Visual Statistics: Materials for a short course

Pedro M. Valero Mora

Eustat
EUSKAL ESTATISTIKA ERAKUNDEA
INSTITUTO VASCO DE ESTADÍSTICA

51

Materials for a short course on Visual Statistics

Pedro M. Valero Mora

Universitat de València

8 March 2010

Lanketa / Elaboración:

Euskal Estatistika Erakundea
Instituto Vasco de Estadística (EUSTAT)

Argitalpena / Edición:

Euskal Estatistika Erakundea
Instituto Vasco de Estadística
Donostia – San Sebastián, 1 – 01010 Vitoria – Gasteiz

Euskal AEko Administrazioa
Administración de la C.A. de Euskadi

Ale-kopurua / Tirada:
500 **ale** / ejemplares

IV-2010

Inprimaketa eta Koadernaketa:
Impresión y Encuadernacion:
Estudios Gráficos ZURE S.A.
Ctra. Lutxana-Asua, 24 A
Erandio-Goikoa (BIZKAIA)

I.S.B.N.: 978-84-7749-461-4
Lege-gordailua / Depósito Legal: BI-827-10

AURKEZPENA

Nazioarteko Estatistika Mintegia antolatzean, hainbat helburu bete nahi ditu EUSTAT-Euskal Estatistika Erakundeak:

– Unibertsitatearekiko eta, batez ere, Estatistika-Sailekiko lankidetza bultzatzea.
– Funtzionarioen, irakasleen, ikasleen eta estatistikaren alorrean interesatuta egon daitezkeen guztien lanbide-hobekuntza erraztea.
– Estatistika alorrean mundu mailan abangoardian dauden irakasle eta ikertzaile ospetsuak Euskadira ekartzea, horrek eragin ona izango baitu, zuzeneko harremanei eta esperientziak ezagutzeari dagokienez.

Jarduera osagarri gisa, eta interesatuta egon litezkeen ahalik eta pertsona eta erakunde gehienetara iristearren, ikastaro horietako txostenak argitaratzea erabaki dugu, beti ere txostengilearen jatorrizko hizkuntza errespetatuz; horrela, gai horri buruzko ezagutza gure herrian zabaltzen laguntzeko.

Vitoria-Gasteiz, 2010eko Martxo

JAVIER FORCADA SAINZ
EUSTATeko Zuzendari Nagusia

PRESENTATION

In promoting the International Statistical Seminars, EUSTAT-The Basque Statistics Institute wishes to achieve several aims:

– Encourage the collaboration with the universities, especially with their statistical departments.
– Facilitate the professional recycling of civil servants, university teachers, students and whoever else may be interested in the statistical field.
– Bring to the Basque Country illustrious professors and investigators in the vanguard of statistical subjects, on a worldwide level, with the subsequent positive effect of encouraging direct relationships and sharing knowledge of experiences.

As a complementary activity and in order to reach as many interested people and institutions as possible, it has been decided to publish the papers of these courses, always respecting the original language of the author, to contribute in this way towards the growth of knowledge concerning this subject in our country.

Vitoria-Gasteiz, March 2010

JAVIER FORCADA SAINZ
General Director of EUSTAT

PRESENTACION

Al promover los Seminarios Internacionales de Estadística, el EUSTAT-Instituto Vasco de Estadística pretende cubrir varios objetivos:

– Fomentar la colaboración con la Universidad y en especial con los Departamentos de Estadística.
– Facilitar el reciclaje profesional de funcionarios, profesores, alumnos y cuantos puedan estar interesados en el campo estadístico.
– Traer a Euskadi a ilustres profesores e investigadores de vanguardia en materia estadística, a nivel mundial, con el consiguiente efecto positivo en cuanto a la relación directa y conocimiento de experiencias.

Como actuación complementaria y para llegar al mayor número posible de personas e Instituciones interesadas, se ha decidido publicar las ponencias de estos cursos, respetando en todo caso la lengua original del ponente, para contribuir así a acrecentar el conocimiento sobre esta materia en nuestro País.

Vitoria-Gasteiz, Marzo 2010

JAVIER FORCADA SAINZ
Director General de EUSTAT

BIOGRAFI OHARRAK

PEDRO M. VALERO-MORA Universitat de València-ko irakasle titularra da, datuen prozesuan. Pertsona-ordenagailuaren arteko elkarreragitean eta estatistikan lan egin du, eta horrek eraman zuen sistema estatistikoetarako ordenagailu-interfazeez arduratzera; mesedegarri, emankor eta interesgarriak nahi zituen.

Azken urteetan interes horrek Forrest W. Young-en ViSta programaren garapenean parte hartzera eraman du.

BIOGRAPHICAL SKETCH

PEDRO M. VALERO-MORA is Professor of Data Processing at Universitat de València. His areas of interest are Person-Computer Interaction changes and Statistics, which led to his designing computer interfaces for statistical systems that were useful, productive and interesting.

In recent years, he went on to work with Forrest W. Young to develop the ViSta program.

NOTAS BIOGRÁFICAS

PEDRO M. VALERO-MORA es Profesor Titular de Proceso de Datos en la Universitat de València. Ha trabajado en los campos de Interacción Persona-Ordenador, y Estadística, lo que le llevó a interesarse en cómo diseñar interfaces de ordenador para sistemas estadísticos que fueran útiles, productivos e interesantes.

En los últimos años, ese interés le ha llevado a colaborar en el desarrollo del programa ViSta de Forrest W. Young.

To Forrest W. Young (1940-2006)

Exploratory data analysis is an attitude, a flexibility, and a reliance on display, NOT a bundle of techniques, and should be so taught.

John W. Tukey,
We need both exploratory and confirmatory,
The American Statistician, 34(1), (Feb., 1980), pp. 23-25.

Contents

Chapter 1 Introduction and History

1.1 Introduction

The best way to introduce the idea of interactive graphics is probably via an example. In this section we show dynamic interactive graphics being used to lay the groundwork needed to develop a medical diagnostic tool.

We use a particular set of data to show how a medical diagnostic tool can be developed using the techniques of visual statistics. The data, which are described in detail below, were obtained from patients examined by a dermatologist about problems with their skin. The data include information obtained during an office visit and the results of a laboratory analysis of a biopsy taken from a patient.

Data. The data are observations of 34 variables obtained from 366 dermatology patients. Twelve of the variables were measured during the office visit, 22 were measured in laboratory tests performed on a skin biopsy obtained during the office visit. Of the 34 variables, 32 were measured on a scale running from 0 to 3, with 0 indicating absence of the feature and 3 the largest amount of it. Of the remaining two variables, *Family history* is binary and *Age* is an integer specifying the age in years. Eight of the patients failed to provide their age. These patients have been removed from the analysis. The data were collected by Nilsel Ilter of the University of Ankara, Turkey (Guvenir et al., 1998).

Two difficulties must be dealt with before we can visualize these data: (1) The data are discrete—All of the variables except *Age* have four or fewer observation categories; and (2) There are too many variables—humans can not understand 34 variables simultaneously.

If we picture the data directly, we soon see the problems. For example, a matrix of scatterplots of the first three variables is shown in Figure 1.1. A scatterplot matrix has variable names on the diagonal and scatterplots off-diagonal. The scatterplots are formed from the variables named on the diagonal of a plot's row and column. The upper-left triangle of the matrix is a mirror image of the lower right.

The discrete nature of the variables means that we have only a few points showing for each scatterplot, and that they are arranged in a lattice pattern that cannot be interpreted. For each plot, each visible point actually represents many observations, since the discrete data make the points overlap each other. In essence, the resolution of the data, which is four values per variable, is too low. The fact that there are 34 variables means that the complete version of Figure 1.1 would be a 34×34 matrix of scatterplots, clearly an impossibly large number of plots for people to visualize. Here, the problem is that the dimensionality of the data, which is 34, is way too high. Thus, the data cannot be visualized as they are, because their resolution is too low and their dimensionality is too high.

Principal components. All is not lost! These two problems can be solved by using principal components analysis (PCA). PCA reduces the dimensionality of data that have a large number of interrelated variables, while retaining as much of the data's original information as is possible. This is achieved by

transforming to a new set of variables, the *principal components*, which are uncorrelated linear combinations of the variables. There is no other set of *r* orthogonal linear combinations that fits more variation than is fit by the first *r* principal components (Jollife, 2002).

Principal components have two important advantages for us. First, since only a few components account for most of the information in the original data, we only need to interpret displays based on a few components. Second, the components are continuous, even when the variables are discrete, so overlapping points are no longer a problem.

Figure 1.2 shows a scatterplot matrix of the five largest principal components. These components account for 63% of the variance in the original 34 variables. The general appearance of Figure 1.2 suggests that the observations can be grouped in a number of clusters. However, the total number of clusters as well as their interrelationships are not easily discerned in this figure because of the limited capabilities of scatterplot matrices and because of the small plot sizes.

Linking. Linking is a powerful dynamic interactive graphics technique that can help us better understand high-dimensional data. This technique works in the following way: When several plots are linked, *selecting* an observation's point in a plot will do more than highlight the observation in the plot we are interacting with—it will also highlight points in other plots with which it is linked, giving us a more complete idea of its value across all the variables. Selecting is done interactively with a pointing device. The point selected, and corresponding points in the other linked plots, are highlighted simultaneously. Thus, we can select a cluster of points in one plot and see if it corresponds to a cluster in any other plot, enabling us to investigate the high-dimensional shape and density of the cluster of points, and permitting us to investigate the structure of the disease space.

Interpretation. Figure 1.3 displays a "comic book"-style account of the process of selecting the groups of skin diseases that were visible in Figure 1.2. Frames of Figure 1.3 are the scatterplots of PC1 versus PC2 to PC5. The frames are to be examined sequentially from left to right and from top to bottom. The last frame is a repetition of the first frame and is intended to show the final result of the different actions carried out on the plots. An explanation of the frames follows.

(1) PC1 vs. PC2: This scatterplot shows quite clearly three groups of points, labeled A, B, and C. Two groups are selected, but the third one remains unselected.

(2) PC1 vs. PC3: Three actions are displayed in this frame. The groups selected in the preceding frame have been marked with symbols: A has received a diamond (‡), and B a cross (+). Also, we can see that dimension PC3 separates group C into two parts: one compact, the other long. We have selected the observations in the long part and called them cluster C_1. At this point we have four clusters: A, B, C_1, and unnamed.

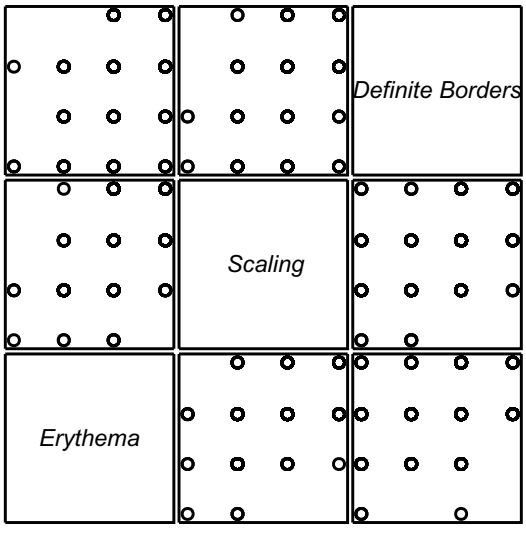

Figure 1.1 Scatterplot matrix for three variables.

(3) PC1 vs. PC4: This plot has the points selected in the preceding frame represented by a square symbol (□). Unassigned observations above and below of squares make two groups. We selected the group with positive values in PC4 and called it C_2, giving us five clusters: A, B, C_1, C_2, and unnamed.

(4) PC1 vs. PC5: We assigned the symbol (×) to the group C_2 and selected the top values of PC5. Notice that this selection involved the reclassification of some points that had previously been assigned to the group C_2. The points selected will define group C_3. Notice that there are still some points that keep the original symbol of the points in the plot [a disk (○)]. We call it group C_4. We now have six clusters: A, B, C_1, C_2, C_3 and C_4.

(5) PC1 vs PC2 again: This frame is the same as the first frame in the sequence except that it shows the plot after steps 1 to 4. Note that we ran out of easily seen symbols for marking C_3, so we used gray to identify points. This frame displays very clearly the three big clusters identified at the beginning and also suggests something of the finer structure inside cluster C.

A downside of the plots in Figure 1.3 is that the four clusters identified as C_1 to C_4 are not visualized very clearly. This problem suggests using *focusing,* a technique described in Chapter 4, to remove from the plot the points in the two largest clusters. As the remaining four subclusters are defined basically using PC3 to PC5, it makes sense to use a 3D plot to visualize them. Figure 1.4 displays such a plot after rotating it manually to find a projection clearly displaying the four clusters. This view suggests that some points actually seem to belong to clusters other than those to which they had previously been assigned. Thus, we reassign them, as explained in the balloons.

Validity. We did not mention it earlier, but the data include a diagnosis made by the doctor of each patient. It is interesting to compare our visual classification with the diagnostic classification. Table 1.1

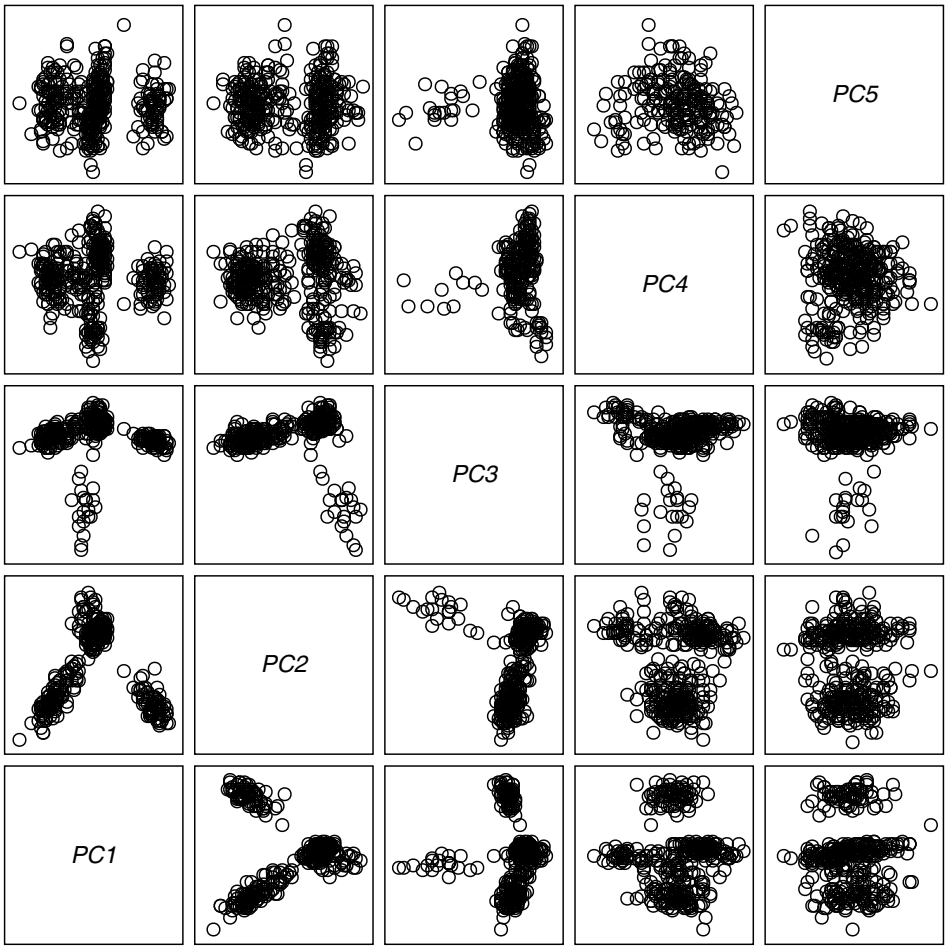

Figure 1.2 Scatterplot matrix for the first five principal components.

presents a confusion matrix, which is a table that shows the frequency with which members of each of our classes were assigned to each of the diagnostic classes. If our visual classification agrees exactly with the diagnostic classification, the confusion matrix will be diagonal, there being zeros in all off-diagonal cells. To the extent that our visual classification is "confused," there will be nonzero frequencies off the diagonal.

In general, we see that our visual classes correspond closely to the diagnostic classes. All of the patients diagnosed with psoriasis and lichen planus were classified visually into groups A and B. The observations in the cluster labelled C are very well separated with respect to clusters A and B, with only one observation out of place. However, some of the subclusters in C, especially C2 and C3 have considerable interchanges between them, suggestion that additional efforts for improving the discrimination between them are nec-

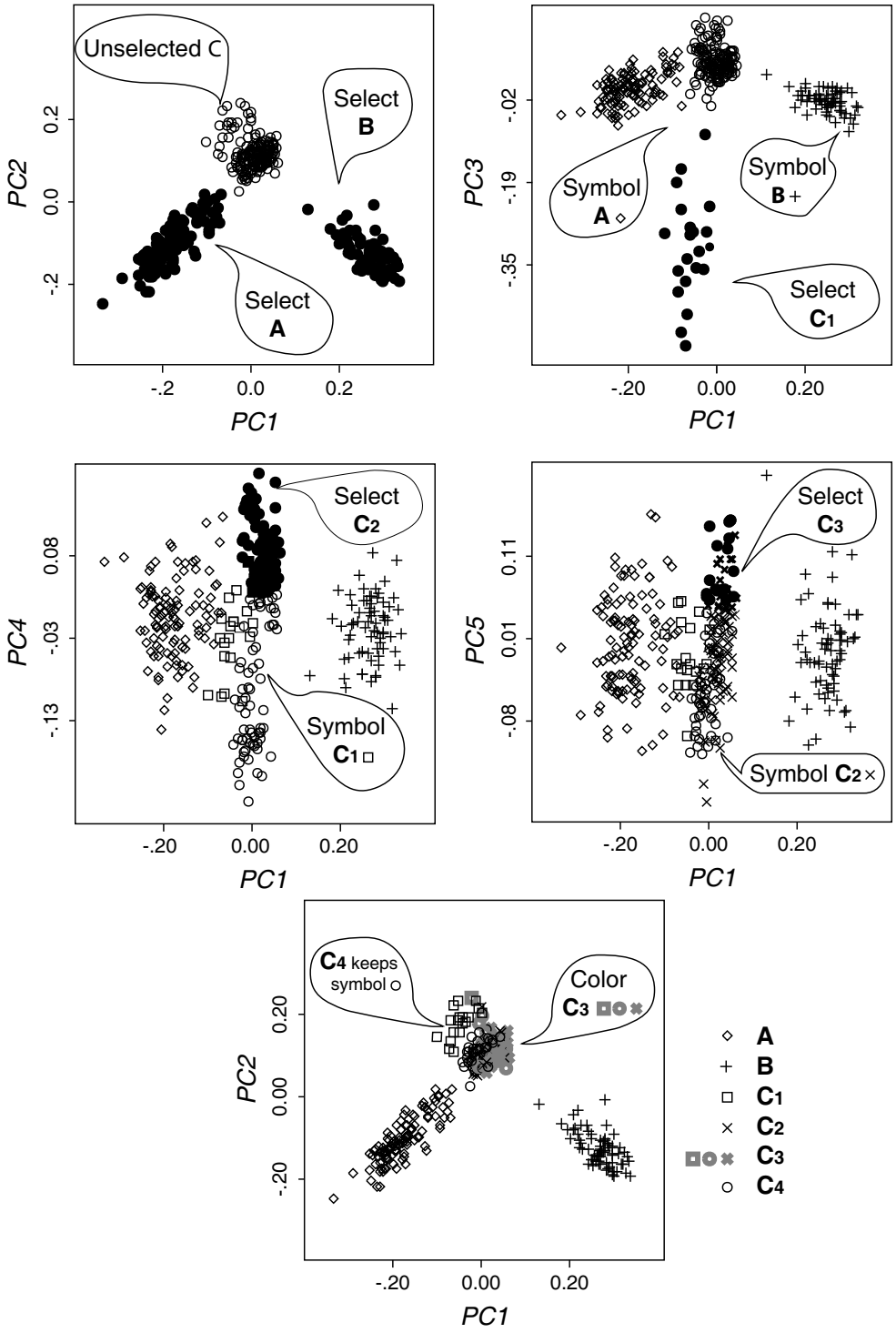

Figure 1.3 Steps in selecting and changing symbols and colors in PCs.

Table 1.1 Confusion Matrix for the Medical Diagnosis Data

	A	**B**	**C_1**	**C_2**	**C_3**	**C_4**	S
Psoriasis	111						**111**
Lichen planus		71					**71**
Pityriasis rubra pilaris			19		1		**20**
Pityriasis rosae				39	8	1	**48**
Seborrheic dermatitis	1			10	48	1	**60**
Chronic dermatitis						48	**48**
	112	**71**	**19**	**49**	**57**	**50**	**358**

essary. This problem can result from the difficulties we had to correctly pinpoint the points in frame 4 of Figure 1.3. Of course, there could also be misdiagnoses by the doctor—we can't tell. Nevertheless, 93.8% of the cases are classified correctly which is only 2.4% lower than the classification made originally using a special algorithm called VFI (Guvenir et al., 1998).

1.2 Software for interactive data analysis

At the time this is being written, there are six statistical visualization systems that seem to us to be the most important: JMP, DataDesk (Velleman and Velleman, 1985), Arc (Cook and Weisberg, 1994), ViSta (Young, 1994; Young and Smith, 1991), GGobi (Swayne et al., 1998, 2003) and Manet (Hofman, 2003; Unwin et al., 1996). Each of these systems uses dynamic interactive graphics to simplify data analysis and to display data structure. Of these, two are commercial systems and four are noncommercial.

The six statistical visualization systems have evolved from research and development in dynamic interactive statistical graphics that began around 1960, with the first commercial systems becoming available in the late 1980s. Some of the major milestones of progress in visual statistics are:

- A direct manipulation system for controlling a power transformation in real time (Fowlkes, 1971), where we first come across the idea that one could connect a parameter and a statistical method manually with an on-screen controller.

- Prim-9, the first system with 3D data rotations (Fisherkeller et al., 1975), demonstrating dynamic interactive graphics for the first time. Other early systems with these capabilities are ORION (McDonald, 1988) and MacSpin (Donoho et al., 1988).

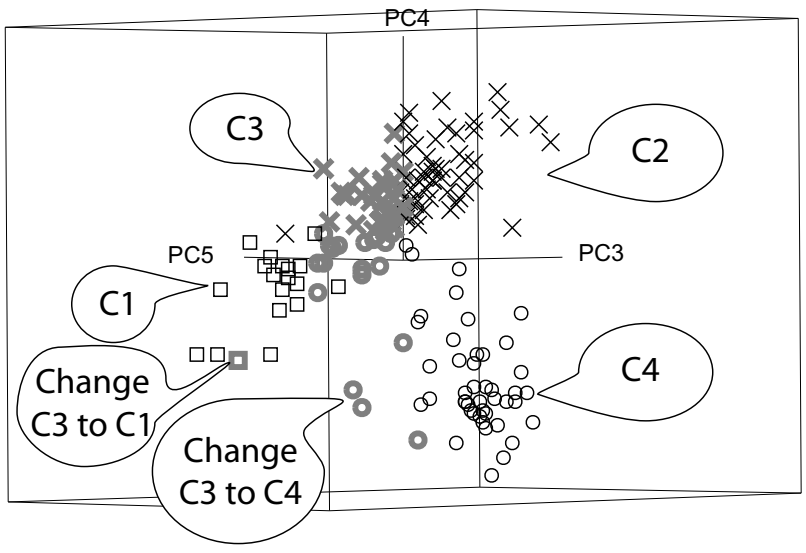

Figure 1.4 Spinplot showing observations in clusters C_1 to C_4

- The dynamic interactive technique known as *brushing* was developed around 1980. This technique allowed the analyst to select regions or points and to see them linked simultaneously in other plots (McDonald, 1982).

- The Grand Tour for visualizing multivariate data sets by creating an animation of data projections by moving a two-dimensional projection plane through *n*-space (Asimov, 1985). The novel idea here was to help the analyst seek informative views of high-dimensional data by defining various criteria of "interestingness."

- A systematic implementation of interactive statistical graphics, including brushing, linking, and other forms of interaction (Becker and Cleveland, 1988).

1.2.1 XLisp-Stat

XLisp-Stat (Tierney, 1988, 1990), has had considerable impact on the development of statistical visualization systems. XLisp-Stat is not, per se, a statistical system, but is a statistically and graphically oriented programming environment designed to facilitate the creation of statistical systems. It is based on a freely available implementation of the Lisp language called XLisp (Betz, 1985) and in ideas borrowed form S.

Tierney (1988, p.4) gave three primary motivations for developing this language: (1) provide a vehicle for experimenting with dynamic graphics and for using dynamic graphics in instruction; (2) explore the use of object-oriented programming ideas for building and analyzing statistical models; and 3) experiment with an environment supporting functional data.

XLisp-Stat provided statisticians with the opportunity to implement ideas related to dynamic graphics in much the way that statistical language *S* had already provided a general statistical programming language (Becker, 1994; Becker and Chambers, 1981; Becker et al., 1988a; Ihaka and Gentleman, 1996). In fact, some of the features of XLisp-Stat were inspired by similar features or functions in *S*. One of the strong points of XLisp-Stat, Tierney and others argued, was the fact that it was based on Lisp, a general-purpose high (and low)-level programming language that was (and is) well known and mature. This guaranteed a solid foundation of technical aspects and a strong set of basic programming tools.

When XLisp-Stat is started by the user, all that is shown to the user is an empty screen with a prompt for typing commands. Using lisp syntax, commands for opening data files and obtaining statistical summaries and plots can be obtained. Also, new functions can easily be written and saved in files, so repeating sessions of analysis or expanding the system could be accomplished.

XLisp-Stat also provides tools for incorporating user interface elements in programs developed by users. In this way, Lisp programs could be turned into direct manipulation systems (Hutchins et al., 1986), the kind of user interaction system popularized by the Macintosh and by Microsoft Windows. These capabilities paved the way to some projects written entirely in XLisp-Stat that provided a user interface for manipulating data, computing results and showing visualizations. These projects are described by their authors in Stine and Fox, 1996.

Note that there is some concern that XLisp-Stat is not a healthy, growing statistical system. Indeed, there is a fairly widespread opinion held by many computation statisticians that it has died, an opinion not shared by the current authors. We have recently published articles about the problems that currently exist (Valero-Mora and Udina, 2005; Molina et al., 2005), as have others intimately involved in the development, popularization, and then demise of the system (de Leeuw, 2005; Tierney, 2005). Perhaps the most important point is made by Weisberg (2005) who states that "solutions to applied statistical problems are framed by the limitations imposed by statistical computing packages and languages." Weisberg goes on to point out that the kinds of solutions supported by XLisp-Stat are very different from those supported by other languages, and that a variety of statistical languages can only benefit, not hurt, the field of statistics.

1.2.2 Commercial Systems

The two commercial systems are DataDesk and JMP, which now have a long history that has made them very solid and complete products that can be used for many everyday computing needs of data analysts as well as for visual statistics.

DataDesk. The DataDesk system (Velleman and Velleman, 1985) included from the very beginning, rotating plots, linking and brushing, and dynamic transformations, among other features. It also provided a visual metaphor for managing datasets, variables, analyses, and so on, and for representing the analysis process.

JMP. (JMP, 2002), which is pronounced "jump," first became available in 1989, providing many of the dynamic interactive features introduced previously in the literature. It did not provide as extensive an environment for data analysis as that provided by DataDesk, and there was no vivid metaphor for the data analysis process itself. Yet JMP provided a system in which every analysis was accompanied automatically by graphs, without having to beg them to show themselves.

1.2.3 Noncommercial Systems

Of the four noncommercial projects, ViSta and Arc were developed using XLisp-Stat. They share the potential of being extended to incorporate new methods or techniques. Also, since XLisp-Stat is available behind the scenes, even though each system is based on a direct manipulation interface, command typing is also available.

Arc. Arc (originally named R-code, but was renamed to avoid confusions with the R software system) (Cook and Weisberg, 1994, 1999) is a regression analysis system that emphasizes regression graphics, many of which are dynamic. *Arc* is one of the reference programs for regression analysis.

ViSta. Vista, the Visual Statistics System (Young and Smith, 1991; Young, 1994), was also developed using the tools provided by XLisp-Stat. Since we use ViSta throughout this book, we describe it more extensively in the next section.

Manet. Manet is one of several statistical visualization programs developed by the computer-oriented statistics and data analysis group of the University of Augsburg (Unwin et al., 1996). Manet originally focused on visual estimation of missing values. It now incorporates many other visualization techniques and is particularly outstanding for visualizing categorical data (Hofman, 2003).

XGobi and GGobi. XGobi, and its recent sibling GGobi, are data visualization systems for exploring high-dimensional data that have graphical views that can be brushed and linked (Swayne et al., 1998). These views include dynamic interactive scatterplots, parallel coordinate plots, and scatterplot matrices. A new system, RGobi, incorporates this features in R

Finally, as the basic ideas of dynamic statistical visualization have become more mainstream, many commercial statistical systems have tried to incorporate them, with, in our opinion, limited success. Programs such as SAS, SPSS, S-Plus, Systat, Statistica, and Minitab have incorporated dynamic interactive graphics techniques, although the complexities involved in integrating these techniques into their parent systems are often problematic and limiting. Also, there have been some attempts to include dynamic plots in R, the free implementation of S (for example iPlots-http://rosuda.org/iplots/).

1.2.4 ViSta

We will use ViSta (Young, 1994) as the primary system to demonstrate the concepts being described. Originally a research project concerning the use of dynamic interactive graphics, ViSta has grown up along the years to cover a large number of statistical techniques. Whereas JMP and DataDesk are both commercial systems with a closed software distribution model, ViSta is a noncommercial, freely available system using a moderated, partially open software distribution model.

ViSta contains the full range of dynamic interactive graphics tools available in other statistical visualization systems. It also includes a visual metaphor for structuring data analysis sessions, a graphical tool for providing the data analyst with expert guidance, and an approach to organizing and coordinating multiple dynamic graphics.

The reason that we have chosen to focus on ViSta is very simple: The first author of this book is the designer and original developer of ViSta, and the second author has made major contributions to the system. As such, we are very familiar with ViSta. We have been using it for visual statistical analysis, for the

development of new visualization methods, and for the development of the software architectures underlying such methods. Together, the authors of this book have about 25 years of first hand knowledge in the theoretical basis of highly interactive, very dynamic statistical graphics, as well as literally decades of 20-hour days spent in the trenches doing the immensely difficult and painstakingly demanding software development underlying ViSta's very dynamic, highly interactive graphics.

Although many of the techniques described in this book are well-known methods for data visualization, we have provided new solutions to some of the problems of data visualization that we believe are of fundamental importance. Our implementations of many of the dynamic statistical graphics tools (described later in the book) would have been difficult at best, and more often impossible, had we used only software provided by others.

ViSta focuses on a single thing—visual statistics—and attempts to do that thing as well as possible. As such, we know that ViSta does not include all the statistical visualization techniques, and it is not difficult to find implementations of certain methods that are better than ours. However, we believe that ViSta provides the user with an excellent structured environment for statistical visualization, which simplifies and clarifies statistical visualization. In our view, the most important aspects of ViSta are its graphical interface, its dynamic interactive graphics, and its multiview graphics.

ViSta is operated by a direct manipulation interface that includes menus, dialog boxes, and icons. This type of interface is appropriate for first-time and occasional users because they can use point and click strategies to control the program. A drawback of direct manipulation is that very often there is no method to record the history of actions of the users. ViSta offers the workmap as a way to solve this problem.

ViSta supports a wide range of dynamic and interactive graphics, including nearly all those mentioned in this book. Essentially all plots support brushing, labeling, and linking. Also, many plots are customized for specific analytical situations.

One of the difficulties of having a large number of plots is organizing the windows on the screen so that they can be contemplated without impediments. ViSta provides spreadplots, discussed above. There is a reasonably simple interface that helps the moderately sophisticated developer write new spreadplots.

ViSta includes a basic set of tools for data management and processing. These allow importing and exporting data to text files, merging files, transforming variables, and more. All together, this set of functions guarantee that many data analysis sessions can proceed in ViSta completely without needing help from other programs.

ViSta supports the advanced user who wishes to enhance or extend its capabilities. This can be done by writing plug-ins, applets, or special-purpose programs. Plug-ins are programs that implement major data analysis and visualization techniques and which are interfaced with ViSta so that they become part of its standard capabilities. *Applets* are small bits of code that are served over the Internet to ViSta, which acts as the client in the standard client–server software distribution model. Special-purpose programs may be written on the fly to be used immediately or saved for later use. No restrictions areimposed.

ViSta can be downloaded from http://www.uv.es/visualstats/Book/. The developer who wishes to extend ViSta's data analysis and visualization capabilities can do so as described above. Those who wish to modify the system have access to portions of the code, with restrictions imposed on critical sections. Even these restrictions may be loosened following procedures defined on the web site. This provides the flexibility of an open system with the reliability of a closed system.

The material in this book is based on version 7.6 which has many capabilities not in 6.4. While essentially all of the figures were prepared using ViSta, some represent specialized programming by the authors and cannot be produced with the point-and-click interface.

But this is not a book about ViSta; it is a book about visual statistics. We focus on the ideas and concepts of visual statistics in and of themselves, not specifically as implemented in ViSta.

Chapter 2 Interactive features in Plots

2.1 Introduction

This chapter presents a number of interactive features that can be implemented in statistical software. The presentations is quite software independent but many of them are present in ViSta. However, some of them may not be present in it. However, before getting into how one interacts with the plots to activate their dynamic features, we present a way of conceptualizing the basic nature of the wide variety of plots that are available to the data analyst. This conceptual scheme organizes the plots according to the nature of the object used to represent the data, a characteristic that has some effect on the methods we can use to interact with the plot.

2.2 Plots

The various graphics that have been proposed all have a specific type of object that is used to visually represent the feature of the data that are being graphed. For example, the scatterplot uses point symbols to represent the data's observations.

We refer to the basic "plot thing" that is used by the plot to represent your data, the plot's glyph (a symbol that imparts information nonverbally), a word that has been used elsewhere with essentially the same meaning. We note that glyphs can be organized according to their dimensionality. Note that we are referring here to the dimensionality of the glyph, not the dimensionality of the plot. This leads us to the following way of thinking about plots:

Points. Many plots use points as the plot's glyph. This is a zero-dimensional glyph which, of course, cannot be seen. So we use tiny point-symbols as the glyph representing the point. An example of a plot of this type is the very familiar scatterplot, where each observation is represented conceptually by a point and is represented physically by a tiny point symbol.

Lines. Some plots use lines as the plot's glyph. Lines are one-dimensional objects that are drawn as such on the screen. Examples that we discuss in this book include the parallel-coordinates plot, the run-sequence plot, and the lag plot.

Areas. Still other plots use two-dimensional geometric figures such as rectangles or wedges as the plot's glyph, where the area represents the observed data—pie charts, bar graphs, and mosaic plots are familiar examples of such graphics.

Schematics. A final type of plot are those that use schematics to represent data. Well-known examples

include boxplots and frequency polygons, to mention just two. (Although it may be stretching the definition, we could call these plots nondimensional, since the dimensionality of the schematic is irrelevant.)

The chapter is written as though all plots are point-based. This is, of course, not true. But since the chapter focuses on point-based plots, we should keep in mind the other families of plots as we proceed through the chapter, asking ourselves whether the particular way of interacting with a dynamic graphic depends on the glyph type of plot under consideration.

When we ask ourselves that question, it seems to us that for the most part, you can substitute *line* for *point* throughout the chapter; however, there are very few line-based plots, and many point-based plots. If dynamic interactive time-series graphics were developed further, line-based plots would be more common. The best example of a line-based plot is the *parallel-coordinates plot* (also known as a *profile plot*). In this plot the line is the fundamental glyph, and there is no consideration of points. On the other hand, the run-sequence plot and the lag plot use lines as the plot's glyph, but unlike parallel-coordinate plots, they are connected-point plots, where the glyph is a line that is point-based and therefore acts totally like a point-based plot.

It also seems to us that quite often you cannot substitute *area* for *point* in this chapter. The difficulty is that in an area-based plot, such as a mosaic plot, the area, which is the building block of the plot, represents several observations, whereas for a scatterplot, for example, the point, which is the plot's glyph, represents a single observation.

We discuss a wide variety of different ways of interacting with dynamic plots. These ways include

- **Activating plot objects:** selecting and brushing, the two major ways in which the window's objects can be activated
- **Manipulating plot objects:** labeling, linking, focusing, and morphing, the major actions that can be taken on the objects in essentially all plots
- **Manipulating plot dimensions:** permuting, changing scale, and changing aspect ratio, all of which have to do with the dimensions of the plot.

We take these up in turn in the coming sections.

2.2.1 Activating Plot Objects

The objects in the plot—usually the points—can be either activated or not activated. Activation is usually denoted by using a visual effect, such as changing color, brightness, or shape. The objects activated can then become the focus of whatever future actions the analyst decides to take.

Activation is often a prerequisite for taking other actions, but it can also be an operation of interest in its own right. The important aspect of activation is that the activated objects are made to stand out from the rest of the objects in some way. Then the activated objects can be used for identifying interesting features of the data.

For example, Figure 2.1 shows a group of activated points in a scatterplot. The visual effect of activation is obtained by changing the color or symbol of the activated points with the color of the points. Other techniques for emphasizing activated points are changing color, size, brightness, or reversing the color and white parts of symbols.

There are two ways of activating an object, by selecting it or by brushing it. We discuss these next.

Selecting . Selecting is a method of interacting with dynamic graphics which causes objects in the graphic to become active. Selection is usually accomplished using the mouse. The standard mouse selection techniques are:

- **Individual selection**: accomplished by single-clicking an object on the screen.
- **Multiple selection**: sually, holding down the meta key (shift or control) while objects are single-clicked.
- **Area selection**: dragging the mouse to create an area of the screen for which contained objects are

selected. The area can be rectangular or free-form (using a lasso-like tool).

- **Multiple area selection**: using the meta key and dragging the mouse to select objects contained within several areas.

Brushing. In this technique, an active area with the shape of a rectangle is defined. Moving the mouse causes the active area to be moved, with objects contained within the active area being activated. Activation is denoted using the same type of visual effects used for selections (i.e., changing color, brightness, shape, etc.). Brushing is usually discussed in connection with linking and can be used, for example, to explore a bivariate plot conditioned by a third variable. Brushing is discussed in several sources (Becker and Cleveland, 1987; Becker et al., 1988; Cleveland, 1994b).

While selecting and brushing are quite similar generally in the effect they produce on the elements on the screen, brushing imposes a bigger workload on the computer. For example, in a scatterplot, in order to get a really smooth effect, it is necessary to refresh the plot each time that a new point enters or goes off the rectangle area. Selecting is a discrete process in which refreshing is necessary only after the user releases the button of the mouse. Consequently, slow computer environments will be able to implement selecting much more easily than brushing.

2.2.2 Manipulating Plot Objects

We will discuss the following actions in this section: Labeling, linking, focusing/excluding, and changing colors/symbols/labels of the plots. Notice that these actions are not performed for aesthetic reasons, but for semantic reasons so the analyst can point out iin some way the interesting elements that is observing in the graphical representation of the data.

Labeling. Placing labels on a plot is a complicated problem for static plots. Numerous heuristic (Noma, 1987) and computational (Christensen et al., 1992, 1995) strategies have been used to produce labels that do not overlap. This is a critical issue for static and printed plots when they need to display all the labels (e.g., in maps).

This problem is solved in dynamic graphics by not showing the labels unless the user carries out actions to reveal them. If the user chooses only one label at a time, overlapping cannot happen. However, when the user wants to view simultaneously labels that correspond to objects that are close in a space, it is possible that some of them will be covered by others. Figure 2.2 shows a scatterplot where three points have been selected. The Buick Estate Wagon can be read even though it goes out of the frame of the plot.

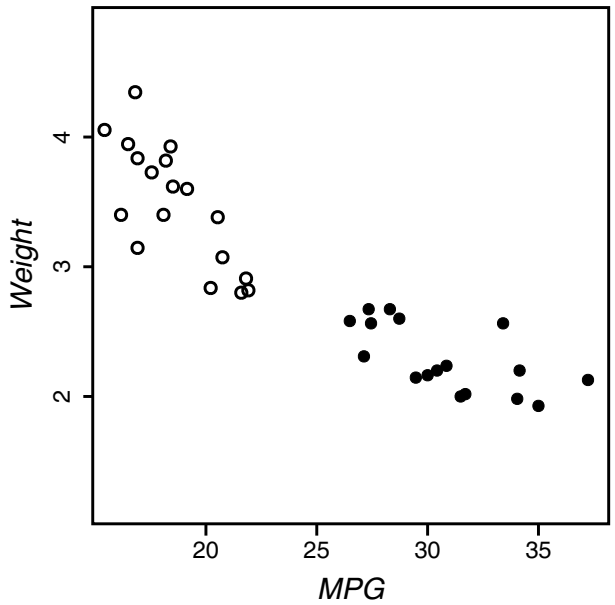

Figure 2.1 Selected points in a scatterplot.

However, the other two are very difficult to read. In this case, the user can reposition the mouse very slightly to try to obtain a different selection, but this can be difficult or impossible in many situations.

Some of the dynamic solutions that have been used for this problem are:

- Small menus or pop-up windows that turn up for overlapping or close points.
- Sequential listing of labels (e.g., each click shows a new label).
- Linking to an external list of labels that shows the observations selected.

The problem remains that if the points are not exactly overlapping and if learning the exact values is of importance, it is still difficult to distinguish one from another. The techniques described in the section on focusing and excluding can be used to enlarge the area of interest and to explore it with less possibility of overlapping labels.

Linking. It is usually the case that a single display cannot show you everything that is interesting about a set of data. Even for the simplest data, we will often see different features of interest in different displays, if only because most graphics are designed to help us explore one (or at most a few) features. Thus, it is usually necessary to examine several different graphics in order to get the full picture of the problem at hand.

The problem is that very quickly there are too many plots to remember, let alone to understand. Linking is one of the important steps in solving this problem. When a plot is linked with other plots, a change in one plot can be made to cause changes in the other plots. The linkage is through the data shared by the plots, with the data's observations being the usual basis for the linkage.

Generally, the units of information that are linked are the observations in the dataset. When observations are linked, then actions carried out on the representation of a observation in one plot are propagated to the representation of the observation in whatever other views are currently displayed.

So, for example, changing the state of a point, its color, symbol, or other aspect in a plot is mirrored in other plots, data listings, or other views. Figure 2.3 shows two scatterplots of four variables for the same observations. When the plots are linked, then if the observation Buick Estate Wagon is selected in the left plot, the same observation is selected automatically in the right plot.

Note that the representations of the observations do not have to be the same in each of the plots. Note also that more than two plots can be linked. Thus, if we linked a histogram (which uses tiles to represent the observations) to the scatterplots in Figure 2.3, then when an observation is selected in any one of the three plots, a tile of the histogram and a point in each scatterplot is highlighted.

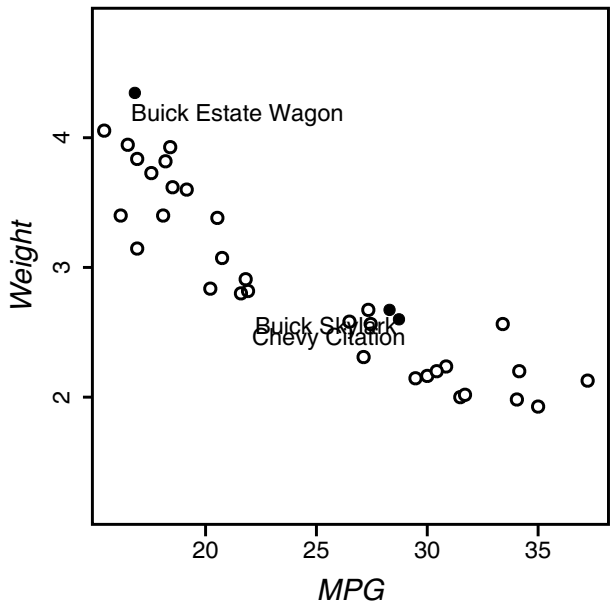

Figure 2.2 Displaying labels in a scatterplot.

Note that linking is a very general concept. Not only can we have multiple plots linked simultaneously, and not only can we have different representations involved in the linkages, but we are also not restricted to a one-to-one relationship. Linkages can be a one-to-many relationship between many plots using many representations. Finally, the linked objects do not have to be observations. They may be variables, or matrices, or whatever is represented by the plot as an object with changeable features.

The main aspect of linked plots is that when a change is made to some aspect of the data as they are represented in one plot, the change can be portrayed in some fashion in additional plots. In a high-interaction, highly dynamic system the changes happen instantaneously in real time. In such a system, the combination of linking and brushing is particularly powerful. With this combination, when one of a set of linked plots is brushed, the appropriate objects are highlighted in the other plots. Linking was first described by Stuetzle (1987) and is implemented in Lisp-Stat (Tierney, 1990), ViSta, DataDesk (Velleman, 1997), Manet (Unwin et al., 1996), and other programs.

Linking is an example of the general problem of coordinating multiple views in a computer system, as discussed by North and Shneiderman (1997), who present a taxonomy of all possible multiple-view coordination techniques.

Focusing and excluding. Figure 2.4 shows an example of focusing on a part of a plot. The cloud of points in the scatterplot on the left looks like it could be split into two separate parts. To look into this, the points with lower *MPG* are selected and a regression line is calculated and shown. The line has a very dif-

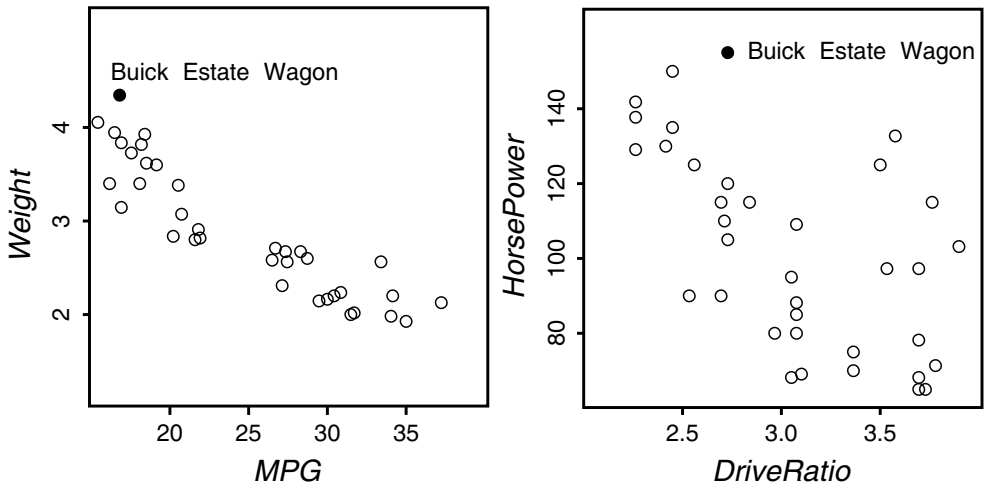

Figure 2.3 Two linked scatterplots.

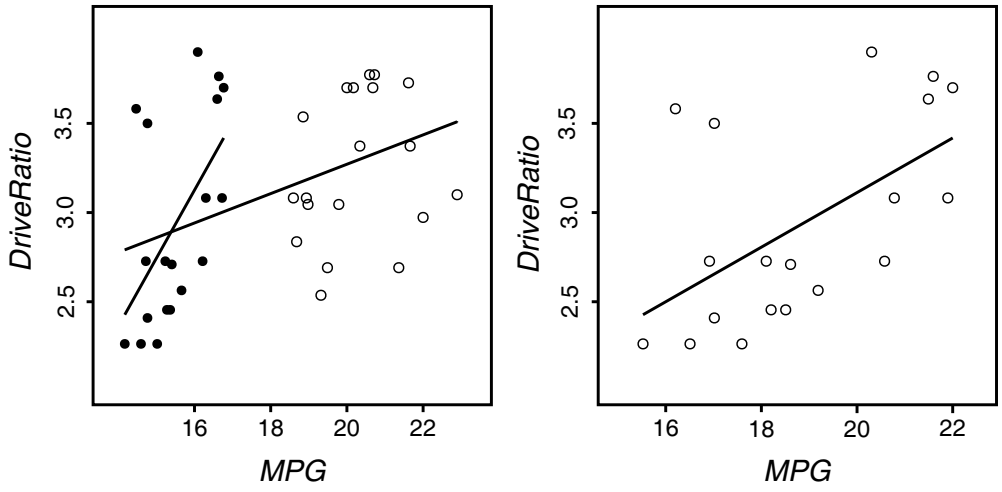

Figure 2.4 Focusing on a part of a plot.

ferent slope than the regression line for all of the points, which is also shown in the figure. Focusing on just the points selected produces the plot on the right, where we can study this part of the plot in more detail.

Focusing refers to the capability to look at just a subset of the data, focusing on a group of points or other elements of interest to get a better understanding of their structure. *Excluding* is just the opposite: removing a group of points or other elements which for some reason seem to distract from seeing the structure that may exist in the rest.

There are two versions of focusing. In one, the scale of the plot is not adjusted automatically after focusing. This lets the user easily compare the focus-on with the focus-off view, but the focus-on view may be very small or located at the edge of the plot. In the other, the scale of the plot changes when the focus is changed so that the plot is scaled and located to take up the entire viewing area. This makes focus-on/ focus-off comparison difficult, but permits full inspection of the structure in either case.

Notice that focusing in statistical graphics does not work like zooming in other computer programs. Zooming enlarges the points in the selected area, whereas focusing leaves the points the same size but moves them farther apart.

Changing colors. Using the capability to change the color of points is a way to emphasize any structure that you think you may see in your data. It is also a way to search through the data to see if the structure is really there.

For most plots, when you first see the plot, all of the points are the same color. Then, while you are exploring your plot, if you see something interesting (a cluster of points or an outlier, for example), you can emphasize it by changing colors of relevant points. This gives you a picture of the data that makes the structure easier to see.

If the plot in which you changed colors is linked to other plots, the colors of the points in the linked plots will also change, allowing you to see if the structure is also revealed in the other plots. This is the way to gather more evidence about the structure that you think may be in the data.

Furthermore, when the points represent the observations in the data (as they almost always do), the color information can be saved as a new categorical variable, using color names as the categories of the variable. These variables can be used subsequently for other analysis (e.g. cross-tabulations, discriminant analysis) to help clarify their meaning.

Color palettes are a very effective way of using color to search for structure, since they make color easy to access and since they have immediate consequences. If a program does not have color palettes, then menus providing access to color should be used. Dialog boxes should be avoided for changing color because they usually obscure the points that are being colored.

Perhaps the main awkwardness with symbols and colors arises when a region of a space is too densely packed with points, making it difficult to identify individual points. Interactive capabilities can be very helpful in dealing with this problem, as shown in Figure 2.5. In this figure there are two plots. In the upper plot the points have been identified with six different symbols. However, only three groups can be clearly perceived. By focusing on this part of the plot we get the view shown in the lower part of Figure 2.5, where it is quite a bit easier to see the various symbols.

Changing point symbols. Note that you can also change the symbols used to represent points, and that everything that was stated above about point color also applies to point-symbols. However, whereas point color is very effective at communicating structure, point symbols are not, because the symbols may superpose and form uninterpretable blobs, making them not very effective at communicating information to the viewer. A possible solution is to use what Cleveland calls *texture symbols*, symbols specially designed to do well in case of overlapping (Cleveland, 1994a).

Changing point labels. Finally, point labels can be a very effective way to identify structure in your data, but in some programs (including ViSta) it is so awkward and clumsy to change the labels that what would otherwise be an effective way of communicating structure becomes ineffective.

2.2.3 Manipulating Plot Dimensions

In this section we discuss techniques for manipulating the dimensions of the plot, such as re-ordering the variables or dimensions or some other feature; changing the absolute or relative size of the axes, etc.

Reordering. There are several examples in the data visualization literature which point out that *reordering* the elements in a plot can determine what we see in a plot. For example, Tufte (1997) explains that the ordering of the information in the plots that were used to decide whether to launch the space shuttle *Challenger* was the cause of not seeing that problems lay ahead: The tables and charts concerning the possibility of failure in cold weather that were prepared by NASA's engineers were sorted by time instead of temperature. When they were resorted by temperature, it was clear that temperature was of great importance.

There are guidelines which suggest how to order information in several situations. For example, it is recommended that bar charts show the bars ordered by the frequency of the categories (Cleveland, 1994a).

Recently, Friendly and Kwan (2003) introduced a principle that is intended to rationally order the presentation of information in complex and high-dimensional plots. They suggest that visual communication

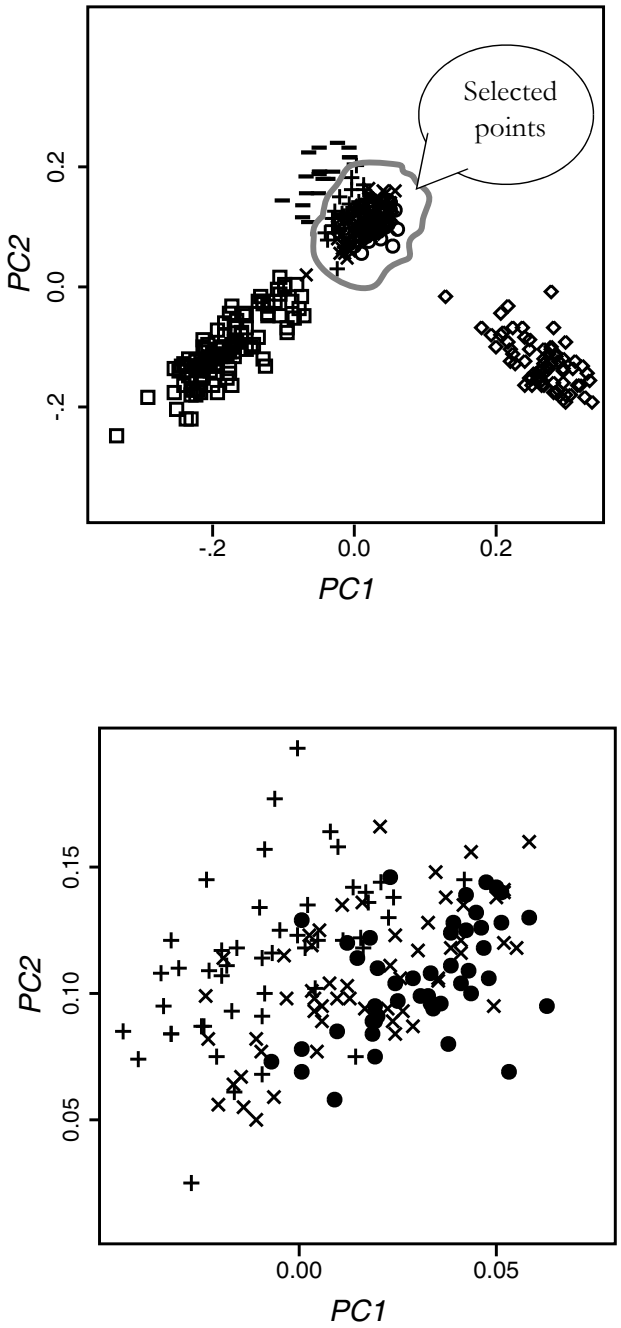

Figure 2.5 Focusing on part of a plot to see overlapping points.

can be facilitated by ordering according to *principal effects*, as they call it, with the definition of principal effects depending on the type of data. In essence, the principal effect is, mathematically, the most important information contained in the data. More precisely, they propose defining principal effects as being the appropriate one of the following four definitions: It is the *main-effects* of *n*-way quantitative data, the *associations among factors* for *n*-way frequency data, the *correlations* for multivariate data, or the *group mean differences* for MANOVA data. In all of these cases, the order of presentation can be optimized by using the values of the singular values to define the order.

Having a well-defined mathematical basis for determining how to order the presentation of information in displays of data is clearly an important first step toward produce compelling visualizations. We can look on such a definition as providing us a rational and (at least mathematically) optimal ordering to start with. The displays, of course, should enable and encourage the analyst to seek new orderings that can be more revealing than those produced by default.

Examples of dynamic reordering include the implementation of mosaic plots in Manet (Hofman, 2003). In these plots, categories of a variable can be ordered by dragging their representations in the plot. This technique would be useful to accomplish the goal mentioned in the preceding paragraph: helping the analysis improve the order shown by default.

Changing scale. The scale of a plot determines the size of the plot. Scatterplots usually choose their initial scale so that all its points will fit in the space available. This is an appropriate default, but it will be inadequate on some occasions, such as, for example, when there is one outlier observation that is very far from the rest of the points, leaving little space for the other points. In this case, focusing can be used to remove the isolated point from the view by modifying the scale of the plot.

2.2.4 Adding Graphical Elements

Another way of adding information to a plot is by using graphical elements such as lines or circles, much as in drawing programs. After all, statistical graphics are a type of graphics, and consequently, it seems natural to draw on them for incorporating remarks, pinpointing elements, and providing explanations. These non-statistical annotations can be very useful, but we do not describe them here because they have been widely described. However, there are similar features that are specifically statistical and are described later in the book, which we preview here.

One statistical use of adding lines is simply to connect the points of a point plot in a specific order, such as in the order they appear in the dataset. This turns a scatterplot into a time-series plot, for example. Many statistical software programs provide such a capability.

Two somewhat more advanced examples are shown in Figure 2.6. On the left we have fit a line to points in a point cloud. We are seeing not only the best-fitting line but the residual lines as well. In the second example (on the right) we have two variables that are very strongly, although not perfectly related by an apparently polynomial function. We have fit a straight line to them, and we have also fit a lowess function to them. Note that the lowess function is not a connect-the-dots function, although it looks like that

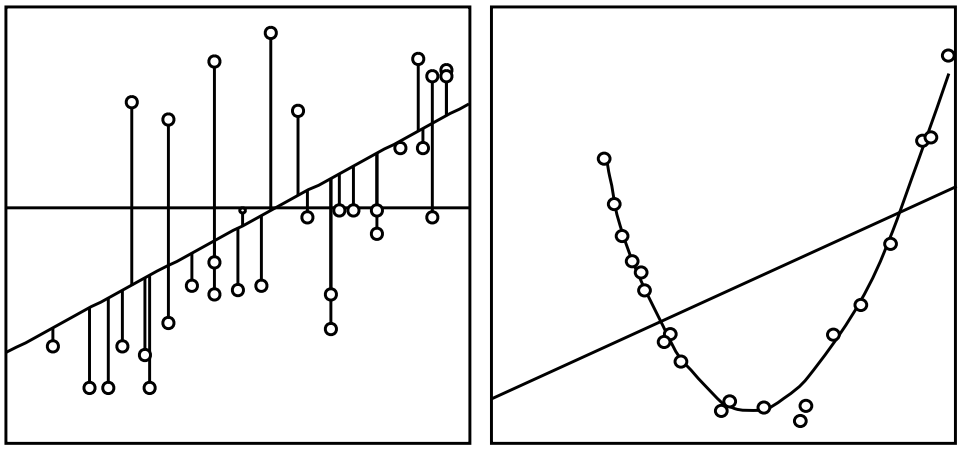

Figure 2.6 Statistical uses for adding drawings.

in much of the plot. But be sure you notice the bottom of the function where there is some scatter around the curve.

Other type of drawing figures can be added. Circles, for example, can be drawn proportionally to a third variable and added to an scatterplot, producing a *bubble* plot. Bubble plots show the relative magnitude of one variable in relation to two other variables.

Some programs offer drawing tools to add ovals, lines, polygons, and so on. These tools are useful for customizing displays to emphasize features found while exploring the data. For example, ovals can be used to surround a cluster of points, text can be added to provide an explanation for an outlier, or arrows can be used to illustrate the relation of one part of the part with another. The goal of this practice can be to make presentations to an audience but it can also be a way of recording information about the analysis for the analyst to use later. JMP is the program that is best in these capabilities.

2.3 Spreadplots

One of the main problems with the visual approach to statistical data analysis is that it is too easy to generate too many plots: We can easily become totally overwhelmed by the shear number and variety of graphics that we can generate. In a sense, we have been too successful in our goal of making it easy for the user: Many, many plots can be generated, so many that it becomes impossible to understand our data.

When we generate too many plots, and if we are restricted to having only one plot per window, we not only have the problem of understanding multiple plots, but we also have the problem of managing multiple windows. In this situation it becomes the case that too much time is spent managing windows, with too little time left for learning about our data.

Several solutions come to mind to improve our task. One solution is to put all of the plots in one big scrollable window, making sure that they don't overlap. Although this may be better than putting every plot in its own window, it still isn't a very good solution: The user now spends too much time scrolling up and down or back and forth trying to locate the right plot. Even simple tasks such as comparing groups can be very time consuming. Another solution is to make sure that the plots all fit on the screen, adjusting size as more are generated. Of course, the plots rapidly become too small to be useful, and just closing them can become a problem.

The spreadplot (Young, et.al., 2003) is our solution to these problems. A spreadplot is a multiview window containing a group of dynamic interactive plots, each of which shows a unique view of the same statistical object. The plots not only interact with the viewer, but also with each other. Since they all provide different views of the same statistical object, they are linked so that changes made in one view are reflected appropriately in the views provided by each of the other plots

Chapter 3 Categorical Data

3.1 Introduction

Categorical data can be displayed with several plots (barcharts, pie charts, etc.) that can be enriched with dynamic-interactive properties such as other plots. However, Mosaic displays (Friendly, 1999) have received considerable attention in the last years and we will focus on them in this chapter. We will use as example a very classic one: the .Berkeley Admissions Data.

Table 3.1 shows data on applicants to graduate school at the University of California at Berkeley for the six largest departments in 1973 classified by *Admission* and *Gender* (Bickel et al., 1975). This dataset has been used as an example of the application of log-linear models in several places (Agresti, 1990; Friendly, 2000; Rindskopf, 1990; Valero-Mora et al., 2004). This dataset has three variables: *Gender* of applicants (male/female), *Department* (A to F), and *Admission* (yes/no). For such data we might wish to study whether there is an association between *Admission* and *Gender*. Are male (or female) applicants more likely to be admitted? The presence of an association might be considered as evidence of gender bias in admission practices.

Table 3.1 Berkeley Admissions Dataset

		Gender			
		Male		Female	
		Admission			
		Yes	No	Yes	No
Department	A	512	313	89	19
	B	353	207	17	8
	C	120	205	202	391
	D	138	279	131	244
	E	53	138	94	299
	F	22	351	24	317

3.2 Mosaic Displays

Mosaic displays are graphical methods for visualizing *n*-way contingency tables and for visualizing models of associations among its variables (Friendly, 1999). The frequencies in a contingency table are portrayed as a collection of reticular tiles whose areas are proportional to the cell frequencies. Additionally, the areas of the rectangles can be shaded or colored to portray quantities of interest, such as residuals from a log-linear model (discussed in Section 3.3).

A mosaic plot can be understood easily as an application of conditional probabilities. For a two-way table, with cell frequencies n_{ij} and cell probabilities $p_{ij} = n_{ij}/n_{++}$, a unit square is first divided into rectangles whose width is proportional to the marginal frequencies n_{i+}, and hence to the marginal probabilities $p_{ij} = n_{i+}/n_{++}$. Each such rectangle is then subdivided horizontally in proportion to the conditional probabilities of the second variable given the first, $p_{j|i} = n_{ij}/n_{i+}$. Hence, the area of each tile is proportional to the cell frequency and probability,

$$p_{ij} = p_i \times p_{j|i} = \frac{n_{i+}}{n_{++}} \times \frac{n_{ij}}{n_{i+}}$$

The steps above are exemplified for the three variables of Berkeley data in Figure 3.1. There are three mosaic plots in this figure. The first step splits the entire rectangle into two areas proportional to the categories of *Gender*, so we can see that there are more males than females in the data. The second step divides the previous tiles (male/female) according to the number of applicants in each department. These new tiles would align vertically in the two columns if the proportion of males and females was the same in all departments. However, the plot reveals that there are more male than female applicants in departments A and B, while departments C to F have relatively fewer male than female applicants. Finally, the third mosaic plot displays *Admission* given the other two variables, *Gender* and *Department*. There is much information in this last display. As an example, the two tiles for the males in department A (marked with a thicker border) show that about two-thirds of males were admitted and one- third rejected at this particular department. This contrasts with the results for females at this department (upper-right corner), which have larger proportions of admission (80%).

Spacing of the tiles of the mosaic provide an important aid for interpretation when there are more than two variables in an axis. This can be observed in Figure 3.1, where the separation between the categories of *Gender* is larger than for the categories of *Admission*, making it easier to see the building blocks of the mosaic display.

A guide for interpreting mosaic displays is how the tiles align at the various splits. Nonaligned tiles mean that there is interaction between the variables, and aligned tiles, that the variables are independent. For example, we know that there is an interaction between *Gender* and *Department* in the second display in Figure 3.1 because the horizontal lines do not align. The lack of alignment of vertical lines in the third display of Figure 3.1 reveals the interaction between *Admission* and *Department* given the categories of *Gender*. Finally, we can compare the profile of these vertical lines for both genders to see if there is interaction between *Gender* and *Admission*. The lines have similar profiles across *Gender* except for department A, sug-

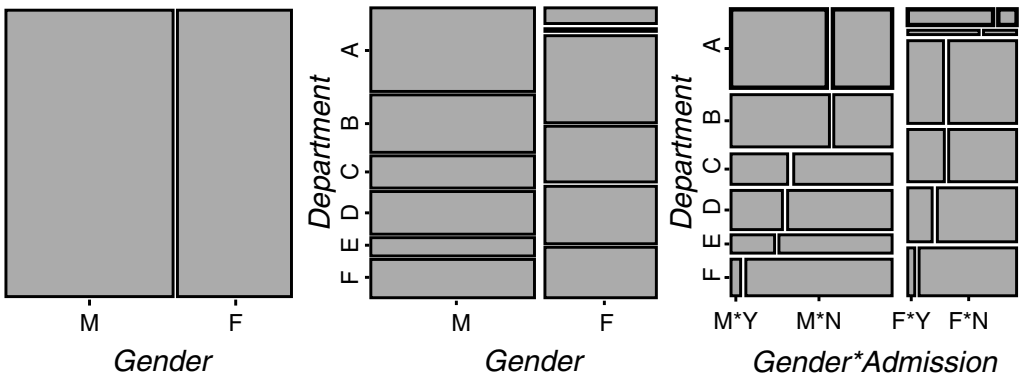

Figure 3.1 Building a mosaic plot for the Berkeley data.

gesting that there is no interaction for the rest of the departments. In other words, fitting a three-way interaction model to the data of all the departments apart from department A would be unnecessary in this case.

Static implementations of mosaic displays are available in some commercial programs (JMP, SAS, SPlus). As well, now there is noncommercial software that surpasses in many respects the commercial software. For example, MOSAICS (Friendly, 1992), which is available as a web page on the Internet, computes mosaic displays, log-linear models, and has interactive capabilities such as querying the mosaic bars to see the frequency or the residual values of the cells. The free R implementation of the S language provides the vcd package containing mosaic plots. Also, two programs developed at the University of Augsburgh, Manet (Hofman, 2003; Unwin et al., 1996) and Mondrian (Theus, 2003), are available as free downloads. Finally, the program ViSta (Valero-Mora et al., 2003, 2004) has interactive mosaic plots and a module for log-linear analysis.

3.3 Visual Fitting of Log-Linear Models

In the previous sections we have shown how to use tables and graphics to describe frequency datasets and to get a first impression of important features of the data such as individual values, patterns, or associations among the variables. As it is usually good practice to spend some time looking at the data with these basic tools in the initial phases of data exploration, we believe that the dynamic interactive extensions introduced above have a considerable interest for those performing statistical analysis with frequency data. However, if we want to test whether our descriptions actually fit the data, we need to go an step further and specify models that can be compared to our data.

Although there are several alternatives available with this same purpose, log-linear models provide the most comprehensive scheme to describe and understand the association among two or more categorical variables. For this reason, log-linear models have gained wide acceptance in recent decades, and every major statistical package now includes capabilities for computing them. Also, there are several excellent textbooks that address this subject (Agresti, 1990; Andersen, 1996; Ato and Lopez, 1996; ; Christensen, 1990), all at intermediate or advanced levels. Indeed, log-linear models for testing hypotheses for frequency data have a status similar to that of classical techniques such as regression or ANOVA for testing hypotheses for numerical data.

Unfortunately, despite the recognition currently enjoyed by log-linear models, dynamic interactive software for computing them is not as advanced as for numerical data. Thus, many of the features that have been implemented successfully for numerical data, discussed in the first part of this book, are absent from the programs for computing log-linear models.

However, the freeware programs mentioned in the introduction to Mosaic displays (Manet/Mondrian, MOSAICS, and ViSta), feature dynamic interaction applied to log-linear models. Thus, all these programs incorporate interactive tools for specifying, computing, and then displaying diagnostics of log-linear models. Furthermore, as these diagnostics may suggest modifications to the model, the programs also make it easy to modify the model and see the consequences in the diagnostics, using what can be called *interactive stepwise procedures*. In summary, these programs illustrate ways that dynamic interactive features can also be applied to the modeling of frequency data.

In the rest of this section, we review the features of the log-linear model module in ViSta. This module, named LoginViSta, works as a plug-in (i.e, it can be modified externally without modifying the internals of ViSta. The main advantage of LoginViSta over programs mentioned previously lies in its flexibility for specifying *all* types of log-linear models. Another unique feature of LoginViSta is a spreadplot that integrates a number of graphic and numerical diagnostics for models, some of which are very innovative. The section is organized according to the elements in the LoginViSta spreadplot and the way they work together to supply an interactive dynamic environment for log-linear models.

3.3.1 Log-Linear Spreadplot

A log-linear model can be understood as a linear model (regression or ANOVA) for the logarithm of the frequencies or counts of the frequency data table:

$$\log m = X\beta$$

where m is a column vector of fitted frequencies, X is the *model matrix* (sometimes called a *design matrix*) and b is a column vector that contains the parameters. As the predictors are all categorical, they have to be coded according to one of the habitual methods. For example, for a 2×2 table, the saturated model (which includes all the main effects and interactions for a dataset) with dummy coding (the one used throughout this section) can be represented as

$$\begin{pmatrix} \log m_{11} \\ \log m_{12} \\ \log m_{21} \\ \log m_{22} \end{pmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} \mu \\ \lambda_1^A \\ \lambda_1^B \\ \lambda_{11}^{AB} \end{pmatrix} \tag{3.1}$$

The formulation of log-linear models in the form of linear model puts them into the framework of generalized linear models (GLM's) (McCullagh and Nelder, 1989), a framework that also includes models such as the logistic regression, Poisson regression, and the standard model with normally distributed errors. This formulation has the advantages that any log-linear model may be expressed in a compact form and that it is easy to express variations with respect to basic models.

The design matrix can be used to set a large variety of models. Of these the most important type are those called *hierarchical models*, which are specified such that when a term is included in the model, all the lower-order effects are also included. *Nonhierarchical models* do not follow the rule of including all the lower-order effects of the terms included in the model. Nonhierarchical models are more flexible than hierarchical models, but those, in turn, are easier to interpret, as in many applications, to include a higher-order interaction without including the lower interactions, is not meaningful (Agresti, 1990; Vermunt, 1997). However, there are occasions where nonhierarchical models can be interpreted in a reasonable manner, often providing alternative explanations to those provided by hierarchical models (Rindskopf, 1990).

Finally, if one of the variables in the study can be considered as dependent and the rest as independents, the model can be formulated as a *logit model*. Logit models are traditionally regarded as different from log-linear models, but it can be proved that there is an equivalent log-linear model for each logit model. Logit models are simpler to analyze than log-linear models because they assume that the main effects and the interactions of the independent variables are already included in the model. This has the advantage that specifying, testing, plotting, and interpreting the results of logit models is considerably easier than doing the same for equivalent log-linear models. All these different models can be specified via the design matrix.

Log-linear models are specified in ViSta using the spreadplot shown in Figure 3.2. We discuss the panels of the spreadplot in the following sections, from left to right, up to down, as follows:

- *Model builder window.* Specification of log-linear models, hierarchical and not hierarchical, is carried out using this window. This panel is discussed in Section 3.3.2.
- *Predicted and observed mosaic plots.*
- *Past models window.* This window gives an overview of the process of modeling. This window can be used to rewind the analysis to any prior model. This pane is discussed in Section 3.3.4.
- *Parameters plot.* Parameters are an important, if cumbersome, element to consider for interpreting a model. This plot provides a visual help for such task. This panel is discussed in Section 3.3.5.

3.3.2 Specifying Log-Linear Models and the Model Builder Window

One of the key points of software programs for fitting log-linear models is how easily they let the user specify the model. Although some programs allow writing the model matrix manually, the process is too time consuming and error-prone to be practical, especially when there are more than two variables to consider and several models to test. Hence, command-oriented software programs for log-linear analysis define special notations that alleviate the task considerably, as they write the model matrix from succinct high-level descriptions of the model. So the saturated model for a dataset with three variables, with names *A*, *B*, and *C* can be indicated simply as $[ABC]$. These command-oriented programs give control over several elements of the model matrix, empowering the user to test many possible variations from the basic models. Examples of programs with this capabilities are GLIM, LEM (Vermunt, 1997), and the procedure

CATMOD in the SAS system. The LOGLINEAR procedure in SPSS can also be used to construct contrasts for testing nonstandard models. Of course, S, R, and S-Plus also have superb capabilities for log-linear modelling.

At this time, however, the programs that use graphical user interfaces do not provide all the capabilities of those based on commands. The number and type of models that can be fitted taking advantage of interactive techniques is typically very low compared with the large number of possible log-linear models discussed in the literature. In particular, hierarchical models are sometimes well covered, but nonhierarchical are usually no covered. The problem seems to stem from the complexity of designing an interactive dynamic visual matrix language for specifying the model.

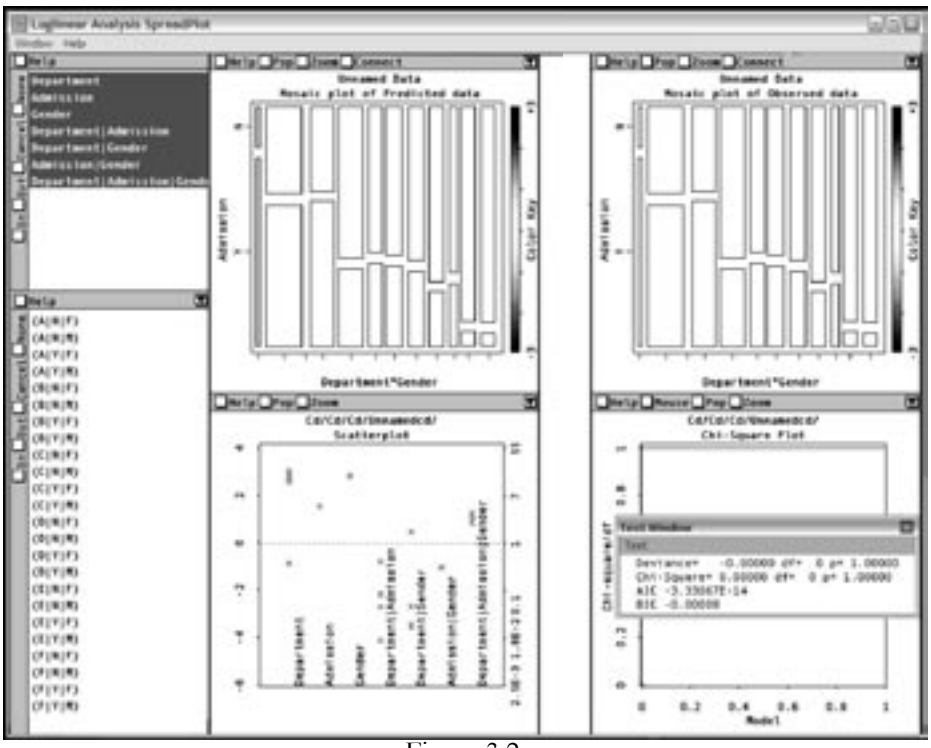

Figure 3.2
Spreadplot for visual fitting of the log-linear model to frequency data.

In the rest of this subsection we show how to specify log-linear models using LoginViSta, using as examples some models that could be considered for the Berkeley dataset introduced in Section .

Model builder window. Figure 3.3 shows the operations that are possible with the model builder window. This figure represent each operation by using two pictures; the first one displays the menu before the user acts, and the second, after. We use the pictures to illustrate the functioning of the model builder window in LoginViSta.

The model builder window lists all the main effects and interactions of the variables in the model. The first illustration shows how the window works *hierarchically* (i.e., clicking on an item selects all the items in the hierarchy included in that item). Thus, clicking on the last item, the highest in the hierarchy, selects all the items. The second illustration provides another example; clicking on the *Gender*×*Department* term also selects the main effects *Gender* and *Department*. The window also has a *non-hierarchical* mode that allows selecting individual items in the list (and consequently, specifying nonhierarchical models). The third illustration illustrates *deselecting* of model terms. Again this function works hierarchically, which hinders deselecting an item if other items higher in the hierarchy are still selected.

Notice that as the process of building log-linear usually starts with the saturated model, deselecting terms is often the most used action. It is usual that the analysis sets off in the saturated model and proceeds by removing terms from the model. The process stops when a parsimonious model with satisfactory fit is reached.

The fourth illustration shows how to add a specific vector to the design matrix. Clicking with the right button on a nonselected item pops up a menu with a list of the parameters for the term. The specific

example in Figure 3.3 displays the parameters for the three-way interaction *Admission* × *Gender* × *Department*. Selecting the parameter for department A adds it to the list of terms and interactions. This makes it possible to fit the nonstandard model shown below.

$$\log m_{ijk} = \mu + \lambda_i^A + \lambda_j^G + \lambda_k^D + \lambda_{ik}^{AD} + \lambda_{jk}^{GD} + \delta_{K=A}\lambda_{ijk}^{ADG} \tag{3.2}$$

where $\delta = 1$ if $k = A$ or 0 otherwise. This model asserts that *Gender* and *Admission* are independent except in department A.

Figure 3.3 Model builder window of LoginViSta.

3.3.3 Evaluating the Global Fit of Models and Their History

For each log-linear model that we may fit a dataset, there are overall or global measures of goodness of fit, described below. In practice, we usually want to explore several alternative models and choose the simplest model that achieves a reasonable goodness of fit. In an interactive setting, we introduce the idea of a history plot, showing a comparative measure of fit for all models contemplated. The interactive features of this history plot also allow you to return to a previous model (by selecting its point) or to make model comparisons.

Tests of global fit. The most commonly used measures of how well the model reproduces the observed frequencies are the familiar χ^2 Pearson statistic,

$$\chi^2 = \sum_i \frac{(n_i - \hat{m}_i)^2}{\hat{m}_i}$$

and the *likelihood ratio* or *deviance statistic*,

$$G^2 = 2\sum n_i \log(n_i / \hat{m})$$

where the n_i are the frequencies observed and the m_i are the expected frequencies given the model considered. Both of these statistics have a χ^2 distribution when all expected frequencies are large. The (residual) degrees of freedom is the number of cells minus the number of parameters estimated. In the saturated model the deviance (or Pearson χ^2) is zero and the degrees of freedom are also zero. More parsimonious models will have positive values of deviance but also more degrees of freedom. As a rule of thumb, non-saturated models fit the data if their deviance is approximately equal to their degrees of freedom (or the ratio $\chi^2 \S df$ is not too large).

The deviance is unique in that it can be used to compare nested models. Two models are nested if one of them is a special case of the other. Comparison of models provides a way of focusing on the additional effects of the terms included in the larger model.

Plot of the history of models. Figure 3.4 is a plot of the history of the values of fit obtained along a session of modeling. The points in the plot represents the values of χ^2/df for five models for the Berkeley data (we could have used the deviance in this plot, too). Values close to 1 mean that the model fits well.

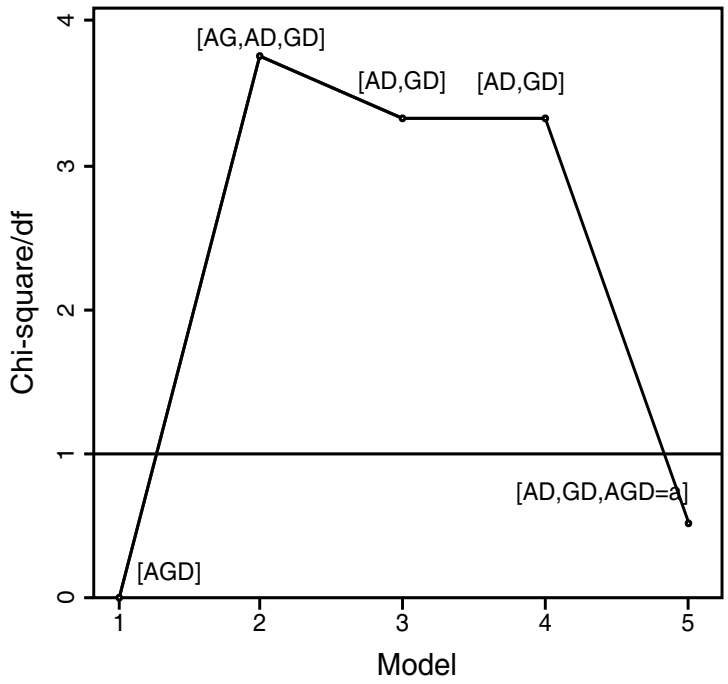

Figure 3.4 History of models applied to Berkeley data.

Labels of the points identify the models considered. This plot allows us to see the process of model fitting at a glance.

The example in Figure 3.4 portrays the following sequence of analysis:

(1) The plot starts with the saturated model[AGD], which fits perfectly.

(2) It continues with the model with all the two-way interactions, which does not fit well.

(3) The model that evaluates the hypothesis of no gender-bias effect is tested (no [AG] term), but its fit is still not satisfactory.

(4) As preparation for testing model 5, the category of reference for the variable *Department*, set at this moment to *A*, was changed to *F*. This produces model 4, which has the same fit and terms as model 3.

(5) Finally, the model testing the interaction of *Gender* and *Admission* was tested only for department A . This model has a good fit as shown in Figure 3.4.

The history plot in LoginViSta is not only a way of representing the successive values of fit of the models considered but a control panel for managing the process. The actions that can be performed from this panel are the following:

- Selecting a point in the plot changes all the plots in the spreadplot of Figure 3.2 to display the values of the model symbolized by such a point. This action allows you to review models fitted in the past.
- Selecting two points (two models) produces a model comparison test. LoginViSta checks if the models are nested before the actual test is accomplished. If the comparison is appropriate, a test of significance of the difference in deviance is displayed in a separate window.

Other actions that could be added to this plot are the capability of removing points, re-arranging them, and supporting different threads (using lines of different colors) and of other goodness-of-fit indexes. In summary, to find an adequate parsimonious model, the plot in Figure 3.4 could be extended to manage completely the process usually followed.

3.3.4 Visualizing Fitted and Residual Values with Mosaic Displays

For a log-linear model, residuals provide important information regarding the adequacy of the model. Outlier cells (with large absolute residuals) may indicate particular levels of some factor that are not well explained by the current model, as we saw in the Berkeley data. More generally, the pattern of signs and magnitudes of residuals may suggest a better model. Mosaic plots are an excellent way to display residuals from a fitted model in the context of a picture of the data.

Another use of mosaic plots is related to understanding the constraints imposed by models. Mosaic plots of fitted values (instead of observed) show the assumed relationship between variables in a graphical way, illustrating the consequences of introducing or removing terms in the model. Also, comparisons of fitted and observed mosaic plots provide a visual check of fit of the model.

In this section, we review the formulas for residuals, including an example of the problems that arise using point plots. Then we show how using predicted and observed mosaic displays can be a good alternative for visualizing these residuals.

Residuals for log-linear models. For a log-linear model, raw residuals are the difference between the values predicted and observed in the cells. However, raw residuals are of little use, as cells with larger expected values will necessarily have larger raw residuals. The *Pearson residuals*

$$e_i = \frac{n_i - \hat{m}_i}{\sqrt{\hat{m}_i}}$$

are properly scaled and approximately normally distributed, but they depend on the leverages of the matrix of the predictors. Therefore, one may define *adjusted residuals* (Agresti, 1990; Friendly, 2000b; Haberman, 1973):

$$r_i = \frac{n_i - \hat{m}}{\sqrt{\hat{m}_i(1-h_{ii})}}$$

(where h_{ii} is the leverage or hat value computed as in any generalized linear model), which are standardized to have unit asymptotic variance.

To check for cells that deviate from normality, the adjusted residuals can be plotted using a normal probability plot. However, the constraints imposed by the model have consequences that make this type of plot less useful than it might be. We will provide an example using the Berkeley data introduced in Section . Table 3.2 contains the residuals of the model $[AD][GD]$, which asserts that there is an interaction for

Table 3.2 Residuals for Model$[AD][GD]$

			Admission	
			Yes	No
Department	A	Female	4.15	4.15
		Male	-4.15	4.15
	B	Female	0.50	-0.50
		Male	-0.50	0.50
	C	Female	-0.87	0.87
		Male	0.87	-0.87
	D	Female	0.55	-0.55
		Male	-0.55	0.55
	E	Female	-1.00	1.00
		Male	1.00	-1.00
	F	Female	0.62	-0.62
		Male	-0.62	0.62

Gender

Admission and *Department*, and for *Department* and *Gender*, but not for *Gender* and *Admission* (i.e., there is no gender bias). As signaled by Agresti (1990), this model imposes the constraint that the sum of interactions predicted between the categories of *Department* and those of *Gender* be the same. A bit of manipulation shows that the residuals for each department are the same in absolute values, but with changed signs. Now, notice that a normal probability plot would be less than ideal in this situation because it would not reflect such a structure in the data. Mosaic displays, discussed in Section 3.2, are a good alternative to Table 3.2 because they can be laid out to reflect such structure.

The Predicted and observed mosaic plots. Figure 2.1 shows two mosaic displays for the model $[AD][GD]$ for Berkeley data.

The mosaic display on the left of Figure 2.1 has tiles proportional to the values predicted for the model, while the mosaic display on the right has tiles proportional to the values observed. Notice that as the model specified excludes interaction between *Gender* and *Admission,* the horizontal lines within each department are parallel in the mosaic plot for values predicted . Comparison of these lines with the lines in the display for the values observed reveals that the predictions are generally correct except for Department A (and perhaps B). This points to the aforementioned lack of gender bias except for department A (and perhaps B).

The previous impression can be reinforced by looking at the residuals of the model. Residuals are typically represented in mosaic displays using two colors (usually red for negative residuals and blue for positive residuals), the greatest intensity of the color representing the extreme of the values. Thus, pure blue

would stand for very extreme positive residuals and pure red for very extreme negative residuals. Finally, cells with null residuals are displayed in white.

Nevertheless, as the previous scheme does not transfer well to black-and-white publications, we have chosen to employ one that uses pure black and pure white to represent the maximum and minimum residual values (those for department A), and a middle gray residual equal to zero. This scheme is not as effective as one based on colors but can work well if the patterns of residuals, as is our case, are simple.

Examination of Figure 2.1 portrays the residuals of the model $[AD][GD]$. Department A has the most extreme residuals under the model specified, being positive for admission of females and rejection of males, and negative, vice versa. The remaining departments are colored with similar, close to intermediate, levels of gray. In summary, the model $[AD][GD]$ copes well with departments B to F, but not with department A.

3.3.5 Interpreting the Parameters of the Model

Parameters of log-linear models are an important tool for interpretation of results based on log-linear analysis. Log-linear models could be understood as linear models of the logarithm of the frequencies. Therefore, in much the same way that parameters provide an essential tool for understanding regression models, they are also fundamental for log-linear analysis.

However, parameters for log-linear have a reputation for being difficult to interpret, as "attested by the frequency of articles on the topic" and that "they devote considerable space to correcting the errors of their predecessors" (Alba, 1987). Therefore, it is not unusual that researchers do not take full advantage of their potentiality and that parameters remain underused.

Software for log-linear analysis has not helped to ease this situation. Many of the programs do not print the coefficients of the parameters for the models very clearly, so that to understand this part of the output, researchers need considerable diligence. The common flaws are excessively long outputs, without proper labels and with no indication of aspects of computation necessary for correct interpretation (e.g., the coding used).

The display included in LoginViSta is, on one hand, an attempt to organize this part of the output of log-linear analysis, but on the other hand, is also a way of supporting the task of interpreting the parameters of log-linear analysis. This last goal is reached by providing a set of interactive tools that interrogate the plot to obtain automatic interpretations of the meanings of the parameters. These explanations decrease the cognitive burden of interpreting the parameters.

Our plan in this section is to review the theory necessary for the interpretation of parameters of log-linear models and then proceed to a description of the plot of parameters.

Interpretation of parameters. The meaning of the parameters in log-linear models depends critically on the type of coding used. Two types of coding, *effect coding* and *dummy coding* are the most commonly used

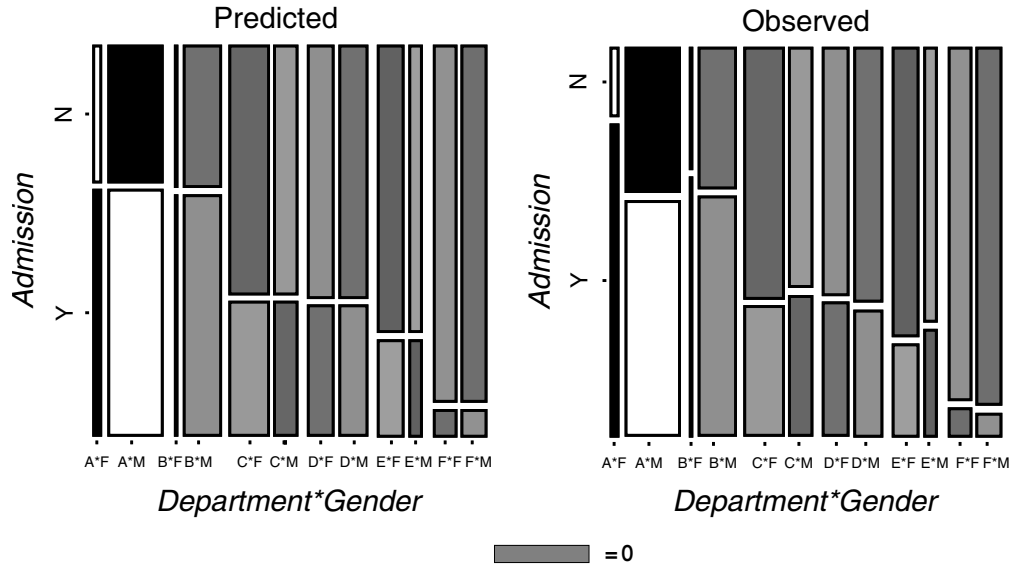

Figure 3.5 Mosaic plots for model $[AD][GD]$ for Berkeley data.

(see, e.g., Rindskopf, 1990). The fitted values are the same under both methods of coding, but the interpretation of parameters depends on the coding used. It is often claimed that dummy coding makes interpretation of parameters easier than effects coding because the parameters are interpreted as comparisons between the categories they represent and the reference category. For such this, we will restrict this exposition to dummy coding.

In dummy coding, one generates a number of vectors such that in any given vector membership in a given category is assigned 1, and nonmembership in the category is assigned 0. To avoid collinearity of the categories, the vector for one of the categories is removed from the design matrix. The category removed is called the *reference category* or *baseline category*. Therefore, the number of vectors for each of the main effects in the model is $k-1$, where k is the number of categories of the variable. Vectors for interaction effects are built by multiplying vectors for main effects.

The parameters created as described above have a simple interpretation as a function of the values predicted for the cells. Using the Berkeley dataset introduced in Section and setting *Admission* = No (*n*), *Department* = F (*f*), and *Gender* = Male (*m*) as reference categories for the saturated log-linear model, the value of the intercept in this model is

$$\mu = \log m^{ADG}_{nfm}$$

(i.e., the logarithm of the value expected for the cell, corresponding to the reference categories for the variables in the model).

The coefficient for the main effect of *Admission* is

$$\lambda^{A}_{y} = \log \frac{m^{ADG}_{yfm}}{m^{ADG}_{nfm}}$$

which is the *odds* of *Admission* (holding the other variables constant at their reference category).

The coefficient for the interaction between *Admission* and *Gender* for *Department* = F is

$$\lambda^{AG}_{yf} = \log \frac{m^{ADG}_{yff} / m^{ADG}_{nff}}{m^{ADG}_{yfm} / m^{ADG}_{nfm}}$$

which is the logarithm of the ratio of odds ratio of admission of males relative to females in department F.

The interaction of *Admission* with *Department* produces five coefficients, which are the product of the $(k-1)(j-1)$ categories of these two variables. As an example, the coefficient for department B turns out to be

$$\lambda^{AD}_{yb} = \log \frac{m^{ADG}_{ybm} / m^{ADG}_{nbm}}{m^{ADG}_{yfm} / m^{ADG}_{nfm}}$$

which is again an odds ratio. Finally, there are five coefficients for the three-way interaction term. As a sample, the coefficient for department B is

$$\lambda^{ADG}_{ybm} = \log \frac{m^{ADG}_{ybf} / m^{ADG}_{nbf}}{m^{ADG}_{ybm} / m^{ADG}_{nbm}} \Big/ \frac{m^{ADG}_{yff} / m^{ADG}_{nff}}{m^{ADG}_{yfm} / m^{ADG}_{nfm}}$$

which is an odds ratio of odds ratios. The same structure can be applied to define higher-order parameters.

As can be seen from the description above, the difficulties of interpreting parameters of log-linear models are not conceptual, as the rules for composing the parameters for hierarchical models are rather simple, but practical, as it is necessary to keep in mind a number of details, such as the reference categories used and the general structure of the model. Also, the number of parameters in a log-linear model is often large, so they can be difficult to visualize readily. As we will see in the next section, dynamic interactive graphics techniques can diminish enormously the burden of interpreting the parameters.

Parameters plot. The parameters plot in LoginViSta is an interactive extension of a plot proposed by Tukey (1977) in the context of ANOVA. Figure 3.6 is a plot of parameters applied to the Berkeley data. In particular, the model displayed is the saturated or complete model. This model includes all the main effects and interactions of the variables and can be a useful starting point in a session of modeling. The interactive tools described below enhance the plot to make it still more helpful for working with log-linear models.

Figure 3.6 displays the parameters grouped by terms. The value of each parameter can be read on the left side of the plot in the log scale, and on the right side in the original scale (odds, or odds ratios). Notice

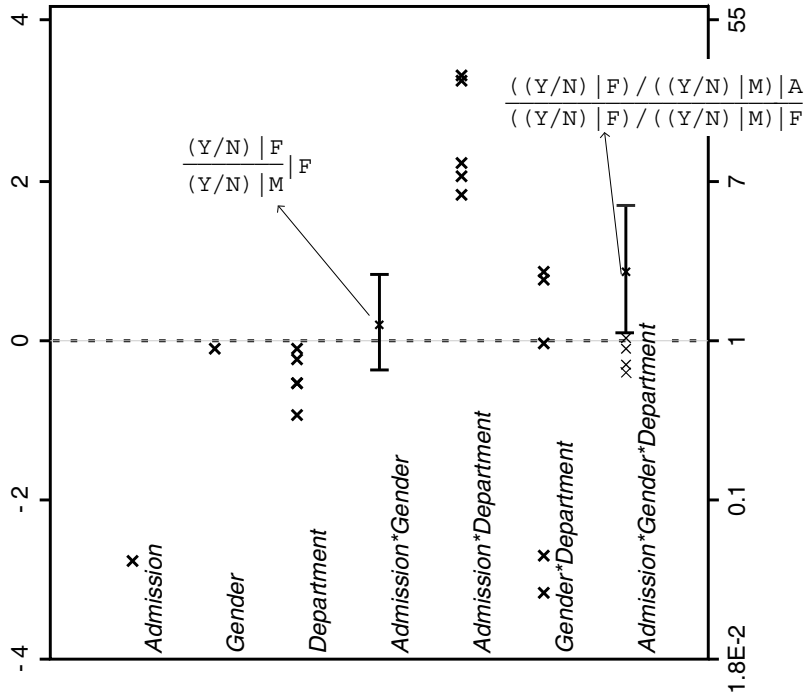

Figure 3.6 Plot of parameters for Berkeley's data-saturated model.

that the parameters excluded to avoid redundancy are set to 0 (1 in the original scale), so they can be visualized as standing on the dashed line.

Selecting a point in the plot reveals a 95% interval of confidence for the parameter. Two points are selected in this manner in Figure 3.6. First, we have selected the only parameter for interaction between *Gender* and *Admission* that happens not to be different from zero. This suggests removing this parameter from the model (but that would make the model nonstandard). The second parameter selected is the one for three-way interaction involving department A. The confidence interval for this parameter does not cross the zero line (i.e., it is different from zero).

The plot also shows the interpretation in terms of the odds ratios of the parameters selected. Focusing on the parameter selected in the three-way interaction term, we can see that this parameter basically tests the admission of females over males in department A versus department F. As mentioned above, this parameter is different (larger) from zero. Therefore, as the rest of parameters for this term are not different from zero (tests not shown), we can infer that the only department with bias gender is department A.

Chapter 4 Numerical Data

4.1 Introduction

Numerical data is the most important case of data and many concepts and tecniques have traditionally been developed first for them before categorical or other types of data. Here, we will consider techniques for univariate numerical data, bivariate, and multivariate.

4.2 Univariate data: Histograms and Frequency Polygons

The histogram and frequency polygons are two closely related schematic representations of the distribution of a numeric variable. They can show us the center and spread of a distribution, can be used to judge the skewness, kurtosis, and modicity of a distribution, can be used to search for outliers, and can help us make decisions about the symmetry and normality of a distribution. Having all of these capabilities, it is no wonder that histograms and frequency polygons have a long history of use and development in statistics, always seeming to be the focus of much interest, among graphics developers and users alike.

Both histograms and frequency polygons represent the relative density of the sample distribution by counting the numbers of observations that fall into intervals of equal width. The intervals, which are called *bins,* are constructed by partitioning the range of the data. The number of observations in each bin is represented by histograms as bars with height proportional to frequency, whereas frequency polygons represent the same information by a polyline of connected dots. ViSta's default initial histogram and frequency polygon are shown in Figure 4.1.

Unfortunately, neither histograms nor frequency polygons provide a very accurate schematic of the data. In fact, they can present a seriously misleading picture of the distribution of a variable's values. However, when compared to other techniques, the histogram and frequency polygon are easy to understand and are easy to explain and communicate. They are particularly useful for understanding the concepts of a variable and a distribution. As a consequence, experts with advanced and thorough understanding of statistics use histograms when they address audiences without experience in statistics, even though the advanced statistician undoubtedly knows that histograms are not the best way of portraying the distribution of a variable's values. The problems with histograms derive from two of the decisions required for its construction: the bin width and the bin origin.

Bin width problem. The bin width problem is illustrated in Figure 4.2 for the data about *MPG*. In this figure we show six histograms, each with a different bin width. We can see that the main feature of the *MPG* variable, the gap in the center of the distribution, is visible in some of the histograms but not in all.

Bin width has an enormous effect on the shape of a histogram (Wand, 1996). Bins that are too wide produce an appearance of solid blocks. There will be no evidence of the gap that we have seen so clearly in the *MPG* data. Bins that are too narrow produce a jagged histogram with too many blocks. There will be almost no information about points that have a higher density of observations.

It is also the case that to produce nice tick marks, changing the width also changes the origin of the bins, producing an additional source of differences among the plots. Furthermore, scales on both axes must be modified to accommodate the bars in the space available.

In short, poor selection of bin width results in oversmoothing or undersmoothing of the histogram. This problem has been dealt with in the statistical literature, and a number of rules have been proposed that provide appropriate bin widths. We discuss some of these rules and we show that having the capability of interactively modifying the histogram can be useful to find values that are subjectively good.

Scott provides a historical account of the bin width selection problem (Scott, 1992). The earliest published rule for selecting bin width seems the be that of Sturges (1926) and amounts to choosing the bin width h as

$$\hat{h} = \frac{\text{range}}{1 + \log_2 n}$$

This rule is used as the default rule for many statistical packages even though it is known that it leads to an oversmoothed histogram, especially for large samples (Wand, 1996).

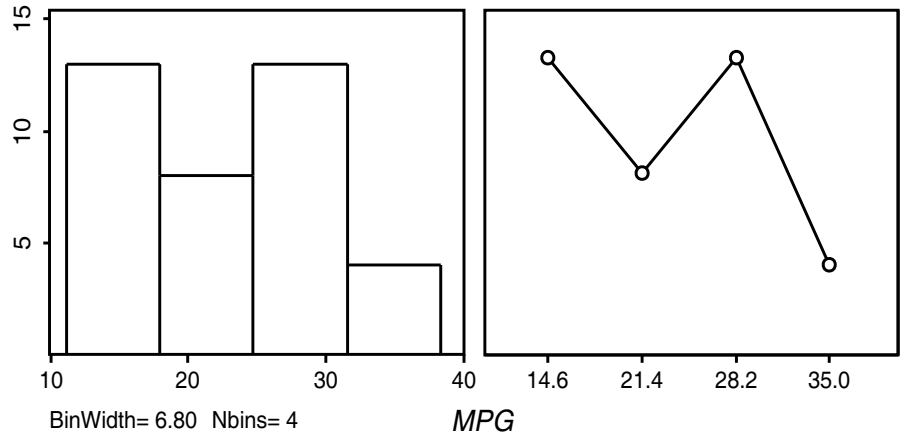

Figure 4.1 ViSta's initial histogram (left) and frequency polygon (right).

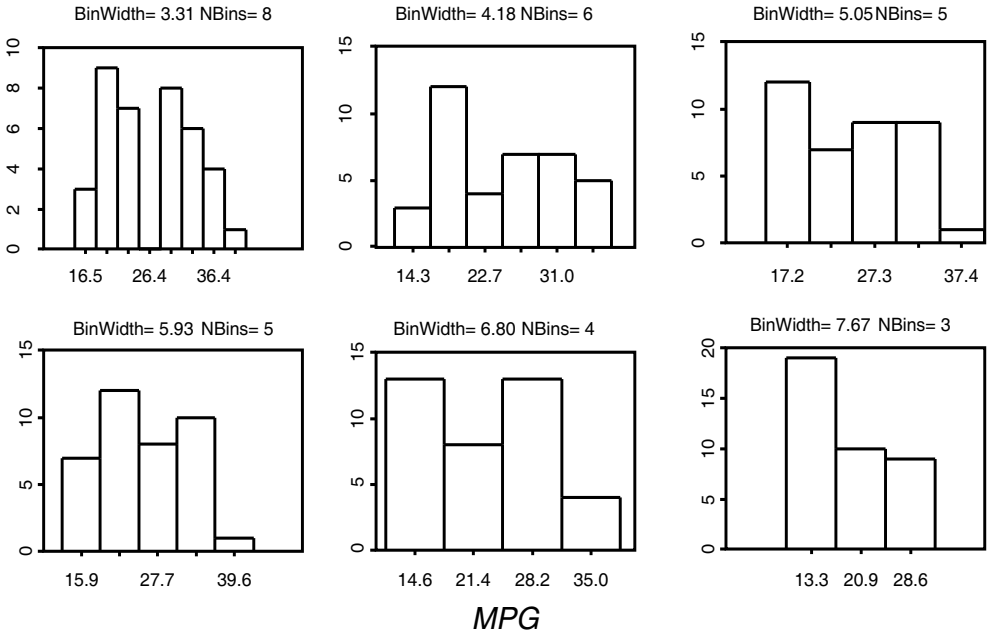

Figure 4.2 Series of histograms for *MPG*.

Scott (1992)) suggested the following rule

$$\hat{h} = 3.49\hat{\sigma}n^{-1/3}$$

This rule is known as the *normal rule* because it is tuned up to the normal distribution with a standard deviation of σ. This histogram is shown in Figure 4.2 in the fifth position, with four bins of width 6.80. Scott (1992) provided modifications of this rule for distributions with varying skewness and kurtosis. Notice that this formula only gives the right answer to the bin width problem given an optimality criterion not discussed here. Other criteria could also be considered, and the data may not follow the normal distribution. Hence, exploring different versions of the histogram may be important to get a more thorough understanding of the data at hand.

Bin origin problem. In addition to the bin width problem, histograms suffer from what is known as the bin origin problem. The *bin origin* is the value at which the lowermost bin's lower boundary is located. Figure 4.3 displays four histograms with the same bin width but with slightly different origins. Notice that in this figure, just as in the preceding one, the main feature of the *MPG* variable, the gap in the center of the distribution, is visible in some of the histograms but not in all.

While the dependence of histograms on different bin widths is easy to comprehend, the dependence of the histogram on the origin of the bins is quite disturbing. Moreover, there does not seem to be any strategy available to cope with this problem. Our recommendation is to use kernel density estimation, an improvement over classical histograms that does not suffer from the bin origin problem. It is also a good idea to use dynamic interactive graphics to test different origins for the histogram as well as different bin widths. This gives one an understanding of the robustness of the features of interest. We also recommend using the shaded histogram notion to see if it reveals anything of interest.

DIG: coping with histogram problems. ViSta provides a histogram that uses as a starting point the bin width given by the preceding formula. Using interactive tools, however, it is possible to explore other bin widths easily. Such an exploration of bin widths may give a more complete picture of the trade-offs required for the various choices. A slider is a particularly effective way to control the bin width of the histogram. Figure 4.2 was created through a selection of the histograms that an analyst could easily review using the slider (actually, only one of four plots is shown in Figure 4.2). Looking at these plots, it is easy to see why the histogram is not, nowadays, considered a very accurate tool for statistical analysis. The differ-

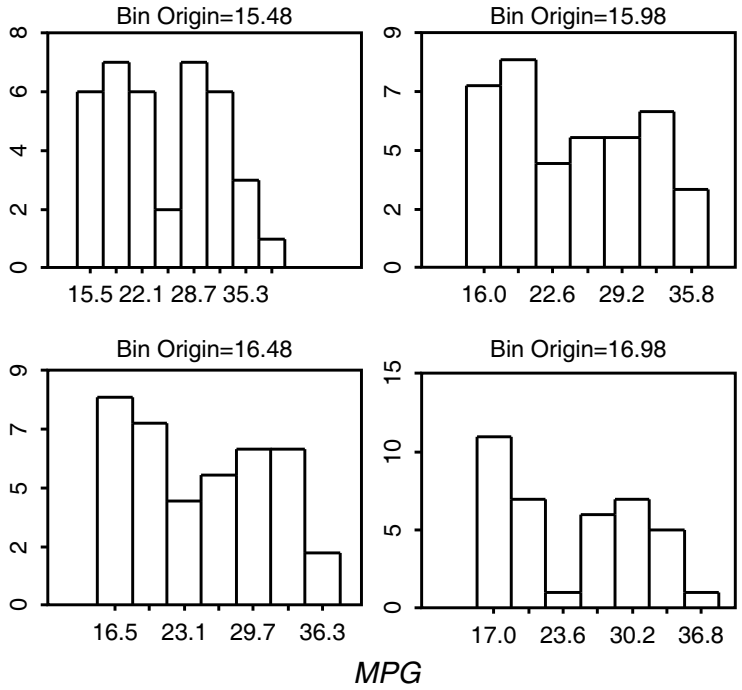

Figure 4.3 Effect of bin origins on the histogram. Bin origins are shown above each panel. All have seven bins with bin widths of 3.31.

ent histograms seem to tell different stories, not all of them coincident. Given that we know that there seems to be a gap in the middle of the data and that perhaps there are two separate distributions, we would choose the first plot left above. However, *not knowing what we know,* there does not seem any good reason to prefer one histogram over the others.

Fitting curves to the histogram. Curves for various functions can be fit to the histogram. These curves include the normal curve, shown on the left side of Figure 4.4, and the kernal density curve, shown on the right side.

Density traces. Density in data can be seen as the relative concentration of observations along the sections of the measurement scale (Chambers et al., 1983). Histograms can be considered a representation of density if they are modified slightly. The modification is to divide the frequencies by the number of cases so that the bars represent proportions or relative frequencies. Since the histograms would be redrawn on the new scale, this change does not alter their shape.

However, histograms are limited in that they inform about the density only in the middle of the interval and produce only one value for the entire interval. A more interesting approach would be to provide values that inform of the density at smaller intervals, while permitting the intervals to overlap. In this way, when we draw a line connecting the midpoint of each interval we will not see drastic changes like those we see with ordinary histograms.

The general formula for kernel density estimation is

$$\hat{f}_h(x) = \frac{1}{nh}\sum_{i}^{n} K\left(\frac{x - X_i}{h}\right)$$

where $\hat{f}_h(x)$ is the density estimated at point x given a size bin h, n is the number of cases in the data, $K(u)$ is a function denominated *kernel function,* and X_i are the sample data. An example of a kernal density curve is shown on the right side of Figure 4.4. The degree of smoothing in this formula depends on two factors:

- The bandwidth h, where shorter intervals will produce more local estimations and a rugged impression in the estimated curve, and longer intervals will produce smoother estimations.
- The kernel function $K(u)$, which is regarded as being a factor of less importance than the width h. These functions basically result in smaller values of $\hat{f}_h(x)$ for observations farther away from x and larger values for observations closer to x. The permissible functions vary in their weightings of the observations. As an example, a function commonly used is the *Gaussian:*

$$\frac{1}{\sqrt{2\pi}} e^{-(1/2)u^2}$$

Selecting an adequate bandwidth is generally seen as being more important than selecting an appropriate kernel. Selecting the best bandwidth can be performed using a slider connected to a display of the density.

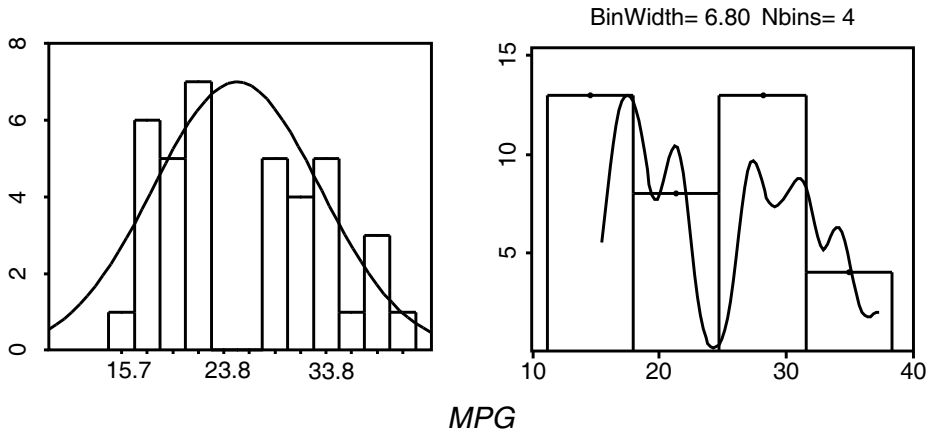

Figure 4.4 Histogram with normal (left) and kernal (right) curves.

This permits the user to rely on intuition, interpretability, and aesthetics. In fact, the process of selecting bandwidth is probably as important as the final value chosen, since it serves to familiarize the user with the data.

Even though interaction techniques are well suited to the task of exploring for appropriate bandwidth values, we start with values that satisfy a given optimality criterion. An estimator that is commonly used is the r*ule of thumb,* which in its robust version has the following formula (Silverman, 1986; Turlach, 1993):

$$\hat{h}_{\text{rot}} = 1.06 \ \min\left(\hat{\sigma}, \frac{\hat{R}}{1.34} n^{1/5} \right)$$

Determining a reasonable interval of values to explore is also a critical decision. Udina (1999) mentions the range $\left[h_0 / 9, \ 4 / h_0 \right]$ as being desirable.

4.3 Bivariate data: Scatterplot

The scatterplot shows the strength and shape of the relationship between two variables of a dataset about characteristics of cars. Two scatterplots are shown in Figure 4.5. Each scatterplot shows the *MPG* (miles per gallon) and *Weight* variables. In each plot the variables are represented by the *X*-axis (drawn horizontally) and the *Y*-axis (drawn vertically), and the observations are represented by the points in the scatterplot, each point being located according to the observation's values on each of the two variables. These values can be approximately determined from the plot by seeing what value of the *X*-axis is below the point and what value of the *Y*-axis is to the left of the point.

4.3.1 What we can see with scatterplots

It is important to pay attention to the first impression one gets from a scatterplot. For the plot showing the relationship between *MPG* and *Weight*, shown as the right image of Figure 4.5, the first impression is that the weight of the cars is inversely related to their efficiency and that the relationship is very strong. It appears that the relationship is nearly, but not quite linear, there being a slight curve in the trend.

The scatterplot can show us information about the shape, strength, direction, and density of the relationship between the two variables. It can also tell us about skedasticity and about outliers and isolates.

Shape. The shape of the relationship between two variables refers to the overall pattern of that relationship. Some of the more common shapes of the many that we may encounter are linear, quadratic, cubic, etc. The shape of the relationship between *Weight* and *MPG* (Figure 4.5) is linear, with a hint of curvilinearity. Note that many statistical tests require linearity.

Strength. The strength of the relationship concerns how close the points in the figure are to their presumed underlying error-free functional relationship. We can see in Figure 4.5 that vehicle weight predicts 81% (= 0.90^2) of the variation in fuel efficiency, assuming that their relationship is linear. But we know that

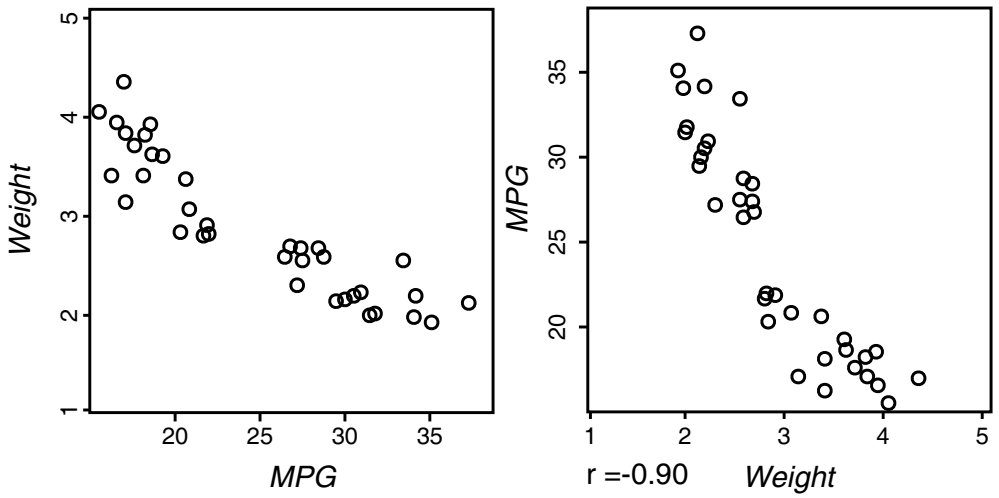

Figure 4.5 Scatterplots for the *Weight* and *MPG* variables.

the assumption of linearity underlying the calculation of the correlation coefficient can make its value very misleading. A quadratic relationship, may a linear correlation coefficient of zero.

Direction. This concept applies unambiguously only to monotonic (strictly increasing or decreasing) relationships, which, of course, include linear relationships. A relationship is positive (or increasing) if as *X* increases, so does *Y*. It is negative (or decreasing) if as *X* increases, *Y* decreases. The top-left plot displays a positive direction. We can also apply the concept without much difficulty to the cubic and exponential (center) figures, both of which have a positive direction. We cannot, however, apply it to the bottom two figures. The relationship between *Weight* and *MPG* in Figure 4.5 is negative or decreasing.

Density. The mathematical concept of density (which we refer to here, and which is not the same as the statistical concept of density) concerns the relative proximity of the points in the point cloud. If it is the case that the relative proximity remains the same throughout the point cloud, the density is said to be "everywhere equally dense." Alternatively, it is often the case that the density gradually tapers off as we proceed from the center to the edge of the cloud, creating what is known as a *tapered density*. Finally, it is not uncommon to see scatterplots where the density is very clumpy. In Figure 4.5 the five simulated distributions are equally dense; the empirical one is clumpy.

Many commonly assumed distributions (normal, for example) are smoothly tapering or equally dense, not clumpy, so clumpiness is a warning that standard assumptions may not apply and that multiple generating functions may be at work. We may wish to form groups of points that seem to form a clump and keep our eye on it as we progress through the analysis.

Outliers/isolates. We should always look for outliers, as they may represent interesting or erroneous data, as discussed in Chapter 5. Note that it is possible to have points that do not appear to be outlying on either of the two variables when they are looked at individually, but will be clearly seen as being outliers when the two variable are looked at together. The points at the top of the cubic and exponential distributions may be outliers. Outlying points can drastically affect the results of analyses, and should be carefully attended to before further analyses are performed.

Skedasticity. *Skedasticity* refers to whether the distributional relationship between two variables has the same variability for all values of *X*. A distribution which does have this characteristic is called *homoskedastic*, one that does not is called *heteroskedastic*. Some statistical tests require homoskedasticity. The linearly based relationship looks homoskedastic, whereas the others do not.

4.3.2 Guidelines

The task of detecting trends in scatterplots can be enhanced by adding lines of various kinds, These lines, which we call guidelines, since they help guide our interpretation of the plots, include the principal axis line, two different kinds of regression lines (one linear and the other monotonic), and two families of smooth lines (Lowess smoothers and kernal density smoothers).

Principal axis. The principal axis line (left panel of Figure 4.6) shows us the linear trend between the *X* and *Y* variables. It is the "longest" direction in the plane: It is the (straight) line such that when the points are projected onto it orthogonally, their projected values have the maximum possible variance.

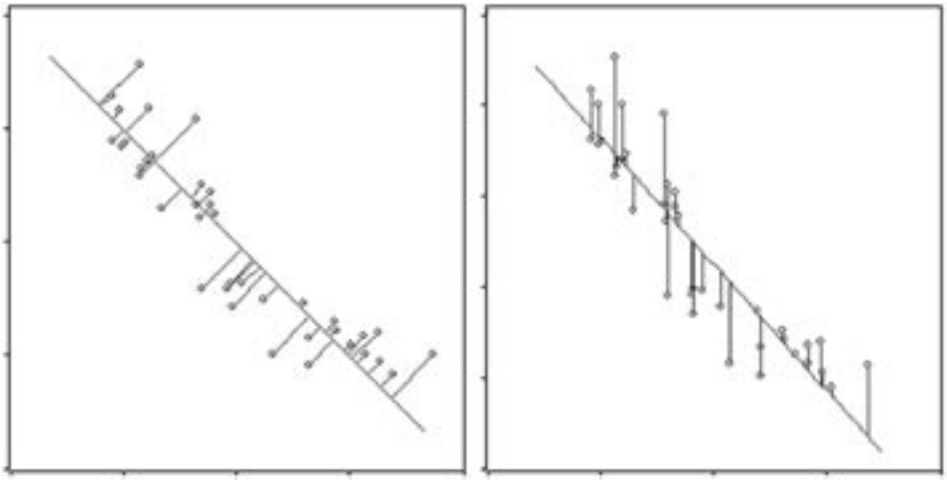

Figure 4.6 Left: principal axis line. Right: regression line.
Each line is shown with its residual lines (the short lines).

There is no other direction in the space which when the points are projected orthogonally onto the line for that direction, would have a greater variance. The principal axis line is also the line that has the shortest possible set of residual lines (these are the short lines drawn at right angles —orthogonal to—the principal axis). There is no other direction through the space that has a smaller sum of squared lengths of the residual lines (i.e., it is the best fitting line). It is the line that is the best linear summary of the *XY* plane.

Regression Lines: Regression lines provide information about how the *Y* variable responds to the *X* variable. These lines only make sense when we believe that the values on the *Y*-axis variable are a response that depends, at least in part, on the values of the *X*-axis variable.

Linear regression. The linear regression line (right panel of Figure 4.6) shows us the least squares linear prediction of the *Y* variable from the *X* variable. This is not the same line as the principal axis, except in unusual circumstances. The regression line maximizes fit to the dependent variable only, whereas the principal axis maximizes fit to both variables simultaneously. Whereas the principal axis measures fit orthogonally to the line (and thus uses information about lack of fit to both *X* and *Y*), the regression line measures fit vertically, only using information about lack of fit to *Y*. This difference is shown in the figure by the differently oriented residual lines: orthogonal to the principal axis in the one case, and vertical in the other case.

Quadratic regression. The quadratic regression line (lower-left panel of Figure 4.7) shows us the least squares quadratic prediction of the *Y* variable from the *X* variable. This regression is done by constructing a variable which is the squares of the *X* values and then fitting *Y* with a linear combination of *X* and X^2.

Cubic regression. The cubic regression line (lower-right panel of Figure 4.7) shows us the least squares cubic prediction of the *Y* variable from the *X* variable. This regression is done by constructing two variables, one of which is the squares of *X* and the other the cubes, and then fitting *Y* with a linear combination of *X*, X^2 and X^3.

Monotone regression. The monotonic regression line (upper-left panel of Figure 4.8). is the line that shows the order-preserving transformation of the *X* variable that has the maximum least squares fit to the *Y* variable. Monotone regression may degenerate into a *step function*, resembling a staircase. In such a case the function may be artifactual or may represent overfitting of the *X*-variable to the *Y*-variable.

Subset regression: Subset regression produces separate regression lines for subsets of the data. The subsets can be based on category variables or on any other groupings the user wishes to make. These groupings can be indicated by point color, point label, or by the name or shape of the object used to represent the point.

Smoothers. Smoothers, including Lowess smoothers and kernel density smoothers, provide us with an approximate idea about where the vertical center of the points is for each part of the scatterplot as we move horizontally from left to right along the *X*-axis. They are, loosely stated, a kind of running weighted regression, where a value of *Y* is computed for each value of *X* by using a weighted regression where nearby observations are most heavily weighted.

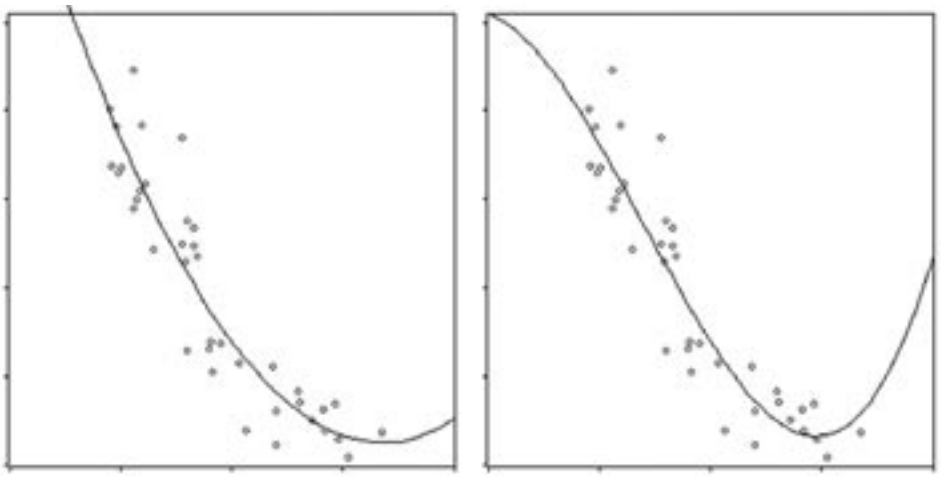

Figure 4.7 Left: quadratic regression line.
Right: cubic regression line.

There are two commonly used smoothers: Lowess smoothers and kernel density smoothers. Each has a parameter that must be manipulated to search for the specific member of the family that seems to be the best. Dynamic graphical techniques can be used for this search.

Lowess is an acronym for *LOcally Weighted regression Scatterplot Smoothing*, a method designed for adding smooth traces to scatterplots that works basically by selecting a strip of neighbors to each Y_i value and then using them to predict Y_i'. The process is iterative, so several passes are necessary to compute the values of the Lowess fit. There is a parameter for Lowess that you can control. It is the proportion of cases included in the strip. Narrow strips means that the smooth follow minor wiggles in the data. Wider strips provide a smooth trace that changes only gradually. The weight versus residuals plot used a strip of 0.75. In general,

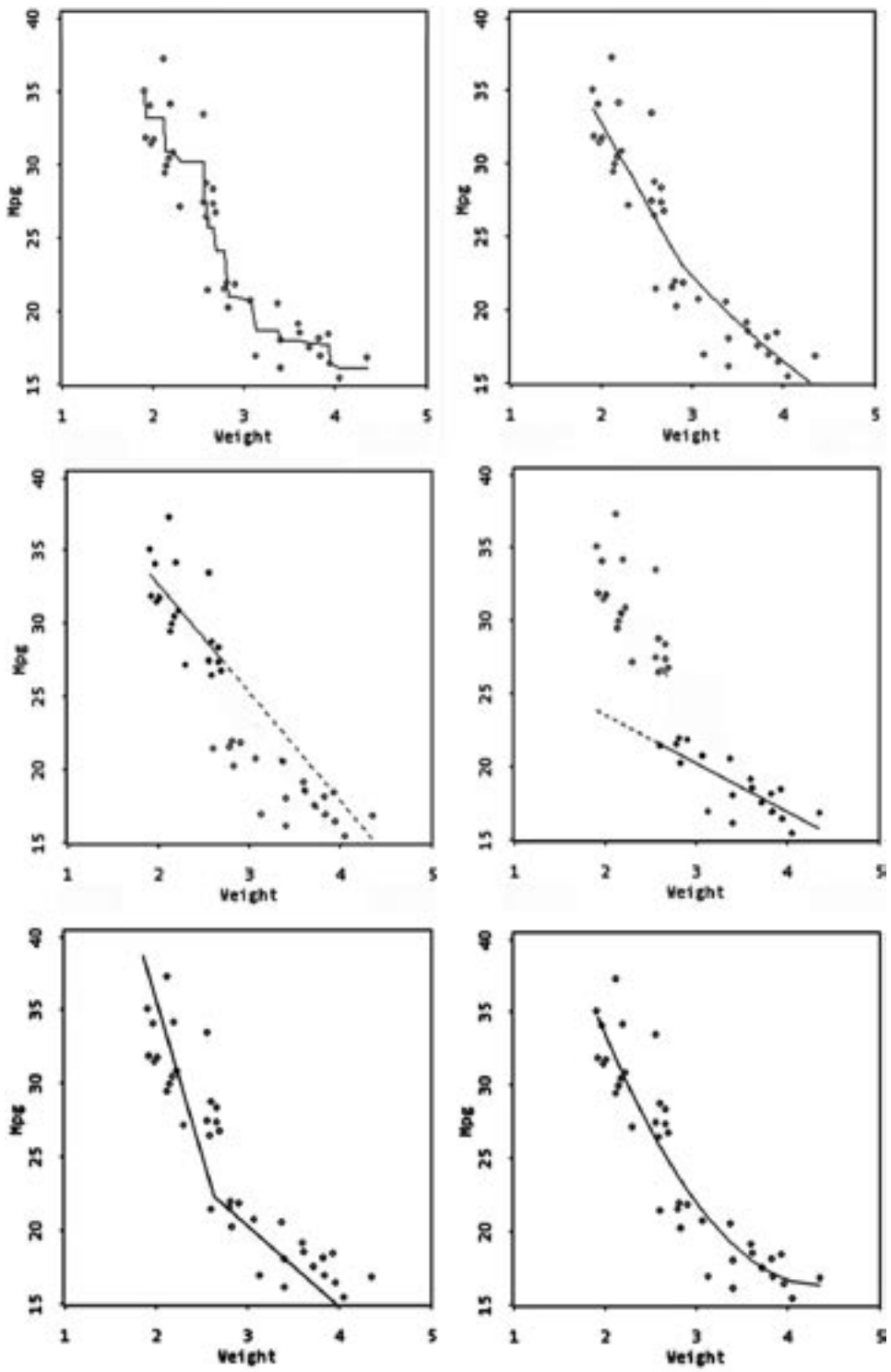

Figure 4.8 Five glimpses of what may be two different linear trends, plus one glimpse suggesting a single nonlinear trend.

what we try with smoothers is to balance smoothness versus goodness of fit. A smoother that follows every spike in the data is not what is normally desired.

A *kernel density smoother* is also a locally weighted regression method which employs a kernel function to generate weights for a series of weighted regression analyses, one performed for each observed value of X, each generating an estimate of Y for that value of X. The kernel function is usually a symmetric probability function such as a normal density function. Although other functions can be used, they generally produce similar results. As with Lowess, kernel density smoothers have a parameter that controls the smoothness of the resulting function. In this case, the parameter controls the width of the kernel function.

Black threads: In their introductory statistics book, Mosteller et al. (1983) argued in favor of fitting lines by eye, their *black thread method*. As they noted, when several points fall on a straight line, the line clearly fits them. Nevertheless, when the points do not fall along a straight line, we may wish to summarize their positions by a straight line. The problem is how to choose the line. Of course, we can use the methods described above, but we should also be able to draw lines by eye. As they note "drawing a straight line 'by eye' can often be as helpful as the formal methods." Their name for the method comes from the fact that they recommended stretching black threads across the plot to locate the best line until it passed the IOI (Interoccular Impact) test. This was, after all, before the advent of computers.

We recommend the availability of a "piecewise" black thread method, one that allows the user to locate connected pieces of black thread, each piece fitting the local points as well as possible, according to the eye. Such a line is shown in the lower-left corner of Figure 4.8.

4.4 Multivariate Data: Parallel plots

In this section we discuss a plots for multivariate data, the parallel-coordinates plot. This plot is based on a non-Cartesian multidimensional coordinate system that opts to represent the dimensions by axes that are mutually *parallel*. The material presented in this section owes its existence to the fundamental work on the parallel-coordinates representation of multivariate information by Inselberg (Inselberg, 1985; Wegman, 1990).

When a dynamic parallel-coordinates plot is brushed, the trace lines flash on and off, with the trace lines of the selected portions of the data being "on" and the trace lines for the unselected portions being "off." When a parallel-coordinates plot is brushed, only the trace lines for the observations selected are shown.

Medical diagnosis data. Of course, we do not usually know the structure ahead of time, and the structure is not usually as clear as that which we built into the data gauge. Thus, we return to the medical diagnosis data for a realistic example.

The parallel-coordinates plot of the principal components of the medical diagnosis data is shown in Figure 4.9. Note that principal components (as will be described shortly) create new variables by obtaining variance-maximizing linear combinations of the original data. Thus, in Figure 4.9 we see that the variance of each variable decreases as we move from the left portion of the figure to the right portion.

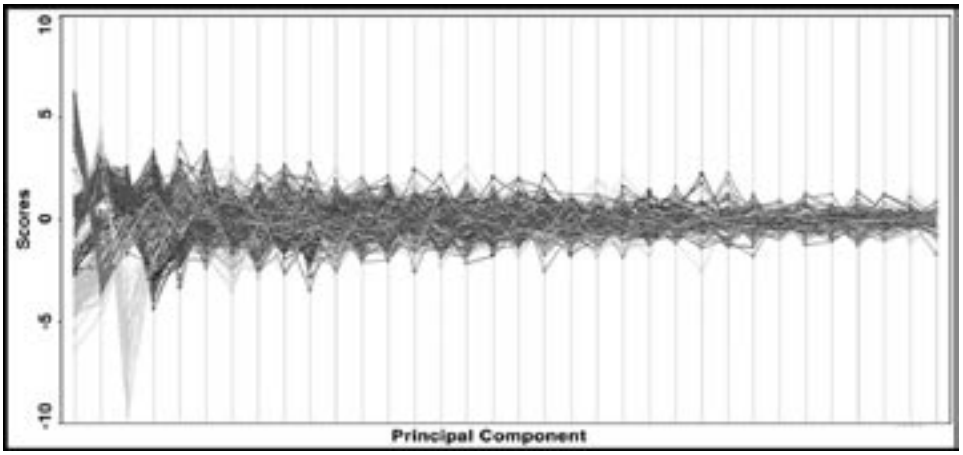

Figure 4.9 Parallel-coordinates plot of the medical diagnosis data.

Actually, Figure 4.9 demonstrates one of the main problems with the parallel-coordinates plot: Even with a fairly small number of observations, the lines overlap so much as to turn the plot into a jumble in which it is difficult to see any structure. This problem can be addressed by repeated application of the following three-step process:

(1) **Brush the plot.** Brush the parallel-coordinates plot in search of subsets of observations that show similar trace-line profiles.

(2) **Change color.** Once a coherent subset of observations is found, change the color of its members. This emphasizes the subset's structure.

(3) **Hide the subse**t. Hiding the subset reduces the clutter of lines so that you can more easily see any

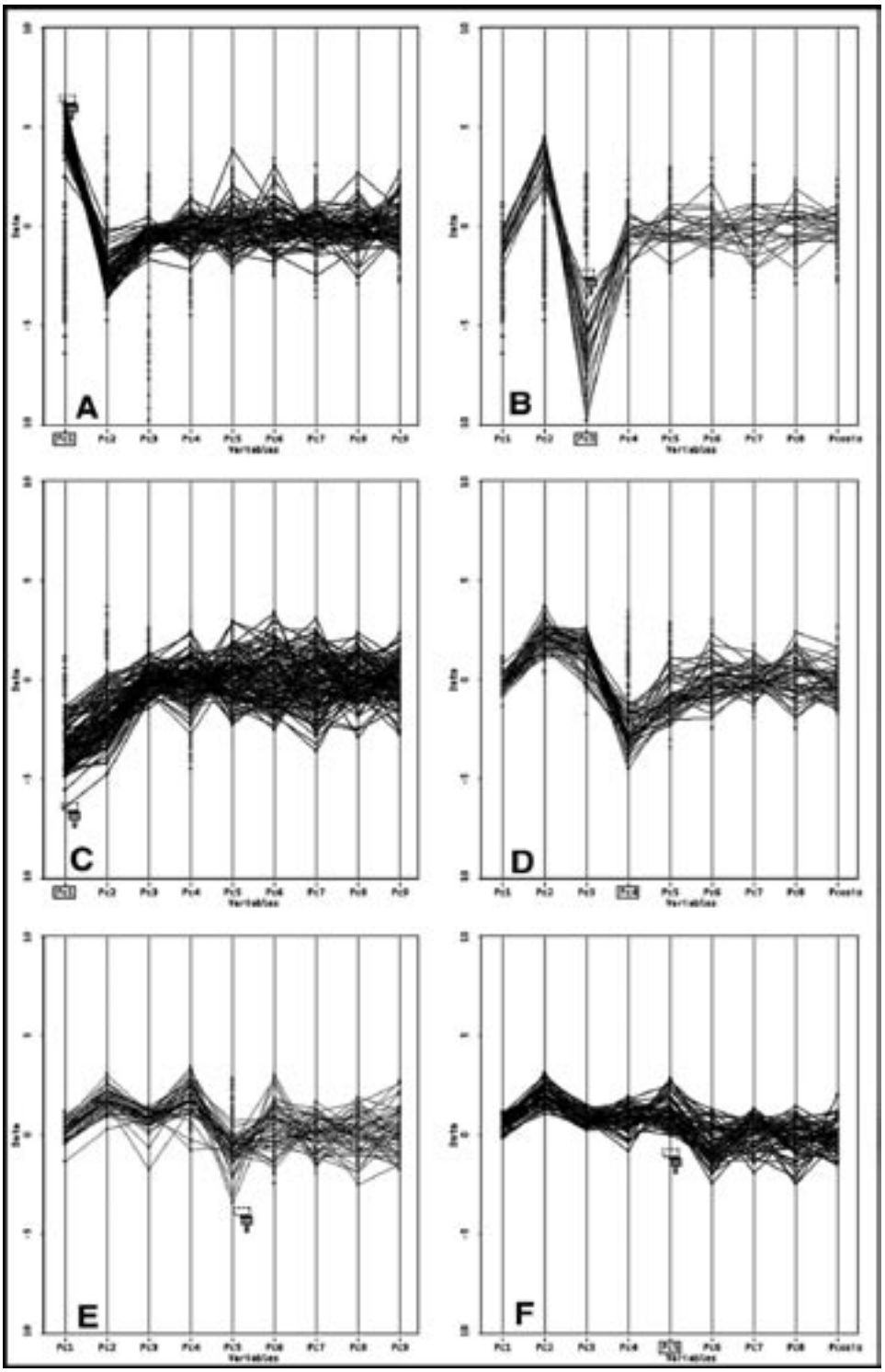

Figure 4.10 Parallel-coordinates plot showing six subsets of observations.

remaining structure. .

We keep cycling through these three steps until no more progress can be made. If we feel that we have hit on a good structure, we can save it; otherwise, we can start over.

We begin by paying particular attention to prominent trace-line features, such as the set of trace lines that are very positive on the first component and those that are very negative on the third component. Brushing each of these prominent features revealed that each feature was part of a pattern shared with numerous other observations, so we formed two subsets, one for each. Each of these can be seen in the top row of Figure 4.10. Once we had identified a subset, we changed the color of its trace lines and hid the subset. This process was repeated several times until we felt that it was complete, either because a good structure was found or because the process had reached a dead end.

We identified six clusters in five cycles of brushing, coloring, and hiding. These clusters are shown in each of the panels of Figure 4.10, which are arranged and alphabetized according to the order in which there were identified: Subset A was identified, colored, and hidden first; subset B was second; and so forth. It certainly would not have been possible to identify subsets E and F without having identified and hidden the first four. The first five principal components were used to identify the six clusters of data.

Chapter 5 Missing values in Data

5.1 Introduction

The function of REM (rapid eye movement) sleep is one of the great mysteries in the field of sleep and wakefulness. It seems that REM sleep must have an important function because almost all mammals have it. One piece of evidence that has been used to investigate its function is the correlation of characteristics of animals with the amount of their REM sleep. Variables that have been found to be correlated with REM sleep in mammals are: measures of the amount of non-REM sleep, safe sleep conditions, and inmaturity at birth (Siegel, 1995). We will use a dataset collected by Allison and Cicchetti (1976) that has variables related to these measures. There are 61 cases in this dataset. The variables are described in Table 5.1.

This dataset has been used to illustrate the technique of multiple regression analysis, where any of the three first variables is taken as the response variable and all or a subset of the other five are used as predictor variables. These data have missing values in some of the variables, but the published analysis usually excludes missing cases. This will result in a loss of cases ranging from 19 (models that use *NonDreaming* as dependent variable and include *LifeSpan* and *Gestation*) to only three (models that use *TotalSleep* as the dependent variable and *BodyWeight* and *BrainWeight*) .

Table 5.1 Variables in the Sleep in Mammals Data[a]

Variables	Description	Number of Missing Values
NonDreaming	Amount of non-REM sleep (hours/day)	13
Dreaming	Amount of REM sleep	11
TotalSleep	Total of sleep (dreaming + nondreaming)	3
BodyWeight	Logarithm of body weight (g)	0
BrainWeight	Logarithm of brain weight (g)	0
LifeSpan	Logarithm of maximun life span (years)	4
Gestation	Logarithm of the gestation time (days)	4
Danger	0 = least danger; 1 = most danger (from other animals)	0

[a] Note that TotalSleep = Dreaming+NonDreaming. Thus, it is always possible to compute the value of an observation for any of these three variables if the value of the other two is known.

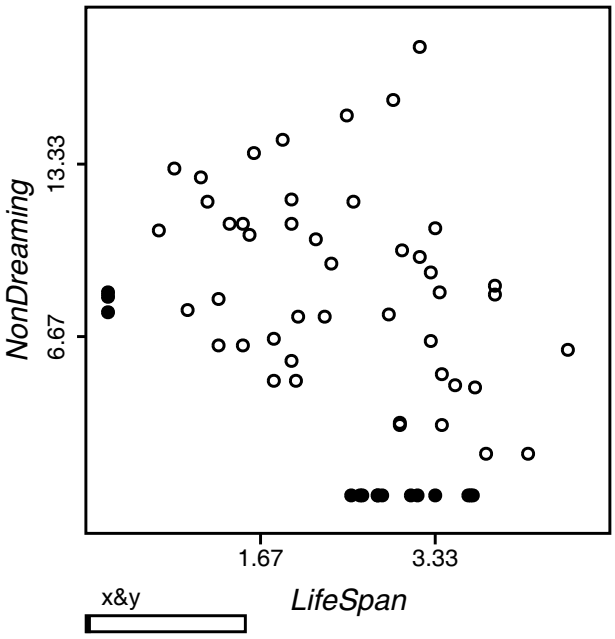

Figure 5.1 Scatterplot with missing values displayed on each axis as filled points and in a separate bar.

5.2 Missing Data Visualization Tools

Conventional graphical techniques often simply ignore missing values. This is unfortunate, as the analyst is left without the opportunity of checking a part of the data that may be informative. In the special case of interactive plots, the consequences of this approach are even more important, as many of the basic techniques may not work properly.

Consider linking. When several linked plots display different variables, each variable having its own unique pattern of missing values, an observation selected in one of the plots will not necessarily link to the same observation in another plot. The problem stems from the lack of response in some of the plots, which have a missing value in some of the variables displayed, but not in others. Also, other interactive actions, such as selecting the missing values in a given variable and seeing their values observed in other variables, are simply not possible.

One solution to these problems is used by the Manet (Missings Are Now Equally Treated) program (Unwin et al., 1996). This program incorporates modifications to basic plots and introduces new plots that can use missing values, thereby ensuring that missing values will always be represented. The specific way that each plot represents the values depends on the characteristics of that type of plot.

Figure 5.1 shows a scatterplot inspired by Manet's proposals. Two variables from the mammals data are displayed in this figure: *LifeSpan* and *NonDreaming*. values observed are represented by circles. Missing values are represented by filled circles. Cases that are missing in both variables simultaneously are displayed in the plot by a separate white rectangle that indicates the proportion of cases with regard to the total. This plot helps us see that missing values for the variable *NonDreaming* happen more often for large values of the variable *LifeSpan*. On the other hand, missing values for the variable *LifeSpan* are located at about the center of the distribution for the variable *NonDreaming*. Finally, we also see that the proportion of missing values occurring simultaneously in both variables is very low because the missing bar chart placed in the lower part of the plot is predominantly white. Manet does not include other plots for numerical data such as 3D plots or scatterplot matrices, but it seems possible to extend the same ideas to these other displays quite easily.

Compared with the traditional way of treating missing values in many statistical systems, Manet's approach is unquestionably sensible. However, using imputation techniques, as we discuss in the following section, we can further improve the plots. Imputation provides us with reasonable values that are com-

puted taking advantage of the observed part of the data. This way, the missing values can be displayed almost as if they were values observed.

5.3 Missing Data Patterns

A pattern of missing data indicates the variables in a group of observations that have missing or values observed. We will see in Section 5.4 that patterns of missing data can be a source of valuable information about data. Thus, for example, two observations in Section 5.2 have values observed in a variable—*Dreaming*—that seem more difficult to record than the variables that had not actually been observed (*TotalSleep* and *NonDreaming*). This strange pattern is eventually explained by the fact that two mammals, Giraffes and Okapis, are able to sleep while they stand up, but they lay down when in REM sleep. Hence, there are more evident cues associated with the state of *Dreaming* than for *TotalSleep,* as it can be difficult to know whether or not an animal is actually sleeping when it stands up.

Exploration of the patterns of missing values in our data may inform about:

- The patterns themselves and the number of cases in each of them.
- The mechanisms that may have produced the missing data.
- The summaries of the values observed given the missing data patterns.

Notice that the number of different patterns of missing data can be large for multivariate datasets and that it is convenient to visualize the observed and imputed part of the data simultaneously. Therefore, to obtain a complete account of the missing data patterns, the analyst can make good use of special tools tuned to this problem. In the following subsections we review some visualizations that have been designed specifically for the exploration of missing data patterns.

5.3.1 Patterns and Number of Cases

The first thing to do with missing data patterns is to list them. Also, the count of cases in each of them is important. A bar chart or a table can be used for this purpose. Table 5.2 shows the patterns for the mammals data. The table has been sorted according to the number of cases in the patterns. Thus, it is posible to see that the pattern for *Complete Data* (no missing values) is the largest in our data, followed by the pattern with missing values in the variables *NonDreaming* and *Dreaming*. Other patterns with fewer cases are listed below.

Table 5.2 Patterns of missing data in the Mammals Data[a]

NonDr	Dream	TotalS	Body	Brain	LifeSp	Gestati	Dange	Count
Y	Y	Y	Y	Y	Y	Y	Y	42
Y	Y	•	•	•	•	•	•	9
•	•	•	•	•	•	Y	•	3
•	•	•	•	•	Y	•	•	2
Y	•	Y	•	•	•	•	•	2
Y	Y	•	•	•	Y	•	•	1
•	•	•	•	•	Y	Y	•	1
Y	Y	Y	•	•	•	•	•	1

[a](Y=observed; •=missing)

As patterns with a high number of missing values are probably affecting more the estimations of parametters than those with fewer, it is important to check if the means, variances, and so on, of the observations in those patterns differ much from the observations in other patterns (especially from the *Complete Data* pattern). Therefore, an important use of Table 5.2 is to identify patterns with many cases.

However, the possibilities of Table 5.2 for data analysis are not limited to show patterns that are large. Thus, we can see in it that there is a mammal so mysterious that researchers have not been able to determine its *LifeSpan* or its *Gestation* time (Desert Hedgehog) and also that there is a mammal whose sleep behaviour is completely unknown (Kangaroo). Therefore, Table 5.2 is a source of valuable insight for data analysis, especially if linked to plots and labels of the observations in the missing data patterns.

Table 5.2 can be re-sorted in other ways, so it is possible to put together patterns with similar variables. Interactive capabilities such as displacing rows manually, or sorting according to one or several variables, make possible explorations of this type.

The mammals data have only eight different patterns of missing data, so their study is easy using a table. However, datasets with many variables may have many more patterns of missing data. Also, to know the count of cases in each pattern is normally only the first step to be undertaken in the exploration of this aspect of the data. Thus, statistics such as means, variances, and covariances displayed by pattern are usually required also. We will see in the following sections visualizations especially tuned to this problem, but first we discuss some preliminary theory that will be of help in that endeavor.

5.3.2 The Mechanisms Leading to Missing Data

One of the main uses of the analysis of the patterns of missing data is to shed light on the mechanisms producing the missingness. Knowledge of those mechanisms is of considerable interest because it may guide the strategy to follow in the analysis of data. In this section we first describe a widely accepted classification of the mechanisms of missing data, and thereafter, a statistical test that has been proposed for checking the mechanisms that underlie the data at hand. As patterns of missing values play a central role in this test, it is interesting to provide visualizations that provide insight about the various factors that take part in the computation of such single summary value. These visualizations are the goal of the Section 5.3.3.

The mechanisms that may have produced the missing data may be classified as (Dempster et al., 1977; Little and Rubin, 1987):

- *Missing completely at random* (MCAR). This mechanism implies that the missing values are a random subsample of the data. This case happens when the probability of response is independent of the values of the variables. Therefore, summary statistics like means or variances should be the same for the data observed and for the non-observed. Likewise, the matrix of variances-covariances for the whole data-matrix should be equal for the complete and for the incomplete data.

- *Missing at Random* (MAR). This MAR mechanism is less restrictive that MCAR. It assumes that the probability of response on a variable Y with missing values may be related to other variables X, but not with itself. Despite its name, MAR does not mean that missing values are a simple random subsample of all values. MAR is less restrictive than MCAR because it requires only that the missing values behave like a random sample of all values within subclassess defined by data observed (Schafer, 1997). In this case, incomplete data have different summary statistics than those of complete data but it is assumed that the data observed possess enough information to recover the lost information, helping to estimate the correct summary statistics. MAR and MCAR are said to be *ignorable* missing data mechanisms.

- *Non Missing at Random* (NMAR). Sometimes, researchers will have additional information on the data that lead them to be suspicious of data being MAR or MCAR. In this case, it is supposed that there is a nonmeasured variable that could be related to the missing values. This is called a *nonignorable missing data mechanism*. This situation is particularly problematic because there is no test that that can be used to evaluate whether or not the missing data mechanism is random.

Most of the literature concerning missing data is based on the ignorable mechanisms of MAR or MCAR. However, it is important to mention that *ignorability* can be regarded as relative. Although a missing data mechanism may not be known by the researcher in full, there can be variables that can explain the *missingness* to a lesser or greater extent. Including a number of the variables in the analysis that explain the missingness will make the assumption of MAR much more plausible (Graham et al., 1996; Schafer, 1997).

Although there are no tests for NMAR, it is possible to check if data are MCAR versus the less restrictive situation of MAR.

One of the most intuitive analyses for MCAR is computing the differences in variables between observed and missing values for the rest of variables. Tests for the differences of means are also sometimes reported. Significant differences for the tests can be taken as evidence that data are not MCAR. However, even though this procedure can be considered informative, it produces a large number of *t*-tests that are difficult to summarize, because they are correlated with a complex structure, depending on the patterns of missing data and the correlation matrix (Little, 1988).

The previous strategy could also be applied to visualizations of data. Plots could be constructed for all the variables for the missing and nonmissing parts of each variable. However, due to the presence of missing values in the variable displayed, the size of the groups would not always be homogeneous. Also, exploring bivariate relationships among variables for missing and nonmissing values of other variables would be more complex, as the plots for different variables might vary widely in the observations displayed.

A test that evaluates if the mechanism of missing values in a dataset is MCAR or MAR was suggested by Little (1988). Little's test evaluates the differences between the means observed for patterns of missing data and the maximum likelihood estimations obtained using the EM algorithm and has the following formula:

$$d^2 = \sum_{j=1}^{J} d_J^2 = \sum_{j=1}^{J} m_j (\overline{y}_{\text{obs},j} - \hat{\mu}_{\text{obs},j}) \hat{\Sigma}_{\text{obs},j}^{-1} (\overline{y}_{\text{obs},j} - \hat{\mu}_{\text{obs},j})^{\text{T}} \tag{3.3}$$

where there are p variables, J patterns of missing values, m_j observations in pattern j, m and Σ are the maximum likelihood estimates of the vector of means and the matrix of covariances obtained using the EM algorithm, and obs, j are the subsets of the parameters corresponding to nonmissing observations for pattern j. Finally, $\overline{y}_{\text{obs},j}$ is the p-dimensional vector for the sample average of data observed in pattern j.

This test has a χ^2 distribution with degrees of freedom equal to

$$\sum_{j=1}^{J} p_j - p$$

where p_j is the number of variables observed in pattern j. Notice that the test above can be regarded as a sum of normalized Mahalanobis distances of each pattern with respect to the maximum likelihood means. Examination of individual contributions to the test can be used to find out which patterns contributed most to the test. However, straightforward comparisons of the sizes of the terms can be misleading, and corrections are necessary to explore these contributions (Hesterberg, 1999).

The test assumes that the matrix of covariances is the same for all patterns. Little (1988) and Kim and Bentler (2002) provide tests for homogeneous means and covariances. However, this test is quite limited, due to patterns with few cases and is not reported os often as the test for means.

Although Little's test offers a convenient summary of the deviations of the parameters from the maximum likelihood estimators, it is always interesting to check the patterns individually. For example, in the mammals data, Little's test for means returns 39.80 with 42 degrees of freedom (p = 0.57) so we woud not reject the MCAR hypothesis. However, looking at Table 5.2, we can see that the variable *LifeSpan* is involved in several patterns with few cases each. But as examination of equation (3.3) makes clear, patterns with few cases can not contribute much to increasing the value of the test, and also, the existence of more patterns increases the degrees of freedom of the test. Therefore, if the variable *LifeSpan* is excluded from the test, we may expect that the MCAR hypothesis will be rejected more easily. Indeed, the output of Little´s test after excluding this variable is 30.54 with 20 degrees of freedom (p = 0.06), which is marginally significant. This result alerts us to the dangers of accepting the results of the test without examining the contributions of the individual patterns.

5.3.3 Visualizing Dynamically the Patterns of Missing Data

We saw in section Section 5.3.2 that the MCAR mechanism involves homogeneity of means and covariances and that a statistical test exists for checking this assumption. However, visualizations of the components of these formula may provide insight into the characteristics of each pattern that would otherwise remain hidden.

In this section we discuss visualizations that address, on the one hand, means and variances, and on the other hand, covariances and correlations of the data. DIG techniques are useful for these visualizations because they allow the user to explore step by step the information in the patterns and focus on those that present more interesting features. The first visualization to be described uses raw data and does not involve imputation. The other two are focused on summaries of the data computed by patterns, such that the summaries can be compared easily among themselves and with the maximum likelihood estimates of the statistics. These two plots are the most closely related to Little's test and hence can be a useful companion to it.

Parallel diamond plots for observations in missing data patterns. Figure 5.2 a shows diamond plots for the observations in two of the missing data patterns of the mammals data. The plots are made in the following way: First, the mean and the standard deviation of each of the variables is computed using the available observations. Second, the variables are standardized using these means and standard deviations to have mean 0 and standard deviation 1. Third, observations in each of the pattern are selected and plotted separately. Fourth, diamond plots for the means and standard deviations of the observations in the pattern are superimposed over the observations in the pattern.

Figure 5.2a shows the observations in the *Complete Data* pattern (no missing values in any of the variables). Figure 5.2b shows the observations in the pattern with variables *Dreaming* and *NonDreaming* missing. The plots can be interpreted as showing if the observations for each variable in a pattern are different from the remainder of the observations. If the observations for a variable in a given pattern are not different, the points will have mean zero and unit standard deviation, and consequently, the diamond for these observations will be centered in the zero line and will have a height of 2. On the contrary, if the observations are different, the diamonds will be uncentered in the zero line or will not have a height of 2. This plot is meant to be used interactively, by selecting one pattern after the other from a list to see the changes in the plot.

Figure 5.2 top shows that the means and the standard deviations for the variables in the *Complete Data* pattern are similar to those for all the available cases. On the other hand, Figure 5.2 bottom, reveals that

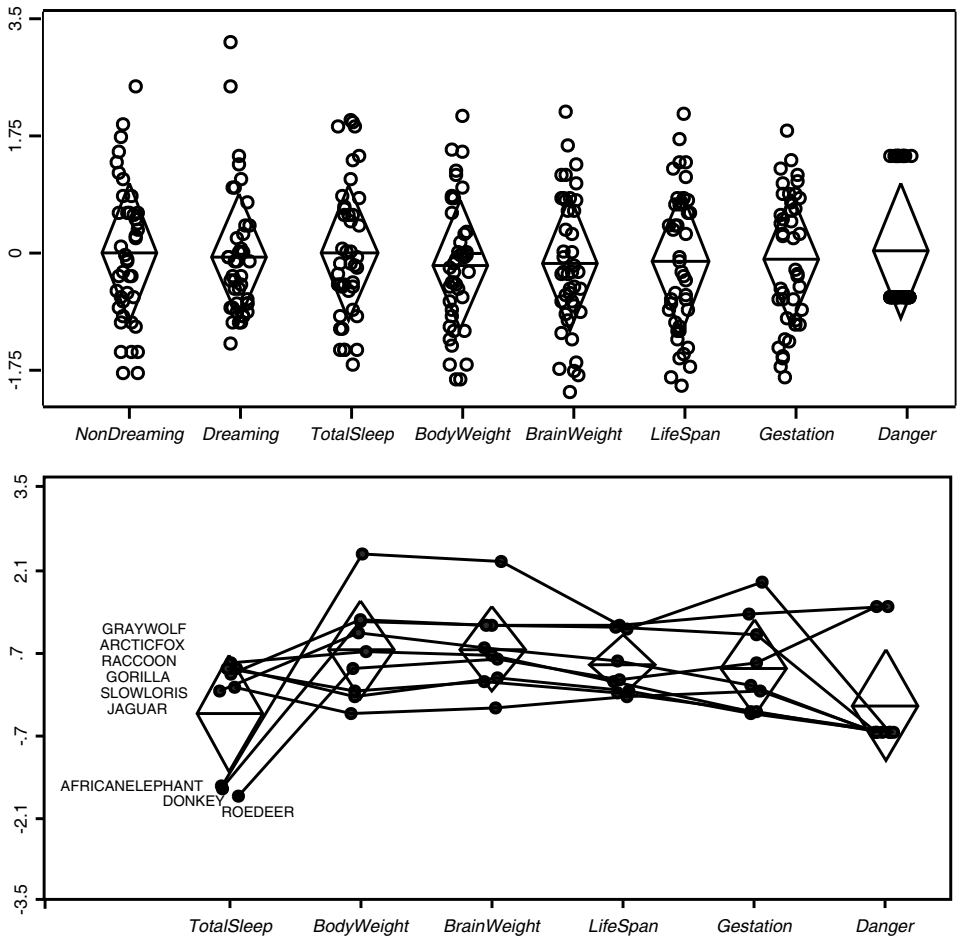

Figure 5.2 Diamonds for observations in the *Complete Data* (top) and *Dreaming/Non Dreaming* (bottom) missing data patterns.

the same statistics for the pattern of missing values *NonDreaming/Dreaming* differ from those for the available cases. The differences are that the mean of the observations for this pattern is lower for the variables *TotalSleep* and *Danger* and higher for the rest of variables than the means for the available data.

The plots discussed here have the appeal of being based on raw data, without resorting to any special procedure or computation.

In the next section we review plots that follow a different strategy. Using imputation to fill the gaps, all the variables and all the patterns are displayed in the plot. This results in a very complete representation of the data that is amenable to more sophisticated exploratory data analysis.

Scatterplot matrices for data with missing values. In Section 5.3.2 we discussed visualizations intended to explore the homogeneity of means and variances by patterns of missing data. In this section we explore visualizations of covariances (or relationships) among variables.

Little (1988) expressed several concerns about the power of the version of the MCAR test if used for checking the homogeneity of both means and covariances among patterns of missing data. In particular, he pointed out that patterns with fewer cases than observed variables cannot be used for this purpose, and that the test is probably too sensitive to departures from the normality assumption to be of practical use. As a consequence, this test is probably not used as often as the test for homogeneity of means. Nevertheless, it seems interesting, from an exploratory point of view, to have the capability of checking the relationships among variables of the observations falling into patterns of missing data.

Figure 5.3 shows a scatterplot of the variables *Gestation* and *LifeSpan* for the mammals data. This scatterplot includes the imputations obtained from the EM algorithm using the symbols described in Section 5.4.1. This scatterplot could be seen as being part of a scatterplot matrix for all the variables in the dataset.

The scatterplot in Figure 5.3 differs from the scatterplot in Figure 5.6 in that it displays a regression line indicating the relationship among the observations *by patterns of missing values*. The solid lines in the plot refer to the regression lines estimated using the covariances computed with the EM algorithm; the dashed lines represents the regression line computed using only the subset of values in the pattern of missing values currently selected (in this case, the pattern is *NonDreaming* and *Dreaming*). An examination of Figure 5.3 shows that, in general, the line for the pattern selected does not differ too much from the maximum likelihood line. Using a scatterplot matrix of all the pairs of variables would allow us to observe whether the MCAR assumption for covariances holds in general for this pattern.

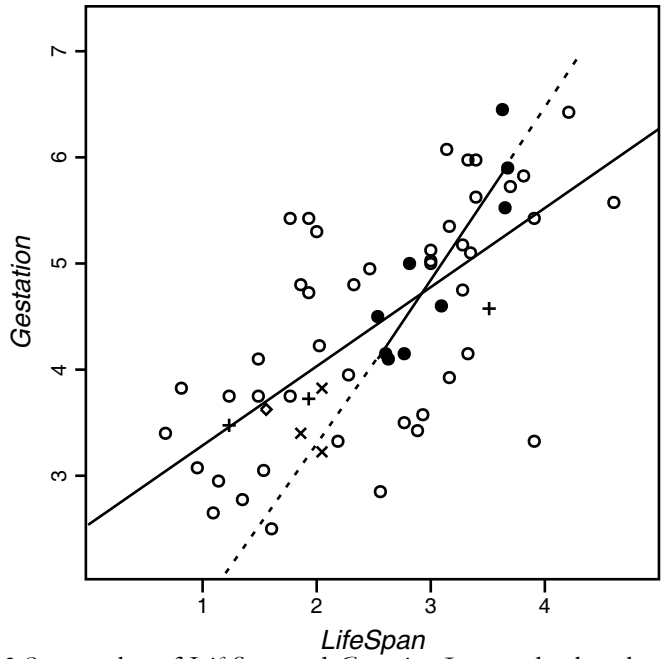

Figure 5.3 Scatterplot of *LifeSpan* and *Gestation*. Imputed values have the following symbols: *LifeSpan* (¥), *Gestation* (+) and both (‡). Points shown with a filled circle are in the missing data pattern *NonDreaming–Dreaming* and have been used to compute the regression line with dashed parts. The other regression line has been computed using the parameters estimated using EM.

5.4 Visualizing Imputed Values

Ad hoc methods for imputation of missing data have been around for many years. However, over the last decade theoretically sound statistical methods for imputing missing values have become available. Furthermore, these methods offer promising approaches for adapting visualization methods to the realities of missing data.

There are a number of issues involved in using imputation methods for statistical visualization:

- Imputed data must be marked uniquely to differentiate them from observed data, and the resulting visualizations must include this marking in order to reflect the special status of the values (Swayne and Buja, 1998) .
- Although there are several imputation methods, not all of them can be recommended. In particular, it is well known that the "quick and dirty methods" cannot be recommended because they are simplistic methods that distort the data. These methods include replacing missing values with unconditional means or removing cases with missing values. These difficulties apply whether the data are imputed for inference or for visualization. Figure 5.4 and the discussion associated with it is an example of the danger of using inappropriate imputation metods.
- Even though good imputations will be less harmful than bad imputations, there is still the issue that the values obtained do not include any indication of the uncertainty of their estimates. Multiple imputation (Rubin, 1987; Schafer, 1997) is a technique that improves over single imputation because it provides information about the variability of imputations. Plots that include this information are of interest because the analysts can obtain insight about the variability associated with the imputed observations.

We discuss these issues in the remainder of this chapter. We use the sleep in mammals data as an example.

5.4.1 Marking the Imputed Values

Before we start discussing methods for imputing reasonable values, we will introduce the way of marking the values in the plots so we do not forget that they are not observed values.

Imputed values can be considered to be reasonable approximations of the unvalues observed. However, it is important not to confuse them with observed values. Therefore, we include in the plot reminders that they are not actually observed and that they should be interpreted with care. The way to remind us of this fact usually consists of marking the imputed values differently from the rest of the values in the plots. This way, the analyst can, on the one hand, obtain general overviews of all the data, imputed and not imputed, and, on the other hand, appreciate the areas where the imputed values tend to behave in an interesting way. Thus, in this section we discuss some ways in that the imputed values can be marked in plots.

The strategies for marking the imputed values in plots generally will differ depending on the types of plots considered. In this section we focus on the plots that use elements for representing each individual observation, such as dotplots, scatterplots, and so on. Other types of plots may require different strategies than used for these plots, but they are not considered here for reasons of space.

Plots based on points can be modified in three ways to indicate that some of them do not represent values observed but that they have been imputed. The modifications consist in using different symbols, colors, and/or sizes. Assuming that we have started with a multivariate data matrix where some of the variables had missing values that have been imputed, these modifications will apply to different plots in the following way:

(1) *Dotplots/boxplots/parallel plots.* In this case the imputed values can be marked with a different feature, such as a different color/symbol/size. The same feature can be used in several plots. Histograms that have individual bricks for each observation can use different colors for displaying imputed values.

(2) *Scatterplots.* In bivariate data, missing values can occur in any of the variables separately and also in both simultaneously. Figure 5.4 shows a scatterplot of the variables *LifeSpan* and time *NonDreaming.*

The method used to impute values is the unconditional means method discussed in section Section 5.4.2. Observed points are displayed as hollow circles. Missing values in the variable *LifeSpan* are displayed with an ¥, and missing values in the variable *NonDreaming* use a +. Finally, observations with missing values in both variables are displayed using a ‡. This plot is in many aspects very similar to the plot in Figure 5.1 but with the difference that it does need any special addendum for missing values in both variables.

(3) *Spinplots*. The same strategy as that used for scatterplots can be applied to spinplots of three variables. However, if all combinations are to be displayed in the plot, eight (rather than four) symbols or colors are needed to represent all types of missing values. In practice, the number of patterns of missing values is probably not going to be so large, so the number of symbols/colors would be lower.

(4) *Scatterplot matrices*. Using different symbols in scatterplot matrices for each pattern of missing values generates too many symbols. A more convenient approach consists of treating each scatterplot individually as described before. For example, we would know that a point marked with an ¥ in one of the scatterplots would correspond to an observation that has a missing value in the variable displayed on the row of the scatterplot matrix in which this scatterplot is located. Notice that this observation would also be marked on the scatterplots along that row with an ¥, except if the variable on the vertical axis is also missing, in which case it would be marked with a ‡ (see Figure 5.6). This marking allows identifying the other variables in which an observation would be missing simultaneously. Interactively, this would be accomplished by selecting an observation and looking along the corresponding row or column of the scatterplot matrix.

Marking the values in the plots is important to avoid the pitfall of interpreting an imputed value as an observed value, as we shall see in the next section.

Even though it is convenient to emphasize that imputed values are not *real* values, we expect them to be credible values, so that we will be able to use them for obtaining insight about our data. However, not all the imputation methods are equally appropriate for producing reasonable values. In the following two sections we review methods appropriate for this endeavor.

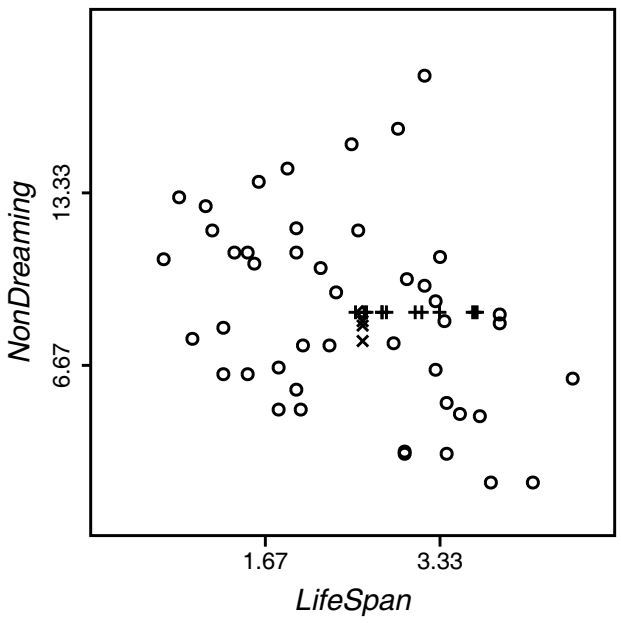

Figure 5.4 Scatterplot with imputation of unconditional means. Imputed values for variables are marked with the symbols: *LifeSpan* (¥), *NonDreaming* (+), and both (‡).

5.4.2 Single Imputation

In this section, we introduce single imputation methods and discuss its application to the sleep in mammals example. The first method of imputation that we explain, *single imputation* (Schafer, 1999), estimates a single value for each missing value. Single mputation is in contrast with *multiple imputation*, a method whereby, using simulation, each missing value receives several imputed values. Multiple imputation has the advantage over single imputation that uncertainty about the missing values can be introduced in the statistical model such that intervals of confidence around the estimations can be computed. On the other hand, single imputation requires less effort and can be also acceptable when the number of missing values is small. Also, as multiple imputation methods can be understood as extensions of single imputation methods, it is useful to review them before discussing the others. The methods are:

(1) *Unconditional means*. This method consists of substituting the mean of the variable for the missing values. This method is available in some statistical packages even though it presents considerable disadvantages and should, in general, be avoided. The reason is that this method inflates the number of values in the center of the distribution and underestimates the variance of the variable (see Little and Rubin, 1987, p. 44). The covariances among the variables are also underestimated. From the point of view of statistical visualization, this method has the disadvantage that the imputed values are laid out in lines parallel to the axis as shown in the Figure 5.4 on page 59 for the variables *LifeSpan* and *Dreaming*.

(2) *Conditional means*. With conditional means, the values used for the imputation are obtained by conditioning on the observed part of the incomplete cases. The method was proposed by Buck (1960, cited in Little and Rubin, 1987) and it consists basically of predicting each missing value using the values observed in a particular pattern of missing values. This method underestimates the variance of the completed data (but less than the unconditional means) and inflates the covariance of the variables. More serious problem is that the imputed data are predicted values without any component of error and give an overoptimistic sense of precision. This suggests inmediately the strategy of adding a suitable random component to the data that is function of the residual variance. Multiple imputation methods (discussed in the next section) can be considered as extending this strategy and combining it with the method in the following paragraph.

(3) *Estimation-maximization (EM) algorithm*. EM can be understood as an extension of the conditional means method. In EM, after the missing values have been substituted by a first round of imputed values, the parameters of the data are recomputed, and a new imputation is carried out. The same process is repeated until convergence. The advantage of the method is that the imputed values are not only a function of parameters of the observed part of the data, but also use the information contained in the nonobserved part *given* the observed part of the data. The EM algorithm is a general strategy apropriate for different types of data (Little and Rubin, 1987) but the one that has been implemented most often corresponds to incomplete multivariate normal samples. Notice that the estimation-maximization algorithm has as its main purpose the computation of maximum likelihood estimates for the parameters of the data (e.g., means, variances, covariances), and not of imputations of values. However, the parameters so computed can be used to compute estimations of missing values similar to the conditional means method, but the imputed values still lack a component of error.

We will show an example of single imputation using the EM algorithm applied to the mammals data. As mentioned in the preceding paragraph, the most common implemented versions of the EM method assume that the data follow approximately the normal multivariate distribution. A way to check that assumption approximately is the scatterplot matrix shown in Figure 5.5. Scatterplot matrices are limited to the univariate and bivariate distributions and do not allow direct inspection of more than two dimensions. However, this can be a reasonable approach to the problem in many cases.

Figure 5.5 shows that all the bivariate plots look well except for the dummy-coded categorical variable *Danger*. This would be a problem if this variable had missing values, because some of the imputed values would probably fall out of the 0 to 1 range. However, *Danger* does not have any missing values, and consequently, this problem does not arise here. Note that as indicated in Table 5.1, some of the variables have been transformed using logarithms in order to improve the symmetry and linearity of the data. Also, it is

important to mention that we assume that the mechanism of missing data is *ignorable*. (See Section 5.3.2 to learn more about assumptions with missing data.)

The scatterplot matrix in Figure 5.5 suggests that it is reasonable to use the EM algorithm to compute the maximum likelihood parameters for the data. Using these parameters, imputed values for the missing data use most of the information in the data observed. Also, once the data have been imputed, it is possible to visualize them using the marking strategies discussed in Section 5.4.1. In our case we examined several scatterplots for pairs of variables until we found one that seemed to us to be of special interest. This scatterplot is shown in Figure 5.6 and corresponds to the variables *LifeSpan* and *NonDreaming*. Note that the imputed values in Figure 5.6 do not lie at the means of the variables as they did in Figure 5.4, which used the unconditional means method. So, although the imputed values actually fall on fitted hyperplanes, they do not portray evidence of regularity when they are displayed in lower dimensionality views.

The special feature we saw in Figure 5.6 is the group of imputed points located at the lowest right side of the plot. These points are marked with a +, denoting that they were observed in the variable *LifeSpan* but not on *NonDreaming*. Selecting and labeling the points in this area provided some additional insight on these observations. So the labels of observations with missing values are the African Elephant (placed very close to the observed Asian Elephant), the Roedder, the Donkey, the Giraffe, and the Okapi. Also, there are five animals selected in Figure 5.6 that have no missing values for the variable *NonDreaming* (Asian Elephant, Goat, Sheep, Horse, and Cow). Interestingly, these mammals are all ruminants, a group of animals that generally have low *TotalSleep;* they are also in the *Danger* species and presumably share other features. So the parallel-boxplots plot in Figure 5.7 displays the profiles of the mammals selected previously along all the variables in the dataset. These profiles reveal that the mammals selected feature low sleep time (low values at *Dreaming*, *NonDreaming,* and *TotalSleep*), large *BodyWeight* and *BrainWeight*, and long *LifeSpan* and *Gestation* time. They also are species in *Danger* (except the African Elephant).

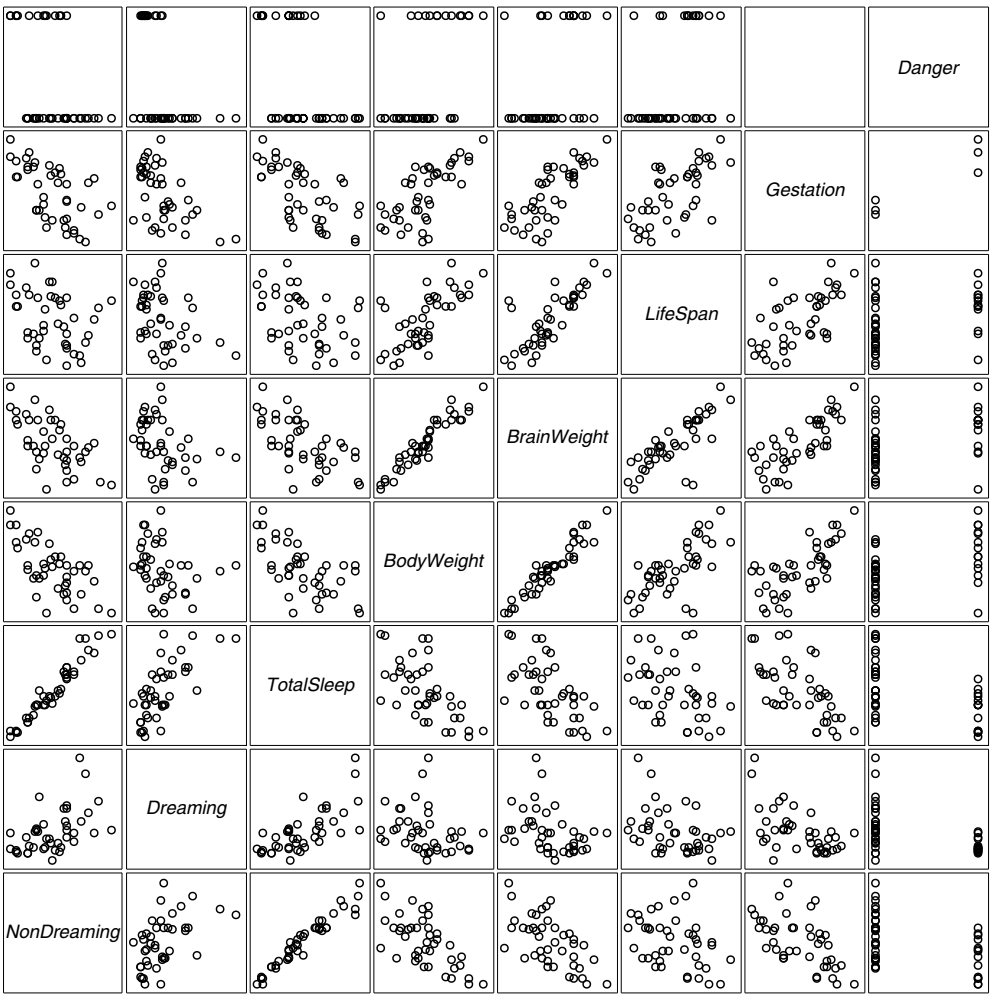

Figure 5.5 Scatterplot matrix for the sleep in mammals data.

The plot in Figure 5.7 can be used for exploring the missingness of the observations selected because they are marked according to the strategies for boxplots outlined in Section 5.4.1. However, this black-and-white printed, static figure suffers from overlapping, a problem that would be solved in the interactive version by selecting each of the observations individually. Also, colors on the computer screen stand out more clearly than do black-and-white symbols. Hence, to improve the effect of this plot, we have enlarged the size of the symbols (the Okapi has a symbol larger than the others).

Note that missing values in the first three variables (*Dreaming, NonDreaming* and *TotalSleep*) must happen in at least two of them at the same time. This is a consequence of the variable *TotalSleep,* being the sum of the variables *Dreaming* and *NonDreaming.* In Figure 5.7 we have selected observations that have missing values in any of these three variables. So the most common pattern of missing values among the selected observations is one in which the variable *TotalSleep* is known, but not its split into time *Dreaming* and *NonDreaming.* Two typical examples of observations in this pattern are the African Elephant and the Donkey. For these animals, it may be difficult to differentiate time *Dreaming* from time *NonDreaming.*

Two of observations selected in Figure 5.7 keep a pattern different from the typical pattern described in the preceding paragraph. The pattern consists of knowing the time *Dreaming* but not the *TotalSleep* or their time *NonDreaming*. Two mammals are in this situation: the Giraffe and the Okapi. This pattern is striking because it seems to us easier to record the variable *TotalSleep* than the variables *Dreaming or NonDreaming* separately. However, this does not seem to be true for Giraffes and Okapis. An explanation of this pattern is that Giraffes spend most of the night standing up, so it is difficult to know if they are sleeping, except in the short periods of time that they place their heads on the ground. This time seems to consist of REM sleep and so can be observed more easily than the total sleep and nondreaming times. Okapis the closest relative of giraffes, are also difficult to observe.

In summary, the plots in Figures 5.6 and 5.7 are useful because they offer a more complete picture of the dataset analyzed than do equivalent plots that exclude the missing values. The advantage stems from two sources:

(1) Cases with missing values are not completely removed from plots.

(2) Although the imputed values are only a reasonable approximation to the "true" values, and their interpretation must be done very cautiously, they can be of enormous help in understanding the missingness and its causes.

Single imputation is an interesting strategy for exploration of data with missing values. However, there is not information about the confidence that can be placed in the imputed values. Multiple imputation, not

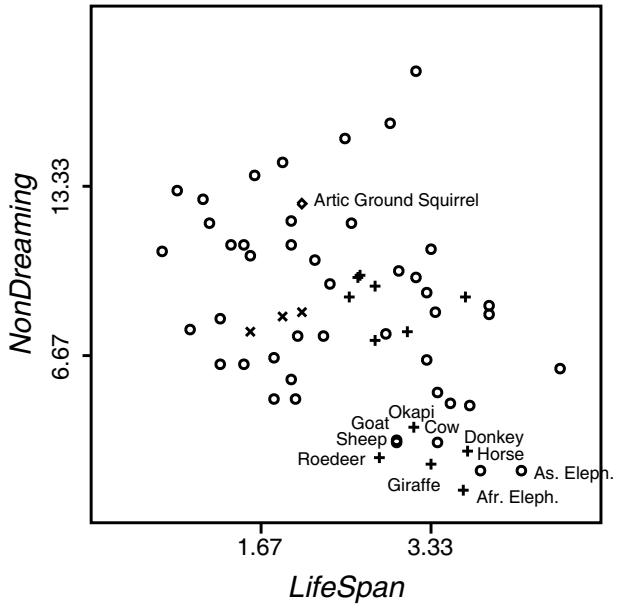

Figure 5.6 Scatterplot with values imputed using the estimation-maximization method.

described in this chapter, offer a better alternative to single imputation because allows computing the uncertainty stemming from imputation.

5.5 Conclusions

The main theme of this chapter is the use of imputation as a way of recovering the missing data and, more important, to illustrate interactive displays that include both imputed and values observed. This strategy requires marking the values imputed differently from the values observed. The visualizations so created can be very useful for the exploration of data. Thus, for the mammals data, a number of features have been

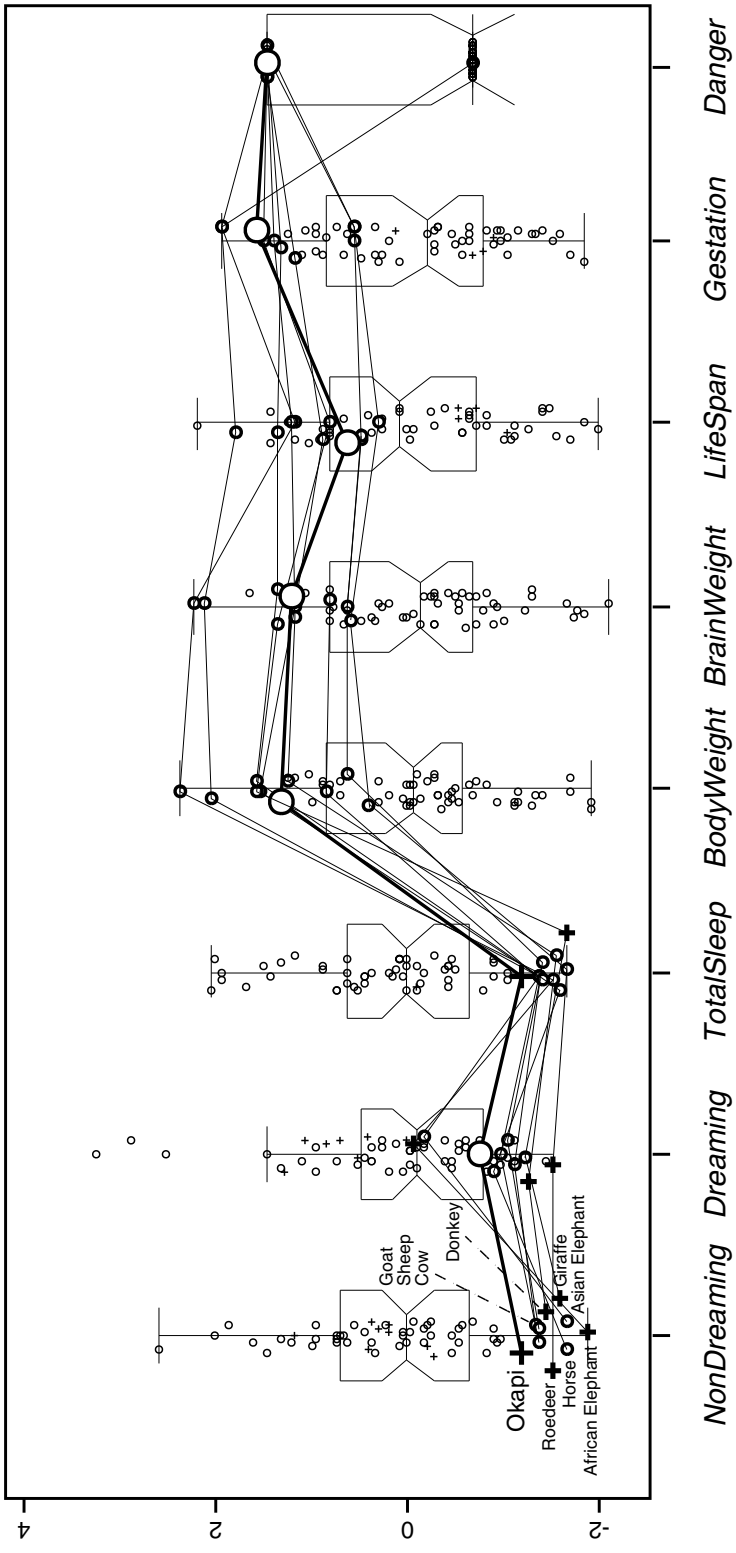

Figure 5.7 Paralell boxplots with values imputed using estimation maximization.

revealed that otherwise would have remained unobserved. These features have lead us to understand the singularities of our data in relation with the missing values. This happened, for example, with the two observations with missing values in the variable *TotalSleep* but values observed in the variable *Dreaming*. It may seem that knowing simply whether or not an animal is sleeping should be easier than detecting whether it is dreaming. However, for mammals that do not necessarily lie down when sleeping but do when dreaming, this seems to be true.

References

Agresti, A. (1990). *Categorical Data Analysis*. Wiley, New York.

Alba, R. D. (1987). Interpreting the parameters of log-linear models. *Sociological Methods and Research*, 16(1):45–77.

Allison, T. and Cicchetti, D. V. (1976). Sleep in mammals: ecological and constitutional correlates. *Science*, 194(12):732–734.

Andersen, E. B. (1996). *Introduction to the Statistical Analysis of Categorical Data*. Springer-Verlag, New York.

Asimov, D. (1985). The Grand Tour: a tool for viewing multidimensional data. *SIAM Journal on Scientific and Statistical Computing*, 6(1):128–143.

Ato, M. and Lopez, J. J. (1996). *Análisis estadísticos para datos categóricos*. Síntesis, Madrid.

Barnett, V. and Lewis, T. (1995). *Outliers in Statistical Data*. Wiley, New York.

Becker, R. (1994). A brief history of S. Bell Laboratories.

Becker, R. A. and Chambers, J. M. (1981). *S: A Language and System for Data Analysis*. AT&T Bell Laboratories, Murray Hill, NJ.

Becker, R. A. and Cleveland, W. S. (1987). Brushing scatterplots. *Technometrics*, 29:127–142.

Bertin, J. (1983). *Semiology of Graphics*. University of Wisconsin Press, Madison, WI. (English translation by W. Berg).

Betz, D. (1985). An XLISP tutorial. *Byte*, 10(3):211–236.

Bickel, P. J., Hammel, J. W., and O'Connell, J. W. (1975). Sex bias in graduate admissions: data from Berkeley. *Science*, 187:398–403.

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

Christensen, J., Marks, J., and Shieber, S. (1992). Labeling point features on maps and diagrams. Technical Report TR-25-92, Harvard University, Cambridge, MA.

Christensen, J., Marks, J., and Shieber, S. (1995). An empirical study of algorithms for point-feature label placement. *ACM Transactions on Graphics*, 14(3):203–232.

Christensen, R. (1990). *Log-Linear Models*. Springer-Verlag, New York.

Cleveland, W. S. (1994a). *The Elements of Graphing Data*, revised edition. Hobart Press, Summit, NJ.

Cleveland, W. S. (1994b). *Visualizing Data*. Hobart Press, Summit, NJ.

Cleveland, W. C. and McGill, M. E. (1988). *Dynamic Graphics for Statistics*. CRC Press, Boca Raton, FL.

Cook, R. D. and Weisberg, S. (1994). *An Introduction to Regression Graphics*. Wiley, New York.

Cook, R. D. and Weisberg, S. (1999). *Applied Regression Including Computing and Graphics*. Wiley, New York.

de Leeuw, J. (2005). On abandoning Xlisp-Stat. *Journal of Statistical Software*, 13(7):1–5.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–37.

Donoho, A. W., Donoho, D. L., and Gasko, M. (1988). MacSpin: dynamic graphics on a desktop computer. *IEEE Computer Graphics and Applications*, 8(4):51–58.

Fisherkeller, M. A., Friedman, J. H., and Tukey, J. W. (1975). Prim-9: a data display and analysis system. In *Pacific Regional Conference of the Association for Computing Machinery*, San Francisco, CA.

Fowlkes, E. B. (1971). User's manual for an on-line interactive system for probability plotting on the ddp-224 computer. Technical memorandum, AT&T Bell Laboratories, Murray Hill, NJ.

Friendly, M. (1999). Extending mosaic displays: marginal, partial, and conditional views of categorical data. *Journal of Computational and Statistical Graphics*, 8:373–395.

Friendly, M. (2000a). Re-visions of Minard. *Statistical Computing and Statistical Graphics Newsletter*, 12(1):13–19.

Friendly, M. (2000b). *Visualizing Categorical Data*. SAS Institute, Cary, NC.

Friendly, M. and Kwan, E. (2003). Effect ordering for data displays. *Computational Statistics and Data Analysis*, 43(4):509–539.

Graham, J., Hofer, S., and MacKinnon, D. (1996). Maximizing the usefulness of data obtained with planned missing value patterns: an application of maximum likelihood procedures. *Multivariate Behavioral Research*, 31(2):197–218.

Guvenir, H. A., Demiroz, G., and Ilter, N. (1998). Learning differential diagnosis of erythemato-squamous diseases using voting feature intervals. *Artificial Intelligence in Medicine*, 13:147–165.

Haberman, S. J. (1973). The analysis of residuals in cross-classification tables. *Biometrics*, 29:205–220.

Hesterberg, T. (1999). A graphical representation of Little's test for MCAR. Technical Report 94. MathSoft, Seattle, WA.

Hofman, H. (2003). Constructing and reading mosaic plots. *Computational Statistics and Data Analysis*, 43(4):565–580.

Hutchins, E. L., Hollan, J. D., and Norman, D. A. (1986). Direct manipulation interfaces. In Norman, D.A. and Draper, S. W., editors, *User Centered System Design: New Perspectives on Human–Computer Interaction*, pages 87–124. Lawrence Erlbaum Associates, Hillsdale, NJ.

Ihaka, R. and Gentleman, R. (1996). R: a language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5(3):299–314.

Inselberg, A. (1985). The plane with parallel coordinates. *The Visual Computer*, 1:69–97.

JMP (1989–2002). *Version 5*. SAS Institute. Cary, NC.

Jollife, I. T. (2002). *Principal Components Analysis*. Springer-Verlag, New York.

Kim, K. H. and Bentler, P. M. (2002). Tests of homogeneity of means and covariance matrixes for multivariate incomplete data. *Psychometrika*, 67(4):609–624.

Little, R. J. A. (1988). A test of missing completely at random for multivariate data with missing values. *Journal of the American Statistical Association*, 83(4):1198–1202.

Little, R. J. A. and Rubin, D. B. (1987). *Statistical Analysis with Missing Data*. Wiley, New York.

Little, R. J. A. and Rubin, D. B. (2002). *Statistical Analysis with Missing Data-second edition*. Wiley, New York.

McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models*. Chapman & Hall, London.

McDonald, J. A. (1982). Interactive graphics for data analysis. Ph.d. dissertation, Stanford University, Stanford, CA.

McDonald, J. A. (1988). Orion I: interactive graphics for data analysis. In Cleveland, W. S. and McGill, M. E., editors, *Dynamic Graphics for Statistics*, pages 179–200, Brooks/Cole, Pacific Grove, CA.

Molina, J. G., Ledesma, R., Valero-Mora, P. M., and Young, F. W. (2005). A video tour through ViSta 6.4. *Journal of Statistical Software*, 13(8):1–13.

Noma, E. (1987). Heuristic method for label placement in scatterplots. *Psychometrika*, 52(3):463–468.

North, C. and Shneiderman, B. (1997). A taxonomy of multiple window coordinations. Technical Report 3854, Computer Science Department, University of Maryland, College Park, MD.

Rindskopf, D. (1990). Non-standard log-linear models. *Psychological Bulletin*, 108(1):150–162.

Rubin, D. (1987). *Multiple Imputation for Nonresponse in Surveys*. Wiley, New York.

Schafer, J. L. (1997). *Analysis of Incomplete Multivariate Data*. Chapman & Hall, London.

Scott, D. W. (1992). *Multivariate Density Estimation*. Wiley, New York.

Siegel, J. M. (1995). Phylogeny and the function of REM sleep. *Behavioural Brain Research*, 69:29–34.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London.

Stine, R. and Fox, J. (1996). *Statistical Computing Environments for Social Research*. Sage, Thousand Oaks, CA.

Stuetzle, W. (1987). Plot windows. *Journal of the American Statistical Association*,82(398): 466-475-

Swayne, D. F. and Buja, A. (1998). Missing data in interactive high-dimensional data visualization. *Computational Statistics*, 13(1):15–26.

Swayne, D., Cook, D., and Buja, A. (1998). XGobi: interactive dynamic data visualization in the X window system. *Journal of Computational and Graphical Statistics* 7(1):113-130.

Swayne, D., Lang, D., Buja, A., and Cook, D. (2003). GGobi: evolving from XGobi into an extensible framework for interactive data visualization. *Computational Statistics and Data Analysis*, 43(4):423.

Theus, M. (2003). Interactive data visualization using Mondrian. *Journal of Statistical Software*, 7(11):1–9.

Tierney, L. (1988). Xlisp-Stat: a statistical environment based on the Xlisp language. Technical Report 528, School of Statistics, University of Minnesota, Minneapolis, MN.

Tierney, L. (1990). *Lisp-Stat: An Object Oriented Environment for Statistical Computing and Dynamic Graphics*. Wiley-Interscience, New York.

Tierney, L. (2005). Some notes on the past and future of Lisp-Stat. *Journal of Statistical Software*, 13(9):1–15.

Tufte, E. R. (1997). *Visual Explanations: Images and Quantities, Evidence and Narrative*. Graphics Press, Cheshire, England.

Turlach, B. A. (1993). Bandwidth selection in kernel density estimation: a review. Discussion paper 9232 Institut of Statistique, Louvain-la-Neuve, Belgium.

Udina, F. (1999). Implementing interactive computing in an object-oriented environment. Economics Working Papers 419. Department of Economics and Business, Universitat Pompeu Fabra, Barcelona, Spain.

Unwin, A. R., Hawkins, G., Hofman, H., and Siegl, B. (1996). Interactive graphics for data sets with missing values: Manet. *Journal of Computational and Graphical Statistics*, 5(2):113–122.

Valero-Mora, P. M. and Udina, F. (2005). The health of Lisp-Stat. *Journal of Statistical Software*, 13(10):1–5.

Valero-Mora, P. M., Young, F. W., and Friendly, M. (2003). Visualizing categorical data in ViSta. *Computational Statistics and Data Analysis*, 43(4):495–508.

Valero-Mora, P. M., Rodrigo, M. F., and Young, F. W. (2004). Visualizing parameters from log-linear models. *Computational Statistics*, 19(1):113–133.

Velleman, P. F. and Velleman, A. Y. (1985). *Data Desk*. Data Description, Ithaka, NY.

Vermunt, J. K. (1997). *Log-Linear Models for Event Histories*. Sage, Thousand Oaks, CA.

Wand, M. P. (1996). Data-based choice of histogram bin width. *Statistical Computing and Graphics*, 51(1):59–64.

Wegman, E. J. (1990). Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, 85(411):664–675.

Weisberg, S. (2005). Lost opportunities: Why we need a variety of statistical languages. *Journal of Statistical Software*, 13(1):1–12.

Young, F. W. (1994). ViSta: The visual statistics system. Research Memorandum 94-1 (revised 1996). L.L. Thurstone Psychometric Laboratory, Chapel Hill, NC.

Young, F., Valero-Mora, P., Faldowski, R., and Bann, C. M. (2003). Gossip: The arquitecture of spreadplots. *Journal of Computational and Graphical Statistics*, 12(1):80–100.

Young, F.W., Valero-Mora, P. & Friendly, M. (2006) *Visual Statistics: Seeing Data with Dynamic Interactive Graphics*. Wiley, New York.

Author Index

Subject Index

P

R

S

U

V

X

Erakunde autonomiaduna
Organismo Autónomo del

EUSKO JAURLARITZA
GOBIERNO VASCO

Eustat
EUSKAL ESTATISTIKA ERAKUNDEA
INSTITUTO VASCO DE ESTADÍSTICA

www.eustat.es