High Interaction Graphics

Stephen G. Eick and Graham J. Wills
AT&T Bell Laboratories, 1000 E. Warrenville Road, Naperville, IL 60566, USA
Email: eick@research.att.com and gwills@research.att.com

Abstract: Examining data using graphical tools, such as histograms, quantile plots, scatterplots and the like, is a necessary part of any serious analysis effort. With the advent of inexpensive graphics-capable desktop computing, such tools are generally available. But the use of computers enables more than simply reproducing static plots on a display; it allows users to interact with plots, changing parameters, querying, zooming and linking plots together so that interesting features of one plot can be seen in the light of the others. In this paper we discuss the core features of interactive graphics, investigate how familiar plots can be made interactive and show examples of interactive graphics for general and specific data analysis.

1.0 Introduction

Looking at data to provide information is an old subject. Hieroglyphics from prehistoric times show that data analysis was important to early man. Abstract displays of information (graphs, plots, etc.) are a more recent innovation at around 1750-1800 (Tufte, 1983). A still more recent development is the concept of interactive displays of information, displays which users can change with minimal effort and small latency. These interactive graphical tools have only recently been realizable, as they require the use of a computer. Now, with the advent of powerful desktop machines capable of rendering tens of thousands of symbols in a fraction of a second, these techniques can be used by a wide range of people for a variety of real-world tasks.

1.1 Definition

We define *Interactive Graphical Methods* as the class of techniques for exploring data that allow the user to manipulate a graphical representation of the data. Becker et al. (1987) defines the process thus:

> "The data analyst takes an action through manual manipulation of an input device and something happens on the screen. These computing capabilities provide a new medium for the invention of graphical methods for data analysis."

The roots of interactive graphics research lie in *Exploratory Data Analysis (EDA)* (Tukey, 1977); a set of techniques for investigating data to spot trends, patterns, errors and features. A key feature of EDA is the use of simple, robust plots to show characteristics of the data. According to Tukey EDA is about "looking at data to see what it seems to say" (p. v). "It is detective work - numerical detective work - or counting detective work - or graphical detective work". (p. 1) and "Unless exploratory data analysis uncovers indica-

tions, usually quantitative ones, there is likely to be nothing for confirmatory data analysis to consider" (p. 3). For Tukey, the burden of discovering information in the data falls on EDA, whereas the burden of proving that the information is not spurious falls on the traditional data analysis methods.

Interactive Graphics (also known as Direct Manipulation Graphics or Dynamic Graphics) provide a satisfying extension of the principles of EDA to the computing environment. Many of the graphic facilities developed for exploratory graphic analysis are present in interactive graphic analysis, but in an enhanced form. We define an *Interactive Graphic View* as a pictorial representation of some form of data or information which the analyst can manipulate in real time[1]. By using a pointing device such as a mouse, pressing keys on the keyboard (but not typing in commands) or via some other such input device, the analyst can specify, usually visually, areas of the plot and effect changes in the display method instantaneously. The effect is that the analyst is able to manipulate the parameters of the plot directly, giving the impression that they are almost 'touching' the data.

1.2 Advantage of Interactive Graphics

A typical analysis consists of examining a set of data in order to examine its properties or to answer some specific question, commonly about the relationship of one or two variables to the rest. The analyst's task is to discover those properties or to answer the questions about the data. One informative paper in this respect is the meta-analysis paper of Hoaglin and Velleman (1991) in which the authors look at 15 different attempts at analyzing a set of data relating to baseball. This data set consists of a number of variables recorded for 439 baseball players and the purpose of the analysis was to answer the question "Are players paid according to their performance?"

Hoaglin and Velleman note that "almost everyone began by displaying the data" and that most groups decided to transform salary and use log(salary) instead, based on the displays created. Similarly, looking at the skewness inherent in career runs invited several groups to re-express this as an annual rate (career runs / years) or to use a root transform. Hoaglin and Velleman further state that

> "those working with log(salary) and with annual rate predictors fared better than those who worked with raw forms of these variables. The models built were more successful at prediction, at identifying errors in the data, and at providing interpretable choices of predictors".

This highlights the use of graphics for exploratory work; to see what model is most suitable for the data. A related pre-modelling use is to indicate data errors.

After fitting the model the analyst will then typically employ further graphical tools to check the validity of his model and look for possible outliers. In the baseball example, two erroneous salary levels were identified and also one value which was an outlier, but not erroneous; this data value was for a player who was both a coach and a player.

1. By "real-time" we mean that the response rate of the system must be fast. To maintain the illusion of manipulating the data directly, a response time of 50 ms or less is required. For continuously changing displays, such as rotating views, this equates to drawing speeds of 20 frames per second.

Having given the case for general graphical methods, why should the graphics be interactive? There are several reasons why interactivity significantly improves static displays:

Clarity. By allowing users to change what they see by manipulating the display, and by showing associated information only on demand, displays can be made cleaner, focusing on the information displayed and allowing a range of options. Consider Figure 1 below, in which a scatterplot produced by a data analysis program is compared with one from a presentation graphics program. The shadowing of points in the latter does nothing to aid our understanding - indeed it may hinder our ability to recognize patterns in the display. The grid lines and heavily labelled axes are useful for static displays as allow the user to read the values precisely, but in an interactive display the user can more easily point to the dot and have the computer display its values. Thus we can dispense with the gridlines to achieve (a).

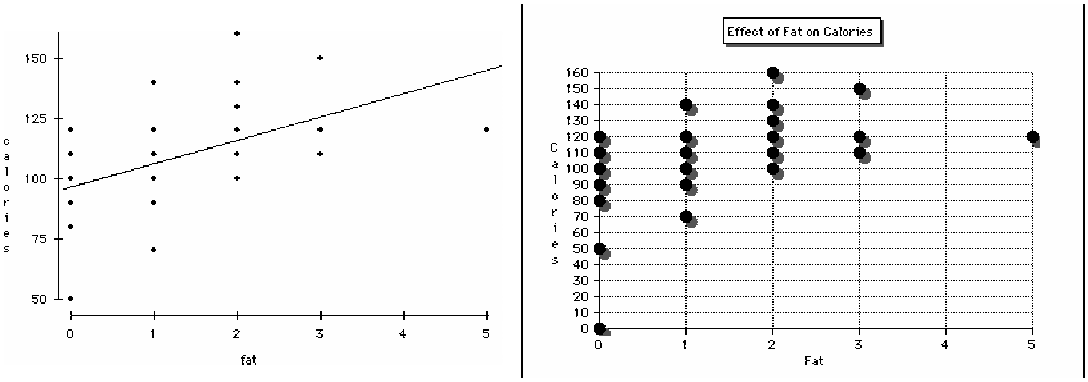

FIGURE 1. (a) A simple scatterplot as created by an exploratory data analysis program. (b) A less clear scatterplot from a presentation program.

Robustness. Since the user can modify the display simply and easily, he can examine a wide range of ways of viewing the data, thus avoiding drawing inferences from only one view of the data. Even as simple a view as a histogram can be very misleading if the intervals are chosen in unfortunate ways. Allowing the user to vary the number of bars might detect regularity in the data that would otherwise be hidden. In Figure 2(a) we show a histogram of a rational variable, drawn using the program default for the number of bars displayed. If we interactively increase the number of bars, as in (b), we see gaps appear in the histogram, indicating some form of dicretization not apparent in (a).

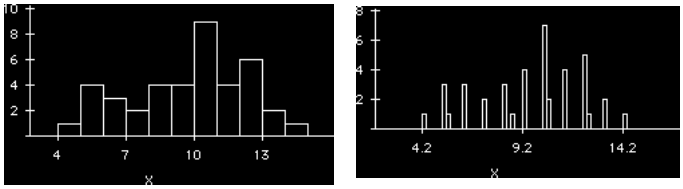

FIGURE 2. A Histogram drawn with (a) few bars and (b) many bars.

Power. Using interactivity allows the user to combine views and leverage the value of one view in another, creating a more powerful exploratory engine. It effectively elevates the utility of the whole to more than the sum of the parts involved. In Figure 3 we have used the mouse to select an area of the barchart corresponding to cereals displayed on the middle shelf in the supermarket. These cereals are differentiated in the scatterplot and are

clustered together. When we discuss linking in section 2.4, we will examine the linking concept in more depth.

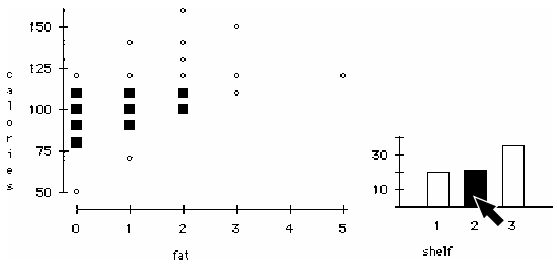

FIGURE 3. Linking from a Bar Chart to a Scatterplot. The bar chart shows which shelf a cereal product has been displayed on and the scatterplot shows the relationship between fat and calory content for the cereals.

Possibility. Some data is very hard to display without using dynamic or interactive facilities. Three-dimensional scatterplots are a well-known example. In Figure 4 the data form a three-dimensional shape which is extremely hard to see from these plots, whereas a cursory examination of the dynamic version immediately makes the parabolic nature of the surface on which these points fall apparent.

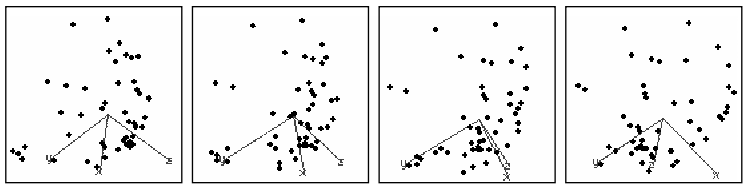

FIGURE 4. Four snapshots of a rotating plot in motion.

Other types of data are naturally shown by an interactive display. For example, Tufte (1983, p170) shows 23 similar colored contour maps of pollution in Los Angeles, describing the change in pollution levels across the city on an hour by hour basis. He notes that:

> "Small multiples resemble the frames in a movie: a series of graphics, showing the same combination of variables, indexed by changes in another variable. The design remains constant through all the frames, so that attention is devoted entirely to shifts in the data."

This is a revealing statement. The comparison with a movie is apt. If we could, we would like to view the frames as a movie, so that we would not have to move our focus of attention, but could just register the changes in the spatial distribution. Of course, we would also want controls to allow us to step forward and back and compare frames which were not adjacent; for example, the initial and final frames. Tufte, in 1983, is calling for the ability to animate a series of plots, each drawn as similar as possible, given that one parameter (time in this case) is changing.

1.3 History

Dynamic display and high-interaction techniques for the analysis of statistical data are recent phenomena. One of the earliest pieces of work in this area was by J. Tukey, called PRIM-9, which was a system which enabled the user to view a multi-dimensional cloud of

data and rotate it on the computer screen in real-time. PRIM stands for Picturing, Rotation, Isolation and Masking, the four major components of its operation, and was “a result of a continuing program of research into techniques for applying computer graphics to exploratory data analysis” (Fisherkeller, Friedman and Tukey, 1974; reprinted in Cleveland and McGill, 1988). In this early work, interaction was necessarily somewhat crude, being by means of an alphanumeric keyboard, light-pen and a function keyboard with 32 buttons, but the principle was of lasting importance. In the conclusions to the paper referenced above, the authors state they “now recognize that these details of control can make or break such a [dynamic pictorial] system” and such has been our experience. It is for this reason that the details of a high-interaction environment are discussed in some length in the sections that follow.

In more recent years other interactive techniques such as dynamic scatterplot brushing have been introduced into statistical thought. Dynamic scatterplot brushing is implemented by the creation of a matrix of $p(p-1)/2$ scatterplots for each pair of variables x_i,x_j in a data set consisting of p variables. Then, using a pointing device, a small region is moved over one of the scatterplots, with the result that points that lie within this region are highlighted both in that scatterplot and in the other scatterplots which comprise the matrix.

This early work in the field of interactive graphics focussed on individual plots. The idea was to create unique views with which the user would interact to gain a more thorough understanding of the data. More recently the emphasis has been on making interaction pervasive throughout the analysis system. Environments such as Data Desk (Velleman 1988), XGOBI (Swayne et al. 1991) and JMP (part of the SAS system) have provided general purpose statistical environments where interaction is an integral part of the analysis. Furthermore, there has been much research into designing specific interactive systems to solve more narrowly defined problems. In section 4.0 we examine some of these in detail.

2.0 Principles

2.1 Simple, easily interpretable views

Since interactive graphics methods are mainly exploratory and diagnostic in intent, the graphical displays must be easy to interpret. Preferably, the results of an analysis should ‘jump out’ at you, as in the case of a bivariate outlier in a scatterplot. A good display will

a) Be obvious as to what it is displaying
b) Focus attention on the data
c) Give indications of scale and location of the data

Unnecessary and distracting features include: (a) over-labelling the data; (b) ‘decorative’ graphical elements which do not aid understanding of the data (termed “chartjunk” by Tufte); (c) redundantly showing the same information; (d) negligent use of color.

The purpose of an interactive graphical display is to use graphical elements to encode the data in such a way as to make patterns apparent and invite exploration and understanding

of the data by manipulating its appearance. To make this encoding obvious, the views should be as easy to decode as possible and therefore use graphical elements which are as simple as possible for the human perceptual system to decode. For example, a bar chart is a better way to compare counts of a categorical variable than a pie-chart because lengths are easier for people to compare than angles (Bertin, 1973). Cleveland (1985, p254) gives the following ordering of difficulty of decoding visual cues, from easiest to hardest:

a) Position along a common scale
b) Position along identical, nonaligned scales
c) Length
d) Angle
e) Area
f) Volume
g) Color hue
h) Color saturation
i) Density

He states that:

> "The ordering is based on theory of visual perception, on experiments in graphical perception, and on informal experimentation. An important principle of data display is that we should encode data on a graph so that the visual decoding involves tasks as high in the ordering as possible."

Since Cleveland is discussing static graphical views, his list is limited to that domain. But there are other cues commonly used in interactive graphics. How these are ranked is an open question:

a) Sonic cues (pitch, timbre, duration, rhythm, etc.)
b) Motion cues (flicker, motion parallax, velocity)
c) Stereo depth cuing

Furthermore, since the user can typically manipulate elements of an interactive graphic in real-time, we must be concerned with the *dynamic* aspects of all these attributes. In general it seems fair to say that if we can compare lengths better than we can compare hues, we should be able to compare changes in length better than changes in hues. But we must be very careful in our generalizations, as the inter-relationships within our perceptual systems for analyzing size, color and motion are poorly understood, and so such generalizations must be made with the understanding that they are provisional only.

2.2 Information Hiding

In Cleveland (1993, pp. 172-176) we see a good example of how simple design and information hiding work together to make data exploration easier. Cleveland uses an historic data set consisting of brain weight and body weight for a variety of animals. He uses a simple scatterplot to show the linear relationship between log(brain weight) and log(body weight). Each animal's statistics are plotted using a circle on the scatterplot. The information being hidden here is which animal is represented by which circle. Traditionally, this

information would be shown by labelling each plotted circle, but Cleveland's figure 3.88 shows that this creates an unreadable mess of overplotted labels. The interactive solution Cleveland proposes to avoid this is to use the mouse to indicate which symbol is of interest, by moving it over that symbol, and then having the label appear beside the symbol.

This is an example of the basic principle of information hiding. Unless information is always required for interpreting a data view, it should be hidden until the user requests it. Furthermore, the user should be able to request effortlessly information only for items deemed interesting by his analysis. Using another example, the Iris data set made famous by Fisher (Fisher, 1971), we look at the relationship between sepal width and length for three varieties of sepal. Figure 5(a) shows the width vs. length scatterplot where the variety of Iris is indicated by plotting each measurement as '1', '2' or '3'.

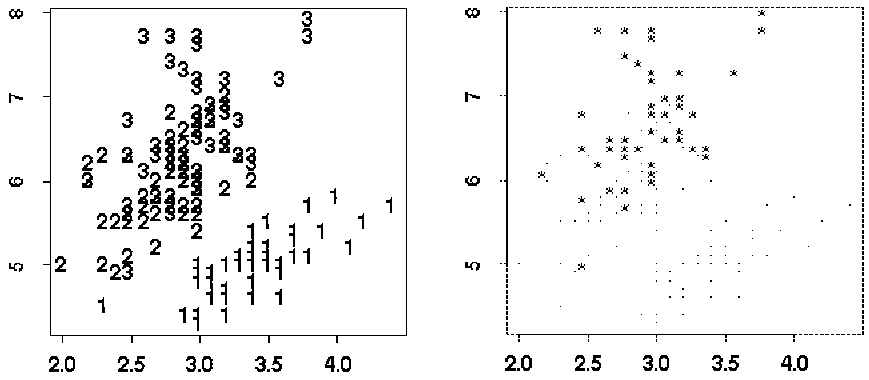

FIGURE 5. Plots of Sepal lengths against widths for three varieties of Iris; (a) displays members of each variety as a number, (b) displays members of varieties 1 and 2 as dots, and 3 as stars.

Although the group formed by variety 1 stands out clearly, the group formed by varieties 2 and 3 are confused. Showing all the information makes understanding harder. In Figure 5(b), we have hidden the variety information for varieties 1 and 2, showing only variety 3 differently (as stars). The information about variety3 relative to the others is now easier to ascertain. If we provide some intuitive method for allowing the user to use these hiding/revealing methods, such as the concept of linking discussed in Section 2.4, the user can navigate easily even through information-rich data sets.

2.3 Direct Manipulation

Moving the mouse over a point to change the display is an example of direct manipulation. When the user clicks on a rotating 3d point cloud and drags to rotate it, he is also using direct manipulation to change the representation. In both cases the key feature is that the user is directly interacting with the view of the data to alter its appearance. The data display responds to their interaction in real-time, so that the user gains the impression that they are 'touching' the data. The importance of real-time feedback is crucial here. If the user does not receive this rapid response, the effect is more of setting a parameter and seeing what the new plot is like. It makes it harder to adjust parameters to see what the most informative view is and it makes the user aware that they are dealing with an interface to the data, rather than dealing with the data itself.

Figure 6 shows two snapshots of direct manipulation in action. The data being graphed are statistics calculated on telephone traffic which indicate how much traffic there was between each pair of individuals. The data have been summarized by a smoothed histogram, where the level of smoothing is under the users control. By dragging a control bar, the user can vary the degree of smoothing between the two extrema shown in Figure 6. In this case, the highly smoothed version shows the Poisson nature of the distribution, whereas the unsmoothed version shows that the data are quite discrete in nature, with some form of transformation applied giving increasingly spaced discrete components. Both pieces of information are important for a thorough understanding of the data.

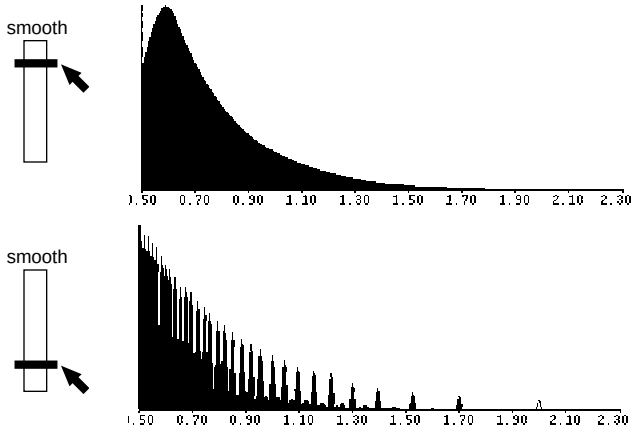

FIGURE 6. Direct Manipulation of a smoothed histogram for telephone connection data.

2.4 Linked Views

Linking is an old concept, as can be seen from examination of the relevant literature. For static graphics a number of methods have been employed; possibly the most widespread of which is to split points into a number of classes and to assign a unique symbol or color to each class and use that symbol or color for points within that class in each plot that is shown. An excellent example of this is shown on page 172 of Chambers et al. (1983) where the Iris data set has been displayed by a scatterplot matrix of petal width, petal length, sepal width and sepal length, with the symbol used to plot each point in each scatterplot being a dot, circle or cross (‘.’, ‘o’ or ‘x’) depending on the variety of iris on which the measurements were made. This coding links the scatterplots together in a way which augments the natural linking caused by the juxtaposition of the scatterplots within the matrix structure. As an aside, note that Cleveland and McGill (1984) point out that plotting symbols of different colors provide the most acceptable coding mechanism.

Linking is an intuitively attractive idea for increasing the information content of a set of data plots. Linking shows visually which parts of one data plot correspond to that of another.This allows the interesting or anomalous areas of one view of the data to be seen in the context of other views of the data in a rapid and intuitive fashion. This is a general concept, but for data analysis one common method used is to associate a state with each datum, either *highlighted* or *unhighlighted*. The subset of data which is marked highlighted is assumed to be the focus of interest to the user. The user should be able to high-

light a subset of data identified in one plot and see the results of that selection in another plot.

For data analysis using computers, the obvious generalization of this static method of linking plots is to allow the user to select a subset to be drawn in a different fashion interactively. A common method of doing this is known as scatterplot brushing, mentioned in section 4.3. In this paradigm, the user moves a region (typically a rectangle, although Tukey, commenting on Becker, Cleveland and Wilks, 1987, notes the possibility of using other brush shapes) known as the brush around the scatterplot matrix, with the effect that any points which fall within this area are highlighted, that is, they are drawn in a manner which separates them from the rest of the data. This highlighting occurs in each plot of the matrix simultaneously, so that by brushing an interesting area of one plot the user can see whether the subset of points corresponding to that area has other features indicated by the other scatterplot panels.

Although a useful tool for scatterplot examination, scatterplot brushing restricts the value of linking considerably. Why restrict linking to just one kind of plot - the scatterplot? To expand the linking paradigm for more general situations a more universal form of linking is required. Such linking was first introduced in a commercial statistical environment by Velleman in the program DataDesk (Velleman, 1988) for a number of simple types of plot, such as lists, histograms, pie charts, bar charts and scatterplots. In this program a number of different types of data plots and lists are linked to one another so that, for example, clicking on a column of a bar chart causes the corresponding points in a scatterplot view to be drawn differently from the non-selected points. Figure 5(b) showed an example of this functionality. A scatterplot of sepal width against length was created and a bar chart of type was created. By clicking on the bar representing variety 3, the appropriate points in the scatterplot were highlighted by drawing them as stars, rather than dots.

Figure 12 below shows the use of linking in the spatial domain, where selecting a number of bars from a histogram highlights the corresponding areas on a map.

There are a number of important issues concerning linking, including problems such as how to *un*highlight highlighted data items and the exact method of click and dragging to highlight data items corresponding to parts of a display, but these topics are too involved for a survey paper. Chambers et al. (1983), Cleveland and McGill (1984) and Becker et al. (1987) discuss these issues in greater depth.

3.0 Basic Interactive Graphics Views

In this section, we examine *general* interactive data views. These are representations which require the data to satisfy very few requirements and so are applicable to a wide variety of analyses. These views are based on static exploratory views of proven value, such as histograms, bar charts and scatterplots, and we examine how interactivity can add to their ability to help the analyst in exploring data.

3.1 Univariate views

When examining one statistic, the key question to answer is “What?”. What is the distribution? What are these strange values? When linking a univariate view to other views, there are two directions to consider. If we have identified an interesting subset in another view, we want to see what that subset looks like in comparison to the overall data. If, on the other hand, we have identified an interesting area in the univariate view, we want to let the user select that subset in as natural a way as possible. Although there are many possible types of univariate plot, we consider three common univariate data views and examine how they lead naturally to interactive versions.

3.1.1 Lists

A list of values forms a very simple data view. The most obvious way to show highlighted data items is to use a different font, color or background to display them. An alternative is to re-order the list so that they appear at the top. Clicking and dragging on items of the list should highlight them, so that, for example, we can click on a label we are interested in and see where that point appears in other plots.

3.1.2 Histograms

The histogram is another plot which is comparatively easy to make interactive. We want to draw a modified histogram which will compare the highlighted subset with the overall data. Figure 7 shows a number of ways of doing this.

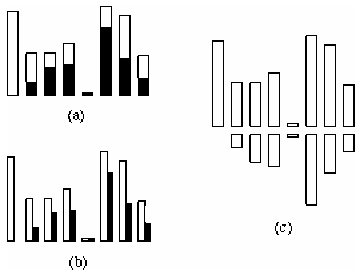

FIGURE 7. Possible methods of comparing two bar charts (a) Superimposition (b) Juxtaposition (c) Hanging from the baseline.

It is hard to compare the histograms in (c), leaving the choice between (a) and (b). We prefer (b) as it contains fewer graphical elements, presenting a simpler view.

Highlighting bars from the interactive histogram is simple. We click on a bar to highlight all data items contributing to that bar and drag the mouse to select multiple adjacent bars. Further interactivity can be added by allowing the number of bars in the histogram to be changed in real time, enabling easy re-parametrization of the plot as shown in Figure 2.

3.1.3 Boxplots

A histogram is a good visualization of a single data variable. One drawback, however, is that it takes up a lot of space on the screen, and so displaying a lot simultaneously is difficult. The boxplot is a more compact summary of the data, showing measure of location and spread for a set of data. Figure 8 shows one way in which we can compare a subset with the entire data set using boxplots.

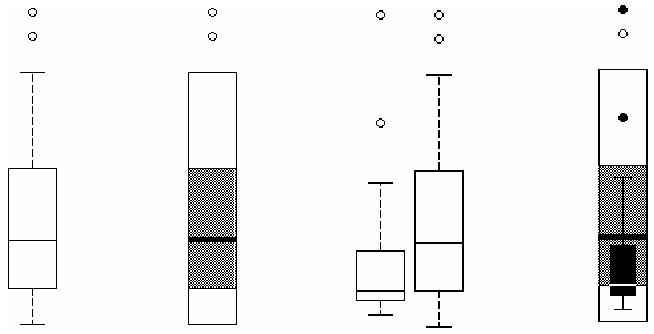

FIGURE 8. Boxplots. (a) A standard boxplot. (b) The modified version for use in linking. (c) Standard boxplots comparing the selection to the overall data. (d) The modified version for use in linking

Figure 8 (a) shows a standard boxplot, and (c) shows the same boxplot beside a boxplot to the same scale for a subset of the data. Comparing the two is not hard, but if there were many such pairs, side by side, it becomes difficult to resolve. Figure 8(b) shows a simplified version of (a), and (d) shows the superposition of a standard boxplot for the subset on this modified boxplot.

As the user selects from another plot, the base boxplot (b) will not change, but the boxplot for the selected variable will move up and down, showing level shifts, and expand or shrink, showing range shifts. As an example of the utility of these boxplots, Figure 9 shows an example taken from a study of bird breeding in Scotland. Boxplots have been constructed for the amount of temporary grass, permanent grass, barley and wheat in a number of areas of Scotland, and a histogram created for the mean temperature. Selecting bars of the histogram dynamically indicates what sort of plant favors given altitudes.

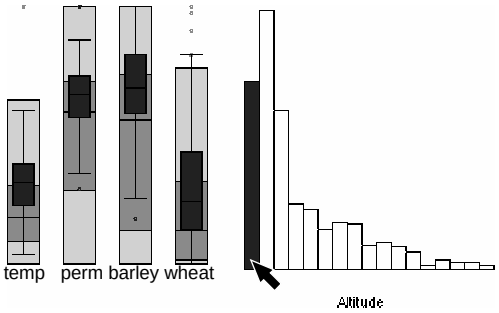

FIGURE 9. Linking altitude to grass and grain types in Scottish districts.

Selecting from the boxplot can be done by allowing the user to select points within a range by clicking and dragging over that area of the boxplot, or by clicking on outliers to see if

they are unusual in other plots also. Because the boxplot summarizes a lot of data, it is important to be able to obtain that information from the display. The ability to query outliers via the mouse and see which labels they correspond to is clearly useful, both for the set of highlighted points and the overall data.

3.2 Multivariate views

We now examine a family of interactive views deriving from the single most useful view for comparing two rational variables; the scatterplot. In previous sections we have noted that much of the motivation for interactive linking of plots came from the scatterplot brushing paradigm. In the remainder of this section we examine scatterplots and scatterplot matrices and briefly discuss three-dimensional scatterplots and associated ideas.

3.2.1 Scatterplot and Scatterplot Matrices

A scatterplot is an ideal tool for examining the relationships between two variables. In Figure 5 we showed an example of how this can be used within a multi-view linking environment. In this case we drew each highlighted point using a star rather than as a dot, thus showing the results of highlighting a group of data items from another plot. There are numerous ways of highlighting items using the scatterplot. In Figure 10 we show a matrix

of scatterplots, where each pane of the window (apart from the diagonal) shows a scatterplot of two of the three variables.

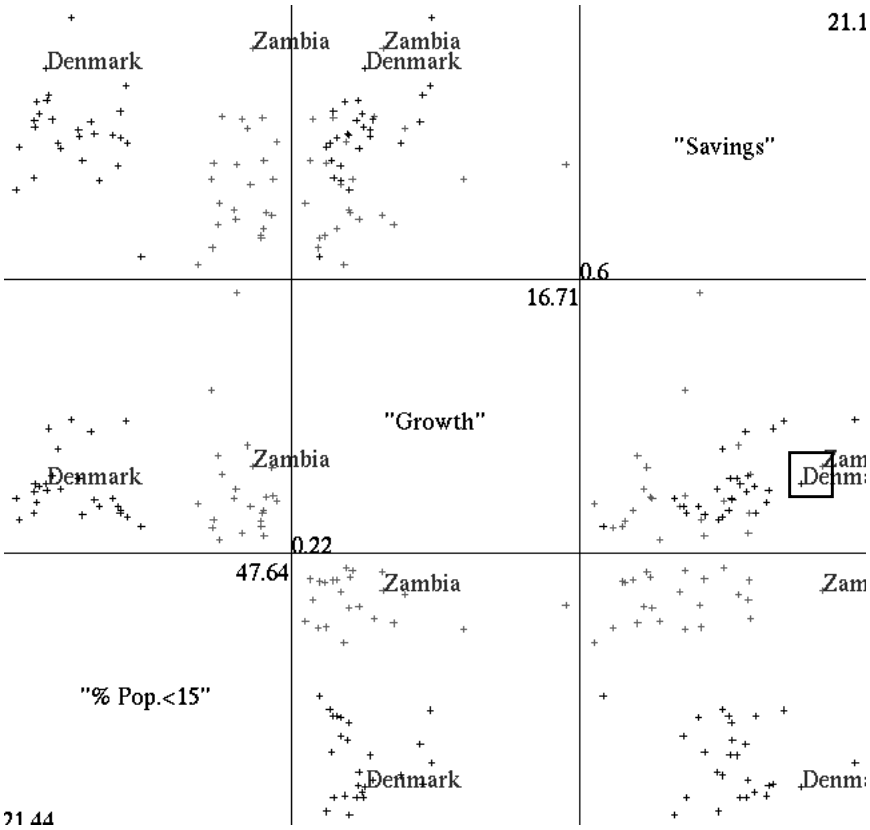

FIGURE 10. A Scatterplot Matrix for levels of Savings, Growth and Percentage of Children for 50 countries world-wide.

In this case the results of a highlight are shown in each plot by writing the names of the highlighted variables beside the associated point. The user has specified the selection by dragging a rectangular shaped *brush* across the “Savings” vs. “Growth” scatterplot and all points under the brush are highlighted. In this case *Zambia* and *Denmark*. This simple example shows that even though the two countries are similar in terms of savings and growth patterns, they are not similar at all in terms of population age.

There have been many papers relating to scatterplot brushing and scatterplot matrices in general. For a longer introduction to such methods, papers such as Becker, Cleveland and Wilks (1987) and Cleveland and McGill (1988) are particularly useful.

3.2.2 3D Scatterplots

In Figure 4 we attempted to show a three-dimensional plot of points forming a surface. Three dimensional point clouds are particularly useful for examining interactions which cannot be expressed by decomposing them into 2-way relationships. However it is all but impossible to appreciate their utility without actually using them. Since the 3D scatterplot appears on the screen as a 2D projection, selection and display are handled similarly to the 2D scatterplot case. The added interaction for these plots is the ability to spin them, and

the perceived three-dimensional motion allows the visual perceptive system to detect patterns in the data.

The control of these plots has been much examined, and current trends seem to suggest that the most effective method (for explanatory work) is to consider that the data is embedded in a clear ball, which has been set in a mounting (much like a trackball or certain styles of globe). When the user clicks on the ball and drags the ball spins in its mountings in the direction in which the drag moved. A further refinement is for the speed of rotation to be related to the speed of the dragging motion.

One statistical technique which has not been regarded as interactive is *projection pursuit*. Here, the idea is to consider a large number of variables and discover a projection (into 2 or 3 dimensions) which is interesting, for some definition of interesting. Figure 11 below shows a clever interactive extension to this idea. The authors of the program display a three-dimensional projection of the data as a rotating point cloud, and have the projection continuously changed so that the cloud changes as the user watches. This change in projection is done by moving to locally 'interesting' projections as defined by the pursuit algorithm. The basic idea is that by watching this the user will get shown a series of potentially revelatory projections of the data, and the continuous motion will allow him to retain his sense of position.

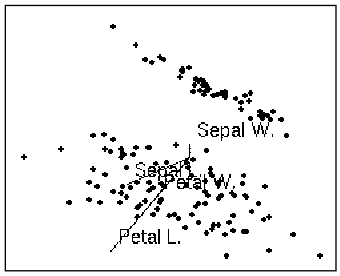

FIGURE 11. A 3D rotating point cloud showing an interesting view of the Iris data. The 4 variables are measurements of width and height on sepals and petals for 150 iris plants.

4.0 Specific Application Areas

Section 3.2.2 presented an example of a general purpose view being used for a specific task; to help understand and work with projection pursuit. Since specific and unusual forms of data are often most hard to work with analytically, and since an obvious model for the data is not always available, these custom-built interactive graphic environments can make a substantial contribution in understanding data in these areas. It would be impossible to deal with the contribution of interactive graphics to every such area, so in the following sections, we restrict ourselves to two areas.

4.1 Spatial Data Visualization

Spatial data consist of data that has some geographic or location component. A common example is measurements of several variables at each of a number of spatial locations. Interactive graphic tools are of particular use in the analysis of such data. This is a view supported by Cressie (1991, p30), talking about exploratory data analysis:

> “It is my view that geostatistics can profit considerably from the philosophy and methods of modern data analysis - A lot of time and effort is put into geological exploration, but only recently has the value of good statistical exploration been realized.”

There are several good reasons why EDA and interactive graphical tools are so useful for spatial data:

a) Analysis of spatial data constantly refers to the locations of the data. When looking at local smooths, residuals or other calculated variables the analyst will always want to see how they are distributed over space. Linking information on the data to the map is a requirement of any spatial exploratory technique.
b) In general, there are more assumptions made about spatial data than about non-spatial data and thus more diagnostic plots are required.With regard to outliers, it is not only necessary to check for unusually large or small points, but also for those points and sub-areas that are unusual with respect to other points or areas near them. Interactive techniques are very good at spotting complex types of outliers.
c) Interactive graphical tools can spot features that are not as obvious from static plots. A shift in level halfway across a field can easily be mistaken for a trend, whereas interactive tools designed for spatial data have a good chance of picking up the level shift. A good example of this can be seen in Bradley (1991).
d) It may be very hard to decide on a useful model for a particular data set. Interactive graphics help by giving a qualitative view of the data. They suggest relationships and lines of approach and can be the most important part of the analysis for some data sets, especially when the object of the analysis is to answer a qualitative question.
e) Domain experts can understand interactive graphics displays more clearly than they can analyses based entirely on classical methodology and their input is extremely valuable to the analyst.

The key to good exploratory tools for spatial data is in how the linking to the map is achieved. This varies depending on the form of data being analyzed. Figure 12 shows an example of how a map of regions may be linked to other views. The user has selected a section of a histogram corresponding to areas with high average incomes, which is instantly reflected in the map by showing those areas in a different color or with different thickness boundary lines. Experience has shown that dragging the pointer over the histogram to select high values, then medium, then low is an excellent way of aiding the user

understand the trend of the data before attempting to model it. Wills et al. (1991) and Haslett et al. (1990) give good examples of this technique in use.

FIGURE 12. Districts of the city of Dublin showing areas with high levels of average income.

Another important spatial display is of network traffic within or between regions. In Becker et. al. (1991) telephone traffic data within the US is analyzed. By dragging the time indicator to show the traffic for a certain time, and by modifying the amount of data being shown (for example, showing only the high traffic densities), it is easy to build up a clear picture of how people communicate.

4.2 Software visualization

One of the biggest challenges in computing is that of software productivity. Nowhere is this more important than in large multi-programmer, multi-year projects, where hundreds or thousands of programmers work together on huge systems. A big problem in these systems is understanding the source code, its history, and how it all fits together.

This is particularly difficult when the programmers who maintain the code are different from those who wrote it, perhaps due to staff turnover or new project assignments. Programmers, given requests for additional functionality, must study the current code to determine which files contain the existing functionality and which lines to change within these files. This task is often difficult and time consuming.

In the area of software visualization, interactive graphics techniques are useful to understand:

a) The code itself. Even a medium sized computer program comprises between 10,000 and 100,000 lines of source code. For large projects it becomes almost impossible to understand the programs without better visualizations than simple code listings. From a management perspective, other questions about who wrote which parts of the code, who modified it, for what purpose and when assume importance. To manage software production properly, you need tools for understanding the end product.
b) The data structures. Especially with programs that use databases, it is important to be able to visualize the structure of the data as well as the data themselves. Indeed, you often must learn about the structure *before* you can learn about the data. Within an object-oriented framework, both this concern and (a) above are tightly wedded together, and ‘object browsers’ are an important tool in this domain

c) The code execution. Code profiling is an important tool for examining the performance of code, but the simple method of looking at the most executed spots is infeasible for large programs and will only find ‘hot-spots’ of poor performance, rather than entire sections with below-average performance. A way of visualizing the entire performance is required. Also, for debugging, examining the execution paths is necessary to see exactly how failures occur. In real-time systems and for parallel programs, these problems become very difficult and visualizing becomes necessary rather than merely useful.
d) The processes. There are many software engineering models for how software is specified, built, debugged and amended, and it is important that these models are clearly examined for how correct and applicable they are.

In Figure 13 we show the Seesoft program (described in detail by Eick et al., 1992) being used to satisfy objective (c) above. In this view each column represents a file of data, with each line in the file being drawn with a grey line, the darkness of which indicates the frequency with which each line is called. The user can interact with the view to show only lines with given ranges of frequencies, and clicking on an area in a file causes a linked text window to show the same region of the same file (not shown in the figure). the lines are colored similarly in both views to reinforce the linking. With this system it is obvious which areas of which files are problematic and the linking allows you to refer back to the code itself in a natural way.

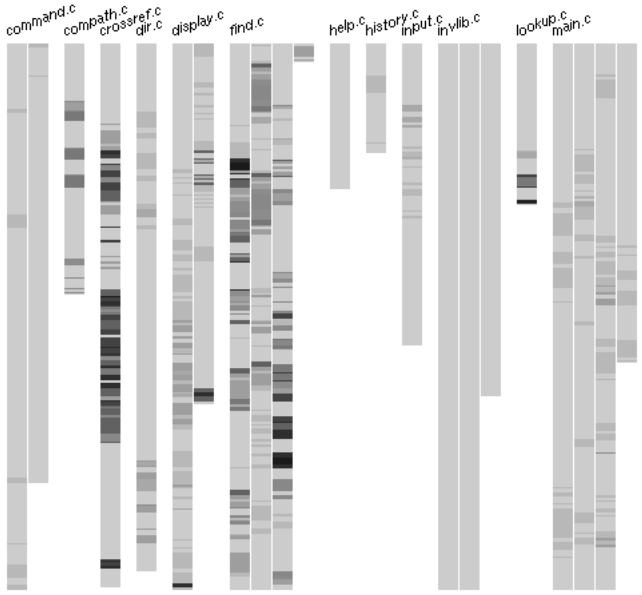

FIGURE 13. A seesoft display showing the frequency with which lines of code within a program are executed. Dark lines are executed more often than lighter ones.

5.0 Conclusions

There are a range of statistical data analysis programs around, some of which emphasize interactive graphics more than others. Most established statistical environments were implemented when interactive graphics would not have been possible on the machines on

which they were designed to run. Batch jobs rather than interactivity were the focus of these early programs. For these older systems, implementing interactive techniques has been a process of adding statistical plots to the pre-existing set of possible displays. Typically available are scatterplot matrix brushing and rotating 3-dimensional point clouds. These systems have the advantage of being complete, tried-and-tested programs, with a wide range of statistical functionality.

Data analysis programs of a more recent origin have made interactivity a core part of their functionality. The designers had the advantage of knowing that their programs would be running on interactive workstations and personal computers, allowing them to assume, for example, that the user will be familiar with a mouse. The main drawback to these environments is that they tend to have fewer traditional statistical tools than the older packages, often focusing on exploratory analysis to the detriment of confirmatory analysis. Fortunately both software designers and computer manufacturers are becoming more comfortable with the idea of data being passed around from program to program and machine to machine, allowing the best tools to be used throughout the process.

Interactive graphics elements are frequently seen in domain specific tools. The linking paradigm is so flexible and intuitive that it is almost always possible to improve the users ability to understand data by adding it to a display system. In emerging fields especially, the need to explore and understand data and processes are vital, and the effort of embodying interactive graphics is small in proportion to its rewards.

Easily interpretable, color-coded displays allow us to show the data clearly and robustly, while allowing deeper exploration via manipulation and information hiding. Linking gives us the ability to relate different displays and so synthesize a more thorough understanding of the data. For these reasons, we believe that interactive graphics are important both within specific domains and in general, and that interactive graphics techniques should be employed by anyone trying to transform data to information.

6.0 References

Becker, R.A., Chambers, J.M. and Wilks, A.R. (1988). *The New S Language.* Wadsworth And Brooks, California.

Becker, R.A., Chambers, J.M. and Wilks, A.R. (1991) “Dynamic Graphics As A Diagnostic Monitor For S” . *ASA 1991 Proceedings Of The Section On Statistical Graphics.*

Becker, R.A., Cleveland, W.S. and Wilks, A.R. (1987) “Dynamic Graphics For Data Analysis”. *Statistical Science 2 Pp 355-395.*

Becker, R.A., Eick. S.G. and Wilks, A.R. (1991) “Basics of Network Visualization”. *IEEE Computer Graphics and Applications,* Vol. 11 Pp 12-14.

Bertin, J.B. (1973). *Semiologie Graphique.* Paris.

Bradley, R.R. (1992). *Exploratory And Diagnostic Methods For The Analysis Of Spatially Referenced Data.* Doctoral Thesis, Trinity College Dublin.

Buckland, S.T., Bell, M.V., and Picozzi, N. (1990). The Birds Of North-East Scotland North-East Scotland Bird Club, Aberdeen

Chambers, J.M., Cleveland, W.S., Kleiner, B., and Tukey, P.A. (1983). *Graphical Methods For Data Analysis.* Wadsworth International, California.

Cleveland, W.S. (1985). *The Elements Of Graphing Data.* Wadsworth, California

Cleveland, W.S. and McGill, R. (1984) "The Many Faces Of A Scatterplot". *Journal Of The American Statistical Association, 79 Pp 807-822.*

Cleveland, W.S. and McGill, R. (1984) "Graphical Perception: Theory, Experimentation, And Application To The Development Of Graphical Methods". *Journal Of The American Statistical Association, 79 Pp 531-554.*

Cleveland, W.S. and McGill, R., eds. (1988). *Dynamic Graphics For Statistics.* Wadsworth & Brooks, California

Cleveland (1993). *Visualizing Data.* Hobart Press, Summit, New Jersey.

Cressie, N. (1984) "Towards Resistant Geostatistics" . *Geostatistics For Natural Resources Characterisation, Part 1.* Eds Verly, G., David, M., Journel, A.g. and Marechal, A.; Reidel, Dordrecht; Pp 21-44

Cressie, N. (1991). *Statistics For Spatial Data.* Wiley, New York

Eick, S.G., Steffen, J.L. and Sumner, E.E. Jr. (1992) "Seesoft - A Tool For Visualizing Line Oriented Software Statistics". *IEEE Transactions On Software Engineering Vol. 18(11) Pp. 957-968* .

Fisher, R.A. (1971). *The Design of Experiments.* Hafner, New York.

Fisherkeller, M.A., Friedman, J.H., and Tukey, J.W. (1974) "Prim-9: An Interactive Multidimensional Data Display And Analysis System". *Slac-Pub-1408.* Slac Publications Office, Stanford, California

Haslett, J., Wills, G. and Unwin, A. (1990)"Spider - An Interactive Statistical Tool For The Analysis Of Spatial Data". *Int.J. Geographical Information Systems 4 No. 3, Pp 285-296.*

Haslett, J.; Bradley, R.; Craig, P.S; Wills, G.; and Unwin, A.R. (1991) "Dynamic Graphics For Exploring Spatial Data, With Application To Locating Global And Local Anomalies". *American Statistician45 No. 3 Pp 234-242.*

Hoaglin, D.C. and Velleman, P.F. (1991) "A Critical Look At Some Analyses Of Major Baseball Salaries". *Asa 1991 Proceedings Of The Section On Statistical Graphics.*

Keller, P. and Keller, M. *Visual Cues.* IEEE Computer Society Press, California.

Sibley, D (1988). *Spatial Applications Of Exploratory Data Analysis.* Geo Books

Silverman, B.W. (1986). *Density Estimation.* Chapman and Hall, New York.

Stevens, S.S. (1975). *Psychophysics.* Wiley & Sons, New York.

Swayne, D.F., Cook, D. and Buja, A. (1991) "Xgobi: Interactive Graphics In The X Window System With A Link To S". *American Statistical Association 1991 Proceedings Of The Section On Statistical Graphics.* Asa, Virginia.

Stuetzle, W. (1987) "Plot Windows". *Journal of the American Statistical Association 82, Pp.466-475.*

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

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

Tukey, J. and Tukey, P. (1990) "Strips Displaying Empirical Distributions: I Textured Dot Strips". *Bellcore Technical Memorandum.*

Tukey, J.W. (1977). *Exploratory Data Analysis.* Addison-Wesley, Reading, Massachusetts.

Velleman, P.F. (1988). *The Datadesk Handbook.* Odesta Corporation

Wills, G., Unwin, A. and Haslett, J. (1991) "Spatial Interactive Graphics Applied To Irish Socioeconomic Data". . *American Statistical Association 1991 Proceedings Of The Section On Statistical Graphics.* Asa, Virginia.