 92a. ALDOUS, D. (April 1987). The Harmonic mean formula for probabilities of Unions: Applications to sparse random graphs.
93. DABROWSKA, D.M. (June 1987). Nonparametric quantile regression with censored data.
94. DONOHO, D.L. & STARK, P.B. (June 1987). Uncertainty principles and signal recovery.
95. RIZZARDI, F. (Aug 1987). Two-Sample t-tests where one population SD is known.
96. BRILLINGER, D.R. (June 1987). Some examples of the statistical analysis of seismological data.  
To appear in *Proceedings, Centennial Anniversary Symposium, Seismographic Stations, University of California, Berkeley.*
97. FREEDMAN, D.A. and NAVIDI, W. (June 1987). On the multi-stage model for cancer.
98. O'SULLIVAN, F. and WONG, T. (June 1987). Determining a function diffusion coefficient in the heat equation.

99. O'SULLIVAN, F. (June 1987). Constrained non-linear regularization with application to some system identification problems.
100. LE CAM, L. (July 1987, revised Nov 1987). On the standard asymptotic confidence ellipsoids of Wald.
101. DONOHO, D.L. and LIU, R.C. (July 1987). Pathologies of some minimum distance estimators.
102. BRILLINGER, D.R., DOWNING, K.H. and GLAESER, R.M. (July 1987). Some statistical aspects of low-dose electron imaging of crystals.
103. LE CAM, L. (August 1987). Harald Cramér and sums of independent random variables.
104. DONOHO, A.W., DONOHO, D.L. and GASKO, M. (August 1987). Macspin: Dynamic graphics on a desktop computer.
105. DONOHO, D.L. and LIU, R.C. (August 1987). On minimax estimation of linear functionals.
106. DABROWSKA, D.M. (August 1987). Kaplan-Meier estimate on the plane: weak convergence, LIL and the bootstrap.
107. CHENG, C-S. (August 1987). Some orthogonal main-effect plans for asymmetrical factorials.
108. CHENG, C-S. and JACROUX, M. (August 1987). On the construction of trend-free run orders of two-level factorial designs.
109. KLASS, M.J. (August 1987). Maximizing  $E \max_{1 \leq k \leq n} S_k^+ / ES_n^+$ : A prophet inequality for sums of I.I.D. mean zero variates.
110. DONOHO, D.L. and LIU, R.C. (August 1987). The "automatic" robustness of minimum distance functionals.
111. BICKEL, P.J. and GHOSH, J.K. (August 1987). A decomposition for the likelihood ratio statistic and the Bartlett correction — a Bayesian argument.
112. BURDZY, K., PITMAN, J.W. and YOR, M. (September 1987). Some asymptotic laws for crossings and excursions.
113. ADHIKARI, A. and PITMAN, J. (September 1987). The shortest planar arc of width 1.
114. RITOV, Y. (September 1987). Estimation in a linear regression model with censored data.
115. BICKEL, P.J. and RITOV, Y. (September 1987). Large sample theory of estimation in biased sampling regression models I.
116. RITOV, Y. and BICKEL, P.J. (September 1987). Unachievable information bounds in non and semiparametric models.
117. RITOV, Y. (October 1987). On the convergence of a maximal correlation algorithm with alternating projections.
118. ALDOUS, D.J. (October 1987). Meeting times for independent Markov chains.
119. HESSE, C.H. (October 1987). An asymptotic expansion for the mean of the passage-time distribution of integrated Brownian Motion.
120. DONOHO, D. (October 1987). Geometrizing rates of convergence, II.
121. BRILLINGER, D.R. (October 1987). Estimating the chances of large earthquakes by radiocarbon dating and statistical modelling. To appear in *Statistics a Guide to the Unknown*.
122. ALDOUS, D., FLANNERY, B. and PALACIOS, J.L. (November 1987). Two applications of urn processes: The fringe analysis of search trees and the simulation of quasi-stationary distributions of Markov chains.
123. DONOHO, D.L. and TAYLOR, B.M. (November 1987). Minimax risk for hyperrectangles.
124. ALDOUS, D. (November 1987). Stopping times and tightness II.

125. HESSE, C.H. (November 1987). The present state of a stochastic model for sedimentation.
126. DALANG, R.C. (December 1987). Optimal stopping of two-parameter processes on hyperfinite probability spaces.
127. DONOHO, D. and GASKO, M. (December 1987). Multivariate generalizations of the median and trimmed mean I.
128. DONOHO, D. and GASKO, M. (December 1987). Multivariate generalizations of the median and trimmed mean II.
129. SMITH, D.L. (December 1987). Vapnik-Cervonenkis classes.
130. STONE, C.J. (November 1987). Uniform error bounds involving logspline models.
131. HESSE, C.H. (December 1987). Numerical and statistical aspects of neural networks.
132. HESSE, C.H. (December 1987). A Bahadur - Type representation for empirical quantiles of a range class of stationary, possibly infinite - variance, linear processes

Copies of these Reports plus the most recent additions to the Technical Report series are available from the Statistics Department technical typist in room 379 Evans Hall or may be requested by mail from:

Department of Statistics  
University of California  
Berkeley, California 94720

Cost: \$1 per copy.