Interactive and Dynamic Graphics for Data Analysis

With Examples Using R and GGobi

Dianne Cook Deborah F. Swayne Andreas Buja
with contributions from Duncan Temple Lang and Heike Hofmann

DRAFT

Contents

List of Figures

Preface

This book is about using interactive and dynamic plots on a computer screen to look at data. Interactive and dynamic graphics has been an active area of research in Statistics since the late 1960s. Originally it was closely associated with exploratory data analysis, as it remains today, but it now makes substantial contributions in the emerging fields of data mining, especially visual data mining, and information visualization.

The material in this book includes:

- An introduction to data visualization, explaining how it differs from other types of visualization.
- A description of our toolbox of interactive and dynamic graphics.
- An explanation of the use of these tools in statistical data analyses such as cluster analysis, supervised classification, longitudinal data analysis, and microarray analysis.
- An approach for exploring missing values in data.
- A strategy for making inference from plots.

The book's examples use the software R and GGobi. R is a free software environment for statistical computing and graphics; it is most often used from the command line, provides a wide variety of statistical methods and includes high quality static graphics. GGobi is free software for interactive and dynamic graphics; it can be operated using a command-line interface or from a graphical user interface (GUI). When GGobi is used as a stand-alone tool, only the GUI is used; when it is used with R, a command-line interface is used.

R was initially developed by Robert Gentleman and Ross Ihaka, of the Statistics Department of the University of Auckland, and is now developed and maintained by a global collaborative effort. R can be considered to be a different implementation of S, a language and environment developed by John Chambers and colleagues at Bell Laboratories (formerly AT&T, now Lucent Technologies). GGobi is a descendant of two earlier programs: XGobi (written

by Deborah Swayne, Dianne Cook and Andreas Buja) and Dataviewer (written by Andreas Buja and Catherine Hurley). Many of the examples might be reproduced with other software such as Splus, JMP, DataDesk, Mondrian, MANET, and Spotfire. However, GGobi is unique because it offers tours (rotations of data in higher than 3D), complex linking between plots using categorical variables, and the tight connection with R.

The web site which accompanies the book contains sample data sets and R code, movies demonstrating the interactive and dynamic graphic methods, and additional chapters not included in this book:

`http://www.public.iastate.edu/~dicook/ggobi-book/ggobi.html`

The web sites for the software are

`http://www.R-project.org` R software and documentation

`http://www.ggobi.org` GGobi software and documentation

Both web sites include source code as well as binaries for various operating systems (linux, Windows, OSX); users can sign up for mailing lists and browse mailing list archives.

The language in the book is aimed at the level of later year undergraduates, beginning graduate students and graduate students in any discipline needing to analyze their own multivariate data. It is suitable reading for an industry statistician, engineer, bioinformaticist or computer scientist with some knowledge of basic data analysis and a need to analyze high-dimensional data. It also may be useful for a mathematician who wants to visualize high-dimensional structures.

The end of each chapter contains exercises to support the use of the book as a text in a class on statistical graphics, exploratory data analysis, visual data mining or information visualization. It might also be used as an adjunct text in a course on multivariate data analysis or data mining.

Way to use this book.... people should follow along.. with ggobi on their computers re-doing the book examples, or by watching the movies.

1

Introduction

In this technological age we live in a sea of information. We face the problem of gleaning useful knowledge from masses of words and numbers stored in computers. Fortunately, the computing technology that produces this deluge also gives us some tools to transform heterogeneous information into knowledge. We now rely on computers at every stage of this transformation: structuring and exploring information, developing models, and communicating knowledge.

In this book we teach a methodology that makes visualization central to the process of abstracting knowledge from information. Computers give us great power to represent information in pictures, but even more, they give us the power to interact with these pictures. If these are pictures of data, then interaction gives us the feeling of having our hands on the data itself and helps us to orient ourselves in the sea of information. By generating and manipulating many pictures, we make comparisons between different views of the data, we pose queries about the data and get immediate answers, and we discover large patterns and small features of interest. These are essential facets of data exploration, and they are important for model development and diagnosis as well.

In this first chapter we sketch the history of computer-aided data visualization and the role of data visualization in the process of data analysis.

1.1 Data Visualization: Beyond the Third Dimension

So far we have used the terms "information", "knowledge" and "data" informally. From now on we will use the following distinction: "data" refers to information that is structured in some schematic form such as a table or a list, and knowledge is derived from studying data. Data is often but not always quantitative, and it is often derived by processing unstructured information. It always includes some attributes or variables such as the number of hits on web sites, frequencies of words in text samples, weight in pounds, mileage in gallons per mile, income per household in dollars, years of education, acidity

on the pH scale, sulfur emissions in tons per year, or scores on standardized tests.

When we visualize data, we are interested in portraying abstract relationships among such variables: for example, the degree to which income increases with education, or the question of whether certain astronomical measurements indicate grouping and therefore hint at new classes of celestial objects. In contrast to this interest in abstract relationships, many other areas of visualization are principally concerned with the display of objects and phenomena in physical 3-D space. Examples are volume visualization (e.g., for the display of human organs in medicine), surface visualization (e.g., for manufacturing cars or animated movies), flow visualization (e.g., for aeronautics or meteorology), and cartography. In these areas one often strives for physical realism or the display of great detail in space, as in the visual display of a new car design, or of a developing hurricane in a meteorological simulation. The data visualization task is obviously different from drawing physical objects.

If data visualization emphasizes abstract variables and their relationships, then the challenge of data visualization is to create pictures that reflect these abstract entities. One approach to drawing abstract variables is to create axes in space and map the variable values to locations on the axes, then render the axes on a drawing surface. In effect, one codes non-spatial information using spatial attributes: position and distance on a page or computer screen. The goal of data visualization is then not realistic drawing, which is meaningless in this context, but translating abstract relationships to interpretable pictures.

This way of thinking about data visualization, as interpretable spatial representation of abstract data, immediately brings up a limitation: Plotting surfaces such as paper or computer screens are merely 2-dimensional. We can extend this limit by simulating a third dimension: The eye can be tricked into seeing 3-dimensional virtual space with perspective and motion, but if we want an axis for each variable, that's as far as we can stretch the display dimension.

This limitation to a 3-dimensional display space is not a problem if the objects to be represented are 3-dimensional, as in most other visualization areas. In data visualization, however, the number of axes required to code variables can be large: five to ten are common, but these days one often encounters dozens and even hundreds. This then is the challenge of data visualization: to overcome the 2-D and 3-D barriers. To meet this challenge, we use powerful computer-aided visualization tools. For example, we can mimic and amplify a paradigm familiar from photography: take pictures from multiple directions so the shape of an object can be understood in its entirety. This is an example of the "multiple views" paradigm which will be a recurring theme of this book. In our 3-D world the paradigm works superbly: the human eye is very adept at inferring the true shape of an object from just a few directional views. Unfortunately, the same is often not true for views of abstract data. The chasm between different views of data, however, can be actively bridged with additional computer technology: Unlike the passive paper medium, computers

allow us to manipulate pictures, to pull and push their content in continuous motion like a moving video camera, or to poke at objects in one picture and see them light up in other pictures. Motion links pictures in time; poking links them across space. This book features many illustrations of the power of these linking technologies. The diligent reader may come away "seeing" high-dimensional data spaces!

1.2 Statistical Data Visualization: Goals and History

Data visualization has homes in several disciplines, including the natural sciences, engineering, computer science, and statistics. There is a lot of overlap in the functionality of the methods and tools they generate, but some interesting differences in emphasis can be traced to the research contexts in which they were incubated. For example, the natural science and engineering communities rely on what is called "scientific visualization," which supports the goal of modeling physical objects and processes. The database research community creates visualization software which grows out of their work on the efficiency of data storage and retrieval; their graphics often summarize the kinds of tables and tabulations that are common results of database queries. The human-computer interface community produces software as part of their research in human perception, human-computer interaction and usability, and their tools are often designed to make the performance of a complex task as straightforward as possible.

The statistics community creates visualization systems within the context of data analysis, so the graphics are designed to help answer the questions that are raised as part of data exploration as well as statistical modeling and inference. As a result, statistical data visualization has some unique features. Statisticians are always concerned with variability in observations and error in measurements, both of which cause uncertainty about conclusions drawn from data. Dealing with this uncertainty is at the heart of classical statistics, and statisticians have developed a huge body of inferential methods that help to quantify uncertainty. Inference used to be statisticians' sole preoccupation, but this changed under John W. Tukey's towering influence. He championed "exploratory data analysis" (EDA) which focuses on discovery and allows for the unexpected. This is different from inference, which progresses from pre-conceived hypotheses. EDA has always depended heavily on graphics, even before the term "data visualization" was coined. Our favorite quote from John Tukey's rich legacy is that we need good pictures to "force the unexpected upon us." In the past, EDA and inference were sometimes seen as incompatible, but we argue that they are not mutually exclusive. In this book, we present some visual methods for assessing uncertainty and performing inference, that is, deciding whether what we see is "really there."

Most of the visual methods we present in this book reflect the heritage of research in computer-aided data visualization that began in the early 1970's.

The seminal visualization system was PRIM-9, the work of Fisherkeller, Friedman and Tukey at the Stanford Linear Accelerator Center in 1974. PRIM-9 was the first stab at an interactive tool set for the visual analysis of multivariate data. PRIM-9 was followed by further pioneering systems at the Swiss Federal Institute of Technology (PRIM-ETH), at Harvard University (PRIM-H) and Stanford University (ORION), in the late 1970s and early 1980s.

Research picked up in the following few years in many places. The authors themselves were influenced by work at AT&T Bell Laboratories, Bellcore, the University of Washington, Rutgers University, the University of Minnesota, MIT, CMU, Batelle Richmond WA, George Mason University, Rice University, York University, Cornell University, Trinity College, and the University of Augsburg, among others.

1.3 Getting Down to Data

Here is a very small and seemingly simple dataset we will use to illustrate the use of data graphics. One waiter recorded information about each tip he received over a period of a few months working in one restaurant. He collected several variables:

- *tip* in dollars,
- *bill* in dollars,
- *sex* of the bill payer,
- whether there were *smokers* in the party,
- *day* of the week,
- *time* of day,
- *size* of the party.

In all he recorded 244 tips. The data was reported in a collection of case studies for business statistics (Bryant & Smith 1995). The primary question related to the data is: *What are the factors that affect tipping behavior?*

This is a typical (albeit small) dataset: there are seven variables, of which two are numeric (*tip*, *bill*), the others categorical or otherwise discrete. In answering the question, we are interested in exploring relationships that may involve more than three variables, none of which is about physical space. In this sense the data are high-dimensional and abstract.

We first have a look at the variable of greatest interest to the waiter: *tip*. A common graph for looking at a single variable is the histogram, where data values are binned and the count is represented by a rectangular bar. We first chose a bin width of one dollar and produced the first graph of Figure 1.1. The distribution appears to be unimodal, that is, it has one peak, the bar representing the tips greater than one dollar and less than or equal two dollars. There are very few tips of one dollar or less. The number of larger tips trails off rapidly, suggesting that this is not a very expensive restaurant.

The conclusions drawn from a histogram are often influenced by the choice of bin width, which is a parameter of the graph and not of the data. Figure 1.1 shows a histogram with a smaller bin width, 10*c*. At the smaller bin width the shape is multimodal, and it is clear that there are large peaks at the full dollars and smaller peaks at the half dollar. This shows that the customers tended to round the tip to the nearest fifty cents or dollar.

This type of observation occurs frequently when studying histograms: A large bin width smooths out the graph and shows rough or global trends, while a smaller bin width highlights more local features. Since the bin width is an example of a graph parameter, experimenting with bin width is an example of exploring a set of related graphs. Exploring multiple related graphs can lead to insights that would not be apparent in any single graph.

So far we have not addressed the waiter's question: what relationships exist between *tip* and the other variables? Since the tip is usually calculated based on the bill, it is natural to look first at a graph of *tip* and *bill*. A common graph for looking at a pair of continuous-value variables is the scatterplot, as in Figure 1.2. We see that the variables are quite correlated, confirming the idea that tip tends to be calculated from the bill. Disappointingly for the waiter, there are many more points below the diagonal than above it: there are many more "cheap tippers" than generous tippers. There are a couple of notable exceptions, especially one party who gave a $5.15 tip for a $7.25 bill, a tip rate of about 70%.

We said earlier that an essential aspect of data visualization is capturing relationships among many variables: three, four, or even more. This dataset, simple as it is, illustrates the point. Let us ask, for example, how a third variable such as *sex* affects the relationship between *tip* and *bill*. As *sex* is categorical, binary actually, it is natural to divide the data into female and male payers and generate two scatterplots of *tip* versus *bill*. Let us go even further by including a fourth variable, *smoking*, which is also binary. We now divide the data into four parts and generate the four scatterplots seen in Figure 1.3. Inspecting these plots reveals numerous features: (1) for smoking parties, there is almost no correlation between the size of the tip and the size of the bill, (2) when a female non-smoker paid the bill, the tip was a very consistent percentage of the bill, with the exceptions of three dining parties, (3) larger bills were mostly paid by men.

Taking Stock

In the above example we gained a wealth of insights in a short time. Using nothing but graphical methods we investigated univariate, bivariate and multivariate relationships. We found both global features and local detail: we saw that tips were rounded, then we saw the obvious correlation between the tip and the size of the bill but noticed a scarcity of generous tippers, and finally we discovered differences in the tipping behavior of male and female smokers and non-smokers.

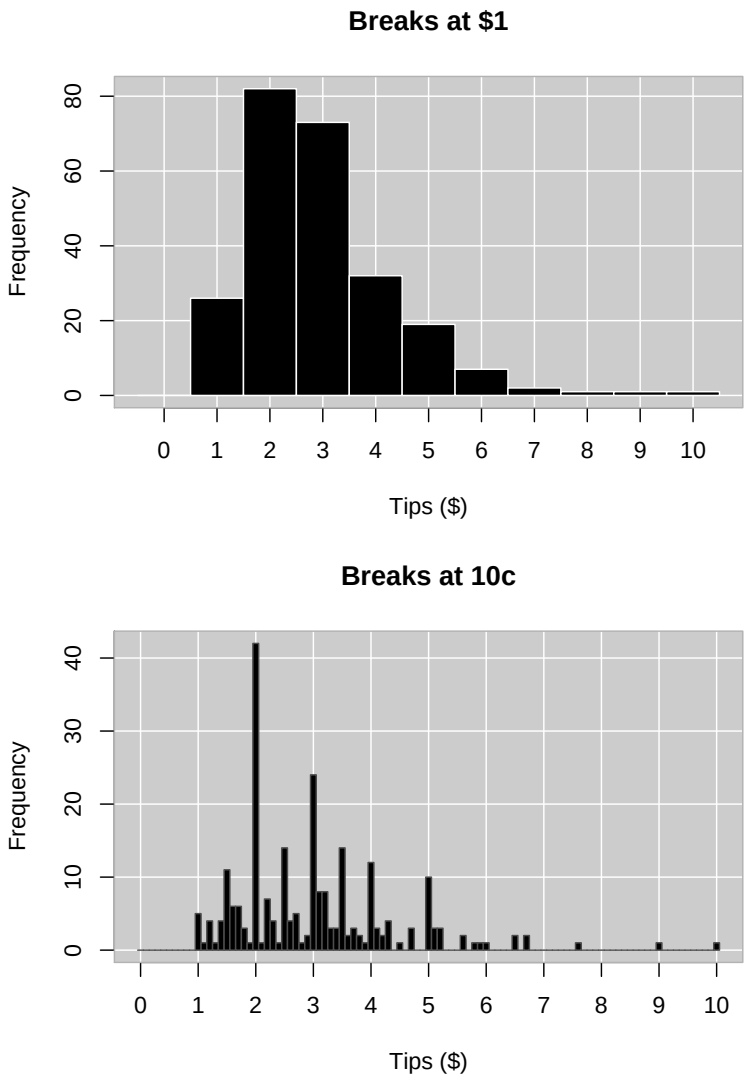

Fig. 1.1. Histograms of actual tips with differing barwidth: $1, 10c. The power of an interactive system allows bin width to be changed with slider.

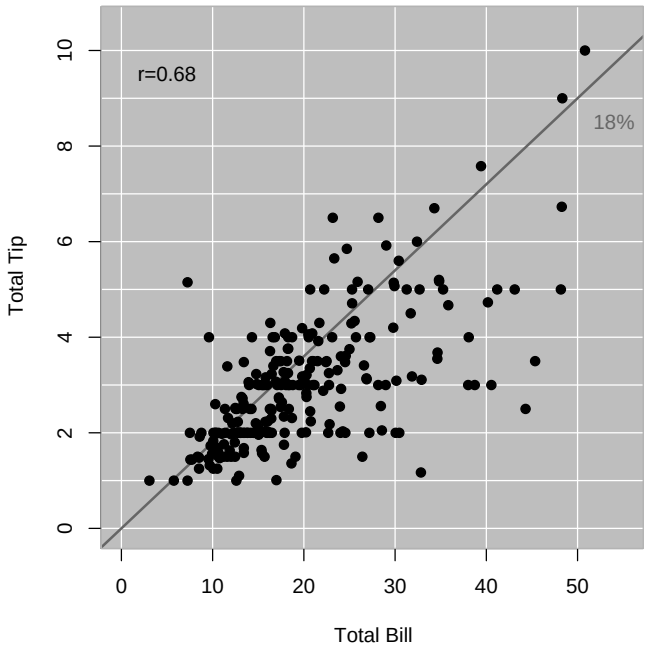

Fig. 1.2. Scatterplot of Total Tip vs Total Bill: More points in the bottom right indicate more cheap tippers than generous tippers.

Notice that we used very simple plots to explore some pretty complex relationships involving as many as four variables. We began to explore multivariate relationships for the first time when we produced the plots in Figure 1.3. Each plot shows a subset obtained by partioning the data according to two binary variables. The statistical term for partitioning based on variables is "conditioning". For example, the top left plot shows the dining parties that meet the condition that the bill payer was a male non-smoker: *sex* = male and *smoking* = False. In database terminology this plot would be called the result of "drill-down". The idea of conditioning is richer than drill-down because it involves a structured partitioning of *all* the data as opposed to the extraction of a single partition.

Having generated the four plots, we arrange them in a two by two layout to reflect the two variables on which we conditioned. While the axes in each individual plot are *tip* and *bill*, the axes of the overall figure are *smoking* (vertical) and *sex* (horizontal). The arrangement permits us to make several kinds of comparisons and make observations about the partitions. For example, comparing the rows shows that smokers and non-smokers differ in the

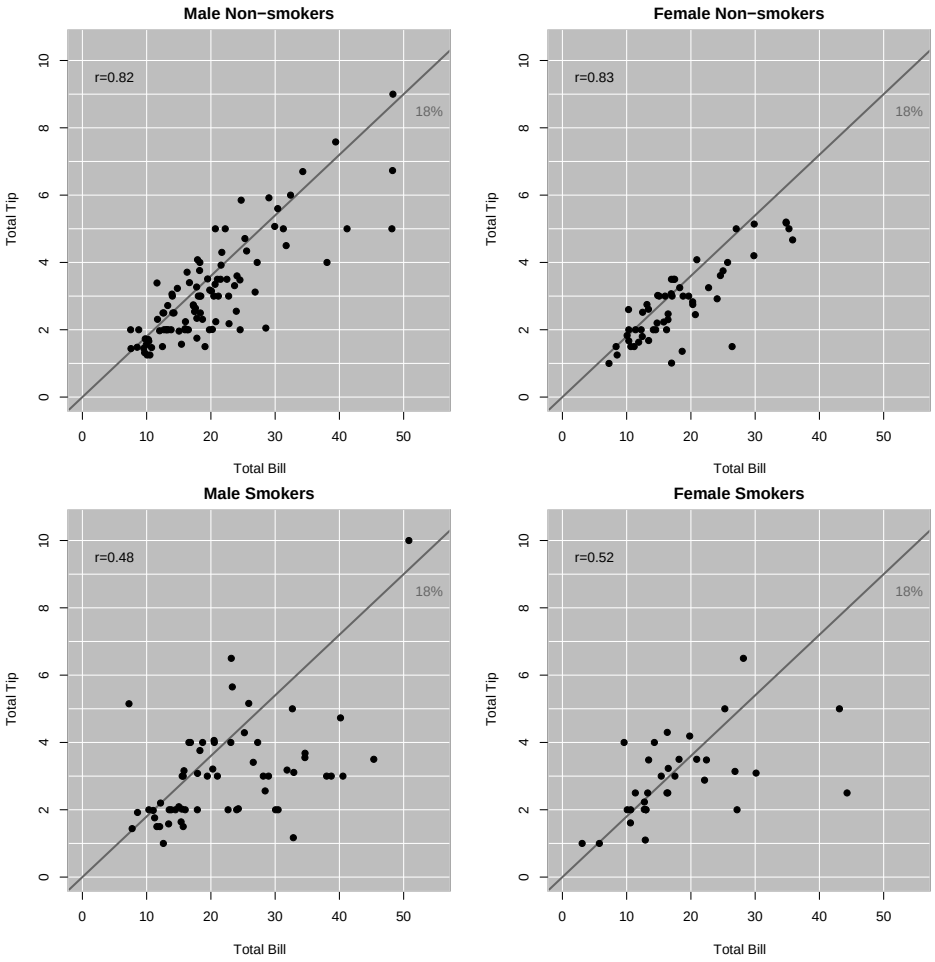

Fig. 1.3. Total Tip vs Total Bill by Sex and Smoker: There is almost no association between tip and total bill in the smoking parties, and, with the exception of 3 dining parties, when a female non-smokers paid the bill the tip was extremely consistent.

strength of the correlation between *tip* and *bill*, and comparing the plots in the top row shows that male and female non-smokers differ in that the larger bills tend to be paid by men. In this way a few simple plots allow us to reason about relationships among four variables!

By contrast, an old-fashioned approach without graphics would be to fit some regression model. Without subtle regression diagnostics (which rely on graphics!), this approach would miss many of the above insights: the rounding of tips, the preponderance of cheap tippers, and perhaps the multivariate relationships involving the bill payer's sex and the group's smoking habits.

1.4 Getting Real: Process and Caveats

The preceding explanations may have given a somewhat misleading impression of the process of data analysis. In our account the data had no problems; for example, there were no missing values and no recording errors. Every step was logical and necessary. Every question we asked had a meaningful answer. Every plot that was produced was useful and informative. In actual data analysis nothing could be further from the truth. Real data are rarely perfect; most choices are guided by intuition, knowledge and judgment; most steps lead to dead ends; most plots end up in the wastebasket. This may sound daunting, but even though data analysis is a highly improvisational activity, it can be given some structure nonetheless.

To understand data analysis, and how visualization fits in, it is useful to talk about it as a process consisting of several stages:

- The problem statement
- Data preparation
- Exploratory data analysis
- Quantitative analysis
- Presentation

The problem statement: Why do you want to analyze this data? Underlying every data set is a question or problem statement. For the tipping data the question was provided to us from the data source: "What are the factors that affect tipping behavior?" This problem statement drives the process of any data analysis. Sometimes the problem is identified prior to a data collection. Perhaps it is realized after data becomes available because having the data available has made it possible to imagine new issues. It may be a task that the boss assigns, it may be an individual's curiosity, or part of a larger scientific endeavor to find a cure. Ideally, we begin an analysis with some sense of direction, as described by a pertinent question.

Data preparation: In the classroom, the teacher hands the class a single data matrix with each variable clearly defined. In the real world, it can take a great deal of work to construct a clean data matrix. For example, data may be missing or misrecorded, they may be distributed across several sources, and the variable definitions and data values may be inconsistent across these sources. Analysts often have to invest considerable time in learning computing tools and domain knowledge before they can even ask a meaningful question about the data. It is therefore not uncommon for this stage to consume most of the effort that goes into a project. And it is also not uncommon to loop back to this stage after completing the following stages, to re-prepare and re-analyze the data.

In preparing the tipping data, we would create a new variable called tip rate, because when tips are discussed in restaurants, among waiters, dining parties, and tourist guides, it is in terms of a percentage of total bill. We may

also create several new dummy variables for the day of the week, in anticipation of fitting a regression model. We didn't talk about using visualization to verify that we had correctly understood and prepared the tipping data. For example, that unusually large tip could have been the result of a transcription error. Graphics identified the observation as unusual and the analyst might use this information to search the origins of the data to check the validity of the numbers for this observation.

Exploratory data analysis: We gave you some of the flavor of this stage in the analysis of the waiter's tips. We checked the distribution of individual variables, we looked for unusual records, we explored relationships among multiple variables, and we found some unexpected patterns. To complete this exploration, we would also add numerical summaries to the visual analysis.

It is this stage in the analysis that we make time to "play in the sand" to allow us to find the unexpected, and come to some understanding of the data we're working with. We like to think of this as a little like travelling. We may have a purpose in visiting a new city, perhaps to attend a conference, but we need to take care of our basic necessities, such as, find eating places, shops where we can get our supplies, a gas station to fill up at. Some of the direction will be determined, guided by the concierge, or other locals, but some of the time we wander around by ourselves. We may find a cafe with just the type of food that we like instead of what the concierge likes, or a gift shop with just the right things for a family member at home, or we might find a cheaper gas price. This is all about getting to know the neighborhood. At this stage in the data analysis we relax the focus on the problem statement, and explore broadly different aspects of the data. For the tipping data, although the primary question was about the factors affecting tip behavior, we found some surprising general aspects of tipping behavior, beyond this question: the rounding of tips, the prevalence of cheap tippers, and heterogeneity in variance between groups.

Exploratory data analysis has evolved with the evolution of fast, graphically enabled desktop computers, into a highly interactive, real-time, dynamic and visual process. Exploratory data analysis takes advantage of technology, in a way that Tukey envisioned and experimented with on specialist hardware 40 years ago: "Today, software and hardware together provide far more powerful factories than most statisticians realize, factories that many of today's most able young people find exciting and worth learning about on their own" (Tukey 1965). It is characterized by direct manipulation, and dynamic graphics: plots that respond in real time to an analyst's queries and change dynamically to re-focus, linking information from other sources, and re-organize information. The analyst is able to work thoroughly over the data rapidly, slipping out of dead-ends, and chasing down new leads. The high-level of interactivity is enabled by fast, decoration-devoid, graphics, which are generally not adequate for presentation purposes. In general this means that it is nec-

essary to re-create the revealing plots in a more exacting and static form to communicate results.

Quantitative analysis: This stage consists of statistical modeling and statistical inference. It is where we focus in on the primary question of interest. With statistical models we summarize complex data. Models often help us decompose data into estimates of signal and noise. With statistical inference, we try to assess whether a signal is real. It is widely accepted that data visualization is an important part of exploratory data analysis, but it's not as well understood that it also plays an important role at this stage. The role played is both in diagnosing a model in relation to the data, and to better understand a model.

For the tips data, we haven't yet addressed the primary question of interest. To do this we'd likely fit a regression model using tip rate as the response and the remaining variables (except tip, total bill) as the explanatory variables (Sex, Smoker, Size, Time, Day). When we do this, of all the variables only Size has a significant regression coefficient, resulting in the model $\hat{TipRate} = 0.18 - 0.01 \times Size$ which explains just 2% of the variation in tip rate. The model says that starting from a baseline tip rate of 18% the amount drops by 1% for each additional diner in a party. This is the model answer in Bryant & Smith (1995). Figure 1.4 shows this model, and the underlying data. The data is jittered horizontally to alleviate overplotting from the discreteness of the Size variable. The data values are spread widely around the model. And there are very few data points for parties of size one, five and six, which makes us question the validity of the model in these regions. What have we learned about tipping behavior? Size of the party explains only a very small amount of the variation in tip rate. The signal is very weak relative to the noise. Is it a useful model? It is used: most restaurants today factor the tip into the bill automatically for larger dining parties.

Most problems are more complex than the tips data, and the typical models commonly used are often more sophisticated. Fitting a model produces its own data, in the form of model estimates and diagnostics. Often we can simulate from the model giving samples from posterior distributions. The model outputs are data that can be explored for the pleasure of understanding the model. We may plot parameter estimates and confidence regions. We may plot the posterior samples.

Plotting the model in relation to the data is important, too. There is a temptation to ignore the data at this point, in favor of the simplification provided by a model. But a lot can be learned from what's left out of the model: We would never consider teaching regression analysis without teaching residual plots. A model is a succinct explanation of the variation in the data, a simplification. With a model we can make short descriptive statements: As the size of the dining party increases an additional person the tip rate decreases by 1%. Pictures can help to assess if a model is too simple for the data, because a well-constructed graphic can provide a digestible summary of

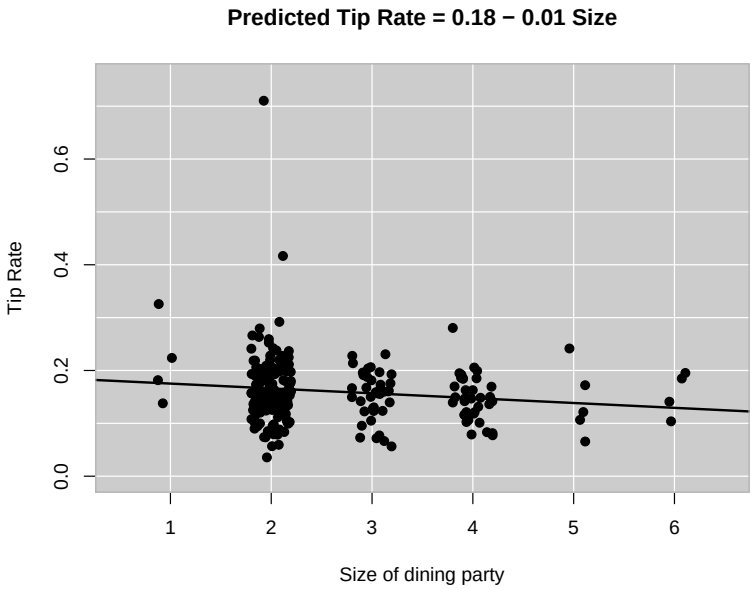

Fig. 1.4. What are the factors that affect tipping behavior? This is a plot of the best model, along with the data. (Points are jittered horizontally to alleviate overplotting from the discreteness of the Size variable.) There is a lot of variation around the regression line: There is very little signal relative to noise. In addition there are very few data points for parties of size 1, 5, 6, raising the question of the validity of the model in these extremes.

complex structure. A problem with a model may be immediately obvious from a plot. Graphics are an essential part of model diagnostics. A graphic should be self-explanatory, but it is usually assisted by a detailed written or verbal description. "A picture saves a thousand words!" Or does it take a thousand words to explain? The beauty of a model is that the explanation is concise, and precise. But pictures are powerful tools in a data analysis that our visual senses embrace, revealing so much that a model alone cannot.

The interplay of EDA and QA: Is it data snooping?

Exploratory data analysis can be difficult to teach. Says Tukey (1965) "Exploratory data analysis is NOT a bundle of techniques.... Confirmatory analysis is easier to teach and compute...." In the classroom, the teacher explains a method to the class and demonstrates it on the single data matrix, and then repeats this with another method. Its easier to teach a stream of seemingly disconnected methods, applied to data fragments than to put it all together. EDA, as a process, is very closely tied to data problems. There usually isn't time to let students navigate their own way through a data analysis, to spend a long time cleaning data, to make mistakes, recover from them,

and synthesize the findings into a summary. Teaching a bundle of methods is an efficient approach to covering substantial quantities of material. But its useless unless the student can put it together. Putting it together might be simply a matter of common sense. Yet, common sense is rare. Probably it should be taught explicitly.

Because EDA is very graphical, it brings rise to a suspicion of data "snooping". With the tipping data, from a few plots we learned an enormous amount of information about tipping: there is a scarcity of generous tippers, that the variability in tips increases extraordinarily for smoking parties, and that people tend to round their tips. These are very different types of tipping behaviors than what we learned from the regression model. The regression model was not compromised by what we learned from graphics. We snooped into the data. In reality, making pictures of data is not necessarily data snooping. If the purpose of an analysis is clear then making plots of the data is "just smart", and we make many unexpected observations about the data, resulting in a richer and more informative analysis. We particularly like the quote by Crowder & Hand (1990): "The first thing to do with data is to look at them.... usually means tabulating and plotting the data in many different ways to 'see whats going on'. With the wide availability of computer packages and graphics nowadays there is no excuse for ducking the labour of this preliminary phase, and it may save some red faces later."

Presentation: Once an analysis has been completed, the results must be reported, either to clients, managers or colleagues. The results probably take the form of a narrative, and include quantitative summaries such as tables, forecasts, models, and graphics. Quite often, graphics form the bulk of the summaries.

The graphics included in a final report may be a small fraction of the graphics generated for exploration and diagnostics. Indeed, they may be different graphics altogether. They are undoubtedly carefully prepared for their audience. The graphics generated during the analysis are meant for the analyst only and thus need to be quickly generated, functional but not polished. This is a dilemma for these authors who have much to say about exploratory graphics, but need to convey it in printed form. We have carefully re-created every plot in this book!

As we have already said, these broadly defined stages do not form a rigid recipe. Some of the stages overlap, and occasionally some are skipped. The order is often shuffled and groups of steps reiterated. What may look like a chaotic activity is often improvisation on a theme loosely following the "recipe".

1.5 Interactive Investigation

Thus far, all the observations on the tipping data have been made using static graphics - the purpose up to this point has been to communicate the importance of plots in the context of data analysis. Although we no longer hand-draw plots, static plots are computer-generated for a passive paper medium, to be printed and stared at by the analyst. Computers, however, allow us to produce plots for active consumption. This book is about interactive and dynamic plots, the material forming the following chapters, but we will give a hint as to how interactive plots enhance the data analysis process we've just described.

The tips data is simple. Most of the interesting features can be discovered using static plots. Yet, interacting with the plots reveals more and enables the analyst to pursue follow-up questions. For example, we could address a new question, arising from the current analysis, such as "Is the rounding behavior of tips predominant in some demographic group?" To investigate we probe the histogram, highlight the bars corresponding to rounded tips, and observe the pattern of highlighting in the linked plots (Figure 1.5). Multiple plots are visible simultaneously, and the highlighting action on one plot generates changes in the other plots. The two additional plots here are spine plots (), used to examine the proportions in categorical variables. For the highlighted subset of dining parties, the ones who rounded the tip to the nearest dollar or half-dollar, the proportion of bill paying males and females is roughly equal, but interestingly, the proportion of smoking parties is higher than non-smoking parties. This might suggest another behavioral difference between smokers and non-smokers: a larger tendency for smokers than non-smokers to round their tips. If we were to be skeptical about this effect we would dig deeper, making more graphical explorations and numerical models. By pursuing this with graphics we'd find that the proportion of smokers who round the tip is only higher than non-smokers for full dollar amounts, and not for half-dollar amounts.

This is the material that this book describes: how interactive and dynamic plots are used in data analysis.

1.6 What's in this book?

We have just said that visualization has a role in most stages of data analysis, all the way from data preparation to presentation. In this book, however, we will concentrate on the use of graphics in the exploratory and diagnostic stages. We concentrate on graphics that can be probed and brushed, direct manipulation graphics, and graphics that can change temporally, dynamic graphics.

The reader may note the paradoxical nature of this claim about the book: Once a graphic is published, is it not by definition a presentation graphic?

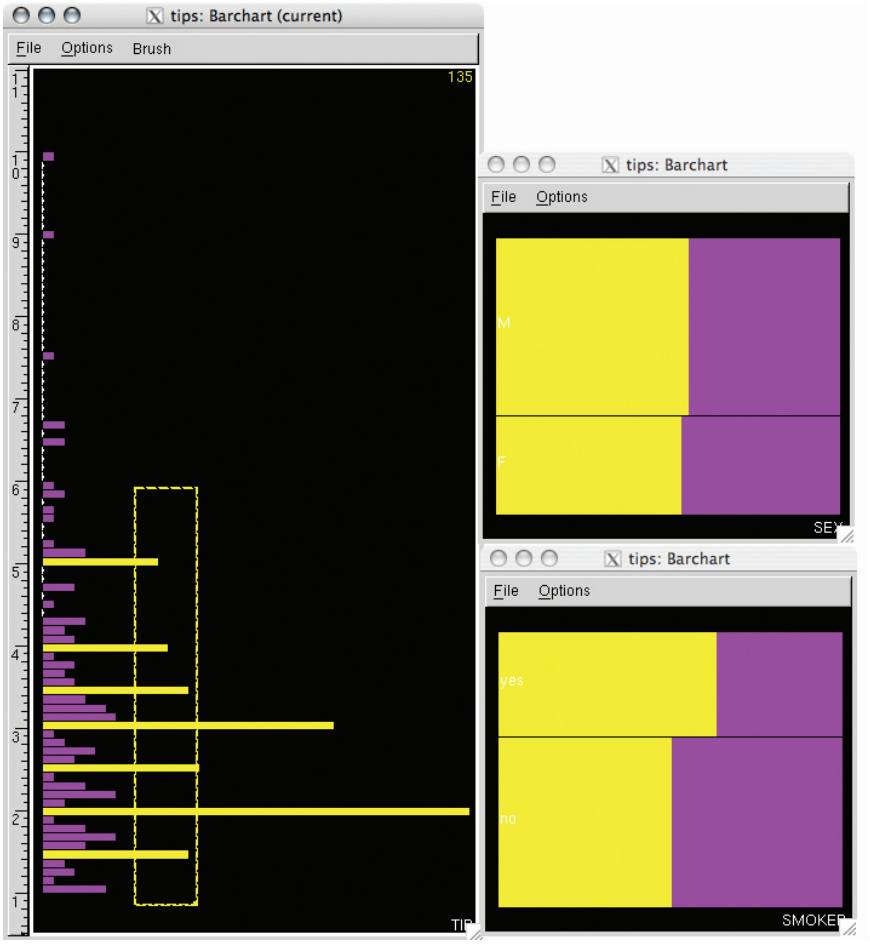

Fig. 1.5. Bins of whole and half-dollar amounts are highlighted. This information is linked to spine plots of gender of the bill payer and smoking status of the dining party. The proportion of males and females in this group that rounds tips is roughly equal, but interestingly the proportion of smoking parties who round their tips is higher than non-smoking parties.

Yes and no: as in the example of the waiter's tips, the graphics in this book have all been carefully selected, prepared, and polished, but they are shown as they appeared during our analysis. Only the last figure for the waiter's tips is shown in raw form, to introduce the sense of the rough and useful nature of exploratory graphics.

The first chapter opens our toolbox of plot types and direct manipulation modes. The missing data chapter is the material most related to a data preparation stage. It is presented early because handling missing values is one of

the first obstacles in analysing data. The chapters on supervised classification and cluster analysis have both exploratory and diagnostic material. A chapter on inference hints at ways we can assess our subjective visual senses.

1

Introduction

In this technological age we live in a sea of information. We face the problem of gleaning useful knowledge from masses of words and numbers stored in computers. Fortunately, the computing technology that causes this deluge also gives us some tools to transform heterogeneous information into knowledge. We now rely on computers at every stage of this transformation: structuring and exploring information, developing models, and communicating knowledge.

In this book we teach a methodology that makes visualization central to the process of abstracting knowledge from information. Computers give us great power to represent information in pictures, but even more, they give us the power to interact with these pictures. If these are pictures of data, then interaction gives us the feeling of having our hands on the data itself and helps us to orient ourselves in the sea of information. By generating and manipulating many pictures, we make comparisons between different views of the data, we pose queries about the data and get immediate answers, and we discover large patterns and small features of interest. These are essential facets of data exploration, and they are important for model development and diagnosis as well.

In this first chapter we sketch the history of computer-aided data visualization and the role of data visualization in the process of data analysis.

1.1 Data Visualization: Beyond the Third Dimension

So far we have used the terms "information" and "data" informally. From now on we will use the following distinction: "data" refers to information that is structured in some schematic form such as a table or a list. Data is often but not always quantitative, and it is often derived by processing unstructured information. It always includes some attributes or variables such as the number of hits on web sites, frequencies of words in text samples, weight in pounds, mileage in gallons per mile, income per household in dollars, years

of education, acidity on the PH scale, sulfur emissions in tons per year, or scores on standardized tests.

When we visualize data, we are interested in portraying abstract relationships among such variables: for example, the degree to which income increases with education, or the question of whether certain astronomical measurements indicate grouping and therefore hint at new classes of celestial objects. In contrast to this interest in abstract relationships, many other areas of visualization are principally concerned with the display of objects and phenomena in physical 3-D space. Examples are volume visualization (e.g., for the display of human organs in medicine), surface visualization (e.g., for manufacturing cars or animated movies), flow visualization (e.g., for aeronautics or meteorology), and cartography. In these areas one often strives for physical realism or the display of great detail in space, as in the visual display of a new car design, or of a developing hurricane in a meteorological simulation.

If data visualization emphasizes abstract variables and their relationships, then the challenge of data visualization is to create pictures that reflect these abstract entities. This task is obviously different from drawing physical objects. One approach to drawing abstract variables is to create axes in space and map the variable values to locations on the axes, then render the axes on a drawing surface. In effect, one codes non-spatial information using spatial attributes: position and distance on a page or computer screen. The goal of data visualization is then not realistic drawing, which is meaningless in this context, but translating abstract relationships to interpretable pictures.

This way of thinking about data visualization, as interpretable spatial representation of abstract data, immediately brings up a limitation: Plotting surfaces such as paper or computer screens are merely 2-dimensional. We can extend this limit by simulating a third dimension: The eye can be tricked into seeing 3-dimensional virtual space with perspective and motion, but if we want an axis for each variable, that's as far as we can stretch the display dimension.

This limitation to a 3-dimensional display space is not a problem if the objects to be represented are 3-dimensional, as in most other visualization areas. In data visualization, however, the number of axes required to code variables can be large: five to ten are common, but these days one often encounters dozens and even hundreds. This then is the challenge of data visualization: to overcome the 2-D and 3-D barriers. To meet this challenge, we use powerful computer-aided visualization tools. For example, we can mimic and amplify a paradigm familiar from photography: take pictures from multiple directions so the shape of an object can be understood in its entirety. This is an example of the "multiple views" paradigm which will be a recurring theme of this book. In our 3-D world the paradigm works superbly: the human eye is very adept at inferring the true shape of an object from just a few directional views. Unfortunately, the same is often not true for views of abstract data. The chasm between different views of data, however, can be actively bridged with additional computer technology: Unlike the passive paper medium, computers

allow us to manipulate pictures, to pull and push their content in continuous motion with a similar effect as a moving video camera, or to poke at objects in one picture and see them light up in other pictures. Motion links pictures in time; poking links them across space. This book features many illustrations of the power of these linking technologies. The diligent reader may come away "seeing" high-dimensional data spaces!

1.2 Statistical Data Visualization: Goals and History

Data visualization has homes in several disciplines, including the natural sciences, engineering, computer science, and statistics. There is a lot of overlap in the functionality of the methods and tools they generate, but some interesting differences in emphasis can be traced to the research contexts in which they were incubated. For example, the natural science and engineering communities rely on what is called "scientific visualization," which supports the goal of modeling physical objects and processes. The database research community creates visualization software which grows out of their work on the efficiency of data storage and retrieval; their graphics often summarize the kinds of tables and tabulations that are common results of database queries. The human-computer interface community produces software as part of their research in human perception, human-computer interaction and usability, and their tools are often designed to make the performance of a complex task as straightforward as possible.

The statistics community creates visualization systems within the context of data analysis, so the graphics are designed to help answer the questions that are raised as part of data exploration as well as statistical modeling and inference. As a result, statistical data visualization has some unique features. Statisticians are always concerned with variability in observations and error in measurements, both of which cause uncertainty about conclusions drawn from data. Dealing with this uncertainty is at the heart of classical statistics, and statisticians have developed a huge body of inference methods that allow us to quantify uncertainty. Inference used to be statisticians' sole preoccupation, but this changed under John W. Tukey's towering influence. He championed "exploratory data analysis" (EDA) which focuses on discovery and allows for the unexpected, unlike inference, which progresses from pre-conceived hypotheses. EDA has always depended heavily on graphics, even before the term "data visualization" was coined. Our favorite quote from John Tukey's rich legacy is that we need good pictures to "force the unexpected upon us." In the past, EDA and inference were sometimes seen as incompatible, but we argue that they are not mutually exclusive. In this book, we present some visual methods for assessing uncertainty and performing inference, that is, deciding whether what we see is "really there."

Most of the visual methods we present in this book reflect the heritage of research in computer-aided data visualization that began in the early 1970's.

The seminal visualization system was PRIM-9, the work of Fisherkeller, Friedman and Tukey at the Stanford Linear Accelerator Center in 1974. PRIM-9 was the first stab at an interactive tool set for the visual analysis of multivariate data. PRIM-9 was followed by further pioneering systems at the Swiss Federal Institute of Technology (PRIM-ETH), at Harvard University (PRIM-H) and Stanford University (ORION), in the late 1970s and early 1980s.

Research picked up in the following few years in many places. The authors themselves were influenced by work at AT&T Bell Laboratories, Bellcore, the University of Washington, Rutgers University, the University of Minnesota, MIT, CMU, Batelle Richmond WA, George Mason University, Rice University, York University, Cornell University, and the University of Augsburg, among others; they contributed to work at the first four institutions.

1.3 Getting Down to Data

Here is a very small and seemingly simple example of a dataset that exhibits the features we mentioned. One waiter recorded information about each tip he received over a period of a few months working in one restaurant. He collected several variables:

- *tip* in dollars,
- *bill* in dollars,
- *sex* of the bill payer,
- whether there were *smokers* in the party,
- *day* of the week,
- *time* of day,
- *size* of the party.

In all he recorded 244 tips. The data was reported in a collection of case studies for business statistics (Bryant & Smith 1995). The primary question related to the data is: *What are the factors that affect tipping behavior?*

This is a typical (albeit small) dataset: there are seven variables, of which two are numeric (*tip*, *bill*), the others categorical or otherwise discrete. In answering the question, we are interested in exploring relationships that may involve more than three variables, none of which is about physical space. In this sense the data are high-dimensional and abstract.

We first have a look at the variable of greatest interest to the waiter: *tip*. A common graph for looking at a single variable is the histogram, where data values are binned and the count is represented by a rectangular bar. We first chose a bin width of one dollar and produced the first graph of Figure 1.1. The distribution appears to be unimodal, that is, it has one peak, the bar representing the tips greater than one dollar and less than or equal two dollars. There are very few tips of one dollar or less. The number of larger tips trails off rapidly, suggesting that this is not a very expensive restaurant.

The conclusions drawn from a histogram are often influenced by the choice of bin width, which is a parameter of the graph and not of the data. Figure 1.1 shows a histogram with a smaller bin width, 10*c*. At the smaller bin width the shape is multimodal, and it is clear that there are large peaks at the full dollars and smaller peaks at the half dollar. This shows that the customers tended to round the tip to the nearest fifty cents or dollar.

This type of observation occurs frequently when studying histograms: A large bin width smooths out the graph and shows rough or global trends, while a smaller bin width highlights more local features. Since the bin width is an example of a graph parameter, experimenting with bin width is an example of exploring a set of related graphs. Exploring multiple related graphs can lead to insights that would not be apparent in any single graph.

So far we have not addressed the waiter's question: what relationships exist between *tip* and the other variables? Since the tip is usually calculated based on the bill, it is natural to look first at a graph of *tip* and *bill*. A common graph for looking at a pair of variables is the scatterplot, as in Figure 1.2. We see that the variables are quite correlated, confirming the idea that tip tends to be calculated from the bill. Disappointingly for the waiter, there are many more points below the diagonal than above it: there are many more "cheap tippers" than generous tippers. There are a couple of notable exceptions, especially one party who gave a $5.15 tip for a $7.25 bill, a tip rate of about 70%.

We said earlier that an essential aspect of data visualization is capturing relationships among many variables: three, four, or even more. This dataset, simple as it is, illustrates the point. Let us ask, for example, how a third variable such as *sex* affects the relationship between *tip* and *bill*. As *sex* is categorical, even binary, it is natural to divide the data into female and male payers and generate two scatterplots of *tip* versus *bill*. Let us go even further by including a fourth variable, *smoking*, which is also binary. We now divide the data into four parts and generate the four scatterplots seen in Figure 1.3. Inspecting these plots reveals numerous features: (1) for smoking parties, there is almost no correlation between the size of the tip and the size of the bill, (2) when a female non-smoker paid the bill, the tip was a very consistent percentage of the bill, with the exceptions of three dining parties, (3) larger bills were mostly paid by men.

Taking Stock

In the above example we gained a wealth of insights in a short time. Using nothing but graphical methods we investigated univariate, bivariate and multivariate relationships. We found both global features and local detail: we saw that tips were rounded, then we saw the obvious correlation between the tip and the size of the bill but noticed a scarcity of generous tippers, and finally we discovered differences in the tipping behavior of male and female smokers and non-smokers.

Notice that we used very simple plots to explore some pretty complex relationships involving as many as four variables. We began to explore multivari-

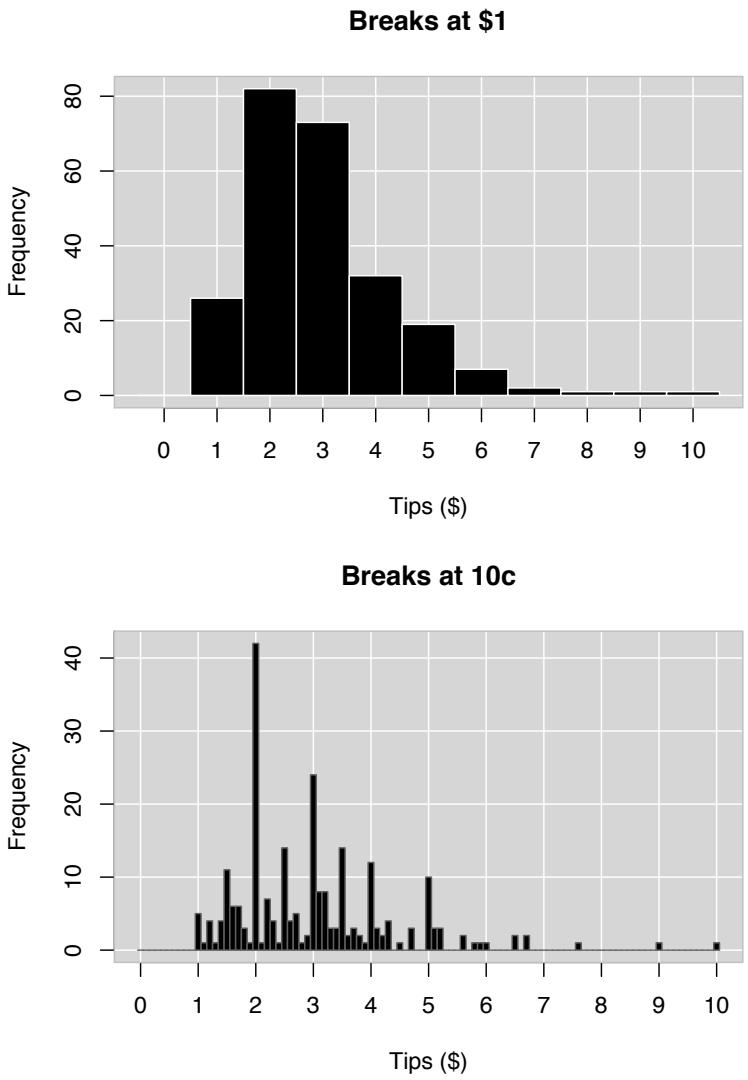

Fig. 1.1. Histograms of actual tips with differing barwidth: $1, 10c. The power of an interactive system allows bin width to be changed with slider.

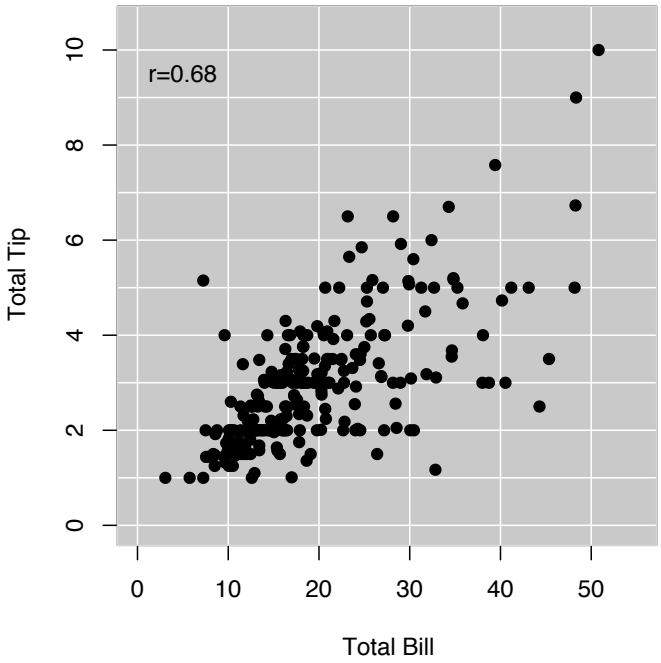

Fig. 1.2. Scatterplot of Total Tip vs Total Bill: More points in the bottom right indicate more cheap tippers than generous tippers.

ate relationships for the first time when we produced the plots in Figure 1.3. Each plot shows a subset obtained by partioning the data according to two binary variables. The statistical term for partitioning based on variables is "conditioning". For example, the top left plot shows the dining parties that meet the condition that the bill payer was a male non-smoker: *sex* = male and *smoking* = False. In database terminology this plot would be called the result of "drill-down". The idea of conditioning is richer than drill-down because it involves a structured partitioning of *all* the data as opposed to the extraction of a single partition.

Having generated the four plots, we arrange them in a two by two layout to reflect the two variables on which we conditioned. While the axes in each individual plot are *tip* and *bill*, the axes of the overall figure are *smoking* (vertical) and *sex* (horizontal). The arrangement permits us to make several kinds of comparisons and make observations about the partitions. For example, comparing the rows shows that smokers and non-smokers differ in the strength of the correlation between *tip* and *bill*, and comparing the plots in the top row shows that male and female non-smokers differ in that the larger

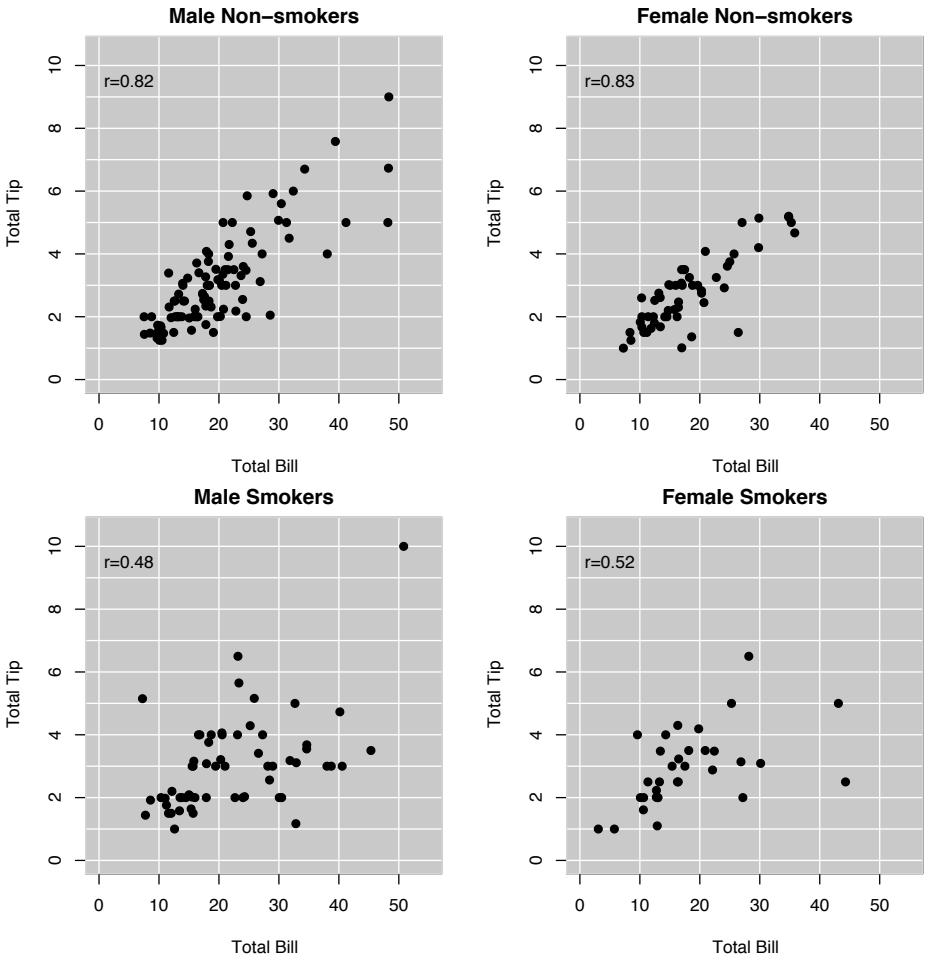

Fig. 1.3. Total Tip vs Total Bill by Sex and Smoker: There is almost no association between tip and total bill in the smoking parties, and, with the exception of 3 dining parties, when a female non-smokers paid the bill the tip was extremely consistent.

bills tend to be paid by men. In this way a few simple plots allow us to reason about relationships among four variables!

By contrast, an old-fashioned approach without graphics would be to fit some regression model. Without subtle regression diagnostics (which rely on graphics!), this approach would miss many of the above insights: the rounding of tips, the preponderance of cheap tippers, and perhaps the multivariate relationships involving the bill payer's sex and the group's smoking habits.

1.4 Getting Real: Process and Caveats

The preceding sections may have given a somewhat misleading impression of the process of data analysis. In our account the data had no problems; for example, there were no missing values and no recording errors. Every step was logical and necessary. Every question we asked had a meaningful answer. Every plot that was produced was useful and informative. In actual data analysis nothing could be further from the truth. Real data are rarely perfect; most choices are guided by intuition, knowledge and judgment; most steps lead to dead ends; most plots end up in the wastebasket. This may sound daunting, but while data analysis is a highly improvisational activity, it can be given some structure nonetheless.

To understand data analysis, and how visualization fits in, it is useful to talk about it as a process consisting of several stages:

- The problem statement
- Data preparation
- Exploratory data analysis
- Quantitative analysis
- Presentation

The problem statement: Why do you want to analyze this data? Underlying every data set is a question or problem statement. For the tipping data the question was provided to us from the data source: "What are the factors that affect tipping behavior?" This problem statement drives the process of any data analysis. Sometimes the problem is identified prior to a data collection. Perhaps it is realized after data becomes available because having the data available has made it possible to imagine new issues. It may be a task that the boss assigns, it may be an individual's curiosity, or part of a larger scientific endeavor to find a cure. Ideally, we begin an analysis with some sense of direction, as described by a pertinent question.

Data preparation: In the classroom, the teacher hands the class a single data matrix with each variable clearly defined. In the real world, it can take a great deal of work to construct clean data matrices. For example, data may be missing or misrecorded, they may be distributed across several sources, and the variable definitions and data values may be inconsistent across these sources. Analysts often have to invest considerable time in learning computing tools and domain knowledge before they can even ask a meaningful question about the data. It is therefore not uncommon for this stage to consume most of the efforts that go into a project. And it is also not uncommon to loop back to this stage after completing the following stages, to re-prepare and re-analyze the data.

In preparing the tipping data, we would create a new variable called tip rate, because when tips are discussed in restaurants, among waiters, dining parties, and tourist guides, it is in terms of a percentage of total bill. We may

also create several new dummy variables for the day of the week, in anticipation of fitting a regression model. We didn't talk about using visualization to verify that we had correctly understood and prepared the tipping data. For example, that unusually large tip could have been the result of a transcription error. Graphics identified the observation as unusual and the analyst might use this information to search the origins of the data to check the validity of the numbers for this observation.

Exploratory data analysis: We gave you some of the flavor of this stage in the above analysis of the waiter's tips. We checked the distribution of individual variables, we looked for unusual records, we explored relationships among multiple variables, and we found some unexpected patterns. To complete this exploration, we would also add numerical summaries to the visual analysis.

It's this stage in the analysis that we make time to "play in the sand", to allow us to find the unexpected, and come to some understanding of the data we're working with. We like to think of this as a little like travelling. We may have a purpose in visiting a new city, perhaps to attend a conference, but we need to take care of our basic necessities, such as, find eating places, shops where we can get our supplies, a gas station to fill up at. Some of the direction will be determined, guided by the concierge, or other locals, but some of the time we wander around by ourselves. We may find a cafe with just the type of food that we like instead of what the concierge likes, or a gift shop with just the right things for a family member at home, or we might find a cheaper gas price. This is all about getting to know the neighborhood. At this stage in the data analysis we relax the focus on the problem statement, and explore broadly different aspects of the data. For the tipping data, although the primary question was about the factors affecting tip behavior, we found some surprising aspects generally about tipping behavior, beyond this question: the rounding of tips, the prevalence of cheap tippers, and heterogeneity in variance corresponding to covariates.

Exploratory data analysis has evolved with the evolution of fast, graphically enabled desktop computers, into a highly interactive, real-time, dynamic and visual process. Exploratory data analysis takes advantage of technology, in a way that Tukey envisioned and experimented with on specialist hardware 40 years ago: "Today, software and hardware together provide far more powerful factories than most statisticians realize, factories that many of today's most able young people find exciting and worth learning about on their own" (Tukey 1965). It is characterized by direct manipulation graphics, and dynamic graphics: plots that respond in real time to an analyst's queries and change dynamically to re-focus, linking information from other sources, and re-organize information. The analyst is able to work thoroughly over the data in a rapid time, slipping out of dead-ends, and chasing down new leads. The high-level of interactivity is enabled by fast to compute, decoration-devoid, graphics, which are generally not adequate for presentation purposes in the later stage of data analyis. In general this means that it is necessary to re-

create the revealing plots in a more exacting and static form to communicate results.

Quantitative analysis: This stage consists of statistical modeling and statistical inference. It is where we focus in on the primary question of interest. With statistical models we summarize complex data; models often help us decompose data into estimates of signal and noise. With statistical inference, we try to assess whether a signal is real. It's widely accepted that data visualization is an important part of exploratory data analysis, but it's not as well understood that it also plays an important role at this stage. The role played is both in diagnosing a model in relation to the data, and to better understand a model.

For the tips data, we haven't yet addressed the primary question of interest. To do this we'd likely fit a regression model using tip rate as the response and the remaining variables (except tip, total bill) as the explanatory variables (Sex, Smoker, Size, Time, Day). When we do this, of all the variables only Size has a significant regression coefficient, resulting in the model $\hat{TipRate} = 0.18 - 0.01 \times Size$ which explains just 2% of the variation in tip rate. The model says that starting from a baseline tip rate of 18% the amount drops by 1% for each additional diner in a party. This is the model answer in Bryant & Smith (1995). Figure 1.4 shows this model, and the underlying data. The data is jittered horizontally to alleviate overplotting from the discreteness of the Size variable. The data values are spread widely around the model. And there are very few data points for parties of size 1, 5, 6, which makes us question the validity of the model in these regions of the data space. What have we learned about tipping behavior? Size of the party explains only a very small amount of the variation in tip rate. The signal is very weak relative to the noise. Is it a useful model? Its used: most restaurants today factor the tip into the bill automatically for larger dining parties.

Most problems are more complex than the tips data, and the models commonly used are often more sophisticated. Fitting a model produces its own data, in the form of model estimates and diagnostics. Many models involve simulation from the model giving samples from posterior distributions. The model outputs are data that can be explored for the pleasure of understanding the model. We may plot parameter estimates and confidence regions. We may plot the posterior samples.

Plotting the model in relation to the data is important, too. There is a temptation to ignore the data at this point, in favor of the simplification provided by a model. But a lot can be learned from whats left out of the model: We would never consider teaching regression analysis without teaching residual plots. A model is a succinct explanation of the variation in the data, a simplification. With a model we can make short descriptive statements: As the size of the dining party increases an additional person the tip rate decreases by 1%. Pictures can help to assess if a model is too simple for the data, because a well-constructed graphic can provide a digestible summary of

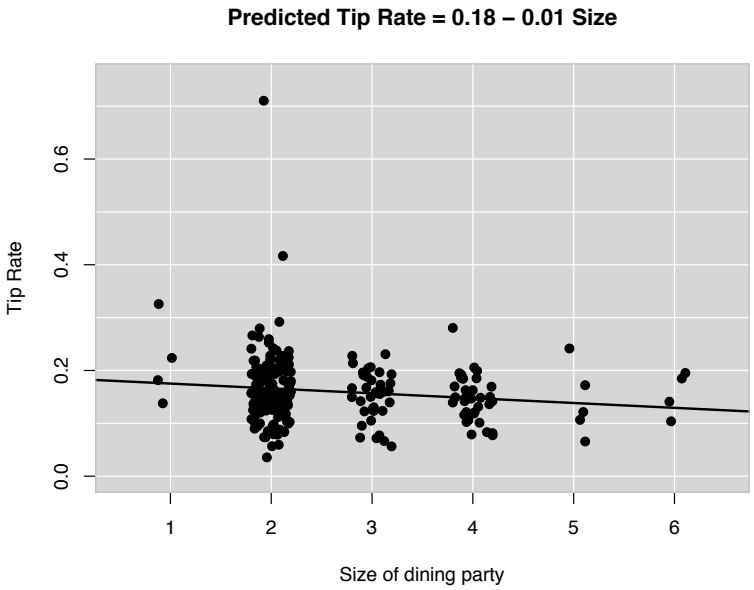

Fig. 1.4. What are the factors that affect tipping behavior? This is a plot of the best model, along with the data. (Points are jittered horizontally to alleviate overplotting from the discreteness of the Size variable.) There is a lot of variation around the regression line: There is very little signal relative to noise. In addition there are very few data points for parties of size 1, 5, 6, raising the question of the validity of the model in these extremes.

complex structure. A problem with a model may be immediately obvious from a plot. Graphics are an essential part of model diagnostics. A graphic should be self-explanatory, but it is usually assisted by a detailed written or verbal description. "A picture saves a thousand words!" Or does it take a thousand words to explain? The beauty of a model is that the explanation is concise, and precise. But pictures are powerful tools in a data analysis, that our visual senses embrace, revealing so much that a model alone cannot.

The interplay of EDA and QA: Is it data snooping?

Exploratory data analysis can be difficult to teach. Says Tukey (1965) "Exploratory data analysis is NOT a bundle of techniques.... Confirmatory analysis is easier to teach and compute...." In the classroom, the teacher explains a method to the class and demonstrates it on the single data matrix, and then repeats this with another method. Its easier to teach a stream of seemingly disconnected methods, applied to data fragments, than put it all together. EDA, as a process, is very closely tied to data problems. There usually isn't time to let students navigate their own way through a data analysis, to spend a long time cleaning data, to make mistakes, recover from them, and

synthesize the findings into a summary. Teaching a bundle of methods is an efficient approach to covering substantial material. But its useless unless the student can put it together. Putting it together as being simply a matter of common sense. Yet, common sense is rare.

Because EDA is a very graphical activity, it bring rise to a suspicion of data snooping. With the tipping data, from a few plots we learned an enormous amount of information about tipping: that there is a scarcity of generous tippers, that the variability in tips increases extraordinarily for smoking parties, and that people tend to round their tips. These are very different types of tipping behaviors than we learned from the regression model. The regression model was not compromised by what we learned from graphics. We snooped into the data. In reality, making pictures of data is not necessarily data snooping. If the purpose of an analysis is clear then making plots of the data is "just smart", and we make many unexpected observations about the data, resulting in a richer and more informative analysis. We particularly like the quote by Crowder & Hand (1990): "The first thing to do with data is to look at them.... usually means tabulating and plotting the data in many different ways to 'see whats going on'. With the wide availability of computer packages and graphics nowadays there is no excuse for ducking the labour of this preliminary phase, and it may save some red faces later."

Presentation: Once an analysis has been completed, the results must be reported, either to clients, managers or colleagues. The results probably take the form of a narrative, and include quantitative summaries such as tables, forecasts, models, and graphics. Quite often, graphics form the bulk of the summaries.

The graphics included in a final report may be a small fraction of the graphics generated for exploration and diagnostics. Indeed, they may be different graphics altogether. They are undoubtedly carefully prepared for their audience. The graphics generated during the analysis are meant for the analyst only and thus need to be quickly generated, functional but not polished. This is a dilemma for these authors who have much to say about exploratory graphics, but need to convey it in printed form. We have carefully re-created every plot in this book!

As we have already said, these broadly defined stages do not form a rigid recipe. Some of the stages overlap, and occasionally some are skipped. The order is often shuffled and groups of steps reiterated. What may look like a chaotic activity is often improvisation on a theme loosely following the "recipe".

1.5 Interactive Investigation

Thus far, all the observations on the tipping data have been made using static graphics - the purpose up to this point has been to communicate the impor-

tance of plots in the context of data analysis. Although we no longer hand-draw plots, static plots are computer-generated for a passive paper medium, to be printed and stared at by the analyst. Computers, though, allow us to produce plots for active consumption. This book is about interactive and dynamic plots, which is the material in the following chapters, but we will give a hint to the way interactive plots enhance the data analysis process we've just described.

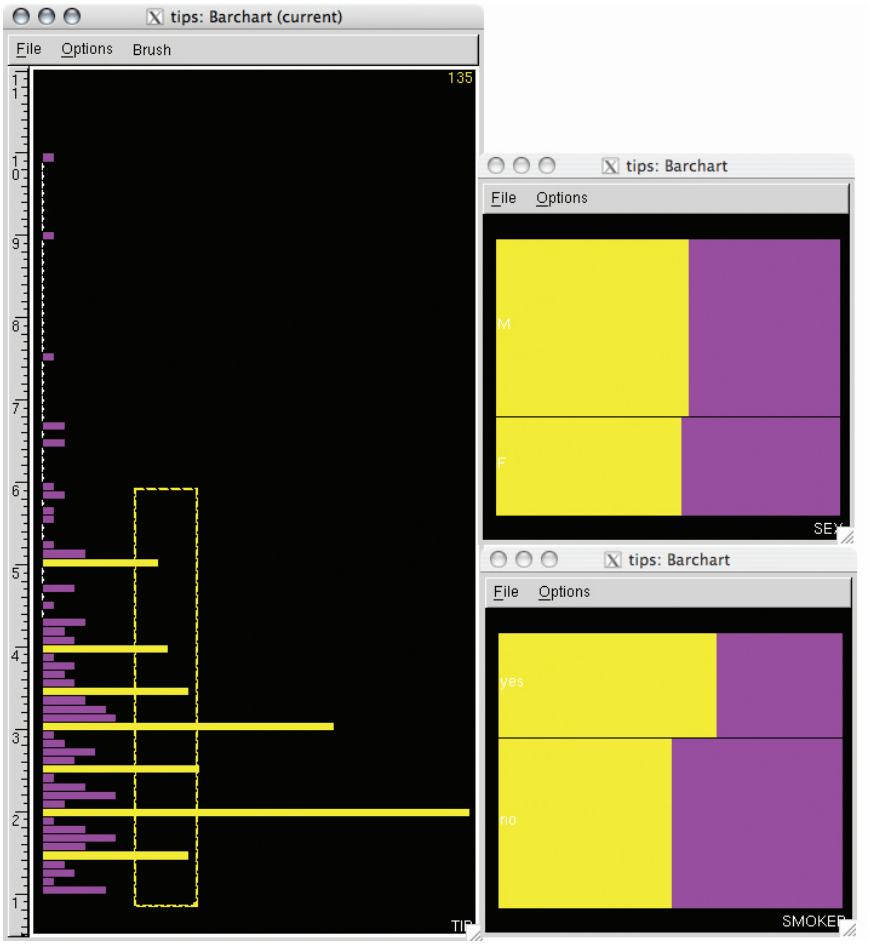

Fig. 1.5. Bins of whole and half-dollar amounts are highlighted. This information is linked to spine plots of gender of the bill payer and smoking status of the dining party. The proportion of males and females in this group that rounds tips is roughly equal, but interestingly the proportion of smoking parties who round their tips is higher than non-smoking parties.

The tips data is simple. Most of the interesting features can be discovered using static plots. Yet, interacting with the plots reveals more and enables the analyst to pursue follow-up questions. For example, we could address a new question, arising from the current analysis, such as "Is the rounding behavior of tips predominant in some demographic group?" To investigate we probe the histogram, highlight the bars corresponding to rounded tips, and observe the pattern of highlighting in the linked plots (Figure 1.5). Multiple plots are visible simultaneously, and the highlighting action on one plot generates changes in the other plots. The two additional plots here are spine plots (), used to examine the proportions in categorical variables. For the highlighted subset of dining parties, the ones who rounded the tip to the nearest dollar or half-dollar, the proportion of bill paying males and females is roughly equal, but interestingly, the proportion of smoking parties is higher than non-smoking parties. This might suggest another behavioral difference between smokers and non-smokers: a larger tendency for smokers than non-smokers to round their tips. If we were to be skeptical about this effect we would dig deeper, make more graphical explorations and numerical models. By pursuing this with graphics we'd find that the proportion of smokers who round the tip is only higher than non-smokers for full dollar amounts, and not for half-dollar amounts.

This is the material that this book describes: how interactive and dynamic plots are used in data analysis.

1.6 Whats in this book?

We have just said that visualization has a role in most stages of data analysis, all the way from data preparation to presentation. In this book, however, we will concentrate on the use of graphics in the exploratory and diagnostic stages. We concentrate on graphics that can be probed and brushed, direct manipulation graphics, and graphics that can change temporally, dynamic graphics.

The reader may note the paradoxical nature of this claim about the book: Once a graphic is published, is it not by definition a presentation graphic? Yes and no: as in the example of the waiter's tips, the graphics in this book have all been carefully selected, prepared, and polished, but they are shown as they appeared during our analysis. Only the last figure for the waiter's tips is shown in raw form, to introduce the sense of the rough and useful nature of exploratory graphics.

The first chapter opens our toolbox of plot types and direct manipulation modes. The missing data chapter is the material most related to a data preparation stage. It is presented early because handling missing values is one of the first obstacles in analysing data. The chapters on supervised classification and cluster analysis have both exploratory and diagnostic material. A chapter on inference hints at ways we can assess our subjective visual senses.

Chapter 3

The Toolbox

The methods used throughout the book are based on a small set of types of plot and user direct manipulation. We call these our tools. In this chapter, we open our toolbox and take a peek at the tools. These are the basics from which we construct graphics and connect multiple plots to see into high-dimensional spaces.

3.1 Notation

It will be helpful to have a shorthand for describing what information is used to generate a plot, and what is shared between plots when the user changes elements of a plot. We'll introduce this notation using the Australian crab data. Its a table of numbers of the form:

sp	sex	FL	RW	CL	CW	BD
1	1	8.1	6.7	16.1	19.0	7.0
1	1	8.8	7.7	18.1	20.8	7.4
1	1	9.2	7.8	19.0	22.4	7.7
1	1	9.6	7.9	20.1	23.1	8.2
1	2	7.2	6.5	14.7	17.1	6.1
1	2	9.0	8.5	19.3	22.7	7.7
1	2	9.1	8.1	18.5	21.6	7.7
2	1	9.1	6.9	16.7	18.6	7.4
2	1	10.2	8.2	20.2	22.2	9.0
2	2	10.7	9.7	21.4	24.0	9.8
2	2	11.4	9.2	21.7	24.1	9.7
2	2	12.5	10.0	24.1	27.0	10.9

The table can be considered to be a data matrix having n observations and p variables denoted as:

$$\mathbf{X} = [\mathbf{X}_1\ \mathbf{X}_2\ \ldots\ \mathbf{X}_p] = \begin{bmatrix} X_{11} & X_{12} & \ldots & X_{1p} \\ X_{21} & X_{22} & \ldots & X_{2p} \\ \vdots & \vdots & & \vdots \\ X_{n1} & X_{n2} & \ldots & X_{np} \end{bmatrix}_{n \times p}$$

The first and second columns of crabs data $(\mathbf{X}_1, \mathbf{X}_2)$ are the values for species and sex, which are the two categorical variables in the data. The subsequent five columns $(\mathbf{X}_3, \ldots, \mathbf{X}_7)$ are the physical measurements taken on each crab. Thus $p = 7$ for the crabs data, and there are $n = 12$ observations shown in the table above.

For this data we are interested in understanding the variation in the five physical variables, particularly if the variation is different depending on the two categorical variables. In statistical language, we may say that we are interested in the *joint distribution* of the five physical measurements *conditional* on the two categorical variables. A plot of one column of numbers displays the *marginal distribution* of one variable. Similarly a plot of two columns of the data displays the marginal distribution of two variables. Ultimately we want to describe the distribution of values in the five-dimensional space of the physical measurements.

Building insight about structure in high-dimensional spaces starts simply. We build from univariate and bivariate plots up to multivariate plots. Real-valued and categorical variables need different handling. The next sections describe the tools for plotting real-valued and categorical variables from univariate to multivariate plots.

3.2 Plot Types

3.2.1 Real-Valued Variables

1-D Plots

The 1-D plots such as histograms, box plots or dot plots are important to examine the marginal distributions of the variables. What is the shape of spread of values, unimodal, multimodal, symmetric or skewed? Are there clumps or clusters of values? Are there values extremely different from most of the values? These types of observations about a data distribution can only be made by plotting the data. There are two types of univariate plots for real-valued variables that are regularly used in this book: the textured dot plot and an average shifted histogram (ASH) dot plot. These univariate plots preserve the individual observation: one row of data generates one point on the plot. This is useful for linking information between plots using direct manipulation, which is discussed later in the chapter. Conventional histograms, where bar height represents the count of values within a bin range, are used occasionally. These are considered to be area plots because one or more observations are pooled into each bin and the group represented by a rectangle: the individual case identity is lost. Each of the univariate plot types uses one column

of the data matrix, $\mathbf{X}_i, i = 1, \ldots, p$. The plots in Figures 3.1, 3.2 show the column $\mathbf{X}_3$.

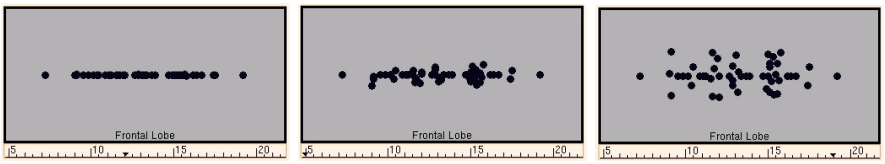

Figure 3.1. *Textured dot plot, unjittered at left, and then with different amount of jitter center and right. Without jittering overplotting can obscure the density of points. Textured dot plots use a combination of random and constrained placement of points. In the frontal lobe (FL) variable of the crabs data we can see a bimodality in the distribution of values, with a lot of cases clustered near 15 and then a gap to a further cluster of values below 12.*

The textured dot plot Figure 3.1 uses a method described in Tukey & Tukey (1990). In a dot plot each case is represented by a dot. The values are binned, incorporating the size of the plot window so it will fit, and the dot plot is calculated on the binned data. This means that there are common values, or ties. In a traditional dot plot, when there are several cases with the same value they will be overplotted at the same location in the plot, making it difficult to get an accurate read of the distribution of the variable. One fix is to *jitter*, or *stack* the points giving each point its own location on the page. The textured dot plot is a variation of jittering, that spreads the points in a partly constrained and partly random manner. When there are very few cases with the same data value (< 3) the points are placed at constrained locations, and when there are more than three cases with the same value the points are randomly spread. This approach minimizes artifacts due purely to the jitter.

The ASH plot in Figure 3.2 is due to Scott (1992). In this method, several histograms are calculated using the same bin width but different origins, and the averaged bin counts at each data point are plotted. His algorithm has two key parameters: the number of bins, which controls the bin width, and the number of histograms to be computed. The effect is a smoothed histogram - a histogram that allows us to retain case identity so that the plots can be linked case by case to other scatterplots.

2-D plots

Plots of two variables are important for examining the joint distribution of two variables. This may be a marginal distribution of a multivariate distribution, as is the case here with the Australian crabs data, where the two variables are a subset of the five physical measurement variables. Each point represents a case. When we plot two variables like this we are interested in detecting and describing the dependence between the two variables, which may be linear or non-linear or non-existent, and the deviations from the dependence such as outliers or clustering or

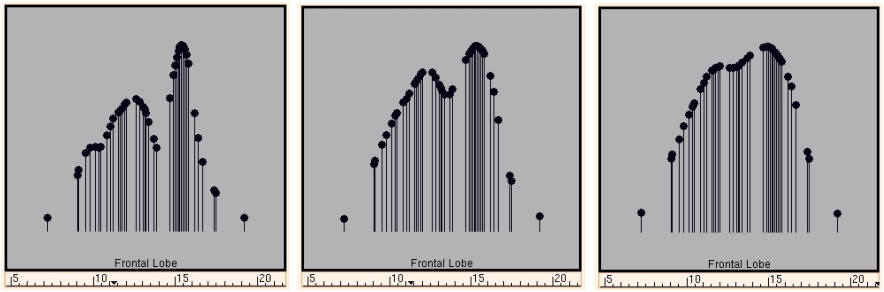

Figure 3.2. *Average shifted histograms, using 3 different smoothing parameter values. The variable frontal lobe appears to be bimodal, with a cluster of values near 15 and another cluster of values near 12. With a large smoothing window (right plot) the bimodal structure is washed out to result in a near univariate density. As we have seen in the tip example in Chapter 1 examining variables at several sizes of bin width can be useful for uncovering different aspects of a distribution.*

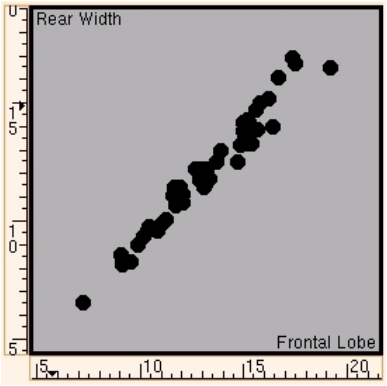

Figure 3.3. *Scatterplot of two variables.*

heterogeneous variation. In this book scatterplots are used, but in general we'd like the ability to overlay density information using contours, color or grey scale. In the situation where one variable might be considered a response and the other the explanatory variable it may be useful to add regression curves or smoothed lines.

p-D Plots

Parallel coordinate plots

Trace plots display each case as a line trace. The oldest method developed was Andrews curves, where the curves are generated by a Fourier decomposition of the variables $x_t = x_1/\sqrt{2} + x_2 \sin t + x_3 \cos t + x_4 \sin 2t + \ldots \quad -\pi < t < \pi$ There is a close connection between Andrews curves and motion graphics such as the tour (discussed later). If we fix t, then the coefficients $(1/\sqrt{2}, \sin t, \cos t, \ldots)$

effectively define a projection vector. So we have a continuous time sequence of 1-D projections. In an Andrews curve plot the horizontal axis is used to display time, and the vertical axis shows the projected data value, and the sequence of projections of each case is shown as a curve. The main problem with Andrews curves is that the Fourier decomposition doesn't uniformly reach all possible 1-D projections.

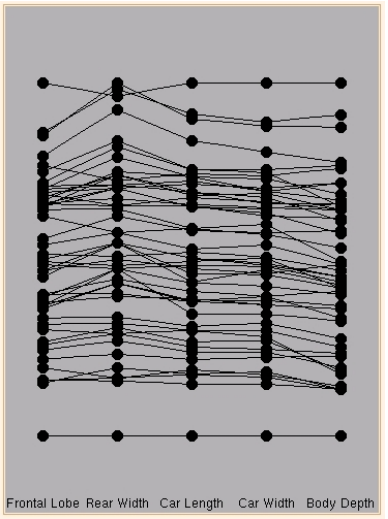

Figure 3.4. *Parallel coordinate plot of the five physical measurement variables of the Australian crabs data. From this plot we see two major points of interest: one crab is uniformly much smaller than the other crabs, and that for the most part the traces for each crab are relatively flat which suggests that the variables are strongly correlated.*

Parallel coordinate plots (Inselberg 1985, Wegman 1990) are increasingly commonly used for data visualization. They are constructed by laying out the axes in a parallel manner rather than the usual orthogonal axes of the Cartesian coordinate system. Cases are represented by a line trace connecting the case value on each variable axis. There is some neat high-level geometry underlying the interpretation of parallel coordinate plots. It should be also be noted that the order of laying out the axes can be important for structure detection, and it may be that re-ordering the layout will allow different structure to be perceived. Parallel coordinates are similar to profile plots, especially common in plotting longitudinal data and repeated measures, or interaction plots, when plotting experimental data with several factors. They may actually date back as far as d'Ocagne (1885), which showed that a point on a graph of Cartesian coordinates transforms into a line on an alignment chart, that a line transforms to a point, and, finally, that a family of lines or a surface transforms into a single line (Friendly & Denis 2004). Figure 3.4 shows the five physical measurement variables of the Australian crabs data as a parallel coordinate plot. From this plot we see two major points of interest: one crab is uniformly much smaller than the other crabs, and that for the most part the

traces for each crab are relatively flat which suggests that the variables are strongly correlated.

Tours

Motion is one of the basic visual tools we use to navigate our everyday environment. When we play hide-and-seek we may search for signs of a person such as a slight movement of a curtain, or a glimpse of an arm being whipped behind the door. To cross a street safely we can gauge quickly if a car is moving towards us or away from us. Motion is used effectively in computer graphics to represent 3D scenes. For data tours can be used to generate motion paths of projections of the p-D space. Tours are created by generating a sequence of low-dimensional projections of a high-dimensional space. Let $\mathbf{A}$ be a $p \times d$ projection matrix where the column are orthonormal, then a d-D projection of the data is defined to be

$$\mathbf{XA} = \begin{bmatrix} X_{11}A_{11} + \ldots + X_{1p}A_{p1} & \ldots & X_{11}A_{1d} + \ldots + X_{1p}A_{pd} \\ X_{21}A_{11} + \ldots + X_{2p}A_{p1} & \ldots & X_{21}A_{1d} + \ldots + X_{2p}A_{pd} \\ \vdots & & \vdots \\ X_{n1}A_{11} + \ldots + X_{np}A_{p1} & \ldots & X_{n1}A_{1d} + \ldots + X_{np}A_{pd} \end{bmatrix}_{n \times d}$$

Here are several examples. If $d = 1$ and $\mathbf{A} = (1 \;\; 0 \;\; \ldots \;\; 0)'$ then variable, of the data,

$$\mathbf{XA} = [X_{11} \;\; X_{21} \;\; \ldots \;\; X_{n1}]'.$$

If $\mathbf{A} = (\frac{1}{\sqrt{2}} \;\; \frac{-1}{\sqrt{2}} \;\; 0 \;\; \ldots \;\; 0)'$ then

$$\mathbf{XA} = \left[\frac{(x_{11} - x_{12})}{\sqrt{2}} \;\; \frac{(x_{21} - x_{22})}{\sqrt{2}} \;\; \ldots \;\; \frac{(x_{n1} - x_{n2})}{\sqrt{2}} \right]',$$

which is a contrast of the first two variables in the data table. If $d = 2$ and

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \text{ then } \mathbf{XA} = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \\ \vdots & \vdots \\ X_{n1} & X_{n2} \end{bmatrix},$$

the first two columns of the data matrix. Generally the values in $\mathbf{A}$ can be any values between $[-1, 1]$ with the constraints that the squared values for each column sum to 1 (normalized) and the inner product of two columns sums to 0 (orthogonal). The sequence must be dense in the space, so that all possible low-dimensional projections are equally likely to be chosen. The sequence can be viewed over time, like a movie, hence the term motion graphics, or if the projection dimension is 1, laid out into tour curves, similar to Andrews curves.

The plots in Figure 3.5 show several 2D projections of the five physical measurement variables in the Australian crabs data. The left-most plot shows the

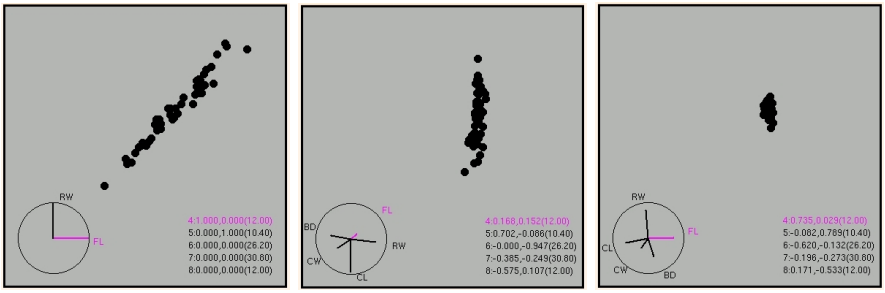

Figure 3.5. *Three tour 2D projections of the Australian crabs data.*

projection of the 5D data into the first two variables. The values of the projection matrix, **A**, are shown at right in the plot, along with the range of the data values for that variable in parentheses. Columns 4-8 of the data matrix are included as active variables in the tour. The data values for each variable, each column of the data matrix, are scaled to range between 0 and 1 using the minimum and maximum. For example, the first column of numbers is scaled using $(X_{i1} - min\{X_{11}, \ldots, X_{n1}\})/range\{X_{11}, \ldots, X_{n1}\}, \quad i = 1, \ldots, n$. Thus to reproduce the plots above we would scale each row of projection matrix **A** by dividing by the range of each variable. The plot at right is produced by setting

$$\mathbf{A} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0.735/12.0 & 0.029/12.0 \\ -0.082/10.4 & 0.789/10.4 \\ -0.620/26.2 & -0.132/26.2 \\ -0.196/30.8 & -0.273/30.8 \\ 0.171/12.0 & -0.533/12.0 \end{bmatrix}$$

and similarly for the other two plots. The circle in the bottom left of each plot displays the axis of the data space. In this data the data space is 5D, defined by 5 orthonormal axes. This may be hard to picture, but this conceptual framework underlies much of multivariate data analysis. The data and algorithms operate in p-D Euclidean space. In this projection just the two axes for the first two variables are shown because the other three are orthogonal to the projection. (The purple color indicates that this variable is the manipulation variable. Its projection coefficient can be manually controlled, which is discussed later in this section.) What can we see about the data? The two variables are strongly linearly related: crabs with small frontal lobe also have small rear width. The middle and right-side plots show arbitrary projections of the data. What we learn about the crabs data from running the tour on the five physical measurement variables is that the points lie on a 1D line in 5D.

Figure 3.6 shows two 1-D projections of the five physical measurement variables of the Australian crabs data. The vertical direction is used to display the

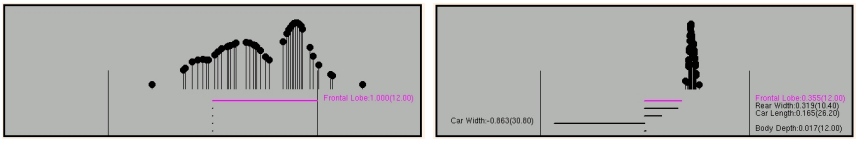

Figure 3.6. *Two tour 1D projections of the Australian crabs data.*

density (ASH plot) of the projected data. The left-most plot is the projection into the first axis of the data space, yielding the first column of the data table, the first variable. The right-most plot shows an arbitrary projection when the points are almost all at the same position, that is, in this direction there is virtually no variance in the data.

Figure 3.7. *Two tour 2x1D projections of the Australian crabs data.*

Figure 3.7 shows two 2x1D projections of the Australian crabs data. This method is useful when there are two sets of variables in the data, one or more response variables and several explanatory variables. The vertical direction in these plots are used for just one variable, species, and the horizontal axis is used to project the five physical measurement variables. Here we would be looking for a combination of the five variables which generates a separation of the two species.

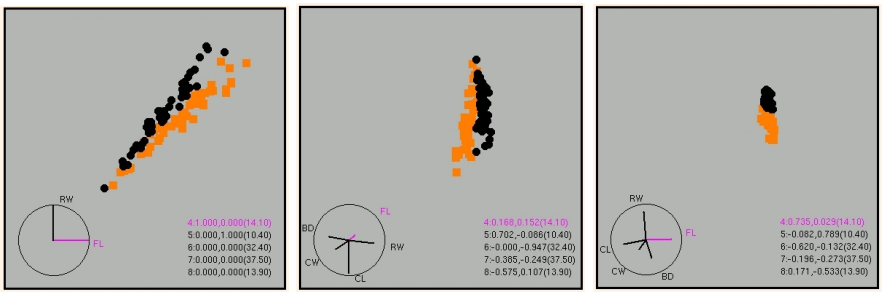

Figure 3.8. *Three tour 2D projections of the Australian crabs data, where two different species are distinguished using color and glyph.*

Note about categorical variables: Generally the tour is a good method for finding patterns in real-valued variables. It is not generally useful to include categorical

variables in the selection of variables used in the tour. There are a few exceptions. But as mentioned earlier the tour is giving the analyst insight into p-D Euclidean space, and categorical variables typically inhabit something different to Euclidean space. If you view categorical variables the dominant pattern will be the gaps between data values that are due to the discreteness of the data. This can distract attention from finding interesting patterns. If there are categorical variables in the data then the best way to code them into the tour is to use color or glyph to represent the different categories Figure 3.8.

How do we choose the projection to show?

There are three ways in our toolbox: random, projection pursuit and manual. The default method for choosing the new projection to view is to use a random sequence, which we call the grand tour. The projections are determined by randomly selecting a projection (anchor basis) from the space of all possible projections, and interpolating along a geodesic path from the current projection to the newly selected projection (target basis), showing all the intermediate projections. It may be considered to be an interpolated random walk over the space of all projections. This method is discussed in detail in Asimov (1985), more simply in Buja & Asimov (1986), and more technically in Buja, Cook, Asimov & Hurley (1997). The algorithm that we use creates an interpolation between two planes and follows these steps:

1. Given a starting $p \times d$ projection $\mathbf{A}_a$, describing the starting plane, create a new target projection $\mathbf{A}_z$, describing the target plane. The projection may also be called an orthonormal frame. A plane can be described by an infinite number of frames. To find the optimal rotation of the starting plane into the target plane we need to find the frames in each plane which are the closest.

2. Determine the shortest path between frames using singular value decomposition. $\mathbf{A}_a'\mathbf{A}_z = \mathbf{V}_a\Lambda\mathbf{V}_z'$, $\Lambda = \text{diag}(\lambda_1 \geq \ldots \geq \lambda_d)$, and the principal directions in each plane are $\mathbf{B}_a = \mathbf{A}_a\mathbf{V}_a, \mathbf{B}_z = \mathbf{A}_z\mathbf{V}_z$. The principal directions are the frames describing the starting and target planes which have the shortest distance between them. The rotation is defined with respect to these principal directions. The singular values, $\lambda_i, i = 1, \ldots, d$, define the smallest angles between the principal directions.

3. Orthonormalize $\mathbf{B}_z$ on $\mathbf{B}_a$, giving $\mathbf{B}_*$, to create a rotation framework.

4. Calculate the principal angles, $\tau_i = \cos^{-1}\lambda_i, i = 1, \ldots, d$.

5. Rotate the frames by dividing the angles into increments, $\tau_i(t)$, for $t \in (0, 1]$, and create the i^{th} column of the new frame, $\mathbf{b}_i$, from the i^{th} columns of $\mathbf{B}_a$ and $\mathbf{B}_*$, by $\mathbf{b}_i(t) = \cos(\tau_i(t))\mathbf{b}_{ai} + \sin(\tau_i(t))\mathbf{b}_{*i}$. When $t = 1$, the frame will be $\mathbf{B}_z$.

6. Project the data into $\mathbf{A}(t) = \mathbf{B}(t)\mathbf{V}_a$.

7. Continue the rotation until $t = 1$. Set the current projection to be $\mathbf{A}_a$ and go back to step 1.

In the grand tour the target projection is chosen randomly, by standardizing a random vector from a standard multivariate normal distribution. Sample p values from a standard univariate normal distribution, resulting in sample from a standard multivariate normal. Standardizing this vector to have length equal to one gives a random value from a $(p-1)$-dimensional sphere, that is, a randomly generated projection vector. Do this twice to get a 2D projection, where the second vector is orthonormalized on the first.

In a projection pursuit guided tour (Cook, Buja, Cabrera & Hurley 1995*a*) the next target basis is selected by optimizing a function defining interesting projections. Projection pursuit seeks out low dimensional projections that expose interesting features of the high dimensional point cloud. It does this by optimizing a criterion function, called the projection pursuit index, over all possible d-dimensional (d-d) projections of p-dimensional (p-d) data,

$$\max f(\mathbf{XA}) \quad \forall \mathbf{A}$$

subject to the orthonormality constraints on $\mathbf{A}$. Projection pursuit results in a number of static plots of projections which are deemed interesting, in contrast to the dynamic movie of arbitrary projections that is provided by a grand tour. Combining the two in an interactive framework (guided tour) provides both the interesting views and the context of surrounding views allowing better structure detection and better interpretation of structure.

Most projection pursuit indices (for example, (Jones & Sibson 1987); (Friedman 1987); (Hall 1989); (Morton 1989); (Cook, Buja & Cabrera 1993); (Posse 1995)) have been anchored on the premise that to find the structured projections one should search for the most non-normal projections. Good arguments for this can be found in Huber (1985) and (Diaconis & Freedman 1984)). (We should point out that searching for the most non-normal directions is also discussed by Andrews, Gnanadesikan & Warner (1971) in the context of transformations to enhance normality of multivariate data.) This clarity of purpose makes it relatively simple to construct indices which "measure" how distant a density estimate of the projected data is from a standard normal density. The projection pursuit index, a function of all possible projections of the data, invariably has many "hills and valleys" and "knife-edge ridges" because of the varying shape of underlying density estimates from one projection to the next.

The projection pursuit indices in our toolbox include Holes, Central Mass, LDA, PCA (1D only). These are defined as follows:

Holes :

$$I_{Holes}(\mathbf{A}) = \frac{1 - \frac{1}{n}\sum_{i=1}^{n} \exp(-\frac{1}{2}\mathbf{y}_i\mathbf{y}_i')}{1 - \exp(-\frac{p}{2})}$$

where $\mathbf{y} = \mathbf{XA}$ is a $n \times d$ matrix of the projected data. For simplicity in these formula, it is assumed that $\mathbf{X}$ is sphered, has mean zero and variance-covariance equal to the identity matrix.

Central Mass :

$$I_{CM}(\mathbf{A}) = \frac{\frac{1}{n}\sum_{i=1}^{n}\exp(-\frac{1}{2}\mathbf{y}_i\mathbf{y}_i') - \exp(-\frac{p}{2})}{1-\exp(-\frac{p}{2})}$$

where $\mathbf{y} = \mathbf{XA}$ is a $n \times d$ matrix of the projected data. For simplicity in these formula, it is assumed that $\mathbf{X}$ is sphered, has mean zero and variance-covariance equal to the identity matrix.

LDA :

$$I_{LDA}(\mathbf{A}) = 1 - \frac{|\mathbf{A}'\mathbf{W}\mathbf{A}|}{|\mathbf{A}'(\mathbf{W}+\mathbf{B})\mathbf{A}|}$$

where $\mathbf{B} = \sum_{i=1}^{g} n_i(\bar{\mathbf{X}}_{i.} - \bar{\mathbf{X}}_{..})(\bar{\mathbf{X}}_{i.} - \bar{\mathbf{X}}_{..})'$, $\mathbf{W}\sum_{i=1}^{g}\sum_{j=1}^{n_i}(\mathbf{X}_{ij} - \bar{\mathbf{X}}_{i.})(\mathbf{X}_{ij} - \bar{\mathbf{X}}_{i.})'$ are the "between" and within sum of squares matrices from linear discriminant analysis, g =number of groups, $n_i, i = 1, ..., g$ is the number of cases in each group.

PCA : This is only defined for $d = 1$.

$$I_{PCA} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{y}_i^2$$

where $\mathbf{y} = \mathbf{XA}$.

All of the projection pursuit indices seem to work best when the data is sphered (transformed to principal component scores) first. Although the index calculations take scale into account, the results don't seem to be as good as with sphered data.

The Holes and Central Mass indices derive from the normal density function. They are sensitive to projections where there are few points, or a lot of points, in the center of the projection, respectively. The LDA index derives from the statistics for MANOVA (Johnson & Wichern 2002), and it is maximized when the centers of colored groups in the data are most far apart. The PCA index derives from principal component analysis and it finds projections where the data is most spread. Figures 3.10, **??** shows some results of projection pursuit guided tours on the crabs data.

The optimization algorithm is very simple and derivative-free. A new target frame is generated randomly. If the projection pursuit index value is larger the tour path interpolates to this plane. The next target basis is generated from a smaller neighborhood of this current maximum. The possible neighborhood of new target bases continues to be shrunk, until no new target basis can be found where the projection pursuit index value is higher than the current projection. This means that the viewer is at a local maximum of the projection pursuit index.

To continue touring, the user will need to revert to a grand tour, or jump out of the current projection to a new random projection.

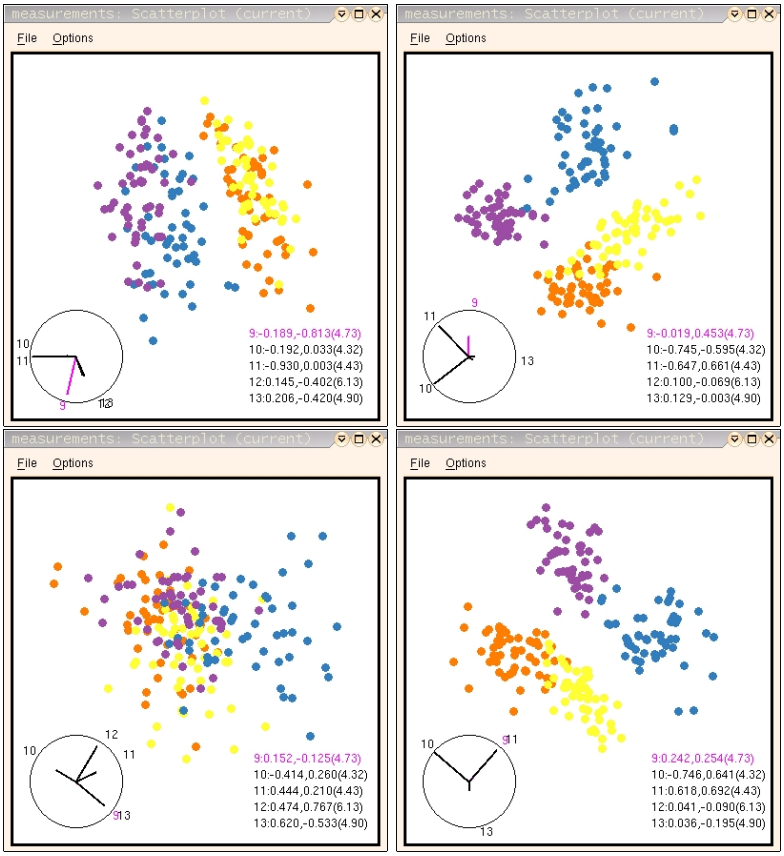

Figure 3.9. *Some results of 2D projection pursuit guided tours on the crabs data. (Top row) Two projections from the holes index show separation between the four colored classes. The holes index doesn't use the group information. It finds projections with few points in the center of the plot, which for this data corresponds to separations between the four clusters. (Bottom left) Projection from the central mass index. Notice that there is a heavier concentration of points inthe center of the plot. For this data its not so useful, but if there were some outliers in the data this index would help to find them. (Bottom right) Projection from the LDA index, reveals the four classes.*

With manual controls one variable is designated as the manip(ulation) variable and with mouse action the projection coefficient for this variable can be controlled, constrained by the coefficients of all the other variables. Manual control is available for 1D and 2D tours. In 1D tours it is straight forward - the manip variable is rotated into or out of the current projection.

For 2D tours, its a little complicated to define manual controls. When there are three variables, manual control works like a track ball. There are three rigid,

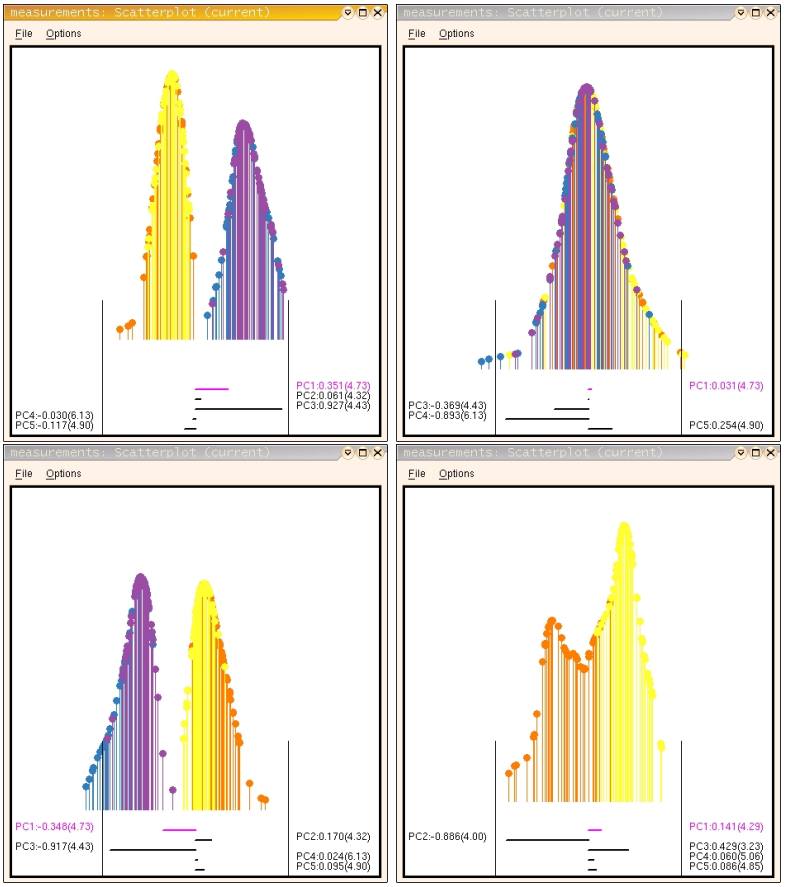

Figure 3.10. *Some results of 1D projection pursuit guided tours on the crabs data. (Top left) Projection from the holes index shows separation between the species. (Top right) Projection from the central mass index, shows a density having short tails. Not so useful for this data. (Bottom row) Two projections from the LDA index, reveals the species separation, which is the only projection found, because the index value for this projection is so much larger than for any other projection. The separation between sexes can only be found by subsetting the data into two separate groups and running the projection pursuit guided tour on each set.*

orthonormal axes, and the projection is rotated by pulling the lever belonging to manip variable (Figure 3.11). With four or more variables, a manipulation space is created from the current 2D projection, and a third axis arising from the manip variable. The 3D manip space results from orthonormalizing the current projection with the third axis. Then the coefficients of the manip variable can be controlled by pulling the lever of the manip belonging to the manip variable. Figure 3.12 illustrates how this works for four variables. This system means that the manip

variable is the only variable where we have full control over the coefficients. Other coefficients are constrained by both their contribution to the 3D manip space and the coefficients of the manip variable.

In practice to manually explore data, the user will need to choose several different variables to be the manip variable. Prior knowledge can be incorporated with manually controlled tours. The user can increase or decrease the contribution of a particular variable to a view to examine how a particular variable contributes to any structure. Manual control allow the user to assess the sensitivity of the structure to a particular variable, or sharpen or refine a structure exposed with the grand or guided tour.

The manual control is not a method that can adequately provide coverage of the space of projections. It is useful for determining how a particular variable affects the structure in a view, that is, assessing the sensitivity of the structure to the variable. It is also useful if you suspect that there is structure in certain variables.

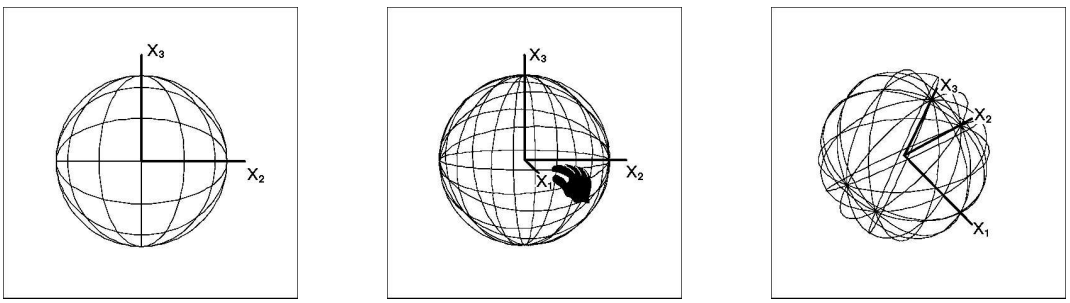

Figure 3.11. *A schematic picture of trackball controls. The semblance of a globe is rotated by manipulating the contribution of X_1 in the projection.*

The types of manual control available in GGobi are "oblique", adjust the variable contributionin any direction, "horizontal", allow adjustments only in the horizontal plot direction, "vertical", allow adjustments only in the vertical plot direction, "radial", allow adjustments only in the current direction of contribution, "angular", rotate the contribution within the viewing plane. These are demonstrated in Figure 3.13.

There is numerous literature on tour methods. Wegman (1991) discusses tours with higher than 2-D projections displayed as parallel coordinates. Wegman, Poston & Solka (1998) discusses touring with spatial data. Tierney (1991) discusses tours in the software XLispStat.

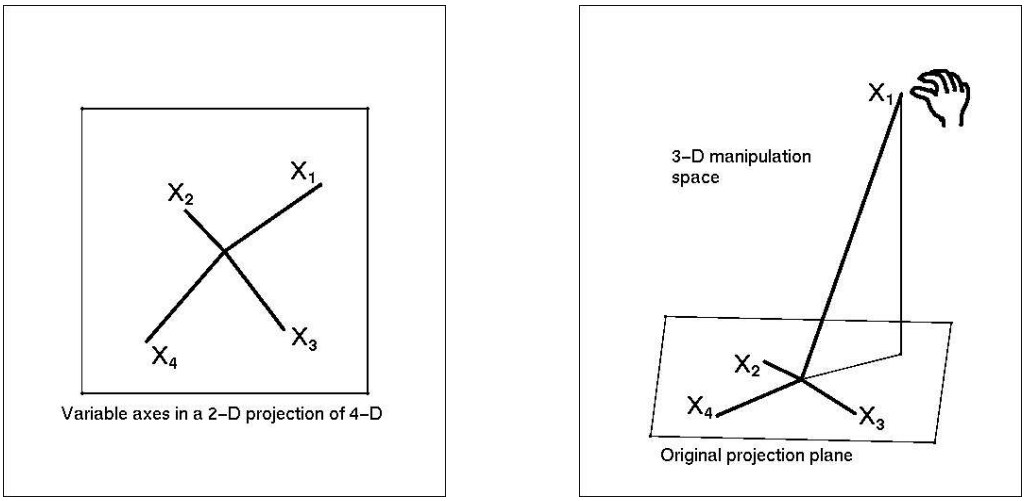

Figure 3.12. *Constructing the 3-dimensional manipulation space to manipulate the contribution of variable 1 in the projection.*

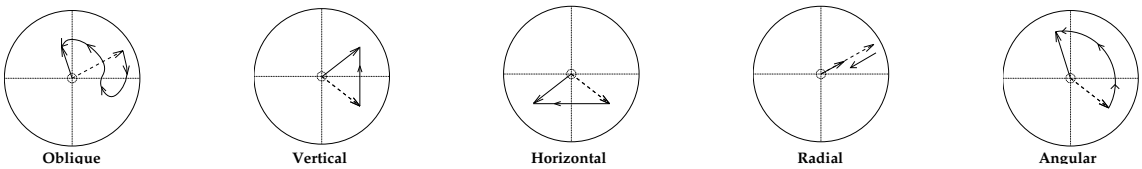

Figure 3.13. *Two-dimensional variable manipulation modes: dashed line represents variable contribution to the projection before manipulation and the solid line is the contribution after manipulation.*

Relationships between tours and numerical algorithms:

There is some relationship between tours and commonly used multivariate analysis methods.

In principal component analysis the principal component is defined to be $\mathbf{Y} = (\mathbf{X} - \bar{\mathbf{X}})\mathbf{A}$ where $\mathbf{A}$ is the matrix of eigenvectors from the eigen-decomposition of the variance-covariance matrix of the data, $\mathbf{S} = \mathbf{A}\Lambda\mathbf{A}'$. Thus a principal component is one linear projection of the data, and could be one of the projections shown by a tour. A biplot (Gabriel 1971) shows a view similar to the tour views: the coordinate axes are added to the plot of points in the first two principal components, which is analogous to the axis tree displayed in the tour plots. These axes are used to interpret any structure visible in the plot in relation to the original variables. (Figure 3.14 shows a biplot of the Australian crabs data alongside a tour plot showing a similar projection, constructed using manual controls.)

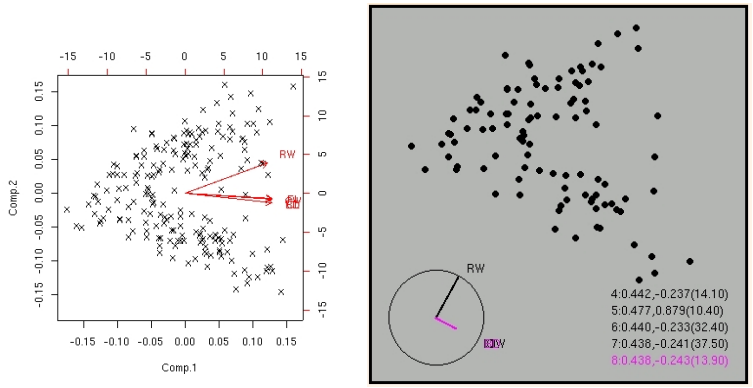

Figure 3.14. *The relationship between a 2-D tour and the biplot. (Left) Biplot of the five physical measurement variables of the Australian crabs data, (Right) the biplot as one projection shown in a tour, produced using the manually controlled tour.*

In linear discriminant analysis, Fishers linear discriminant, which is the linear combination of the variables that gives the best separation of the class means with respect the class covariance, can be considered to be one projection of the data that may be shown in a tour of the data. In support vector machines (Vapnik 1995) projecting the data into the normal to the separating hyperplane would be one projection provided by a tour.

Canonical correlation analysis and multivariate regression are examples of particular projections that may be viewed in a 2x1d tour.

3.2.2 Categorical Variables

To illustrate methods for categorical data we use the tipping behavior data, which has several categorical variables.

1-D plots

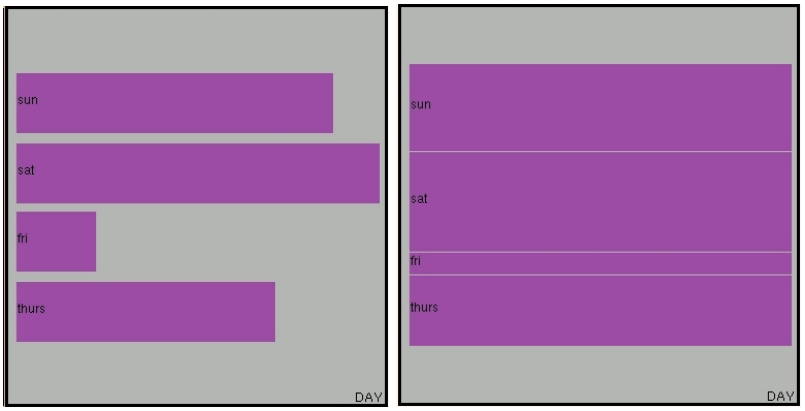

Figure 3.15. *(Left) Barchart of the day of the week in the tipping data. We can see that Friday has fewer diners than other days. (Right) Spine plot of the same variable, where width of the bar represents count.*

The most common way to plot 1-D categorical data is the bar chart (Figure 3.15), where the height of the bar reflects the relative count of each category. Other approaches are pie charts and spine plots. Pie charts use a circle divided like a pie to reflect the relative percentage of each category. Spine plots use the width of the bar to represent the category count.

p-D plots

Mosaic plots are an extension to spine plots, which are useful for examining dependence between several categorical variables. The rectangle of the page is divided on the first orientation into intervals sized by the relative counts of the first variable. Small rectangles result from divided on the second orientation using intervals representing the relative count of the second variable. Figure 3.16 shows a mosaic plot for two variables of the tipping data, day of the week and gender. The height of the boxes for males increase over the day of the week, and conversely it decreases for women, which shows that the relative proportion of males to females dining increases with day of the week.

3.2.3 Multilayout

Laying out plots in an organized manner allows us to examine marginal distributions in relation to one another. The scatterplot matrix is a commonly used multilayout method where all pairwise scatterplots are laid out in a matrix format which matches the correlation or covariance matrix of the variables. Figure 3.17 displays a scatterplot matrix of the five physical measurement variables of the males and females for one of the species of the Australian crabs data. Along the diagonal of

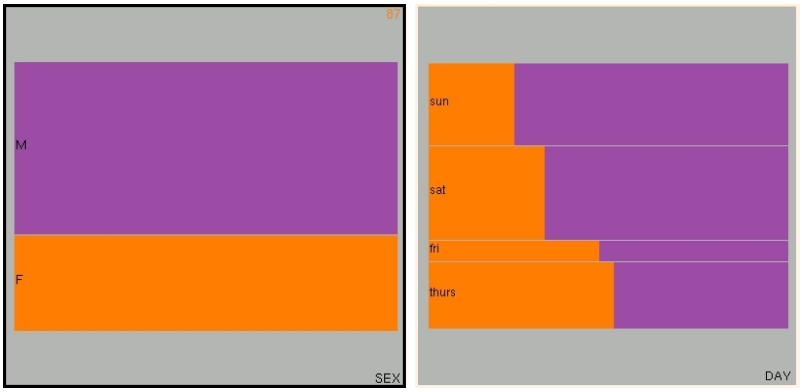

Figure 3.16. *(Left) A spine plot of gender of the bill payer, females are highlighted orange. More males pay the bill than females. (Right) Mosaic plot of day of the week conditional on gender. The ratio of females to males is roughly the same on Thursday but decreases through Sunday.*

this scatterplot matrix is the ASH for each variable. The correlation matrix for this subset of the data is

	FL	RW	CL	CW	BD
FL	1.00	0.90	1.00	1.00	0.99
RW	0.90	1.00	0.90	0.90	0.90
CL	1.00	0.90	1.00	1.00	0.99
CW	1.00	0.90	1.00	1.00	0.99
BD	0.99	0.90	1.00	1.00	1.00

All five variables are strongly linearly related. The two sexes differ in their ratio of rear width to all other proportions! There is more difference for larger crabs.

3.2.4 Mixtures of Continuous and Categorical Variables

Categorical variables are commonly used to layout plots of continuous variables (Figure 1.2) to make a comparison of the joint distribution of the continuous variables conditionally on the categorical variables. Also its common to code categorical data as color or glyph/symbol in plots of continuous variables as in Figures 3.8 and 3.17.

3.3 Direct Manipulation on Plots

3.3.1 Brushing

Brushing means to directly change the glyph and color of plot elements. The brush for painting points is a rectangular-shaped box that is moved over points on the screen to change the glyph. In persistent painting mode the glyphs stay changed after the brush has moved off the points. In transient brushing mode the glyphs

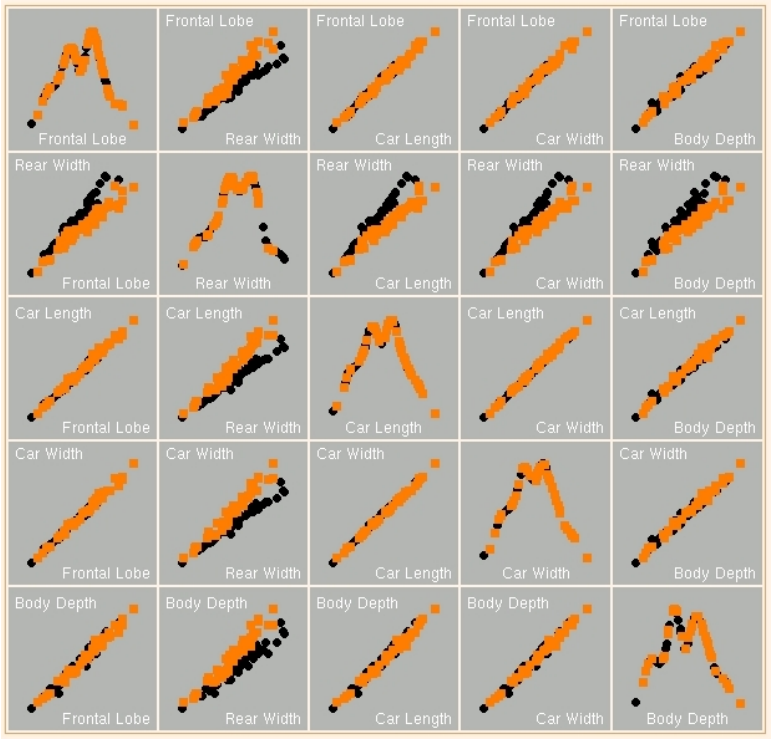

Figure 3.17. *The scatterplot matrix is one of the common multilayout plots. All pairs of variables are laid out in a matrix format that matches the correlation or covariance matrix of the variables. Here is a scatterplot matrix of the five physical measurement variables of the Australian crabs data. All five variables are strongly linearly related.*

revert the the previous glyph when the brush moves on. Figure 3.18 illustrates this.

The brush for painting lines is a cross-hair. Lines that intersect with the cross-hair are considered under the brush. Figure 3.19 illustrates this.

3.3.2 Identification

Attributes of a point in a plot can be ideintified by mousing over the point. Labels can be made sticky by clicking. Figure 3.20 illustrates different attributes shown using identify: row label, variable value and record id. The point highlighted is an OrangeFemale crab, that has a value of 23.1 for Frontal Lobe, and it is the 200 row in the data matrix.

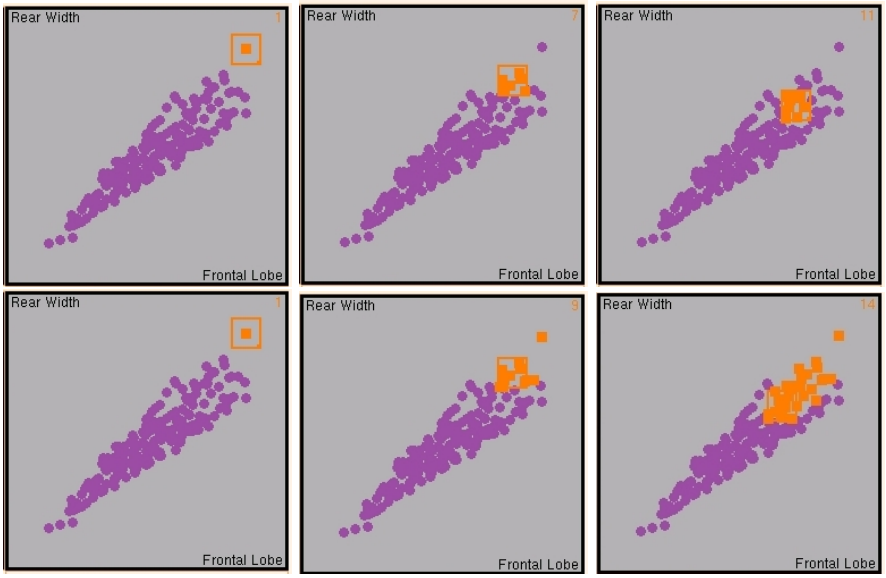

Figure 3.18. *Brushing points in a plot: (Top row) Transient brushing, (Bottom row) Persistent painting.*

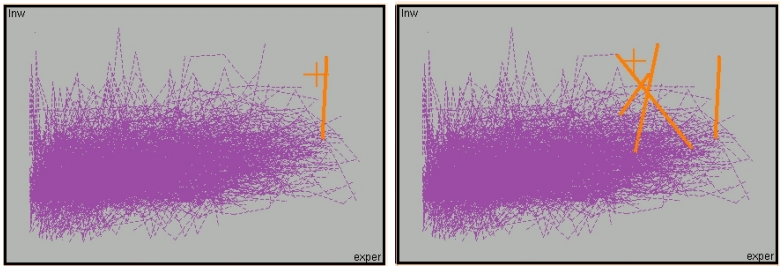

Figure 3.19. *Brushing lines in a plot.*

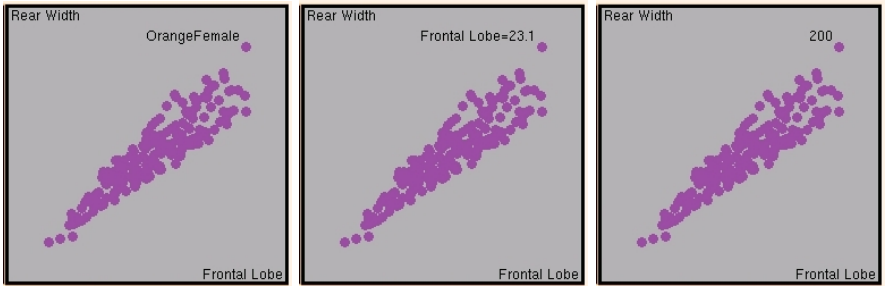

Figure 3.20. *Identifying points in a plot: (Left) Row label, (Middle) Variable value, (Right) Record id.*

3.3.3 Linking

Figure 3.16 shows an example of linked brushing between plots. The females category is highlighted orange and this carries through the plot of dining day, so we can see the proportion of bill payers who were female. In this data the proportion of females paying the bill drops from Thursday through Sunday!

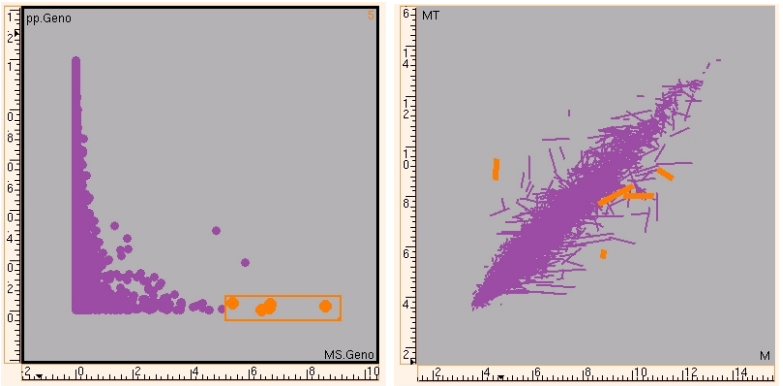

Figure 3.21. *Linking between a point in one plot and a line in another. The left plot contains 8297 points, the p-values and mean square values from factor 1 in an ANOVA model for each gene. The highlighted points are genes that have small p-values but large mean square values, that is, there is a lot of variation in this gene but most of it is due to the treatment. The right plot contains 16594 points, that are paired, and connected by 8297 line segments. One line segment in this plot corresponds to a point in the other plot.*

Linking between plots can be more sophisticated. Lets take a look at the microarray data. Here we have two replicates measured on two factors. We want to examine the variation in the treatments relative to the variation in the replications for each gene. To do this we set up two data sets, one containing both replicates as separate rows in the data matrix, and the other that contains diagnostics from fitting a model for each gene. Figure 3.21 shows the two plots, where a point in one plot corresponding to a diagnostic is linked to a line in the other plot connecting two replicates. The interesting genes are the ones that are most different in treatment (far from the $x = y$ line) but with similar replicate values. The left plot contains 8297 points, the p-values and mean square values from factor 1 in an ANOVA model for each gene. The highlighted points are genes that have small p-values but large mean square values, that is, there is a lot of variation in this gene but most of it is due to the treatment. The right plot contains 16594 points, that are paired, and connected by 8297 line segments. One line segment in this plot corresponds to a point in the other plot.

In general linking between plots can be conducted using any of the categorical variables in the data as the index between graphical elements, or using the ids for each record in the data set.

3.3.4 Scaling

Scaling the aspect ratio of the plot aspects is a good way to explore for different structure in data. Figure 3.22 illustrates this, for a time series. At an equal ratio horizontal to vertical ratio there is a weak up then down trend. When the horizontal axis is wide relative to a short vertical axis, periodicities in the data can be seen.

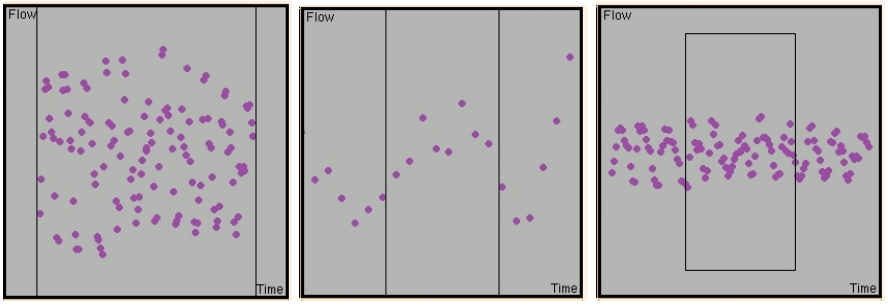

Figure 3.22. *Scaling a plot reveals different aspects: (Left) Original scale, shows a weak global trend up then down, (Middle) horizontal axis stretched, vertical axis shrunk, (Right) both reduced, reveals periodicities.*

3.3.5 Moving points

Chapter 4

Missing Values

4.1 Background

Missing data is a common problem in data analysis. It arises for many reasons: measuring instruments fail, samples get lost or corrupted, patients don't show up to scheduled appointments. Sometimes the missing values occur in association with other factors: a measuring instrument might fail more frequently if the air temperature is high. In some rare circumstances we can simply remove the incomplete cases or incomplete variables and proceed with the analysis, but usually this is not an option, for many reasons. Too many values may be missing, so that almost no case is complete; perhaps only a few values are missing, but distribution of the gaps is correlated with the variables of interest. Sometimes the "missings" are concentrated in some critical subset of the data, and sometimes the data is multivariate and the missings are well-spread amongst variables and cases. Consider the following, constructed data:

Case	X_1	X_2	X_3	X_4	X_5
1	NA	20	1.8	6.4	-0.8
2	0.3	NA	1.6	5.3	-0.5
3	0.2	23	1.4	6.0	NA
4	0.5	21	1.5	NA	-0.3
5	0.1	21	NA	6.4	-0.5
6	0.4	22	1.6	5.6	-0.8
7	0.3	19	1.3	5.9	-0.4
8	0.5	20	1.5	6.1	-0.3
9	0.3	22	1.6	6.3	-0.5
10	0.4	21	1.4	5.9	-0.2

There are only 5 missing values out of the 50 numbers in the data. That is, 10% of the numbers are missing, but 50% of the cases have missing values, and 100% of the variables have missing values.

One our first tasks is to explore the distribution of the missing values, and learn about the nature of "missingness" in the data. Do the missing values appear to occur randomly or do we detect a relationship between the missing values on one variable and the recorded values for some other variables in the data? If the distribution of missings is not random, this will weaken our ability to infer structure among the variables of interest. It will be shown later in the chapter that visualization is helpful in searching for the answers to this question.

In order to explore the distribution of the missing values, it's necessary to keep track of them. (As you will see, we may start filling in some of the gaps in the data, and we need to remember where they were.) One way to do that is to "shadow" the data matrix, with a missing values indicator matrix. Here is the shadow matrix for the constructed data, with a value of 1 indicating that this element of the data matrix is missing.

Case	X_1	X_2	X_3	X_4	X_5
1	1	0	0	0	0
2	0	1	0	0	0
3	0	0	0	0	1
4	0	0	0	1	0
5	0	0	1	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0
10	0	0	0	0	0

In order to model the data, and sometimes even to draw it, it is common to impute new values; i.e. to fill in the missing values in the data with suitable replacements. There are numerous methods for imputing missing values. Simple schemes include assigning a fixed value such as the variable mean or median, selecting an existing value at random, or averaging neighboring values. More complex distributional approaches to missing values assume that the data arises from a standard distribution such as a multivariate normal and samples this distribution for missing values. See Shafer()*** for a description of multiple imputation, and Little and Rubin()*** for a description of imputing using multivariate distributions. Unfortunately, these references spend little time discussing visual methods to assess the results of imputation.

Perhaps this oversight is partly due to the fact that traditional graphical methods, either static or interactive, have not offered many extensions designed for working with missing values. It is common to exclude missings from plots altogether; this is unsatisfactory, as as we can guess from the example above. The user usually needs to assign values for missings to have them incorporated in plots. Once some value has been assigned, the user may lose track of where the missings once were. More recent software offers greater support for the analysis of missings, both for exploring their distribution and for experimenting with and assessing the results of imputation methods (see **?**)).

As an example for working with missing values, we use a small subset of the TAO data: all cases recorded for five locations (latitude 0^o with longitude 110^oW/95^oW, 2^oS with 110^oW/95^oW, and 5^oS with 95^oW) and two time periods (November to January 1997, an El Niño event, and for comparison, the period from November to January 1993, when conditions were considered normal). There are 736 data points, and we find missing values on three of the five variables, as shown in the Table 4.1 below. Table 4.2 shows the distribution of missings on cases: Most cases (77%) have no missing values, and just two cases have missing values on three of the five variables.

Variable	Number of missing values
Sea Surface Temperature	3
Air Temperature	81
Humidity	93
UWind	0
VWind	0

Table 4.1. *Missing values on each variable.*

Number of missing values on a case	Count	%
3	2	0.3
2	2	0.3
1	167	22.7
0	565	76.7

Table 4.2. *Distribution of the number of missing values on each case.*

4.2 Exploring missingness

4.2.1 Getting started: plots with missings in the "margins"

The simplest approach to drawing scatterplots of variables with missing values is to assign to the missings some fixed value outside the range of the data, and then to draw them as ordinary data points at this unusual location. It is a bit like drawing them in the margins, an approach favored in other visualization software. In Figure 4.1, the three variables with missing values are shown. The missings have been replaced with a value 10% lower than the minimum data value for each variable. In each plot, missing values in the horizontal or vertical variable are represented as points lying along a vertical or horizontal line, respectively. A point that is missing on both variables appears as a point at the origin; if multiple points are missing on both, this point is simply overplotted.

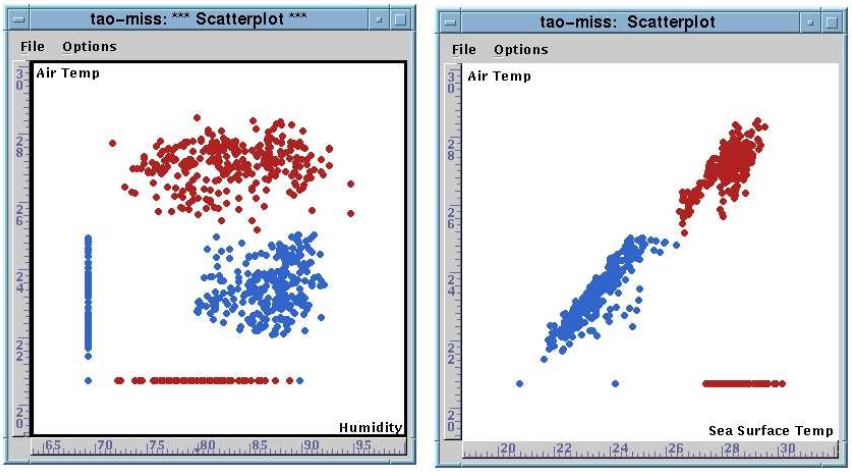

Figure 4.1. *In this pair of scatterplots, we have assigned to each missing value a fixed value 10% below the each variable's minimum data value, so the "missings" fall along vertical and horizontal lines to the left and below the point scatter. The points showing data recorded in 1993 are drawn in blue; points showing 1997 data are in red.*

What can be seen? First consider the righthand plot, Air Temp vs Sea Surface Temp. All the missings in that plot fall along a horizontal line, telling us that more cases are missing for Air Temp than for Sea Surface Temp. Some cases are missing for both, and those lie on the point at the origin. (The live plot can be queried to find out how many points are overplotted there.) We also know that there are no cases missing for Sea Surface Temp but recorded for Air Temp – if that were not true, we would see some points plotted along a vertical line. The lefthand plot, Air Temp vs Humidity, is different: there are many cases missing on each variable but not missing on the other.

Both pairwise plots contain two clusters of data: the blue cluster corresponds to recordings made in 1993, and the red cluster to 1997, an El Niño year. There is a relationship between the variables and the distribution of the missing values, as we can tell both by the color and by the position of the missings. For example, all the cases for which Humidity was missing were recorded in 1993: they're all blue and they all lie close to the range of the blue cluster. This is an important insight, because we now know that if we simply exclude these cases in our analysis, the results will be distorted: we will exclude 93 out of 368 measurements for 1993, but none for 1997, and the distribution of Humidity is quite different in those two years.

4.2.2 A limitation

Populating missing values with constants is a useful way to begin, as we've just shown. We can explore the data that's present and begin our exploration of the

missing values as well, because these simple plots allow us to continue using the entire suite of interactive techniques. Higher-dimensional projections, though, are not amenable to this method, because using fixed values causes the missing data to be mapped onto artificial 2-planes in 3-space, which obscure each other and the main point cloud.

Figure 4.2 shows a tour view of sea surface temperature, air temperature and humidity, with missings set to 10% below minimum. The missing values appear as clusters in the data space, which might be explained as forming parts of three walls of a room with the complete data as a scattercloud within the room.

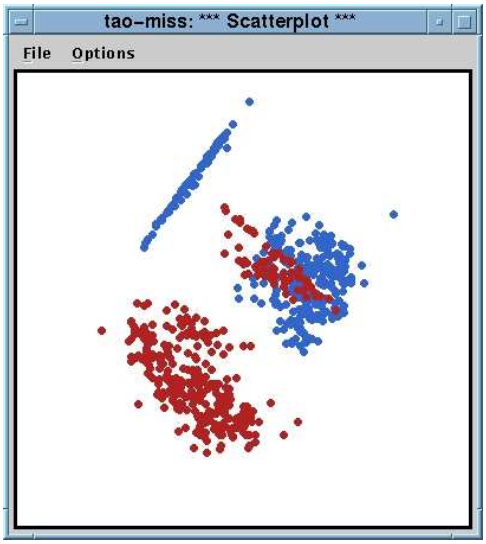

Figure 4.2. *Tour view of sea surface temperature, air temperature and humidity with missings set to 10% below minimum. There appear to be four clusters, but two of them are simply the cases that have missings on at least one of the three variables.*

Figure 4.3 shows the parallel coordinates of sea surface temperature, air temperature, humidity and winds, with missings set to 10% below minimum. The profiles look to split into two groups, for 1993 (blue) on humidity, and for 1997 (red) on air temperature. This is due solely to the manner of plotting the missing values.

4.3 Imputation

4.3.1 Shadow matrix: The missing values data set

While we are not finished with our exploratory analysis of this subset of the TAO data, we have already learned that we need to investigate imputation methods. We know that we won't be satisfied with "complete case analysis": that is, we

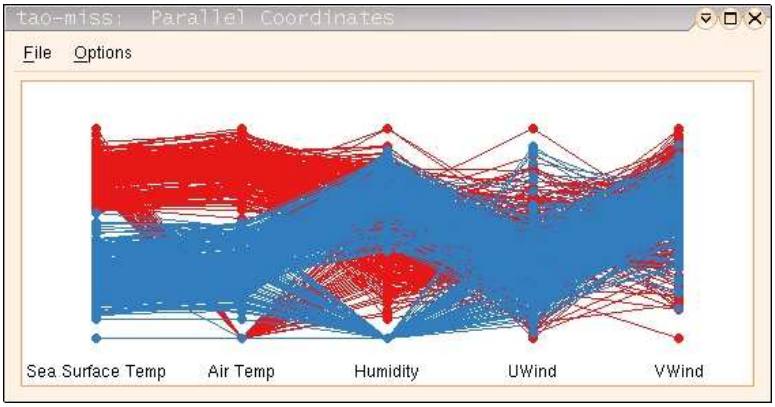

Figure 4.3. *Parallel coordinates of the five variables sea surface temperature, air temperature, humidity and winds with missings set to 10% below minimum. The two groups visible in the 1993 year (blue) on humidity is due to the large number of missing values plotted below the data minimum, and similarly for the 1997 year (red) on air temperature.*

can't safely throw out all cases with a missing value because the distribution of the missing values on at least one variable (Humidity) is strongly correlated with at least one other data variable (Year).

This tells us that we need to investigate imputation methods. As we replace the missings with imputed values, though, we don't want to lose track of their locations. We want to use visualization to help us assess imputation methods as we try them: we want to know that the imputed values have nearly the same distribution as the rest of the data.

In order to keep track of the locations of the missings, we construct a missing values dataset. It has the same dimensionality as the main dataset, but it is essentially a logical matrix representing the presence or absence of a recorded value in each cell. This is equivalent to a binary data matrix of 0s and 1s, with 1 indicating a missing value. In order to explore the data and their missing values together, we will display projections of each matrix in a separate window. In the main window, we show the data with missing values replaced by imputed values; in the missing values window, we show the binary indicators of missingness.

Although it may seem unnatural, we often like to display binary data in scatterplots because scatterplots preserve case identity for pointing operations; by contrast, histograms and other aggregating presentations visualize groups rather than individual cases. When using scatterplots to present binary data, it is natural to spread the points so as to avoid multiple overplotting. This can be done by adding small random numbers to (jittering) the zeros and ones. The result is a view such as the lefthand plot in Figure 4.4. The data fall into four squarish clusters, indicating presence and "missingness" of values for the two selected variables. For instance, the top right cluster consists of the cases for which both variables have missing

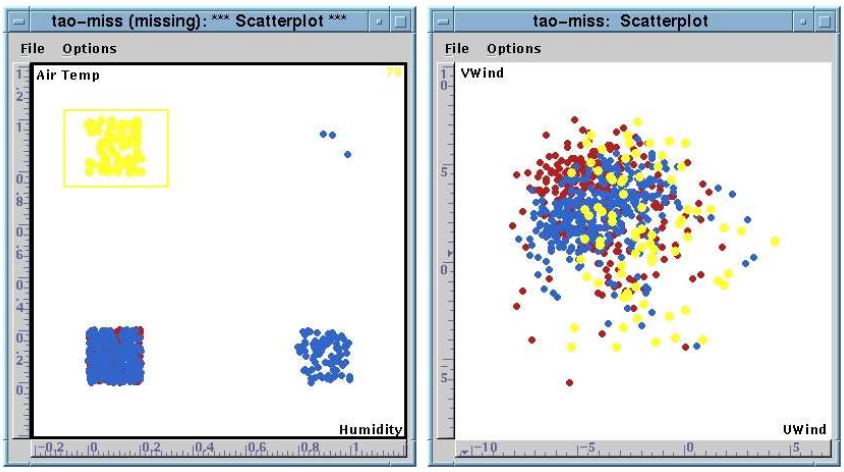

Figure 4.4. *Exploring the data using the missing values dataset. The lefthand plot is the "missings" plot for Air Temp vs Humidity: a jittered scatterplot of 0s and 1s where 1 indicates a missing value. The points that are missing only on Air Temp have been brushed in yellow. The righthand plot is a scatterplot of VWind vs UWind, and those same missings are highlighted. It appears that Air Temp is never missing for those cases with the largest negative values of UWind.*

values, and the lower right cluster shows the cases for which the horizontal variable value is missing but the vertical variable value is present.

Figure 4.4 illustrates the use of the missing values data set to explore the distribution of missing values for one variable with respect to other variables in the data. We have brushed in yellow only the cases in the top left cluster, where Air Temp is missing but Humidity is present. We see in the righthand plot that none of these missings occur for the lowest values of UWind, so we have discovered another correlation between the distribution of missingness on one variable and the distribution of another variable.

We didn't really need the missings plot to arrive at this observation; we could have found it just as well using by continuing to assign constants to the missing values. In the next section, we'll continue to use the missings plot as we begin using imputation.

4.3.2 Examining Imputation

4.3.3 Random values

The most rudimentary imputation method is to fill in the missing values with some value selected randomly from among the recorded values for that variable. In the middle plot of Figure 4.5, we have used that method for the missing values on Humidity. The result is not very good, and we shouldn't be surprised. We already

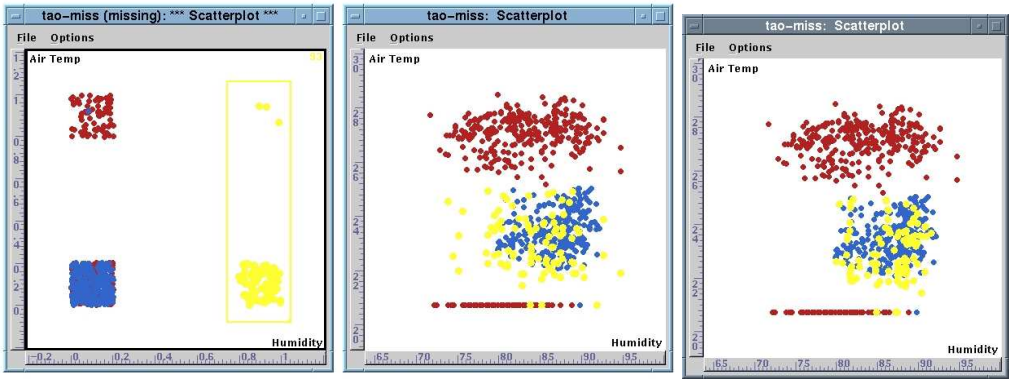

Figure 4.5. *(Middle) Missing values on Humidity were filled in by randomly selecting from the recorded values. The imputed values, in yellow, aren't a good match for the recorded values for 1993, in blue. (Right) Missing values on Humidity have been filled in by randomly selecting from the recorded values, conditional on drawing symbol.*

noted that the missing values on Humidity are all part of the 1993 data, colored in blue, and we can see that Humidity was higher in 1993 than it was in 1997. Simple random imputation ignores that fact, so the missings on Humidity, brushed in yellow, are distributed across the entire range for both years.

A slightly more sophisticated random imputation method conditions on symbol. In this method, since all the Humidity missings are drawn in blue, they are filled in only with values which are also drawn in blue. Thus measurements from 1993 are used to fill in the missing values for 1993, and the results are much better, as we see in the plot at right of Figure 4.5.

Still, the results are far from perfect. In Figure 4.6, missings on all values have been filled in using conditional random imputation, and we're investigating the imputation of Air Temp. Sea Surface Temp and Air Temp are highly correlated but this correlation is much weaker among the imputed values.

4.3.4 Mean values

Using the variable mean to fill in the missing values is also very common and simple. In Figure 4.7, we have substituted the mean values for missing values on sea surface temperature, air temperature and humidity. We shouldn't be surprised to see the cross structure in the scatterplot. These are the the imputed values.

4.3.5 From external sources

In the TAO data we have included a data matrix where the missings are imputed used nearest neighbors. The ten closest points in the 5D data space (sea surface temperature, air temperature, humidity, winds) are averaged to give a value to the

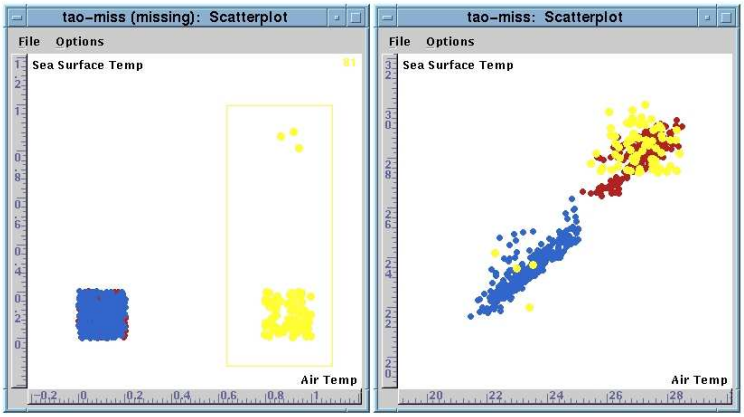

Figure 4.6. *Missing values on all variables have been filled in using random imputation, conditioning on drawing symbol. The imputed values for Air Temp show less correlation with Sea Surface Temp than the recorded values do.*

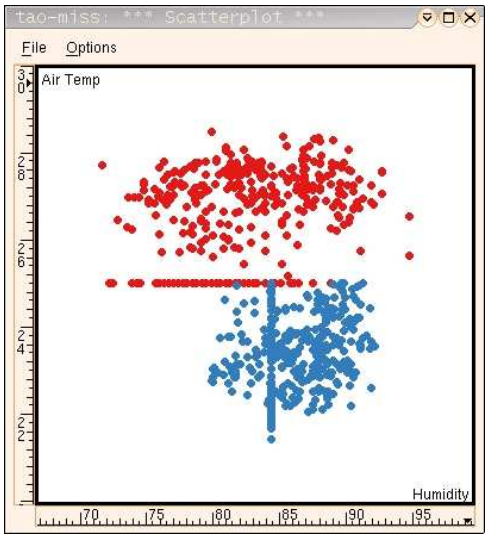

Figure 4.7. *Missing values on all variables have been filled in using variable means. This produces the cross structure in the center of the scatterplot.*

missings. The cases that had missing values on air temperature are highlighted (yellow) in Figure 4.8. The imputed values don't correspond well with the sea surface temperature values. The missings on air temperature mostly occurred at high sea surface temperatures. The imputed air temperature values appear to be too low relative the sea surface temperature.

Imputation can be done from within R and the values dynamically loaded into

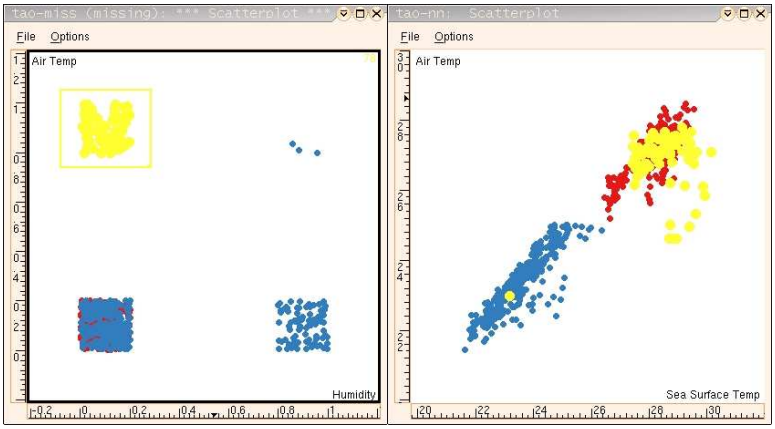

Figure 4.8. *Missing values on the five variables are replaced by a nearest neighbor average. (Left) The cases corresponding to missing on air temperature, but not humidity are highlighted (yellow). (Right) A scatterplot of air temperature vs sea surface temperature. The imputed values are some strange: many are estimated to have much lower sea surface temperature than we'd expect given the air temperature values.*

ggobi. This next example demonstrates this. Here is the code in R:

```
# Load the libraries
library(Rggobi)
library(norm)
# Read in data if its not already in R
d.elnino<-read.table("tao.asc",header=T)
# Calculate preliminary statistics for imputation
d.elnino.nm.97<-prelim.norm(as.matrix(d.elnino[1:368,4:6]))
d.elnino.nm.97$nmis
d.elnino.nm.97$ro # is the row order of the reorganized data.
d.elnino.nm.93<-prelim.norm(as.matrix(d.elnino[369:736,4:6]))
d.elnino.nm.93$nmis
d.elnino.nm.93$ro # is the row order of the reorganized data.
d.elnino.ro<-d.elnino[c(d.elnino.nm.97$ro,368+d.elnino.nm.93$ro),]
# Start ggobi on the row re-ordered data, and set default colors
# and glyphs and different colors for the missing values
ggobi(d.elnino.ro)
setColors.ggobi(rep(2,368),c(1:368))
setColors.ggobi(rep(3,368),c(369:736))
setGlyphs.ggobi(types=rep(5,736),sizes=rep(2,736),which=c(1:736))
indx<-c(1:768)[is.na(d.elnino.ro[,5])]
setColors.ggobi(rep(1,length(indx)),indx) # set missings in air temp
                                          # to be highlighted
```

```
setGlyphs.ggobi(rep(5,length(indx)),rep(2,length(indx)),indx)
# Make a scatterplot of sea surface temperature
# and air temperature, using the ggobi menus.

# multiple imputation
rngseed(1234567) # set the seed
thetahat.97<-em.norm(d.elnino.nm.97,showits=TRUE)
theta.97<-da.norm(d.elnino.nm.97,thetahat.97,steps=100,showits=TRUE)
getparam.norm(d.elnino.nm.97,theta.97,corr=TRUE)
thetahat.93<-em.norm(d.elnino.nm.93,showits=TRUE)
theta.93<-da.norm(d.elnino.nm.93,thetahat.93,steps=100,showits=TRUE)
getparam.norm(d.elnino.nm.93,theta.93,corr=TRUE)
d.elnino.impute.97<-imp.norm(d.elnino.nm.97,theta.97,
  as.matrix(d.elnino.ro[1:368,4:6]))
d.elnino.impute.93<-imp.norm(d.elnino.nm.93,theta.93,
  as.matrix(d.elnino.ro[369:736,4:6]))

# Set the values of missings to be the imputed values.
setVariableValues.ggobi(c(d.elnino.impute.97[,1],d.elnino.impute.93[,1])
  4,1:736)
setVariableValues.ggobi(c(d.elnino.impute.97[,2],d.elnino.impute.93[,2])
  5,1:736)
setVariableValues.ggobi(c(d.elnino.impute.97[,3],d.elnino.impute.93[,3])
# You may need to use the missing values panel to re-scale the plot
# now that missings are no longer low values.
  6,1:736)
```

Figure 4.9 shows plots of the data containing imputed values resulting from multiple imputation. The imputed values for missings on air temperature are highlighted (yellow). There appears to be a mismatch with the complete data: the imputed values have air temperatures too low for the corresponding sea surface temperatures. This can be seen in the scatterplot of air temperature vs sea surface temperature. We use the tour to compare the imputed values with the complete cases in the three variables, sea surface temperature, air temperature and humidity. The imputed and complete on humidity values are quite similar.

4.4 Exercises

1. Describe the distribution of the two wind variables and the two temperature variables conditional on missing in humidity.

2. Describe the distribution of the four variables, winds, temperatures, conditional on missing in humidity, using brushing and the tour.

3. This question uses the `support` data.

   (a) Examine the plot of albumin vs bilirubin with missing values plotted

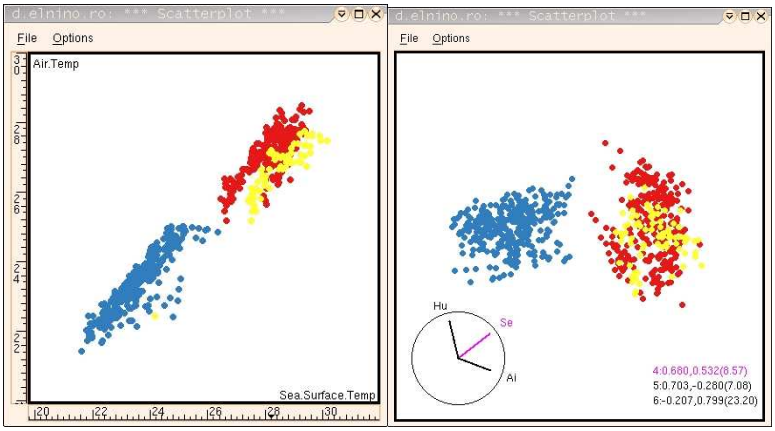

Figure 4.9. *Missing values on all variables have been filled in using multiple imputation. (Left) In the scatterplot of air temperature vs sea surface temperature the imputed values appear to have a different mean than the complete cases: higher sea surface temperature, but lower air temperature. (Right) A tour projection of three variables, sea surface temperature, air temperature and humidity where the imputed values match reasonably.*

in the margins. Describe the distribution between missings and non missings.

(b) Substitute the missing values with those suggested in the description of the data. Assess the result using a scatterplot.

(c) Transform the variables so that they more closely resemble normal distributions. Impute the missing values using multiple imputation, assuming the data arises from a bivariate normal distribution. Assess the results using a scatterplot.

4

Supervised Classification

When you browse your email, you can usually tell right away whether a message is spam or not. Still, you probably don't enjoy spending your time identifying spam, and have come to rely on a filter to do that task for you, either deleting the spam automatically, or filing it in a different mailbox. An email filter is based on a set of rules applied to each incoming message, tagging it as spam or "ham" (not spam). Such a filter is an example of a supervised classification algorithm. It is formulated by studying a training sample of email messages which have been manually classified as spam or ham. Information in the header and text of each message is converted into a set of numerical variables such as the size of the email, the domain of the sender, or the presence of the word "free". These variables are used to define rules which determine whether an incoming message is spam or ham.

An effective email filter must successfully identify most of the spam without losing legitimate email messages: that is, it needs to be an accurate classification algorithm. The filter must also be efficient so that it doesn't become a bottleneck in the delivery of mail. Knowing which variables in the training set are useful and using only these helps relieve the filter of superfluous computations.

Supervised classification forms the core of what we have recently come to call *data mining*. The methods originated in Statistics in the early nineteenth century, under the moniker *discriminant analysis*. An increase in the use of databases in the late twentieth century has inspired a need to extract knowledge from data, contributing to a recent burst of research on new methods, especially on algorithms.

There are now a multitude of ways to build classification rules, each with some common elements. A training sample contains data with known categorical response values for each recorded combination of explanatory variables. The training sample is used to build the rules to predict the response. Accuracy, or inversely error, of the classifier for future data is also estimated from the training sample. Accuracy is of primary importance, but there are many other interesting aspects of supervised classification applications beyond this:

- Are the classes well-separated in the data space, so that they correspond to distinct clusters? If so, what are the shapes of the clusters? Is each cluster sufficiently ellipsoidal so that we can assume that the data arises from a mixture of multivariate normal distributions? Do the clusters exhibit characteristics that suggest one algorithm in preference to others?
- Where does the boundary between classes fall? Are the classes linearly separable, or does the difference between classes suggest a non-linear boundary? How do changes in the input parameters affect these boundaries? How do the boundaries generated by different methods vary?
- What cases are misclassified, or have more uncertainty in the predictions? Are there regions in the data space where predictions are especially good, or indeed, bad?
- Is it possible to reduce the set of explanatory variables?

This chapter discusses the use of interactive and dynamic graphics to investigate these different aspects of classification problems. It is structured as follows: Section 4.1 gives a brief background of the major approaches, Section 4.2 describes graphics for viewing the classes, which is followed by graphics associated with different methods in Section 4.3. A good companion to this chapter is the material presented in Venables & Ripley (2002) which provides data and code for practical examples of supervised classification using R.

4.1 Background

Supervised classification arises when there is a categorical response variable (the output), $Y_{n\times 1}$, and multiple explanatory variables (the input) $\mathbf{X}_{n\times p}$, where n is the number of cases in the data, and p is the number of variables. Because Y is categorical, it may be represented by strings and must be recoded using integers; for example, binary variables may be converted to $\{0, 1\}$ or $\{-1, 1\}$, while multiple classes may be recoded using the values $\{1, \ldots, g\}$. Coding of the response really matters, and can alter the formulation or operation of a classifier.

Since supervised classification is used in several disciplines, the terminology used to describe the elements can vary widely. The explanatory variables may also be called independent variables or attributes. The instances may be called cases, rows or records.

4.1.1 Classical Multivariate Statistics

Discriminant analysis dates to the early 1900s. Fisher's linear discriminant (Fisher 1936) determines a linear combination of the variables which separates two classes by comparing the differences between class means with the

variance of values within each class. It makes no assumptions about the distribution of the data. Linear discriminant analysis (LDA) formalizes Fisher's approach, by imposing the assumption that the data values for each class arise from a p-dimensional multivariate normal with common variance-covariance matrix centered at different locations. Under this assumption, Fisher's linear discriminant gives the optimal separation between the two groups.

For two groups, where Y is coded as $\{0, 1\}$, the LDA rule is:

Allocate a new observation, $\mathbf{X}_0$ *to group 1 if*

$$(\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_2)'\mathbf{S}^{-1}_{pooled}\mathbf{X}_0 \geq \frac{1}{2}(\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_2)'\mathbf{S}^{-1}_{pooled}(\bar{\mathbf{X}}_1 + \bar{\mathbf{X}}_2)$$

else allocate to group 2,

where $\bar{\mathbf{X}}_k$ are the class mean vectors of an $n \times p$ data matrix, $\mathbf{X}_k \;\; (k = 1, 2)$,

$$\mathbf{S}_{pooled} = \frac{(n_1 - 1)\mathbf{S}_1}{(n_1 - 1) + (n_2 - 1)} + \frac{(n_2 - 1)\mathbf{S}_2}{(n_1 - 1) + (n_2 - 1)}.$$

is the pooled variance-covariance matrix, and

$$\mathbf{S}_k = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{X}_{ki} - \bar{\mathbf{X}}_k)(\mathbf{X}_{ki} - \bar{\mathbf{X}}_k)', \quad k = 1, 2$$

is the class variance-covariance matrix. The linear discriminant part of this rule is $(\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_2)'\mathbf{S}^{-1}_{pooled}$ which defines the linear combination of variables which best separates the two groups. Computing the value of the new observation, $\mathbf{X}_0$, on this line and comparing it with the value of the average of the two class means, $(\bar{\mathbf{X}}_1 + \bar{\mathbf{X}}_2)/2$, on this line, gives the classification rule.

For multiple (g) classes, the rule and the discriminant space are constructed using the between-group sum-of-squares matrix,

$$\mathbf{B} = \sum_{k=1}^{g} n_k(\bar{\mathbf{X}}_k - \bar{\mathbf{X}})(\bar{\mathbf{X}}_k - \bar{\mathbf{X}})'$$

which measures the differences between the class means, compared to the overall data mean, $\bar{\mathbf{X}}$, and the within-group sum-of-squares matrix,

$$\mathbf{W} = \sum_{k=1}^{g}\sum_{i=1}^{n_k}(\mathbf{X}_{ki} - \bar{\mathbf{X}}_k)(\mathbf{X}_{ki} - \bar{\mathbf{X}}_k)'$$

which measures the variation of values around each class mean. The linear discriminant space is generated by computing the eigenvectors (canonical coordinates) of $\mathbf{W}^{-1}\mathbf{B}$, and this is the space where the group means are most separated with respect to the pooled variance-covariance. The resulting classification rule is to allocate a new observation to the class with the highest value of

$$\bar{\mathbf{X}}_k' \mathbf{S}_{pooled}^{-1} \mathbf{X}_0 - \frac{1}{2} \bar{\mathbf{X}}_k' \mathbf{S}_{pooled}^{-1} \bar{\mathbf{X}}_k \quad k = 1, ..., g$$

which results in allocating the new observation into the class with the closest mean.

This LDA approach is generally applicable, but it is useful to check the underlying assumptions: (1) that the cluster structure corresponding to each class is elliptically-shaped, consistent with a sample from a multivariate normal distribution, and (2) that the variance of values around each mean is nearly the same. Figure 4.1 illustrates two data sets, one consistent with the assumptions, and the other not. Other parametric models, such as quadratic discriminant analysis or logistic regression, require checking assumptions like these too.

A good treatment of parametric methods for supervised classification can be found in Johnson & Wichern (2002) or similar multivariate analysis text. Missing from these treatments is a good explanation of how to use graphics to check the assumptions underlying the methods, and how to use graphics to explore the results. This chapter does so.

4.1.2 Data Mining

Algorithmic methods have overtaken parametric methods in the practice of supervised classification. A parametric method, such as LDA, yields a set of interpretable output parameters; it leaves a clear trail helping us to understand what was done to produce the results. An algorithmic method, on the other hand, is more or less a black box, with various input parameters that are adjusted to tune the algorithm. The algorithm's input and output parameters do not always correspond in any obvious way to the interpretation of the results. All the same, these methods can be very powerful and their use is not limited by requirements about variable distributions as is the case with parametric methods.

The tree algorithm (Breiman, Friedman, Olshen & Stone 1984) is an example of an algorithmic method. The tree algorithm generates a classification rule by sequentially subsetting or splitting the data into two buckets. Splits are made between sorted data values of individual variables, with the goal being to get pure classes on each side of the split. The inputs for a simple tree classifier commonly include (1) a choice of impurity measures, (2) a parameter that sets the minimum number of cases in a node, or the minimum number of observations in a terminal node of the tree, and (3) a complexity measure that controls the growth of a tree, balancing the use of a simple generalizable tree against an accurate data-tailored tree. When applying tree methods, it is useful to explore the effects of the input parameters on the tree; for example, it helps us to assess the stability of the tree model.

Knowing about the relationship between the cluster structure and the data classes can help to decide if a particular algorithmic method is appropriate for that data. For example, if the variables are not independent within each class

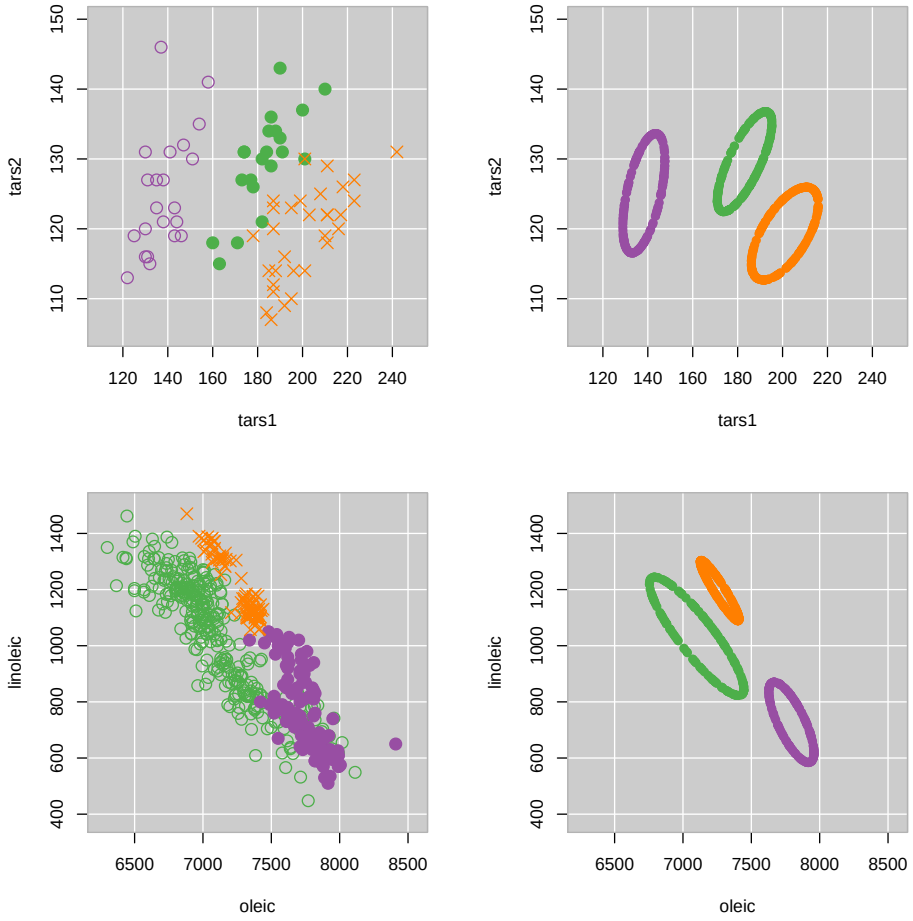

Fig. 4.1. (Top left) Flea beetle data that contains three classes, each one appears to be consistent with a sample from a bivariate normal distribution with equal variance-covariance, (top right) with the correponding estimated variance-coviance ellipses. (Bottom row) Olive oil data that contains three classes clearly inconsistent with LDA assumptions. The shape of the clusters is not elliptical, and the variation differs from cluster to cluster.

then the tree algorithm, which ignores this, might not give the best results. The flea beetle data shown in the top row of plots in Figure 4.1 has this type of structure. Each class corresponds to a cluster in the 2D space that has some positive linear association between the variables. The separations between the clusters are likely to be better if a linear combination of two variables is used rather than vertical or horizontal splits on a single variable. The plots in Figure 4.2 display the class predictions for LDA and a tree. The LDA boundaries which are formed from a linear combination of tars1 and tars2 probably make more practical sense than the straight-up or across boundaries of the tree classifier.

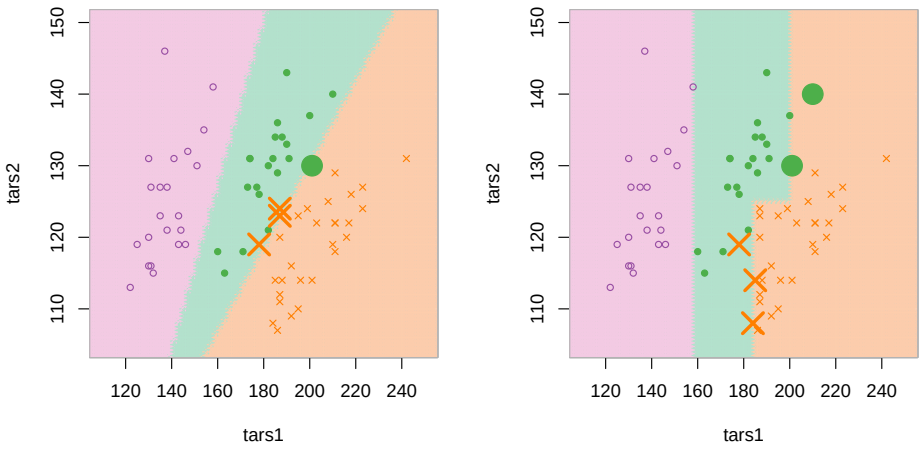

Fig. 4.2. Missclassifications highlighted on plots showing the boundaries between three classes (Left) LDA (Right) Tree.

Hastie, Tibshirani & Friedman (2001) has a thorough discussion of algorithms for supervised classification presented from a modeling perspective with a tendency to the theoretical. Ripley (1996) is an early volume describing and illustrating both classical statistical methods and algorithms for supervised classification. Both volumes contain some excellent examples of how graphics can be used to examine 2D boundaries generated by different classifiers. The discussions in these and other writings on data mining algorithms are missing the treatment of graphics for the high-dimensional spaces in which the classifiers operate, and the exploratory data analysis approach to supervised classification, which are described in this chapter.

4.1.3 Studying the Fit

A classifier's performance is usually assessed by its error, or its converse, the accuracy. The error is calculated by comparing the predicted class with the known true class, using a missclassification table. For example, below are the respective missclassification tables for the LDA and the tree classifier:

LDA

	True			
Predicted	1	2	3	
1	20	0	3	23
2	0	22	0	22
3	1	0	28	29
	21	22	31	74

Tree

	True			
Predicted	1	2	3	
1	19	0	3	22
2	0	22	0	22
3	2	0	28	30
	21	22	31	74

The total error is the number of misclassified samples divided by the total number of cases: $4/74 = 0.054$ for LDA and $5/74 = 0.068$ for the tree classifier. It is really interesting to study which cases are missclassified, or generally, which areas of the data space have more error. The missclassified cases for LDA and tree classifiers are highlighted in Figure 4.2. In the tree classifier (right) some cases that are obviously members of their class, at the top of the green group and bottom of the orange are missclassified. These errors have occurred because of the limitations of the tree algorithm.

Ideally the error estimate reflects the future performance of the classifier on new samples. Error calculated on the same data as the classifier will tend to be too low. There are many approaches to compensating for the double-dipping in the data. A simple approach is to split the data into training sample and test sample. The classifier is built using the training sample and the error is calculated using the test sample. Other cross-validation methods are commonly used.

Ensemble methods build cross-validation into the error calculations. Ensembles are constructed using multiple classifiers and pool the predictions with a voting scheme. A good example of an ensemble is a random forest (Breiman 2001, Breiman & Cutler 2004). A random forest pools the predictions of multiple trees. Different trees are generated by randomly sampling the input variables and sampling the cases. It's the sampling of cases (bagging) that provides the in-built cross-validation because the error can be estimated for each tree by predicting the classes of the casses left out of the fit. Forests provide numerous diagnostics that enable us to inspect the fit very closely.

4.2 Purely Graphics: Getting a Picture of the Class Structure

The approach is simple. Code the response variable, Y, using color and symbol in plots of the explanatory variables, $\mathbf{X}$. There are a couple of cautions about (1) number of colors and glyphs, and (2) number of dimensions. The number of colors and glyphs used needs to be small – it is difficult to digest information from plots having more than 3 or 4 colors. So if there are a lot of classes make it simpler, try to break them into a smaller set of super-classes, or sequentially examine one class against the rest. On the second problem, when there are many variables, thus many dimensions of the data, start simply with plots of individual variables and build up to multivariate plots.

If the number of classes is large, keep in mind that it is difficult to digest information from plots having more than 3 or 4 colors. You may be able to simplify the displays by grouping classes into a smaller set of super-classes. Alternatively, you can examine a small number of classes at a time.

If the number of dimensions is large, it takes much longer to get a sense of the data, and it's easy to get lost in high-dimensional plots. There are many possible low-dimensional plots to examine, and that's the place to start. Explore plots one or two variables at a time before building up to multivariate plots.

When exploring these plots, we are trying to understand how the distinctions between classes arise, and we are hoping to see gaps between clusters of points. A gap indicates a well-defined distinction between groups, and suggests that there will be less error in predicting future samples. We will also study the shape of the clusters.

4.2.1 Overview of Olive Oils Data

The olive oil data has eight explanatory variables (levels of fatty acids in the oils) and nine classes (regions of Italy). The goal of the analysis is to develop rules which reliably distinguish oils from the nine different areas. It is a problem of practical interest, because oil from some regions is more highly valued and unscrupulous suppliers sometimes make false claims about the origin of their oil.

There are many fascinating data sets collected to solve contemporary problems with supervised classification problems, like the spam data. Unfortunately, the olive oils is very old data, but in its defense, it is really is very interesting data. It has an ideal mix of straight forward separations, and difficult separations, and unexpected finds. Olive oil is considered one of the more healthful oils, and some of its constituent fatty acids are considered to be more healthful than others.

We break the classification job into a two stage process, starting by grouping the the nine regions into three super-classes, corresponding to three large

areas of Italy: South, North and Sardinia. We first build a classifier for the three large regions, and then classifiers for the areas within each region.

4.2.2 Classifying Three Regions

Univariate Plots: We first paint the points according to the three large regions. Next, using univariate plots, we look at each explanatory variable in turn, either by manually selecting variables or cycling through the variables automatically, looking for separations between regions. We find that it's possible to cleanly separate the oils of the South (red) from the other two regions using just one variable, eicosenoic acid (Figure 4.3). We learn that the oils from other regions (green, purple) contain no eicosenoic acid.

Next we focus on separating the oils from the North (purple) and Sardinia (green), removing the Southern oils from view. Several variables reveal differences between the regions: for example, oleic and linoleic acid (Figure 4.3). The two regions are perfectly separated by linoleic acid, but there is no gap between the two groups of points. We learn that oils from Sardinia contain lower amounts of oleic acid and higher amounts of linoleic acid than oils from the north.

Bivariate Plots: If one variable is not enough to distinguish northern oils (purple) from Sardinian oils (green), perhaps we can find a pair of variables that will do the job.

Starting with oleic and linoleic acids, which were so promising when taken singly, we look at pairwise scatterplots (Figure 4.4). Unfortunately, the combination of oleic acid and linoleic acid is no more powerful than each one was alone. They are strongly negatively associated, and there is still no gap between the two groups. We explore other pairs of variables.

Something interesting emerges from a plot of arachidic acid and linoleic acid: There is big gap between the points of the two regions! Arachidic acid alone seems to have no power to separate, but it improves the power of linoleic acid. However, the gap between the two groups follows a non-linear, almost quadratic path, so we must do a bit more work to define a functional boundary.

Multivariate Plots: Using a 1D tour on these two variables, we're now looking for a gap using a linear combination of linoleic and arachidic acid. The right plot in Figure 4.4 shows the results. The two regions are separated by a gap using a linear combination of linoleic and arachidic acid (The linear combination as returned by the 1D tour is $\frac{0.985}{1022} \times \text{linoleic} + \frac{0.173}{105} \times \text{arachidic}$).

A parallel coordinate plot can also be used to select important variables for classification. Figure 4.5 shows a parallel coordinate plot for the olive oils data, where the three colors represent the three regions. Here we can see some information about the important variables for separating the regions. Eicosenoic acid is useful for separating southern oils (red) from the others,

Fig. 4.3. Looking for separation between the 3 regions of Italian olive oil data in univariate plots. Eicosenoic acid separates Southern oils from the others. North and Sardinia oils are separated by linoleic acid, although there is no big gap between the two clusters.

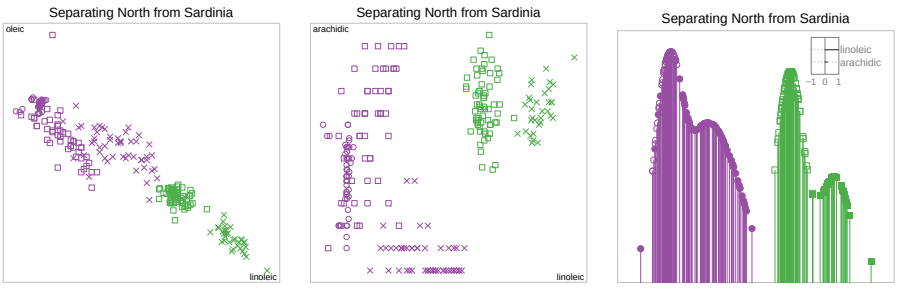

Fig. 4.4. Separation between the Northern (purple) and Sardinian (green) oils in bivariate scatterplots (left, middle) and a linear combination of linoleic and arachidic acids viewed in a 1D tour (right).

because there is a separation of these groups on the last axis corresponding to eicosenoic acid. To some extent palmitic, palmitoleic and oleic acids also distinguish the Southern oils. Southern oils have high values on palmitic, palmitoleic and eicosenoic acids, and low values on oleic acid, relative to the oils of the other regions. Linoleic and oleic acid are important for separating northern oils (purple) from Sardinian oils (green) because we can see separations of these two groups on the two respective axes. Northern oils have high values of oleic and low values of linoleic acid, relative to Sardinian oils. Parallel coordinate plots are not as good as tours for visualizing the shape of the clusters corresponding to classes and the shape of the boundaries between them.

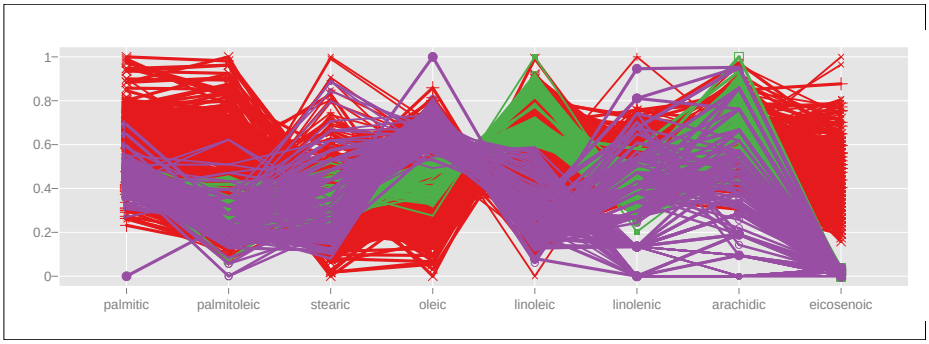

Fig. 4.5. Parallel coordinate plot of the 8 variables of the olive oils data. Color represents the three regions: South (red), North (purple), Sardinia (green). Eicosenoic acid, and to some extent palmitic, palmitoleic and oleic acids distinguish Southern oils from the others. Oleic and linoleic acids distinguish Northern from Sardinian oils.

4.2.3 Separating Nine Areas

It's clear now that the oils from the three larger regions, North, South and Sardinia can be recognized by their fatty acid composition. Within each of these regions we will explore for separations between oils among the areas.

Northern Italy: In this data the North region is composed of three areas in the region, Umbria, East Liguria and West Liguria. We use the same approach for stepping up through the dimensions, univariate, bivariate, to multivariate plots, looking for differences between the oils of the three areas.

In the univariate plots there are no clear separations between areas, although several variables are correlated with area. For example, oil from West Liguria (blue) has higher linoleic acid content than the other two areas (Figure 4.6 top left). In the bivariate plots there are also no clear separations between

areas, but two variables, stearic and linoleic show some differences (Figure 4.6 top right). Oils from West Liguria have the highest linoleic and stearic acid content, and oils from Umbria (pink) have the lowest linoleic and stearic acid content.

Fig. 4.6. Separation in the oils from areas of northern Italy: (top left) West Liguria oils (blue) have a higher percentage of linoleic acid, (top right) stearic acid and linoleic acid almost separate the three areas, (bottom) 1D and 2D linear combinations of palmitoleic, stearic, linoleic and arachidic acids reveals difference the areas.

Starting from these two variables we explore linear combinations using a 1D tour. Projection pursuit guidance using the LDA index was used to find the linear combination shown in Figure 4.6 (bottom left). West Liguria (blue) is almost separable from the other two areas using a combination of palmitoleic, stearic, linoleic and arachidic acids. At this stage we could move the analysis in two different ways. The first step would be to remove the points corresponding

to West Ligurian oils, and look for differences between the other two areas, and the second step would be to look for differences using 2D projections. The bottom right plot in Figure 4.6 shows a 2D linear combination of the same four variables (palmitoleic, stearic, linoleic and arachidic acids) where the oils from the three areas are almost separated. Projection pursuit guidance using the LDA index and manual controls were used to find this view.

What we learn from these plots is that there are clusters corresponding to the three areas but no gaps between the clusters. It may not be possible to build a classifier that perfectly predicts the areas of the north, but the error should be very small.

Sardinia: This is easy! Look at a scatterplot of oleic acid and linoleic acid. There is a big gap between two clusters corresponding to the oils of the two areas in the Sardinia super-class: the coastal and inland areas of Sardinia.

Southern Italy: In this data there are four areas grouped into in the South region. These are North Apulia, South Apulia, Calabria, and Sicily.

Working through the univariate, bivariate and multivariate plots the prospects of finding separations between these four areas looks dismal. In a scatterplot of palmitoleic and palmitic there is a big gap between North (orange) and South (pink) Liguria, with Calabria (red) in the middle, but the oils from Sicily (yellow) overlap all of the three areas (Figure 4.7 top row). Oils from North Liguria have low percentages of palmitic and palmitoleic acids and those from South Liguria have a higher content of both fatty acids.

The pattern is similar when more variables are used. We examine these two variables in combination with other variables in a tour, using projection pursuit with LDA and manual controls, and find that we can find a lot of difference between three areas (Calabria, North and South Apulia) but that oils from Sicily are similar to the oils from every other area (Figure 4.7 bottom row).

4.2.4 Taking stock:

What we have learned from this data is that the olive oils have dramatically different fatty acid composition depending on geographic region. The three larger geographic regions, North, South, Sardinia, are well-separated based on eicosenoic, linoleic and arachidic acids. The oils from areas in northern Italy are mostly separable from each other using all the variables. The oils from the inland and coastal areas of Sardinia have different amounts of oleic and linoleic acids. The oils from three of the areas in southern Italy are almost separable. And one is left with the curious content of the oils from Sicily. Why are these oils indistinguishable from the oils of all the other areas in the South? Is there a problem with the quality of these samples?

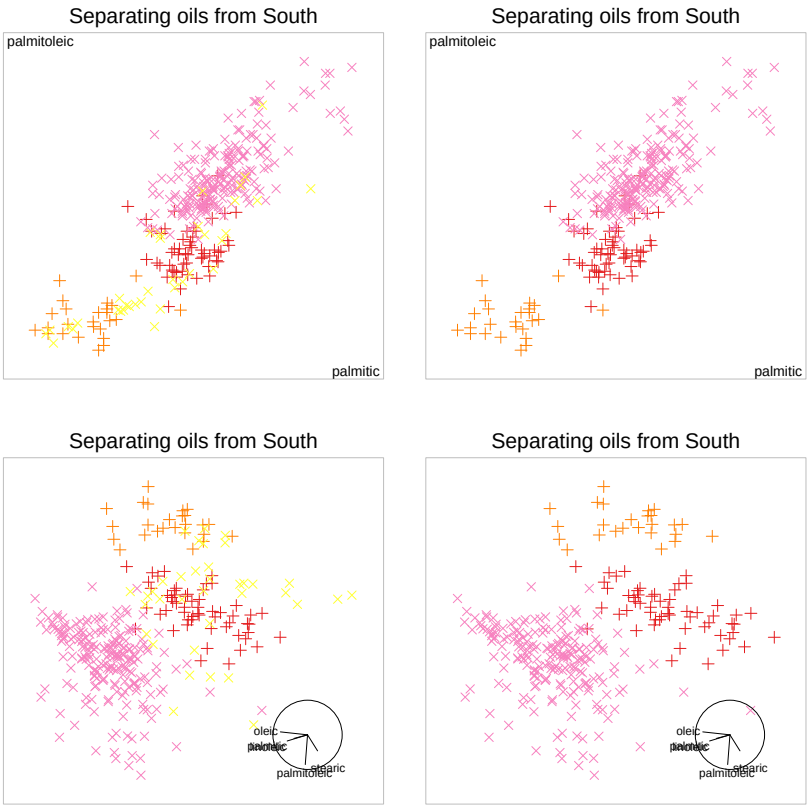

Fig. 4.7. The areas of southern Italy are almost separable, with the exception of the samples from Sicily (yellow) which overlap the points of the other three areas.

4.3 Numerical Methods

4.3.1 Linear discriminant analysis

Linear discriminant analysis (LDA) assumes the data arises from a mixture of multivariate normal distributions with equal variance-covariances. The boundaries between groups is placed mid-way between the class means, relative to the pooled variance-covariance.

For LDA to be appropriate for a particular a dataset we want to examine the variance-covariance of each cluster - it should be ellipsoidal, and equal between clusters. Using a tour we can check if many projections of the data exhibit these qualities. The olive oil data obviously does not have equal elliptical variance-covariance structure within the classes, as can be seen by looking

at only two variables in Figure 4.1. We don't need to do any more work here to realize that LDA is an not appropriate classifier for the olive oils data. However, we'd like to show how to check this assumption in high-dimensions, so we use the flea beetles data to illustrate. The first two variables of the flea beatles data (Figure 4.1) suggests the data conforms to the equal variance-covariance, multivariate normal model. But this data has six variables. Does this assumption hold for these additional variables. We check the assumption by examining the data using a 2D tour: If the data is consistent with the assumption then the clusters should be approximately elliptical and equal in variance in all projections viewed. Figure 4.8 shows two projections from a 2D tour. The projection of the data is shown at left and the projection of the 6D variance-covariance ellipsoid is shown at right. In some projections (bottom row) there are a few slight differences from the equal ellipsoidal structure but the differences are small enough to be due to sampling variability. In all the projections of the data viewed in a tour it looks like the flea beetles data is consistent with the multivariate normal mixture equal variance-covariance model.

LDA is often used to find the best low-dimensional view of the cluster structure (as discussed in Section 4.1.1). This is the linear combination of the variables where the class means are most separated relative to the pooled variance-covariance, called the discriminant space. It is obtained by computing the eigenvectors of $\mathbf{W}^{-1}\mathbf{B}$. This is typically computed and viewed staticly. The discriminant space may not be the perfect view of the cluster structure, but it usually provides a reasonable view of the class clusters. Based on the computation of the discriminant space (Lee, Cook, Klinke & Lumley 2004) proposed the LDA projection pursuit index:

$$I_{LDA}(\mathbf{A}) = \begin{cases} 1 - \frac{|\mathbf{A}'\mathbf{W}\mathbf{A}|}{|\mathbf{A}'(\mathbf{W}+\mathbf{B})\mathbf{A}|} & \text{for } |\mathbf{A}'(\mathbf{W}+\mathbf{B})\mathbf{A}| \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

The LDA index compares the variation of the means, with the pooled within-class variation. Higher values indicate more separation. Using a LDA index guided tour we can usually find a good separation between the class clusters and then use manual controls to explore the neighborhood.

The left plot in Figure 4.9 shows the discriminant space for the olive oil data, which is the projection which maximizes the LDA index. Note that, if LDA was used as a classifier on this data the boundary would be placed too close to the Southern oils (red) resulting in some misclassifications, due to the equal variance-coavariance assumption. But the projection uncovered by the LDA index is very informative. It shows the three regions nicely separated. This projection is a good starting place to manually search the neighborhood for a clearer view, that is, to sharpen the image. With very little effort the projection shown in right plot of Figure 4.9 emerges. In this view the three regions are better separated. Reducing the data to this projection would enable almost all classification methods to write accurate rules. So although we

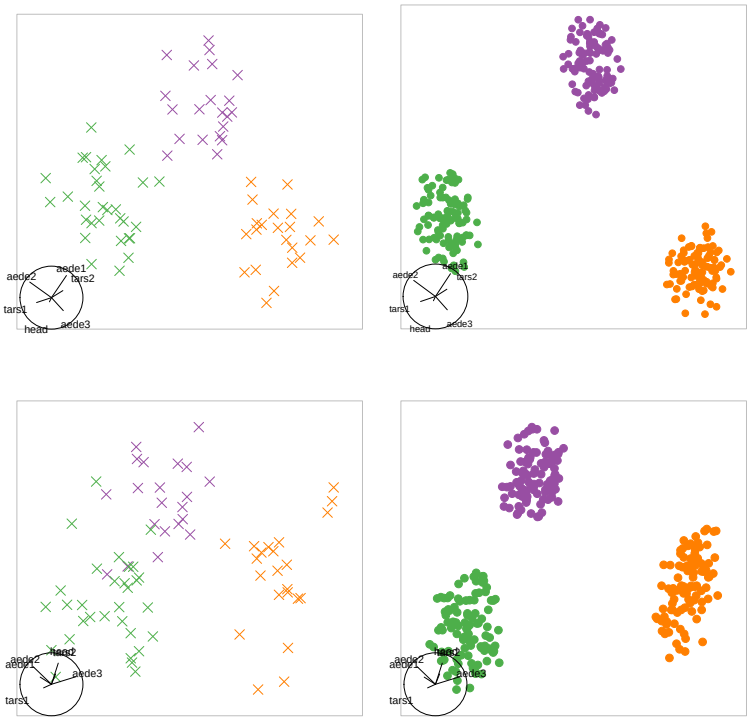

Fig. 4.8. Checking if the variance-covariance of the flea beetles data is ellipsoidal. Two, of the many, 2D tour projections of the flea beetles data viewed and ellipses representing the variance-covariance.

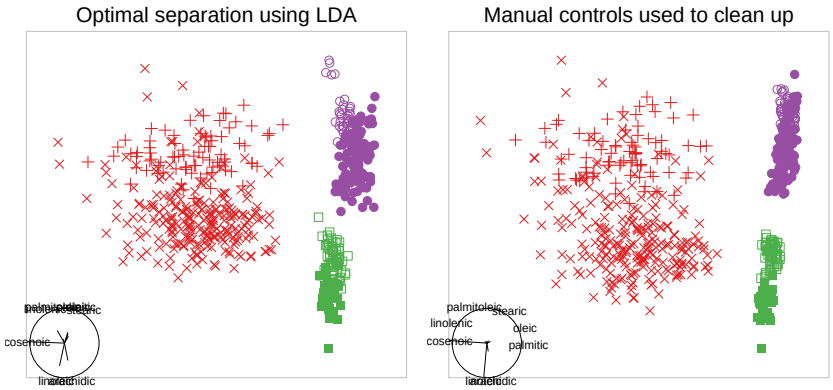

Fig. 4.9. Examining the discriminant space (left) and using manual controls to find a bigger gap between the Northern (purple) and Sardinian (green) oils.

wouldn't use LDA as a classifier here, it does help find a low-dimensional space which separates the classes.

We can learn more about LDA by exploring the misclassifications that occur if LDA is used to make a classifier:

	Predicted Region		
Region	South	Sardinia	North
South	322	0	1
Sardinia	0	98	0
North	0	4	147

Figure 4.10 shows the olive oil data plotted in the 2D discriminant space, with the misclassified samples highlighted using solid circles. In this projection, all misclassified points fall between the clusters. It is not at all surprising to see misclassifications where there is overlap, between the North and Sardinia regions. There is a more egregious misclassification represented by the red circle, showing that, despite the large gap between these clusters, one of the oils from the South was misclassified as an oil from the North. As discussed earlier, LDA is blind to the size of the gap when its assumptions are violated. Since the variance-covariance of these clusters is so different, LDA makes obvious mistakes.

The misclassified samples can be examined further using a tour by linking the misclassification table with other plots of the data (Figure 4.10). The southern oil sample that is misclassified is on the outer edge of the cluster of oils from the South, but it is very far from the points from the other regions. It really shouldn't be confused - it is clearly a southern oil. The four missclassified samples from the North really shouldn't be confused either: they are at one edge of the cluster of northern oils, but still far from the cluster of Sardinian oils.

4.3.2 Trees

The construction of classification trees is easy to explain, and in simple cases, generates a model that is easy to interpret. The values of each variable are searched for points where a split would best separate members of different classes; the best split is chosen. The data is then divided into the two subsets falling on each side of the split, and each subset is then searched for its best split. There are many algorithms for computing tree classifiers; on the olive oils data, using the region as the class variable, they all yield trees like the following:

```
If eicosenoic >= 7 then assign the sample to South.
Else
  If linoleic >= 1053.5 assign the sample to Sardinia.
  Else                  assign the sample to North.
```

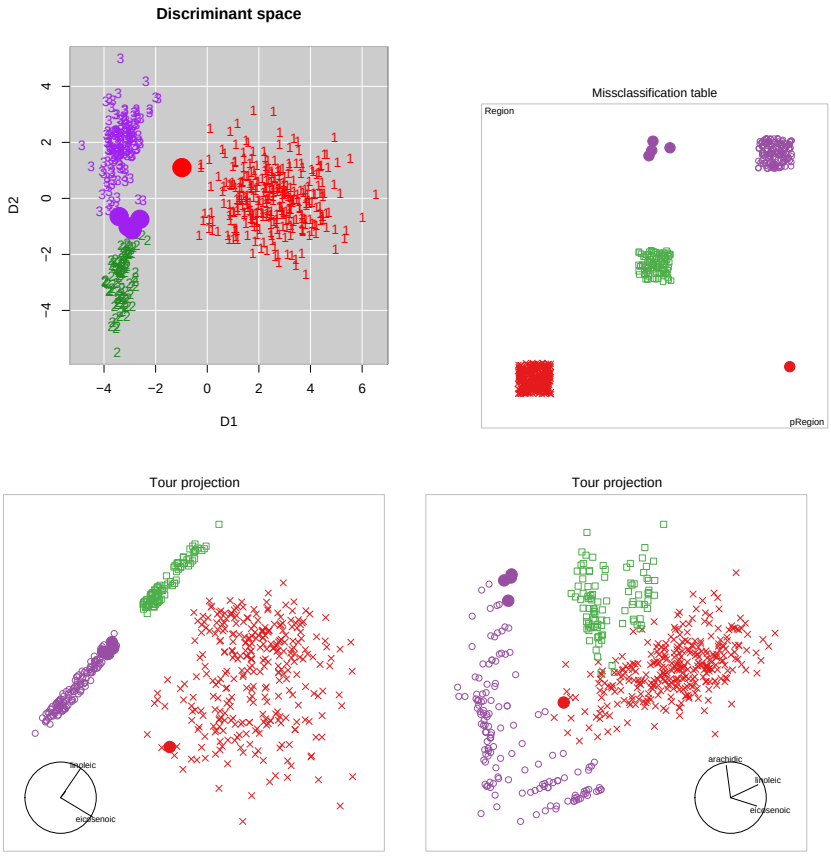

Fig. 4.10. Examining missclassifications from an LDA classifier for the regions of the olive oils data. In the discriminant space (top left) as computed using the `lda` function in the `MASS` library, the one misclassified Southern oil case is on the side of the cluster facing the other clusters. This case is far from the other clusters in other projections shown by the a 2D tour (bottom left). It should be missclassified. Similarly for the three Northern samples which are close to the Sardinian oils in the discriminant space, but far from these clusters in other projections (bottom right).

This tree yields a perfect classification of the data by region. Unlike LDA it doesn't confuse any oils from the South, but like the LDA result there is very little separation between oils from Sardinia and the North. The tree method doesn't consider the variance-covariance of the groups, it simply tries to place a knife between neighboring points to slice apart the classes. Thus it finds the separation between oils from the South and other regions, and slices the data right in the middle of the gap. For the other two regions it also finds a place to slice where oils of the North are on one side and Sardinian oils are on

the other, although there is no gap between these groups. The tree classifier yields a much simpler solution than that of LDA. Only two variables are used instead of a combination of 8 variables.

Tree classifiers effectively single out some of the important variables, but in general, use only one variable at a time to define splits. If linear combinations of variables would improve the model, these classifiers are likely to miss that fact. If a better classification can be found using a combination of variables they might approximate this using many splits along the different variables, zig-zagging a boundary between the clusters.

Accordingly the model produced by a tree classifier can sometimes be improved by exploring the neighborhood using the manual tour controls (Figure 4.11). Starting from the projection of the two variables selected by the tree algorithm, linoleic and eicosenoic acid, we find an improved projection by including just one other variable, arachidic acid. The gap between the North and Sardinian regions is distinctly wider.

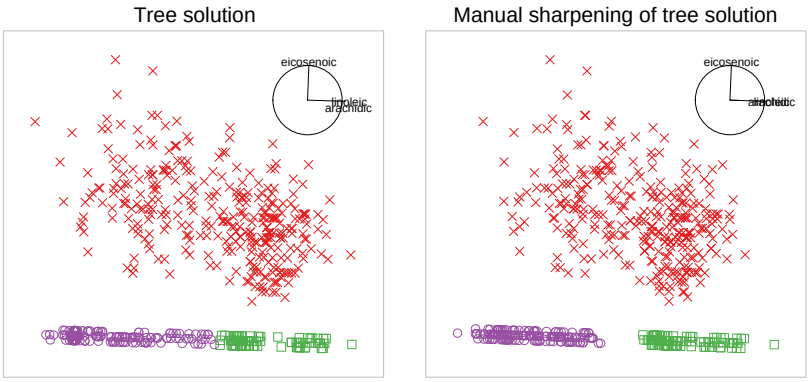

Fig. 4.11. The discriminant space, using only eicosenoic and linoleic acid, as determined by the tree classifier (left), is sharpened using manual controls (right).

We can capture the coefficients of rotation that generate this projection, and create a new variable. We define *linoarach* to be $\frac{0.969}{1022} \times linoleic + \frac{0.245}{105} \times arachidic$. We can add this new variable to the olive oils data, and run the tree classifier on the augmented data. The new tree is:

```
If eicosenoic >= 7 then assign the sample to South.
Else
  If linoarach>=1.09 assign the sample to Sardinia.
  Else               assign the sample to North.
```

Is this tree better than the original? They both have the same error for this data, so there is no difference numerically. Based on the plots in Figure 4.12, however, the sharpened tree looks more stable.

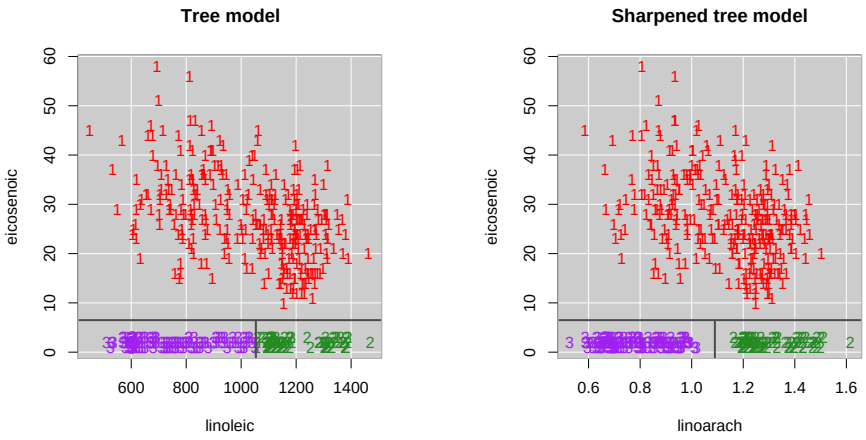

Fig. 4.12. Boundaries drawn in the tree model (left) and sharpened tree model (right).

4.3.3 Random Forests

A random forest (Breiman 2001) is a classifier that is built from multiple trees generated from random sampling the cases, and the variables. Forests are computationally intensive but retain some of the interpretability of trees. The code, documentation and other resources are available at Breiman & Cutler (2004), and there is also an R package, `randomForest` (Liaw 2006). A random forest is an example of a black box classifier, but with the addition of diagnostics that make the algorithm a little less mysterious.

Figure 4.13 illustrates how we would look at the diagnostics from a random forest classifier for the olive oils data. A forest of 500 trees is generated, each built from a random sample of four of the eight variables.

The random sampling of cases for each tree has the fortunate effect of creating a training (in the bag) and test (out of the bag) sample for each tree computed. The class of each case in the out-of-bag sample for each tree is predicted, and the predictions for all the trees are combined into a vote for the class identity. These votes are displayed in Figure 4.13, along with projections from a tour. Since there are three classes the votes data falls into a triangle, with one vertex for each region: South is at the far right, Sardinia is at the top, and North is in the lower left. Samples which are consistently classified

correctly are close to the vertices. Cases which are commonly misclassified are further from a vertex.

The pattern of the votes for the Northern and Sardinian samples suggest that there might be a potential for error in classifying future samples. Although forests perfectly classify this data, there is something interesting to be learned by studying these plots, and also another diagnostic from the forest, variable importance. Forests return two measures for the variable importance. Both measures give similar results. Using the Gini measure the order of importance of the variables is: eicosenoic, linoleic, oleic, palmitic, arachidic, palmitoleic, linolenic, stearic. This should surprise you! Some of the order is as expected, given the initial graphics analysis of the data. Eicosenoic acid is the most important. Yes, that's what we uncovered with graphics. Linoleic acid is next most important. Yes, this should be expected based on the plots we made of the data. The surprise is that arachidic acid is considered to be less important than palmitic.

Did we overlook something important in our earlier investigation? We return to the use of the manual manipulation of the tour to see if palmitic acid does in fact perform better than arachidic at finding a gap between the two regions. But it does not. By overlooking the importance of arachidic acid, the Random Forest never finds an adequate gap between the Northern and Sardinian regions, and that probably explains why there is more confusion about some Northern samples than is necessary.

If we re-build the forest using a new variable constructed from a linear combination of linoleic and arachidic (*linoarach*), just as we did when applying the tree classifier, the confusion between Northern and Sardinian oils disappears (Figure 4.13 bottom right). The new variable becomes the second most important variable according to the importance diagnostic. There is one qualification: The diagnostic for importance is affected by correlation between variables, with correlated variables reducing each other's importance. Because oleic, linoleic and linoarach are strongly correlated, both linoleic and oleic acids need to be removed for linoarch to have an appropriately high importance measure.

Classifying the regions is too easy a problem for forests; they are designed to tackle challenging classification tasks. We'll use them to examine the oils from areas in the southern region (Calabria, Sicily, North and South Apulia). Remember from the initial graphical analysis of the data that the four regions were not separable. The problem appeared to be that the samples from Sicily overlapped with each of the three other samples. We'll use a forest classifier to see how well it can distinguish these areas. Experimenting with several inputs we show the results for a forest of 1000 trees, sampling two variables at each tree node, yielding an out-of-bag error of 0.068. The misclassification table is:

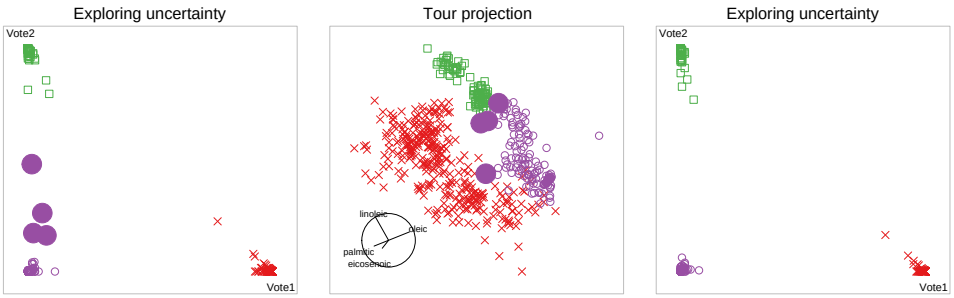

Fig. 4.13. Examining the results of a forest classifier on the olive oils. The votes assess the uncertainty associated with each sample. The corners of the triangle are the more certain classifications into one of the three regions. Points further from the corners are the samples that have been more comonly missclassified. These points are brushed (left plot) and we examine their location using the tour (middle plot). When a linear combination of linoleic and arachidic is entered into the forest there's no confusion between North and Sardinia (right plot).

	Predicted Area				
Area	North Apulia	South Apulia	Calabria	Sicily	Class Error
North Apulia	22	0	2	1	0.120
South Apulia	0	201	2	3	0.024
Calabria	0	2	54	0	0.036
Sicily	3	5	4	24	0.333

The overall error of the forest is surprisingly low, but a pattern can be seen in the error rates for each area. Predictions for Sicily are very poor, 0.33, wrong about a third of the time. Figure 4.14 shows some more interesting aspects of the results. We start with the top row of the figure. The misclassification table is represented by a jittered scatterplot, at the left. Plots of the four voting variables are in the center, and a single projection from a tour of the four most important variables is at right. Because there are four groups, the votes (in the center plot) lie on a 3D tetrahedron (a simplex). At the center is Sicily (blue cross), overlapping with the other three areas.

Remember that when points are clumped at the vertex, class members are consistently predicted correctly. Since this doesn't occur for Sicilian oils, we see that there is more uncertainty in the predictions for this area.

In the tour, we saw that the points corresponding to Sicilian oils overlap with the points from other areas in all projections. Clearly these are tough samples to classify correctly.

We remove these points from the plot so we can focus on the other three areas (bottom row of plots). The points representing North Apulia oils (orange plus) form a very tight cluster at a vertex, with two exceptions. These two points are misclassified as Calabrian (red plus). The pattern of the votes

suggests that there is high certainty in the predictions for North Apulian oils, with the exception of these two samples. Using the tour, we saw that the points do form a distinct cluster in the data space, which confirms this observation about the votes. Using brushing we explore the locations of the two misclassified points with a tour on the four most important variables. These two cases are outliers with respect to other North Apulia points. However they are not so far from their group - it is a bit surprising that the forest has trouble classifying these cases. Rather than exploring the other misclassifications, we leave that to the reader.

In summary, a random forest is a useful method for tackling tough classification problems. Its diagnostics provide a rich basis for graphical exploration, helping us to digest and evaluate the solution.

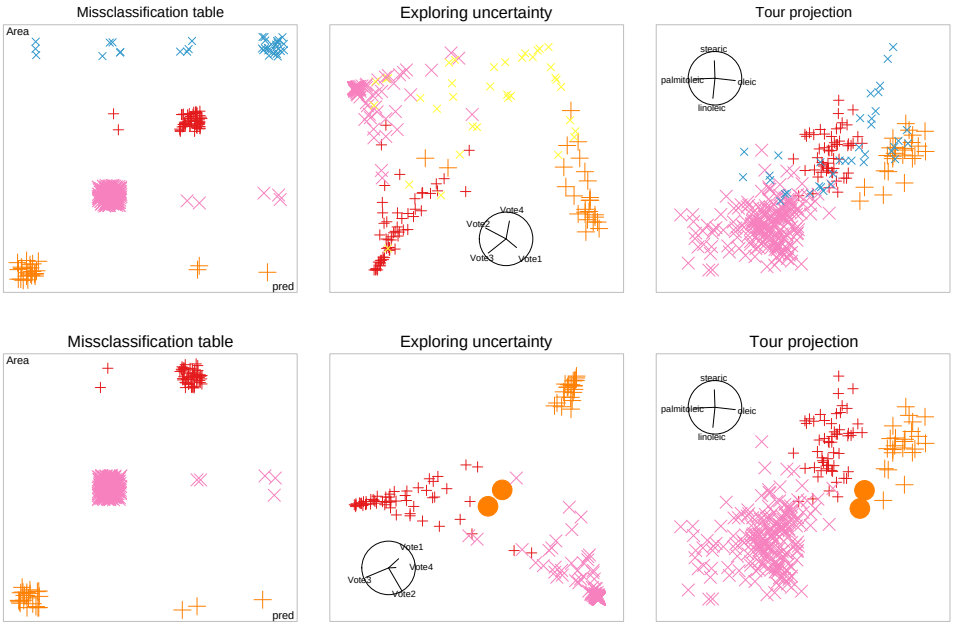

Fig. 4.14. Examining the results of a random forest for the difficult problem of classifying the oils from the four areas of the South. A representation of the missclassification table (left column) is linked to plots of the votes (middle column) and a 2D tour (right column).

In summary, a random forest provides a useful package for tackling tough classification problems. It's healthy collection of diagnostics provides a rich basis for graphical exploration helping to digest the solution.

4.3.4 Neural Networks

Neural networks for classification can be thought of as additive models, where explanatory variables are transformed, usually through a logistic function, added to the other explanatory variables, transformed again, added again to to yield class predictions. A good description can be found in (Cheng & Titterington 1994). The model can be formulated as:

$$\hat{y} = f(\mathbf{x}) = \phi(\alpha + \sum_{h=1}^{s} w_h \phi(\alpha_h + \sum_{i=1}^{p} w_{ih} x_i))$$

where x is the vector of explanatory variable values, y is the target value, p is the number of variables, s is the number of nodes in the single hidden layer and ϕ is a fixed function, usually a linear or logistic function. This model has a single hidden layer, and univariate output values. The model is typically fit by minimizing the sum of squared differences between observed values and fitted values. The minimization may not always converge. Neural networks are a black box method: enter inputs, compute, spit out predictions. With graphics, some insight into the black box can be gained. We use the feed-forward neural network, provided in the `nnet` package of R (Venables & Ripley 2002), to illustrate.

We continue to work with the olive oils data, and we look at the performance of the neural network in classifying the four areas of the South, a difficult challenge. Because the software doesn't include a method for computing the predictive error, we'll break the data into training and test samples so we can better estimate the error. The neural network could be tweaked to perfectly fit the current data, but we'd like to be able to assess how well it would do with new data. We'll use the training subset to build the classifier, and the test subset to compute the predictive error. After trying several values for s, the number of nodes in the hidden layer, we chose $s = 4$, and linear ϕ, $decay = 0.005$, and $range = 0.06$. We optimize the model fit from many random starts, until it finally converged to an accurate solution. Below are the missclassification tables for training and test samples:

Training

	Predicted Area			
Area	Nth Ap	Sth Ap	Calab	Sic
Nth Ap	16	1	0	2
Sth Ap	0	155	1	2
Calab	0	0	42	0
Sic	1	1	1	24

Test

	Predicted Area			
Area	Nth Ap	Sth Ap	Calab	Sic
Nth Ap	3	0	2	1
Sth Ap	0	45	2	1
Calab	0	2	12	0
Sic	1	1	2	5

The training error is $9/246 = 0.037$, and the test error is $12/77 = 0.156$. The missclassifications are explored in Figure 4.15. There are three samples of North Apulia oils (orange plus) that are missclassified: one is incorrectly

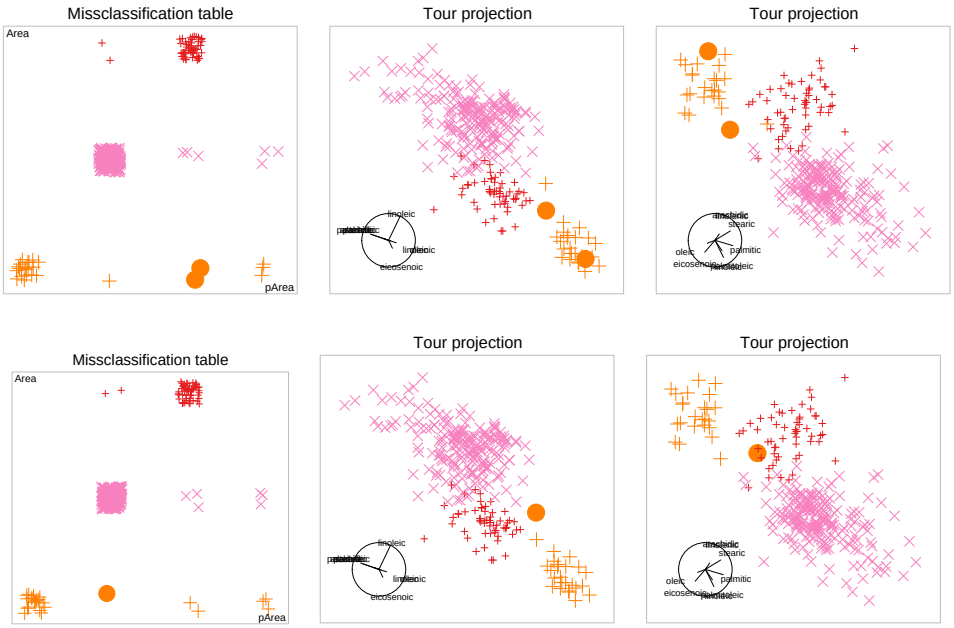

Fig. 4.15. Examining the results of a feed-forward neural network on the problem of classifying the oils from the four areas of the South. A representation of the misclassification table (left column) is linked to projections viewed in a 2D tour.

classified as from South Apulia (pink cross) and two are incorrectly classifed as from Calabria (red plus). The plots in the left side of Figure 4.15 illustrate the missclassification table. We highlight the two North Apulia cases missclassified as Calabrian oils, and observe them in a tour (top row of plots). One of the two is on the edge of the cluster of North Apulia points close to the Calabria cluster. It's understandable that there might be some confusion about this case. The other sample is on the outer edge of the North Apulia cluster, but it is far from the Calabria cluster - this shouldn't have been confused. Next we examine the one North Apulia sample missclassified as South Apulian. It is highlighted in the representation of the missclassification table, and viewed in a tour. This point is on the outer edge of North Apulia cluster but it is closer to the Calabria cluster than the South Apulia cluster. It would be understandable for it to be missclassified as Calabrian, so it's a bit puzzling that it is missclassified as South Apulian.

Our exploration of the misclassifications is shown in Figure 4.15. The plots at the left side of Figure 4.15 show the misclassification table for all four areas. There are three samples of oils from North Apulia (orange plus) that are misclassified: one is incorrectly classified as South Apulian (pink cross) and two are incorrectly classified as Calabrian (red plus). In the misclassification

plot in the upper left, we highlight the two North Apulia cases misclassified as Calabrian oils, and observe them in a tour (see the other two plots in the top row). One of the two is on the edge of the cluster of North Apulian points close to the Calabrian cluster. It is understandable that there might be some confusion about this case. The other sample is on the outer edge of the North Apulian cluster, but it is far from the Calabrian cluster - this shouldn't have been confused.

In the bottom row of plots, we follow the same procedure to examine the one North Apulian sample misclassified as South Apulian. It is highlighted in the misclassification plot, and viewed in a tour. This point is on the outer edge of North Apulia cluster but it is closer to the Calabria cluster than the South Apulia cluster. It would be understandable for it to be misclassified as Calabrian, so it's a bit puzzling that it is misclassified as South Apulian.

4.3.5 Support Vector Machine

A support vector machine (SVM) (Vapnik 1999) is a binary classification method that takes an $n \times p$ data matrix, where each column (variable or feature) is scaled to [-1,1] and each row (case or instance) is labelled as one of two classes ($y_i = +1$ or -1), and finds a hyperplane which separates the two groups, if they are separable. Each row of the data matrix is a vector in p-dimensional space, denoted as

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}$$

and the separating hyperplane can be written as:

$$\mathbf{W}'\mathbf{X} + b = 0$$

where $\mathbf{W} = [w_1 \quad w_2 \quad \ldots \quad w_p]'$ is the normal vector to the separating hyperplane and b is a constant. The best separating hyperplane is found by maximizing the margin of separation between the two classes as defined by two parallel hyperplanes:

$$\mathbf{W}'\mathbf{X} + b = 1, \qquad \mathbf{W}'\mathbf{X} + b = -1.$$

These hyperplanes should maximize the distance from the separating hyperplane, and have no points between them, capitalizing on any gap between the two classes. The distance from the origin to the separating hyperplane is $|b|/||\mathbf{W}||$, thus the distance between the two parallel margin hyperplanes is $2/||\mathbf{W}|| = 2/\sqrt{w_1^2 + \ldots + w_p^2}$. Maximizing this is the same as minimizing

$||\mathbf{W}||/2$. To ensure that the two classes are separated, and that no points lie between the margin hyperplanes we need:

$$\mathbf{W}'\mathbf{X}_i + b \geq 1, \quad \text{or} \quad \mathbf{W}'\mathbf{X}_i + b \leq -1 \quad \forall i = 1, ..., n$$

which corresponds to:

$$y_i(\mathbf{W}'\mathbf{X}_i + b) \geq 1 \quad \forall i = 1, ..., n \tag{4.1}$$

Thus the problem corresponds to:

$$\text{minimizing } \frac{||\mathbf{W}||}{2} \text{ subject to } y_i(\mathbf{X}_i\mathbf{W} + b) \geq 1 \quad \forall i = 1, ..., n.$$

Interestingly, only the points closest to the margin hyperplanes are needed to define the separating hyperplane. We might think of these points as the ones on the convex hull of each cluster, opposing each other. These points are called support vectors, and the coefficients of the separating hyperplane are computed from a linear combination of the support vectors $\mathbf{W} = \sum_{i=1}^{s} y_i\alpha_i\mathbf{X}_i$, where s is the number of support vectors. We could also use of $\mathbf{W} = \sum_{i=1}^{n} y_i\alpha_i\mathbf{X}_i$, where $\alpha_i = 0$ if $\mathbf{X}_i$ is not a support vector. For a good fit the number of support vectors, s, should be small relative to the n. Fitting algorithms can gain achieve gains in efficiency by examining samples of the cases rather than all the data points to find suitable support vector candidates, which is the approach used in SVMLight (Joachims 1999).

In practice, the assumption that the classes are separable classes is unrealistic. Classification problems rarely present a gap between the classes, resulting in no missclassifications. Cortes & Vapnik (1995) relaxed the separable condition to allow some missclassified training points by adding a tolerance value, ϵ_i, to $y_i(\mathbf{W}'\mathbf{X}_i + b) > 1 - \epsilon_i, \epsilon_i \geq 0$. Points that meet this criterion instead of the strictor (4.1) are called slack vectors.

Nonlinear classifiers can be obtained by using nonlinear transformations of $\mathbf{X}_i$, $\phi(\mathbf{X}_i)$ (Boser, Guyon & Vapnik 1992), which is implicitly computed during the optimization using a kernel function, K. Common choices of kernels are linear, $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i'\mathbf{x}_j$, polynomial $K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma\mathbf{x}_i'\mathbf{x}_j + r)^d$, radial basis $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2)$ or sigmoid functions $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma\mathbf{x}_i'\mathbf{x}_j + r)$, where $\gamma > 0, r, d$ are kernel parameters.

The ensuing minimization problem is formulated as:

$$\text{minimizing } \frac{1}{2}||\mathbf{W}|| + C\sum_{i=1}^{n}\epsilon_i \text{ subject to } y_i(\mathbf{W}'\phi(\mathbf{X}) + b) > 1 - \epsilon_i$$

where $\epsilon_i \geq 0$, and $C > 0$ is a penalty parameter guarding against overfitting the training data, ϵ controls the tolerance for missclassification. The normal to the separating hyperplane, $\mathbf{W}$, can be written as $\sum_{i=1}^{n} y_i\alpha_i\phi(\mathbf{X}_i)$, where points other than the support and slack vectors will have $\alpha_i = 0$. Thus the optimization problem becomes:

$$\text{minimizing } \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j K(\mathbf{X}_i, \mathbf{X}_j) + C\sum_{i=1}^{n}\epsilon_i \text{ subject to } y_i(\mathbf{W}'\phi(\mathbf{X}) + b) > 1 - \epsilon_i$$

We use the `svm` function in the `e1071` package (Dimitriadou, Hornik, Leisch, Meyer & Weingessel 2006) of R, which uses `libsvm` (Change & Lin 2006) to classify the olive oils of the four areas in southern Italy, as demonstrated for random forests and neural networks. SVM is a binary classifier but this algorithm overcomes this limitation by comparing classes in pairs, fitting 6 separate classifiers and using a voting scheme to make predictions. To fit the SVM we also need to specify a kernel, or rely on the internal tuning tools of the algorithm to choose this for us. Automatic tuning in the algorithm chooses a radial basis, which actually gives a poorer classification than a linear kernel, from a practical perspective. An earlier visual inspection of the data (Section 4.2) suggests a linear kernel would be sufficient. A linear kernel produces a very good classification, as can be seen in the missclassifcation tables:

Training

	Predicted Area			
Area	Nth Ap	Sth Ap	Calab	Sic
Nth Ap	19	0	0	0
Sth Ap	0	155	0	3
Calab	0	0	42	0
Sic	1	3	2	21

Test

	Predicted Area			
Area	Nth Ap	Sth Ap	Calab	Sic
Nth Ap	6	0	0	0
Sth Ap	0	46	0	2
Calab	1	1	12	0
Sic	1	0	1	7

The training error is $9/246 = 0.037$, and the test error is $6/77 = 0.078$, which lower than the neural network classifier. On closer inspection most of the error is associated with Sicily, which we've already seen from the graphical analysis to be a problematic group. The fatty acid values for some Sicilian oils are more similar to the values from the other three areas. In the training data there are no other errors and in the test data there are just two samples from Calabria (highlighted as solid circles) mistakenly classified. Figure 4.16 illustrates how we examine the missclassified cases. (Points corresponding to Sicily were removed from the plots, to make it easier to digest the results.) Both of these cases are on the edge of their clusters so the confusion of identities is reasonable.

The linear SVM classifier uses 20 support vectors and 29 slack vectors to define the separating planes between the 4 regions. It is interesting to examine which points are selected as support vectors, and where they are located in the data space. Figure 4.17 illustrates this. (The Sicilian points are again removed.) The support vectors are represented by open circles and the slack vectors by open rectangles. We would expect that the support vectors will line up on either side of the margin of separation in some projection. The slack vectors will be closer to the boundary and perhaps mixed in with the points of

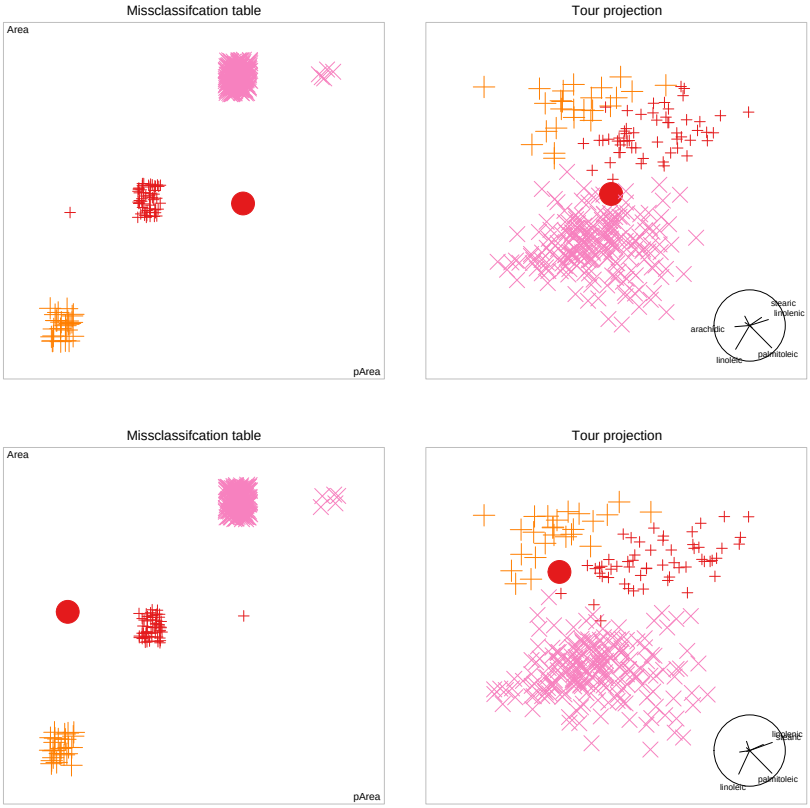

Fig. 4.16. Examining the results of a support vector machine on the problem of classifying the oils from the four areas of the South, by linking the missclassification table (left) with 2D tour plots (right).

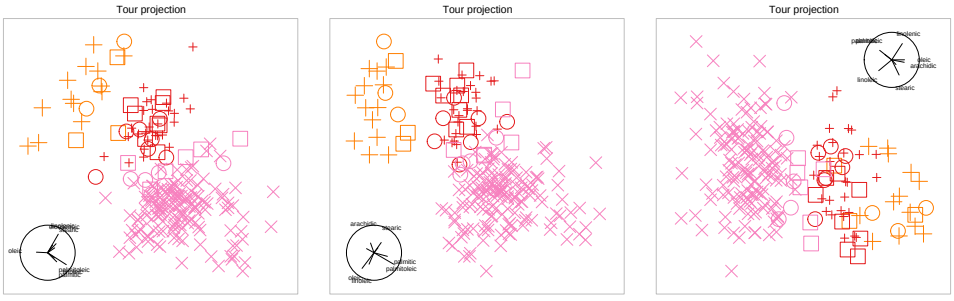

Fig. 4.17. Using the tour to examine the choice of support vectors on the problem of classifying the oils from the four areas of the South. Support vectors are open circles and slack vectors are open rectangles.

other classes. For this problem the expectation seems to hold reasonably well. We check this using the tour, looking at different projections of the data, using the grand tour, and manual controls to try to line up the support vectors. The support vectors are on the opposing outer edge of the point clouds for each cluster.

The linear SVM does a very nice job with this difficult classification. The error is mostly associated with Sicily and the accuracy is almost perfect on the remaining classes, which validates what we observed on the data.

4.3.6 Examining boundaries

For some classification problems it's possible to get a good picture of the boundary between classes. With LDA and SVM classifiers the boundary is described by the equation of a hyperplane. For others the boundary can be determined by evaluating the classifier on points sampled in the data space, either a regular grid, or using a more efficient sampling scheme.

We use the R package `classifly` (Wickham 2006) to explore the boundaries of different classifiers on the olive oils data. Figure 4.18 show some examples of studying the boundary between pairs of groups. In each example the grand tour was used with manual control to focus the view on a projection that revealed the boundary between the two groups. The top two plots show tour projections of the Northern (purple) and Sardinian (green) oils where the two classes are separated and a view of the boundary generated by LDA (left) and SVM (right). The LDA boundary slices too close to the norrthern oils. This might not be unexexpected because LDA assumes equal variance between the groups, and if this is not true, then it places the boundary too close to the group with the larger variance. The SVM boundary is slightly shifted towards the Sardinian oils yet it is still a tad too close to the northern oils. The bottom row of plots examines the more difficult classification of the areas of the South, focusing on separating the South Apulian oils (red), which is the largest sample, from the oils of the other areas (pink). There isn't a perfect separation between the classes. Both plots are tour projections showing SVM boundaries generated by a linear kernel (left) and a radial kernel (right). The radial kernel is chosen by the SVM tuning as the best classification of the two classes. Based on studying the boundaries, though the linear SVM provides a more reasonable boundary between the two groups. The shape of the clusters of the two groups is approximately the same, and there is only a small overlap of the two clusters. The linear boundary fits this structure neatly. The radial kernel wraps the South Apulian oils.

4.4 Reduction

The olive oils example demonstrates how it is possible to get a good mental image of cluster structure in relation to class identities in high dimensional

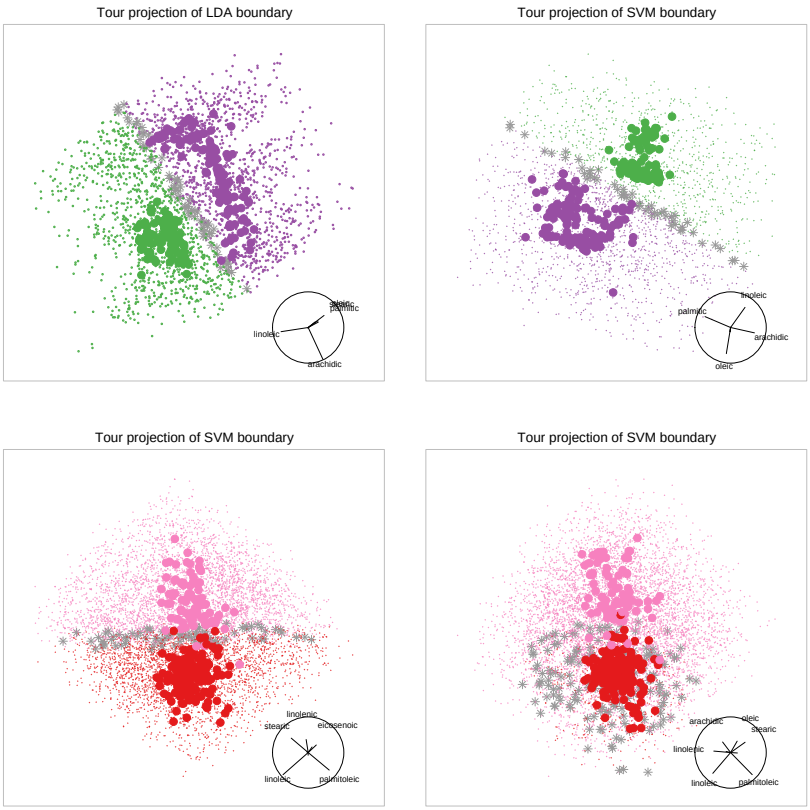

Fig. 4.18. Using the tour to examine the classification boundary. Points on the boundary are grey stars. (Top row) Boundary between North and Sardinian oils (left) LDA (right) linear SVM. Both boundaries are too close to the cluster of northern oils. (Bottom row) Boundary between South Apulia and other Southern area oils using (left) linear SVM (right) radial kernel SVM, as chosen by the tuning functions for the software.

space. This is possible with many multivariate data sets. Having a good mental image of the class structure helps improve a classification analysis in many ways: to choose an appropriate classifier, validate or reject the results of a classification, and simplify the final model. For the olive oils data, we saw that the data had a story to tell: the olive oils of Italy are remarkably different in composition based on geographic boundaries. We also learned that there is something fishy about the Sicilian oils and the most plausible story is that the Sicilian oils used borrowed olives from neighboring regions. This is interesting! Data analysis is detective work.

Visual methods give a richer understanding of how a classifier is performing. It can be very surprising to examine the boundary generated by a classifier

that on the surface of the error rate looks perfect. Quite commonly the boundary is oddly placed. In linear classifiers the separating hyperplane might be too close to one group, or sloped at an odd angle. For non-linear classifiers the boundary often poorly fits regions outside the current data range. We have also seen how visual methods help to discover outliers that might influence a model, and to determine which variables are more important for separating groups.

4.5 Exercises

1. This question uses the flea beetle data.
   a) Generate a scatterplot matrix of the flea beetle data. Which variables would contribute to separating the 3 species?
   b) Generate a parallel coordinate plot of the flea beetle data. Characterize the 3 species by the pattern of their traces.
   c) Watch the flea beetle data in a grand tour. Stop the tour when you see a separation and describe the variables that contribute to the separation.
   d) Using the projection pursuit guided tour, with the holes index, find a projection which neatly separates all 3 species. Put the axes onto the plot and explain the variables that are contributing to the separation. Using univariate plots confirm that these variables are important to separate species.
2. This question is about the Australian crabs data.
   a) From univariate plots assess if any individual variables are good classifiers of species or sex.
   b) From either a scatterplot matrix or pairwise plots, determine which pairs of variables best distinguish the species, and sexes within species.
   c) Examine the parallel coordinate plot of the 5 measured variables. Why isn't a parallel coordinate plot helpful to determine the importance of variables for this data?
   d) Using Tour1D (perhaps with projection pursuit with LDA index) find a 1-dimensional projection which mostly separates the species. Report the projection coefficients.
   e) Now transform the 5 measured variables into principal components and run Tour1D on these new variables. Is a better separation between the species to be found?
3. This question is about the olive oils data.
   a) Split the samples from Northern Italy into 2/3 training and 1/3 test samples for each area.
   b) Build a tree model to classify the three areas of Northern Italy. Which are the most important variables. Make plots of these variables. What is the accuracy of the model for the training and test sets?

c) Build a random forest to classify the three areas of Northern Italy. Compare the order of importance of variables with what you found from a single tree. Make a parallel coordinate plot in the order of the variable importance.
d) Fit a support vector machine model and a feed-forward neural network model to classify the three areas of Northern Italy. Using plots compare the predictions of each point for SVM, FFNN and random forests.

4. This question is about the TAO data. Build a classifier to distinguish between the normal and El Niño years. Depending on the classifier you use you may need to impute the missing values first. Which variables are important?
5. This question is about the spam data.
   a) Create a new variable "Domain.reduced" that reduces the number of categories in the "Domain" variable to be "edu", "com","gov","org", "net","other".
   b) Using the variable "Spam" as the class variable, and explanatory variables are Day.of.Week, Time.of.Day, Size..kb., Box, Domain.reduced, Local, Digits, name, X.capital, Special, credit, sucker, porn, chain, username, Large.text, build a random forests classifier using $mtry = 2$.
   c) What is the order of importance of the variables?
   d) How many non-spam emails are misclassified as spam?
   e) Examine a scatterplot of predicted class against actual class, using jittering to spread the values, and a parallel coordinate plot of the explanatory variables in the order of importance returned by the forest. Brush the cases corresponding to non-spam email that has been predicted to be spam. Describe the types of emails these are? (all frmo the local box, small number of digits, ...) Now look at the emails that are spam and correctly classified as spam. Is there something special about these emails?
   f) Examine the relationship between Spam (actual class) and Spam.Prob (probability of being spam as estimated by ISU's mail facilities). How many cases that are not spam are rated as more than 50% likely to be spam?
   g) Examine the probability rating for cases corresponding to non-spam that random forests classifies as spam. Write a description of the email that has the highest probability of spam and is also considered to be very likely to be spam by random forests.
   h) Which user has the highest proportion of non-spam email classed as spam?
   i) Based on your exploration of this data, which variables would you suggest are the most important in determining if an email is spam or not?
6. This question is about the music data. The goal is to build a a classifier to distinguish Rock from Classical tracks.

a) For the music data there are 70 explanatory variables for 62 samples. Reduce the number of variables to less than 10, that are the best suitable candidates on which to build a classifier. Hint: One of the problems to consider is that there are several missing values. It might be possible to do the variable reduction in a way that also fixes the missing values problem.
b) Split the data into 2/3 training and 1/3 test data. Report which cases are in each sample.
c) Build your best classifier for distinguishing Rock from Classical tracks.
d) Predict the five new tracks as either Rock or Classical.

5

Cluster Analysis

5.1 Background

The aim of unsupervised classification, or cluster analysis, is to organize observations into similar groups. Cluster analysis is a commonly used, appealing and conceptually intuitive statistical method. Some of its uses include market segmentation, where customers are grouped into clusters with similar attributes for targeted marketing; gene expression analysis, where genes with similar expression patterns are grouped together; and the creation of taxonomies of animals, insects or plants. A cluster analysis results in a simplification of a data set for two reasons: first, because each cluster, which is now relatively homogeneous, can be analyzed separately, and second, because the data set can be summarized by a description of each cluster. Thus, it can be used to effectively reduce the size of massive amounts of data.

Organizing objects into groups is a task that seems to come naturally to humans, even to small children, and perhaps this is why it's an apparently intuitive method in data analysis. But cluster analysis is more complex than it initially appears. Many people imagine that it will produce neatly separated clusters like those in the top left plot of Figure 5.1, but it almost never does. Such ideal clusters are rarely encountered in real data, so we often need to modify our objective from "find the natural clusters in this data" to "organize the cases into groups that are similar in some way." Even though this may seem disappointing when compared with the ideal, it is still often an effective means of simplifying and understanding a data set.

At the heart of the clustering process is the work of discovering which variables are most important for defining the groups. It is often true that we only require a subset of the variables for finding clusters, while another subset (called "nuisance variables") has no impact. In the bottom left plot of Figure 5.1, it is clear that the variable plotted horizontally is important for splitting this data into two clusters, while the variable plotted vertically is a nuisance variable. Nuisance is an apt term for these variables that radically change the interpoint distances and impair the clustering process.

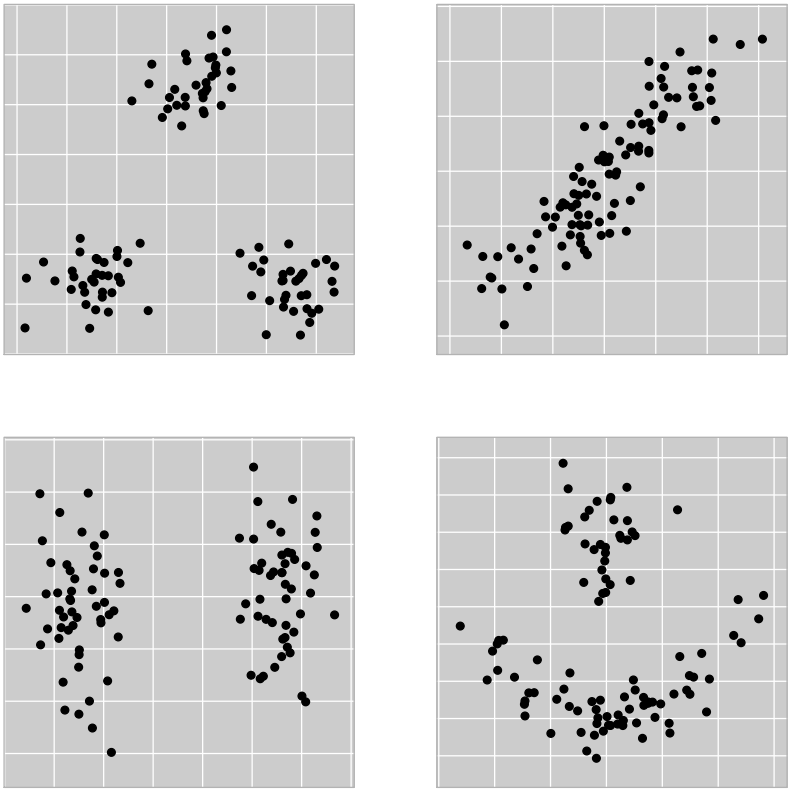

Fig. 5.1. Cluster analysis involves grouping similar observations. When there are well-separated groups the problem is conceptually simple (top left). Often there are not well-separated groups (top right) but grouping observations may still be useful. There may be nuisance variables which don't contribute to the clustering (bottom left), and there may odd shaped clusters (bottom right).

Dynamic graphical methods help us to find and understand the cluster structure in high dimensions. With the tools in our toolbox, primarily tours, along with linked scatterplots and parallel coordinate plots, we can see clusters in high-dimensional spaces. We can detect gaps between clusters, the shape and relative positions of clusters, and the presence of nuisance variables. We can even find unusually shaped clusters, like those in the bottom right plot in Figure 5.1. In simple situations we can use graphics alone to group observations into clusters, using a "spin and brush" method. In more difficult data problems, we can assess and refine numerical solutions using graphics.

Before we can begin finding groups of cases that are similar, we need to decide on a definition of similarity. How is similarity defined? Consider a data set with 3 cases and 4 variables, described in matrix format as:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{X}_3 \end{bmatrix} = \begin{bmatrix} 7.3 & 7.6 & 7.7 & 8.0 \\ 7.4 & 7.2 & 7.3 & 7.2 \\ 4.1 & 4.6 & 4.6 & 4.8 \end{bmatrix}$$

which is plotted in Figure 5.2. The Euclidean distance between two cases (rows of the matrix) is defined as:

$$\begin{aligned} d_{Euc}(\mathbf{X}_i, \mathbf{X}_j) &= \sqrt{(\mathbf{X}_i - \mathbf{X}_j)'(\mathbf{X}_i - \mathbf{X}_j)} \\ &= \sqrt{(X_{i1} - X_{j1})^2 + \ldots + (X_{ip} - X_{jp})^2}, \quad i, j = 1, \ldots, n. \end{aligned}$$

For example, the Euclidean distance between cases 1 and 2 in the above data, is

$$\sqrt{(7.3 - 7.4)^2 + (7.6 - 7.2)^2 + (7.7 - 7.3)^2 + (8.0 - 7.2)^2} = 1.0.$$

For the three cases, the interpoint Euclidean distance matrix is:

$$d_{Euc} = \begin{bmatrix} 0.0 & & \\ 1.0 & 0.0 & \\ 6.3 & 5.5 & 0.0 \end{bmatrix} \begin{matrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{X}_3 \end{matrix}$$

Cases 1 and 2 are more similar to each other than they are to case 3, because the Euclidean distance between cases 1 and 2 is much smaller than the distanced between cases 1 and 3, and cases 2 and 3.

There are many different ways to calculate similarity. In recent years similarity measures based on correlation distance have become common. Correlation distance is typically used where similarity of structure is more important than similarity in magnitude.

As an example, see the parallel coordinate plot of the sample data at the right of Figure 5.2. Cases 1 and 3 are widely separated, but their shapes are similar (low, medium, medium, high). Case 2, while overlapping with Case 1, has a very different shape (high, medium, medium, low). The correlation between two cases is defined as:

$$\rho(\mathbf{X}_i, \mathbf{X}_j) = \frac{(\mathbf{X}_i - c_i)'(\mathbf{X}_j - c_j)}{\sqrt{(\mathbf{X}_i - c_i)'(\mathbf{X}_i - c_i)}\sqrt{(\mathbf{X}_j - c_j)'(\mathbf{X}_j - c_j)}}. \tag{5.1}$$

where c_i, c_j are the sample means $\bar{\mathbf{X}}_i, \bar{\mathbf{X}}_j$ and ρ is the Pearson correlation coefficient. If they are set at 0, as is commonly done, ρ describes the angle

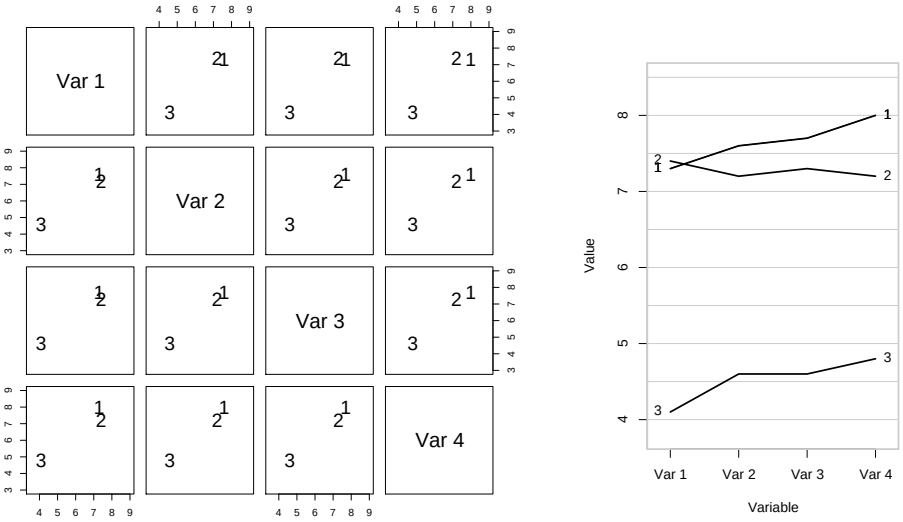

Fig. 5.2. (Left) Scatterplot matrix of example data. (Right) Parallel coordinates of example data.

between the two data vectors. The correlation is then converted to a distance metric; one equation for doing so is this:

$$d_{Cor}(\mathbf{X}_i, \mathbf{X}_j) = 2(1 - \rho(\mathbf{X}_i, \mathbf{X}_j))$$

Distance measures built on correlation are effectively angular distances between points, because for two vectors $\mathbf{X}_i and \mathbf{X}_j$, $\cos(\angle(\mathbf{X}_i, \mathbf{X}_j)) \propto \mathbf{X}_i'\mathbf{X}_j$. The above distance metric will treat cases that are strongly negatively correlated as the most distant.

The interpoint distance matrix for the sample data using d_{Cor} and the Pearson correlation coefficient is:

$$d_{Cor2} = \begin{bmatrix} 0.0 & & \\ 3.6 & 0.0 & \\ 0.1 & 3.8 & 0.0 \end{bmatrix}$$

By this metric, cases 1 and 3 are the most similar, because the correlation distance is smaller between these two cases than the other pairs of cases.

Note that the interpoint distances differ dramatically from those for Euclidean distance. As a consequence, the way the cases would be clustered is also be very different. Choosing the appropriate distance measure is an important part of a cluster analysis.

We've already drawn your attention to the parallel coordinate plot in Figure 5.2. It's a helpful plotting method to use with cluster analysis, both for exploring the data and for assessing the results.

It is actually difficult to determine whether the results of a cluster analysis are good. Cluster analysis is best thought of as an exploratory technique: There are no p-values, and the process tends to produce hypotheses rather than testing them. Even the most determined attempts to produce the "best" results using modeling and validation techniques may result in clusters which, while seemingly significant, are useless for practical purposes. On the other hand, even without formal validation, the results of a cluster analysis may be useful. The context in which the data arises is the key to determining an appropriate distance metric and assessing the usefulness of the results. If a company can gain an economic advantage by using a particular clustering method to carve up the customer database, then that's the method they should use.

The next section describes an example of a purely graphical approach to cluster analysis, the spin-and-brush method, which works for simple clustering problems. In this example we were able to find simplifications of the data that had not been found using numerical clustering methods, and to find a variety of structures in high-dimensional space. Section 5.3 describes methods for reducing the interpoint distance matrix to an intercluster distance matrix using hierarchical algorithms and model-based clustering, and shows how graphical tools are used to assess the results.

5.2 Purely graphics

A purely graphical spin-and-brush approach to cluster analysis works well when there are good separations between groups, even when there are marked differences in variance structures between groups or when groups have nonlinear boundaries. It doesn't work very well when there are classes which overlap, or when there are no distinct classes but rather we simply wish to partition the data. In these situations it may be better to begin with a numerical solution and use visual tools to evaluate it, perhaps making refinements subsequently. Several examples of the spin-and-brush approach are documented in the literature, such as Cook, Buja, Cabrera & Hurley (1995) and Wilhelm, Wegman & Symanzik (1999).

This description of the spin-and-brush approach on particle physics data follows that in Cook et al. (1995). The data contains seven variables. We have no labels for the data, so when we begin, all the points have the same color and glyph. Watch the data in a tour for a few minutes and you'll see that there are no natural clusters, but there is clearly structure.

We'll use the projection pursuit guided tour. We'll rotate the principal components rather than the raw variables, because that improves the performance of the projection pursuit indices. There are two indices that are useful

for detecting clusters: holes and central mass. The holes index is sensitive to projections where there are few points (i.e., a hole) in the center. The central mass index is the opposite: it is sensitive to projections that have too many points in the center. These indices are explained in Chapter 2.

The holes index is usually the most useful for clustering, but not for the particle physics data, because it does not have a "hole" at the center. The central mass index is the most appropriate here. Alternate between optimization (a guided tour) and the unguided grand tour to find local maxima, each of which is a projection which is potentially useful for revealing clusters. The process is illustrated in Figure 5.3.

The top left plot shows the initial default projection, Principal Component 2 plotted against Principal Component 1. The plot next to it shows the projected data corresponding to the first local maximum found by the guided tour. It has three strands of points stretching out from the central clump, and several outliers. We brush the points along each strand, in red, blue, orange, and the outliers are changed to open circles. (See the next two plots.) We continue by choosing a new random start for the guided tour, then waiting until the data has found new territory.

The optimization settles on a projection where there are three strands visible, as seen in the leftmost plot in the second row. Two of the strands have been previously brushed, but a new one has appeared; this is painted yellow.

We also notice that there is another new strand hidden below the red strand. It's barely distinguishable from the red strand in this projection, but the two strands separate widely in other projections. Manual controls are helpful when we want to examine neighboring projections to distinguish the new strand from the red. It's tricky to brush it, because it isn't well separated in this projection. We use a trick: Hide the red points, brush the new strand green, and "unhide" the red points again (middle plot in the second row).

Five clusters have been easily identified; finding more clusters in this data is increasingly difficult. After several more alternations between the grand tour and the guided tour, we find something new (shown in the rightmost plot in the second row): One more strand has emerged, and we paint it pink.

The results at this stage are summarized by the bottom right plot. There is a very visible triangular component (in gray) and two color groups at each vertex. The next step is to clean up this solution, touching up the color groups by continuing to tour, and repainting a point here and there. When we finish, we have found seven clusters in this data that form a very strong geometric object in the data space: a 2-dimensional triangle, with two 1-dimensional strands extending in different directions from each vertex. To confirm our understanding of this object's shape, we can draw lines between some of the points and continue to tour (left two plots in the bottom row of Figure 5.3).

The next stage of cluster analysis is to characterize the nature of the clusters. To do that, we calculate summary statistics for each cluster, and plot them. When we plot the clusters of the particle physics data, we find that the

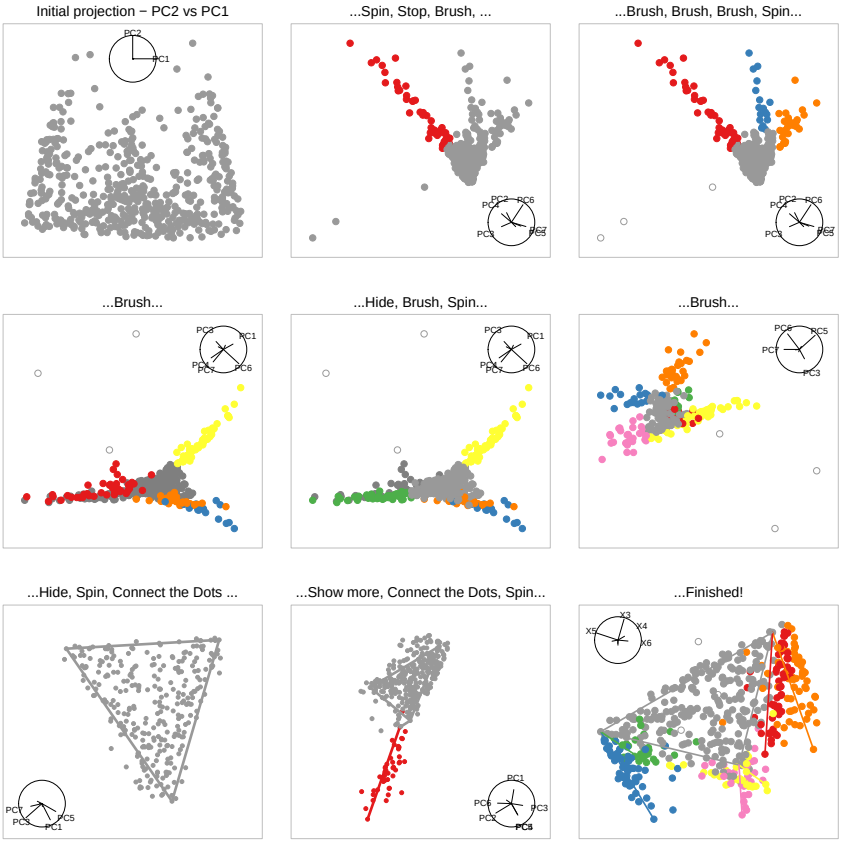

Fig. 5.3. Stages of spin and brush on PRIM7

2D triangle exists primarily in the plane defined by X3 and X5 (Figure 5.4). If you do the same, notice that the variance in measurements for the grey group is large in variables X3 and X5, but negligible in the other variables. The linear pieces can also be characterized by their distributions on each of the variables. With this example, we've shown that it is possible to uncover very unusual clustering in data without any domain knowledge.

Here are several tips about the spin-and-brush approach. Save the data set frequently during the exploration of a complex data set, being sure to save your colors and glyphs, because it may take several sessions to arrive at a final clustering. Manual controls are useful for refining the optimal projection because another projection in the neighborhood may be more revealing. The holes index is usually the most successful projection pursuit index for finding clusters. Principal component coordinates may provide a better starting point than the raw variables. Finally, the spin-and-brush method will not work

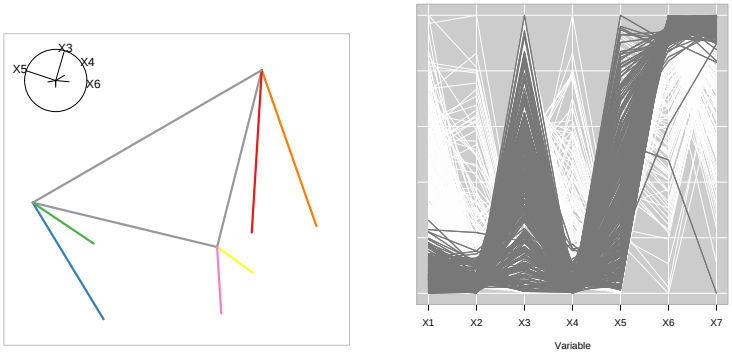

Fig. 5.4. (Left) Final model arrived at using line drawing in high-dimensions. (Right) Characterizing the discovered clusters using a parallel coordinate plot. The highlighted profiles (yellow) are the points in the 2D triangle.

well if there are no clear separations in the data, and the clusters are high-dimensional unlike the low-dimensional clusters found in this example.

5.3 Numerical methods

5.3.1 Hierarchical algorithms

Hierarchical cluster algorithms sequentially fuse neighboring points to form ever-larger clusters, starting from a full interpoint distance matrix. *Distance between clusters* is described by a "linkage method:" for example, single linkage uses the smallest interpoint distance between the members of a pair of clusters, complete linkage uses the maximum interpoint distance and average linkage uses the average of the interpoint distances. A good discussion on cluster analysis can be found in Johnson & Wichern (2002) or Everitt, Landau & Leese (2001).

Figure 5.5 contains several plots which illustrate the results of the hierarchical clustering of the particle physics data; we used euclidean interpoint distances and the average linkage method. The dendrogram at the top shows the result of the clustering process. Several large clusters were fused late in the process, with heights (indicated by the height of the horizontal segment connecting two clusters) well above those of the first joins; we will want to look at these. Two points were fused with the rest at the very last stages, which indicates that they are outliers and have been assigned to singleton clusters.

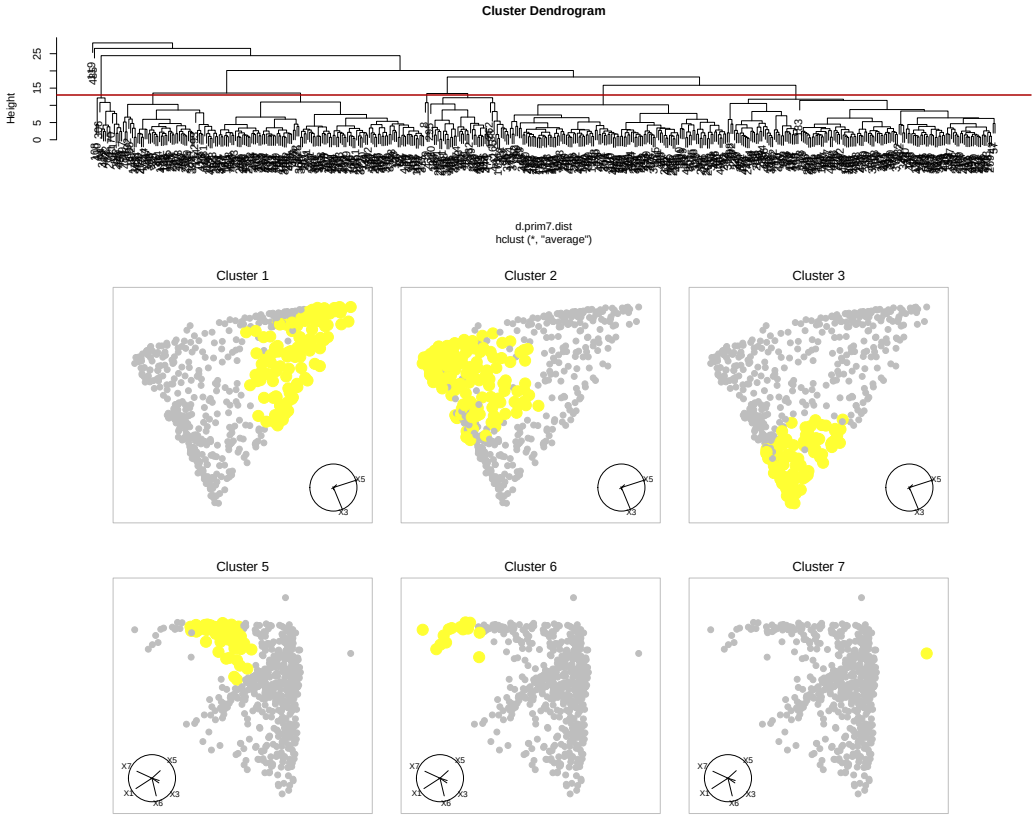

Fig. 5.5. Examining the results of hierarchical clustering using average linkage on the particle physics data using brushing from R linked to a tour in ggobi. (Top) Dendrogram describing the results, cut at 9 clusters. (Middle row) Clusters 1, 3 and 5 carve up the base triangle of the data. (Bottom row) Clusters 4 and 6 divide one of the arms, and cluster 7 is a singleton cluster.

We cut the dendrogram to produce nine clusters because we would expect to see seven clusters and a few outliers based on our observations from the spin-and-brush approach, and our choice looks reasonable given the structure of the dendrogram. (In practice, we would usually explore the clusters corresponding to several different cuts of the dendrogram.) We assign each cluster an integer identifier, and the leftmost plot just under the dendrogram is a plot of this cluster id against one of the original seven variables. In the subsequent plots, you see the results of highlighting one cluster at a time and then running the grand tour to focus on the placement of that cluster within the data. The plot in the upper right is an exception: it highlights both of the two singleton clusters at once, and they are indeed outliers relative to all the data.

The next three plots show, respectively, clusters 1, 2 and 3: these clusters roughly divide the main triangular section of the data into three. The plot at bottom right shows five of the clusters brushed in different colors.

The results are reasonably easy to interpret. Recall that the basic geometry underlying this data is that there is a 2D triangle with two linear strands extending from each vertex. The hierarchical average linkage clustering of the particle physics data using 9 clusters essentially divides the data into three chunks in the neighborhood of each vertex (clusters 1, 2, and 3), three pieces at the ends of the six linear strands (4, 6, and 7), and three clusters containing outliers (5, 8, and 9). This data is a big challenge for any cluster algorithm – low-dimensional pieces embedded in high-dimensional space – and we're not surprised that no algorithm that we have tried will extract the structure we found using interactive tools.

The particle physics data is extremely ill-suited to hierarchical clustering, but this extreme failure is an example of a common problem. When performing cluster analysis, we want to group the observations into clusters without knowing the distribution of the data. How many clusters are appropriate? What do the clusters look like? Could we just as confidently divide the data in several different ways and get very different but equally valid interpretations? Graphics can help us assess the results of a cluster analysis by helping us explore the distribution of the data and the characteristics of the clusters.

5.3.2 Model-based clustering

Model-based clustering (Fraley & Raftery 2002) fits a multivariate normal mixture model to the data. It uses the EM algorithm to fit the parameters for the mean, variance-covariance of each population and the mixing proportion. The variance-covariance matrix is re-parameterized using an eigen-decomposition

$$\Sigma_k = \lambda_k D_k A_k D_k', \quad k = 1, \ldots, g \ \text{ (number of clusters)}$$

resulting in several model choices, ranging from simple to complex:

Name	Σ_k	Distribution	Volume	Shape	Orientation
EII	λI	Spherical	equal	equal	NA
VII	$\lambda_k I$	Spherical	variable	equal	NA
EEI	$\lambda DD'$	Diagonal	equal	equal	NA
VEI	$\lambda_k DD'$	Diagonal	variable	equal	NA
VVI	$\lambda_k D_k D_k'$	Diagonal	variable	variable	NA
EEE	$\lambda DAD'$	Ellipsoidal	equal	equal	equal
EEV	$\lambda DA_k D'$	Ellipsoidal	equal	equal	variable
VEV	$\lambda D_k A D_k'$	Ellipsoidal	variable	equal	variable
VVV	$\lambda_k D_k A_k D_k'$	Ellipsoidal	variable	variable	variable

Note the distribution descriptions "spherical" and "ellipsoidal". This derives from the shape of the variance-covariance for a multivariate normal distribution. A standard multivariate normal distribution has a variance-covariance matrix with zeros in the off-diagonal elements, which corresponds to spherically-shaped data. When the variances (diagonals) are different or the variables are correlated then the shape of data from a multivariate normal is ellipsoidal.

The models are typically scored using the Bayes Information Criterion (BIC), which is based on the log likelihood, number of variables and number of mixture components. They should also be assessed using graphical methods, as we demonstrate using the Australian crabs data. To introduce the methods we first use just two of the five variables (frontal lobe and rear width) and only one species (blue). The goal is to determine whether model-based methods can discover clusters which will distinguish between the two sexes.

Figure 5.6 contains the plots we will use to examine the results of model-based clustering on this reduced data set. The top leftmost plot shows the data, with "M" indicating males, and "F" females. The two sexes correspond to long cigar-shaped objects which have some overlap, particularly for smaller crabs. The "cigars" aren't perfectly regular, either: the variance of the data is smaller at small values for both sexes, so that our cigars are somewhat wedge-shaped. In the models, the ellipse describing the variance-covariance is similar for each class, but oriented differently. With the heterogeneity in variance-covariance, this data doesn't strictly adhere to the multivariate normal mixture model underlying model-based methods, but we hope that the departure from regularity is not so extreme that it prevents the model from working.

The top right plot shows the BIC results for a full range of models, all variance-covariance parameterizations for 1 to 9 clusters. The best model (labeled H, for two clusters) used the EEV (equal volume, equal shape, different orientation) variance-covariance parameterization. This seems to be perfect! We can imagine that this result corresponds to two equally shaped ellipses that intersect near the lowest values of the data, and angle towards higher values.

We turn to the plots in the middle row to assess the model. (The points are plotted using their cluster id.) Surprise! All the small crabs, male and female, have been assigned to cluster 2. In the rightmost plot, we have added ellipses representing the estimated variance-covariances. The ellipses are the same shape, but the ellipse for cluster 1 is shifted towards the large values.

The next two best models, EEV-3 and VVV-2 (H for three clusters and J for two), have similar BIC values. The plots in the bottom row display representations of the variance-covariances for these models. EEV-3 organizes the crabs into two clusters of larger crabs and one cluster of small crabs. VVV-2 is similar to EEV-2.

What solution is the best for this data? If the EEV-3 model had done what we intuitively expected, it would have been ideal: the sexes of smaller

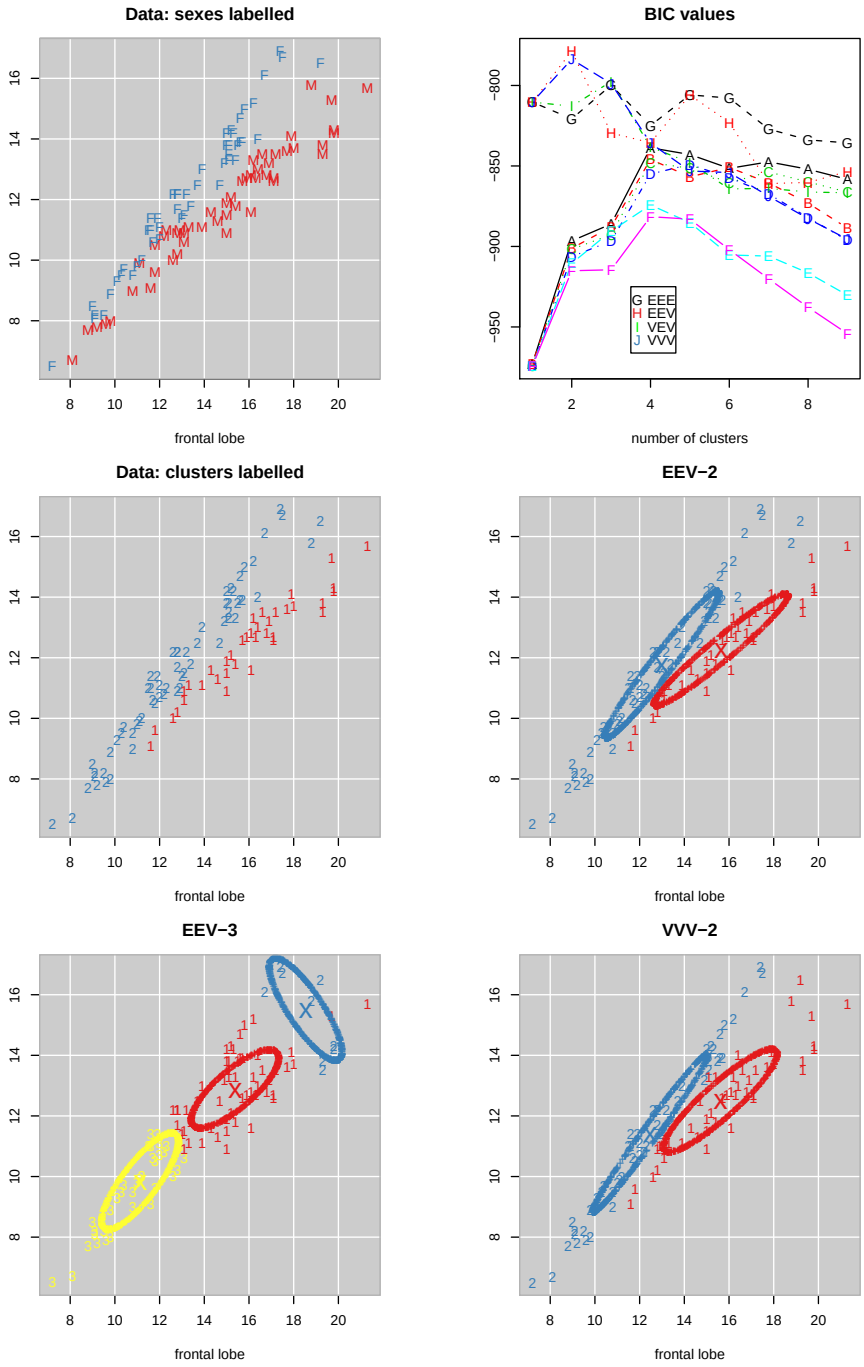

Fig. 5.6. Examining the results of model-based clustering on 2 variables and 1 species of the Australian crabs data: (Top left) Plot of the data with the two sexes labeled; (top right) Plot of the BIC values for the full range of models, where the best model (H) organizes the cases into two clusters using EEV parameterization; (middle left) The two clusters of the best model are labeled; Representation of the variance-covariance estimates of the three best models, EEV-2 (middle right) EEV-3 (bottom left) VVV-2 (bottom right).

crabs are indistinguishable, so they should be afforded their own cluster, while larger crabs could be clustered into males and females. In fact, model-based clustering didn't discover the true gender clusters. Still, it produced a useful and interpretable clustering of the crabs.

Plots are indispensable for choosing an appropriate cluster model. It's easy to visualize the models when there are only two variables, but increasingly difficult as the number of variables grows. Tour methods save us from producing page upon page of plots, They allow us to look at many orthogonal projections of the data, enabling us to conceptualize the shapes and relationships between clusters in more than two dimensions.

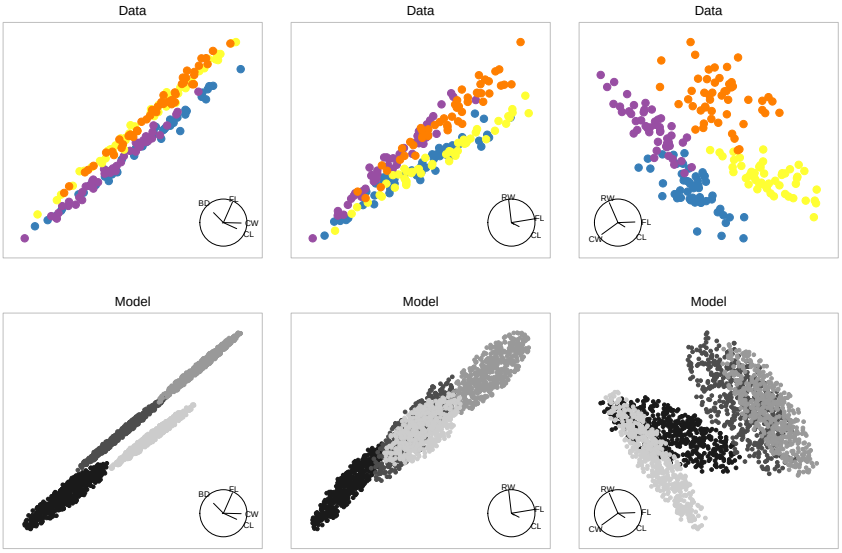

Fig. 5.7. Examining the results of model-based clustering on all 5 variables of the Australian crabs data. Tour projections of the 5D data (top row), and 5D ellipses corresponding to the variance-covariance in the four cluster model (bottom row). The variance-ellipses of the four clusters don't match the four known groups in the data.

Figure 5.7 displays the graphics for the corresponding high dimensional investigation using all five variables and four classes (two species, two sexes) of the Australian crabs. The cluster analysis is much more difficult now. Can model-based clustering uncover these four groups?

In the top row of plots, we display the raw data, before modeling. Each plot is a tour projection of the data, colored according to the four true classes. The blue and purple points are the male and female crabs of the blue species, and the yellow and orange points are the male and female crabs of the orange species.

	Male	Female
Blue Species	blue	purple
Orange Species	yellow	orange

The clusters corresponding to class are long thin wedges in 5D, with more separation and more variability at larger values, as we saw in the subset just discussed. The rightmost plot shows the "looking down the barrel" view of the wedges. At small values the points corresponding to the sexes are mixed (leftmost plot). The species are reasonably well separated even for small crabs (middle plot). The variance-covariance is wedge-shaped rather than elliptical, but again we hope that modeling based on the normal distribution which has elliptical variance-covariance will be adequate.

In the results from model-based clustering, there is very little difference in BIC value for variance-covariance models EEE, EEV, VEV, and VVV, with a number of clusters from 3 to 6. The best model is EEV-3, and EEV-4 is second best. We know that three clusters is insufficient to capture the four classes we have in mind, so we examine the four-cluster solution.

The bottom row of plots in Figure 5.7 illustrates the four-cluster model in three different projections. In each view, the ellipsoids representing the variance-covariance estimates for the four clusters are shown in four shades of grey, because none of these match any actual cluster in the data. Remember that these are two-dimensional projections of five-dimensional ellipsoids. The resulting clusters from the model don't match the true classes. The result roughly captures the two species (see the left plots where the species are separated in the actual data, as are the ellipses also). The grouping corresponding to sexes is completely missed (see the middle plots of both rows, where sexes are separated in the actual data but the ellipses are not separated). Just as in the smaller subset (two variables, one species) discussed earlier, there is a cluster for the smaller crabs of both species and sexes. The results of model-based clustering on the full five-dimensional data are very unsatisfactory.

In summary, plots of the data and parameter estimates for model-based cluster analysis are very useful for understanding the solution, and choosing an appropriate model. Tours are very helpful for examining the results in higher dimensions, for arbitrary numbers of variables.

5.3.3 Self-organizing maps

A self-organizing map (SOM) is constructed using a constrained k-means algorithm. A 1D or 2D net is stretched through the data. The knots in the net form the cluster means, and points closest to the knot are considered to belong to that cluster. The similarity of nodes (and their corresponding clusters) is defined as proportional to their distance from one another on the net.

We'll demonstrate SOM using the music data. The data has 62 cases, each one corresponding to a piece of music. For each piece there are seven variables: the artist, the type of music, and five characteristics, based on amplitude and

frequency, that were computed using the first forty seconds of the piece on CD. The music used included popular rock songs by Abba, the Beatles and Eels, classical compositions by Vivaldi, Mozart and Beethoven, and several new wave pieces by Enya. Figure 5.8 displays a typical view of the results of clustering using SOM on the music data. Each data point corresponds to a piece of music, and is labelled by the title of the piece or by a short code based on the composer's name: for example, SOS is an Abba song, and V1 is a Vivaldi composition.

A SOM is commonly assessed with a 2D map view, like that the left plot in Figure 5.8. Here we have used a 6×6 net pulled through the 5D data. The net that was wrapped through the high-dimensional space is straightened out and laid flat, and the points, like fish in a fishing net, are laid out where they have been trapped. In the plot shown here, the points have been jittered slightly, away from the knots of the net, so that the labels don't overlap too much. If the fit is good, the points that are close together in this 2D map view are close together in the high-dimensional data space, and also close to the net as it was placed in the high-dimensional space.

Much of the structure in the map is no surprise: The rock and classical tracks are on opposing corners, with rock in the upper right and classical in the lower left. The Abba tracks are all grouped at the top and left of the map. Beatles and Eels tracks are mixed. There are also some unexpected assocations: for example, the Beatles song Hey Jude is mixed amongst the classical compositions!

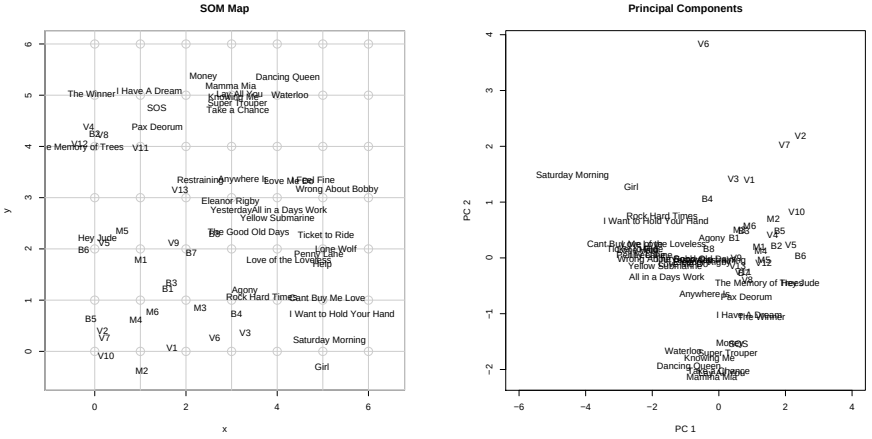

Fig. 5.8. (Left plot) Typical view of the results of clustering using self-organizing maps. Here the music data is shown for a 6×6 map. Some jittering is used to spread tracks clustered together at a node. (Right plot) First two principal components.

Construction a self-organizing map is a dimension reduction method, akin to multidimensional scaling (Borh & Groenen 2005) or principal components analysis (Johnson & Wichern 2002). Using principal component analysis to find a low-dimensional approximation of the similarity between music pieces, we find the right-side plot in Figure 5.8. There are many differences between the two representations. The SOM has a more even spread of music pieces across the grid, contrary to the stronger clumping of points in the PCA view. In contrast, the PCA view has several outliers, such as V6, which could lead us to learn things about the data that we might miss by relying exclusively on the SOM.

Although the reduced dimension view is the common way to graphically assess the SOM results, it is woefully limited. What might appear to be an appealing result from the map view may indeed be a poor fit in the data space. Dimension reduction plots need to be associated with ways to assess their accuracy. PCA suggests a contradictory view, suggesting that the data is clumped with several outliers. Which method yields the more accurate picture of the data structure, SOM or PCA? We can use the grand tour to help us find an answer to that question.

We will use a grand tour to view the net wrapped in amongst the data, hoping to learn how the net converged to this solution, and how it wrapped through the data space. Actually, it is rather tricky to fit a SOM: Like many algorithms, it has a number of parameters and initialization conditions that affect the outcome.

Figure 5.9 shows two different states of the fitting process, and of the SOM net cast through the data. In both fits, a 6×6 grid is used and the net is initialized in the direction of the first two principal components. The top row shows the results of our first SOM fit, which was obtained using the default settings; it gave terrible results. At the left is the map view, in which the fit looks quite reasonable. The points are spread evenly through the grid, with rock tracks (orange) at the upper right, classical tracks (orange) at the lower left, and new wave tracks (purple) in between. The tour view, at the right, shows the fit to be inadequate. The net is a flat rectangle in the 5D space, and has not sufficiently wrapped through the data. This is the result of stopping the algorithm too soon, thus failing to let it converge fully.

The middle and bottom row of plots show our favorite fit to the data. The data was standardized, we used a 6x6 net, and we ran the SOM algorithm for 1000 iterations. The map is in middle left, and it matches the map already shown in Figure 5.8, except for the small jittering of points. The other three plots show different projections from the grand tour. The middle left plot shows how the net curves with the nonlinear dependency in the data. In the middle right plot we see that the net is warped in some directions to fit the variance pattern. At bottom left we see that one side of the net collects a long separated cluster of tracks that correspond to the Abba tracks. We can also see that the net hasn't been stretched out to the full extent of the range of the

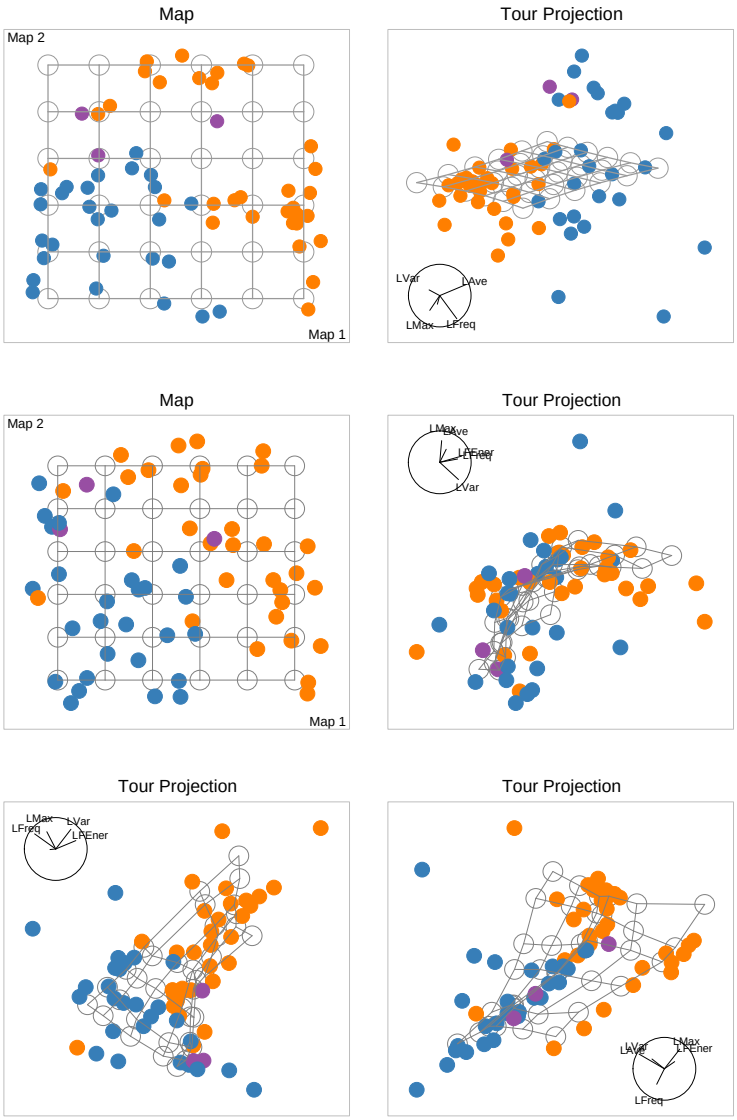

Fig. 5.9. The map view along with the map rendered in the 5D space of the music data. (Top row) The SOM fit is problematic. Although the fit looks quite good from the map view in the data space it is clear that the net has not sufficiently wrapped into the data: the algorithm has not converged fully. (Middle and bottom rows) SOM fitted to standardized data, shown in the 5D data space and the map view. The net wraps through the nonlinear dependencies in the data. It doesn't seem to be stretched out to the full extent of the data, and there are some outliers which are not fit well by the net.

data. It is tempting to manually manipulate the net to stretch it in different directions and update the fit.

It turns out that the PCA view of the data more accurately reflects the structure in the data than the map view. The music pieces really are clumped together in the 5D space, and there are a few outliers.

5.3.4 Comparing methods

To compare the results of two methods we commonly compute a confusion table. For example, Table 5.3.4 is the confusion table for five-cluster solutions for the music data from k-means and Ward's linkage hierarchical clustering. The numerical labels of clusters are arbitrary, so these can be rearranged to better digest the table (right table). There is a lot of agreement between the two methods: the two methods agree on the cluster for 48 tracks out of 62, or 77% of the time. We want to explore the data space to see where the agreement occurs, and where the two methods disagree.

	Wards				
k-means	1	2	3	4	5
1	0	0	3	0	14
2	0	0	1	0	0
3	0	9	5	0	0
4	8	2	1	0	0
5	0	0	3	16	0

Rearrange rows ⇒

	Wards				
k-means	1	2	3	4	5
4	8	2	1	0	0
3	0	9	5	0	0
2	0	0	1	0	0
5	0	0	3	16	0
1	0	0	3	0	14

Figure 5.10 illustrates linking a confusion table for the two clustering methods with plots of the data. The plots in the left column show the confusion table, with jittering used to separate the points in each category combination. The plots in the right column show the data in tour projections. In the top row of plots a cluster of 14 tracks that both methods agree on is brushed in red. Identifying the tracks in this cluster, we learn that it consists of a mix of tracks by the Beatles (Penny Lane, Help, Yellow Submarine, ..) and the Eels (Saturday Morning, Love of the Loveless, ...). From the plot at the right, we see that this cluster is a closely grouped set of points in the data space and they are characterized by high values on LVar (variable 3 in the data); that is, they have large variance in frequency.

In the bottom row of plots, another group of tracks that were clustered together by both methods have been brushed in red. Identifying these eight tracks, we see that they are all Abba songs (Dancing Queen, Waterloo, Mamma Mia, ...). In the plot to the right, we see that this cluster is closely grouped in the data space. Despite that, this cluster is a bit more difficult to characterize. It is oriented mostly in the negative direction of LAve (variable

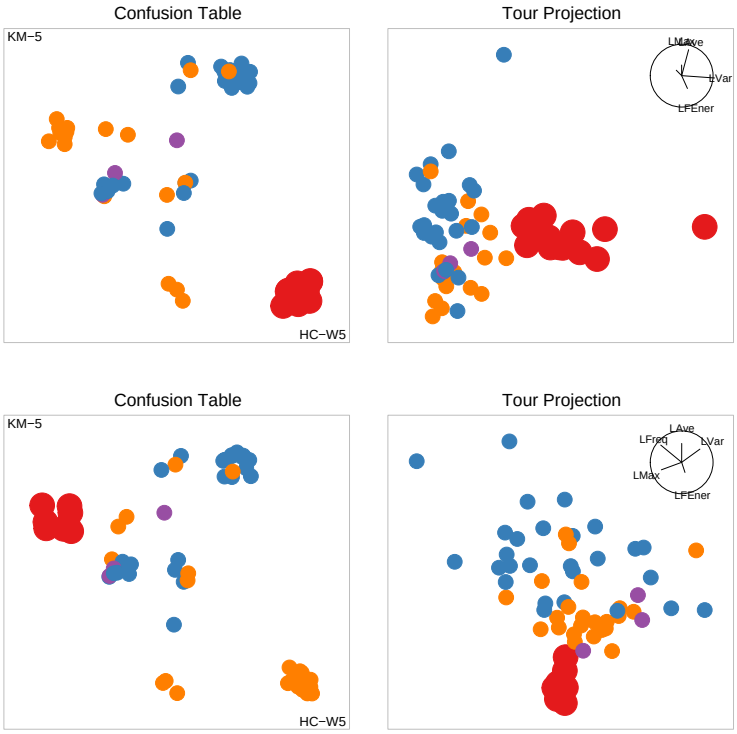

Fig. 5.10. Comparing the five cluster solutions of k-means and Wards linkage hierarchical clustering of the music data. (Left plots) Jittered display of the confusion table with areas of agreement brushed red. (Right plots) Tour projections showing the tightness of each cluster where there is agreement between the methods.

4), so it would have smaller values on this variable. But this vertical direction in the plot also has large contributions from variables 3 (LVar) and 7 (LFreq) also. In similar fashion we could explore the tracks where the methods disagree.

5.4 Recap

Graphics are invaluable during cluster analysis. The spin-and-brush approach can be used to get a gestalt of clustering in the data space. Scatterplots and parallel coordinate plots, in conjunction with a dendrogram, can help us to understand the results of hierarchical algorithms. In model-based cluster analysis we can examine the clusters and the model estimates to understand the solution. For self-organizing maps the tour can assist uncovering problems with the fit, such as when the map wraps in on itself through the data making

it appear that some cases are far apart when they are truly close together. A confusion table can come alive with linked brushing, so that mismatches and agreements between methods can be explored.

5.5 Exercises

1. Using the spin-and-brush method uncover three clusters in the flea data and confirm that these correspond to the three species. (Hint: It helps to transform the data to principal components and enter these variables into the projection pursuit guided tour running the holes index.)
2. Run hierarchical clustering with average linkage on the flea beetle data (excluding the species variable).
   a) Cut the tree at 3 clusters and append a cluster id to the flea data set. How well do the clusters correspond to the species? (Plot cluster id vs species, and use jittering if necessary.) Using brushing in a plot of the cluster id, linked to a tour plot of the six variables examine the beetles that are misclassified.
   b) Now cut the tree at 4 clusters, and repeat the last part.
   c) Which is the better solution, 3 or 4 clusters? Why?
3. This question uses the olive oils data.
   a) Consider the oils from the four areas of Southern Italy. What would you expect to be the results of model-based clustering on the eight fatty acid variables?
   b) Run model-based clustering on the southern oils, with the goal being to extract clusters corresponding to the four areas. What is the best model? Create ellipsoids corresponding to the model and examine these in a tour. Does it match your expectations?
   c) Create ellipsoids corresponding to alternative models and use these to decide on a best solution.
4. This question uses the rat gene expression data.
   a) Explore the patterns in expression level for the functional classes. Can you characterize the expression patterns for each class?
   b) How well do the cluster analysis results match the functional classes? Where do they differ?
   c) Could you use the cluster analysis results to refine the classification of genes into functional classes? How would you do this?
5. In the music data, do more comparisons between the five cluster solutions of k-means and Wards hierarchical clustering.
   a) On what tracks do the methods disagree?
   b) Which track does k-means consider to be a singleton cluster and yet Wards hierarchical clustering group with 12 other tracks?
   c) Identify and characterize the tracks in the four clusters where both methods agree.

6. In the music data, fit a 5×5 grid SOM, and observe the results for 100, 200, 500, 1000 updates. How does the net change with the increasing number of updates?
7. There is a mystery data set in the collection, called `clusters-unknown.csv`. How many clusters in this data?

Chapter 8
Longitudinal Data

8.1 Background

In longitudinal (panel) data individuals are repeatedly measured through time which enables the direct study of change (Diggle, Heagerty, Liang & Zeger 2002). Each individual will have certain special characteristics, and measurements on several topics or variables may be taken each time an individual is measured. The reporting times can vary from individual to individual in number, dates and time between reporting. This deviation from equi-spaced, equal quantity time points, producing a ragged time indexing of the data, is common in longitudinal studies and it causes grief for many data analysts. It may be difficult to develop formal models to summarize trends and covariance, yet there may be rich information in the data. There is a need for methods to tease information out of this type of complex data (Singer & Willett 2003). Most documented analyses discuss equi-spaced, equal quantity longitudinal measurement, but ragged time indexed data is probably more common than the literature would have us believe. This paper discusses exploratory methods for difficult to model ragged time indexed longitudinal data.

The basic question addressed by longitudinal studies is how the responses vary through time, in relation to the covariates. Unique to longitudinal studies is the ability to study individual responses. This is different from repeated cross-sectional studies which take different samples at each measurement time, to measure the societal trends but not individual experiences. Longitudinal studies are similar to time series except that there are multiple time series, one for each individual. Software for time series can deal with one time series or even a couple, but the analysis of hundreds of them is not easily possible. The analysis of repeated measures could be considered to be a subset of longitudinal data analysis where the time points are equal in number and spacing (Crowder & Hand 1990).

Analysts want to explore many different aspects of longitudinal data - the distribution of values, temporal trends, anomalies, the relationship between multiple responses and covariates in relation to time. Exploration, which reveals the unexpected in data and is driven by rapidly changing questions, means it is imperative

to have graphical software which is interactive and dynamic: software that responds in real time to an analyst's enquiries and changes displays dynamically, depending on the analyst's questions. Plots provide insight into multiple aspects of the data, overviews of the general behavior and tracking individuals. Analysts may also want to link recorded events, such as a graduation or job loss, to an individuals' behavior. Even with unequal time points the values for each individual can be plotted, for example, each variable against time, variable against variable with measurements for each individual connected with line segments. Linking between plots, using direct manipulation, enables the analyst to explore relationships between responses and covariates (Swayne & Klinke 1998, Unwin, Hofmann & Wilhelm 2002). Dynamic graphics such as tours (Asimov 1985) will enable the study of multivariate responses.

There is very little in the literature discussing graphical methods for longitudinal data. Both Diggle et al. (2002) and Singer & Willett (2003) state there is a need for graphics but have only brief chapters describing static graphics. Koschat & Swayne (1996) illustrated the use of direct manipulation for customer panel data. They applied tools such as case identification, linking multiple views and brushing on scatterplots, dot plots and clustering trees, and a plot they called the case-profile plot (time series plot of a specific subject). Case-profile plots are also known as parallel coordinates(Inselberg 1985, Wegman 1990), interaction plots, or profile plots in the repeated measures and ANOVA literature. Koschat and Swayne recommended looking at different views of the same data. Sutherland, Rossini, Lumley, Lewin-Koh, Dickerson, Cox & Cook (2000) demonstrate viewing multiple responses in relation to the time context using a tour. Faraway (1999) introduced what he called a graphical method for exploring the mean structure in longitudinal data. His approach fits a regression model and uses graphical displays of the coefficients as a function of time. Thus the method describes graphics for plotting model diagnostics but not the data: graphical method is an inaccurate title.

Longitudinal data analysis, like other statistical methods, has two components which operate side-by-side: exploratory and confirmatory analysis. Exploratory analysis is detective work, comprising of techniques to uncover patterns in data. Confirmatory analysis is like judicial work, weighting evidence in data for, or against hypotheses (Diggle et al. 2002). This chapter concentrates on exploratory data analysis.

8.2 Notation

We denote the *response* variables to be $\mathbf{Y}_{ijt_i}$, and the time-dependent explanatory variables, or *covariates* to be $\mathbf{X}_{ikt_i}$, where $i = 1, \ldots, n$ indexes the number of individuals in the study, $j = 1, \ldots, q$ indexes the number of response variables, $k = 1, \ldots, p$ indexes the number of covariates, and $t_i = 1, \ldots, n_i$ indexes the number of times individual i was measured. Note that n is the number of subjects or individuals in the study, n_i is the number of time points measured for individual i, q is the number of response variables, and p is the number of explanatory variables, measured for each individual and time. The explanatory variables may include

indicator variables marking special events affecting an individual. There may also be time-independent explanatory variables or covariates, which we will denote as $\mathbf{Z}_{il}$, $i = 1, \ldots, n$, $l = 1, \ldots, r$. Simplifications to the notation can be made when the data is more constrained, such as equi-distant, equal number of time points.

8.3 More Background

The immediate impulse is to plot Y_{ij} against t_i, with values for each individual connected by line segments. Figure 8.1, left plot, shows the profiles for the wages data. These plots can be very messy, and practically useless (Diggle et al. 2002).

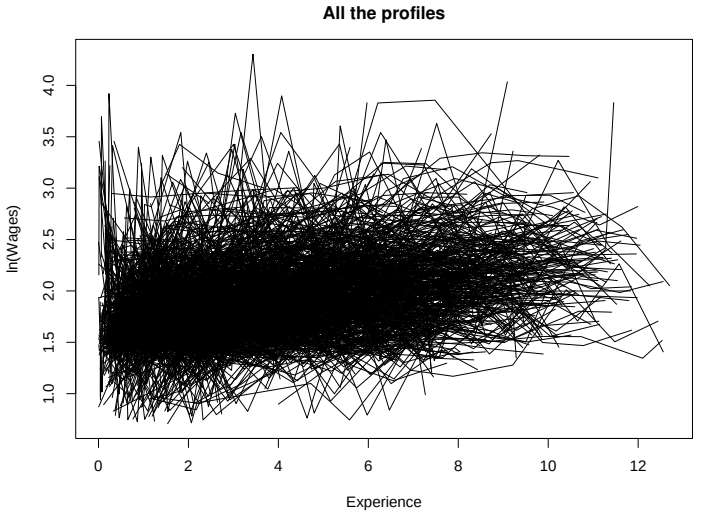

Figure 8.1. *The plot of all 888 individual profiles. Can you see anyhting in this plot? With so much overplotting the plot is rendered unintelligible. Note that a value of* $ln(Wage) = 1.5$ *converts to* $exp(1.5) =$*\$4.48.*

To alleviate overplotting Diggle et al. (2002) suggest plotting a sample, or several samples, of the individuals. The plot at right in Figure 8.2 shows a sample of 50 individuals. Not a lot more can be seen from the sample of 50 profiles. There's considerable variability from individual to individual. There looks to be a slight upward trend.

Another common approach is to animate over all the individuals. We show the first few individuals here as separate plots because we cannot demonstrate an animation (Figure 8.3). There are another 879 profiles to look at. Are you willing to look at this many? Watching an animation of 888 profiles is going to be dizzying rather than insightful. We're in the situation at this stage where we want to look at the data because we don't know much about it. Its messy enough data that we cannot learn much from static plots. And to conduct meaningful analyses we need

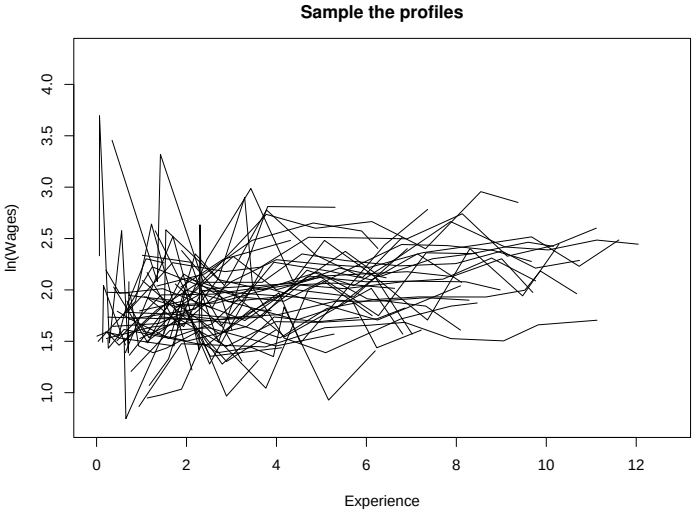

Figure 8.2. *A sample of 50 individual profiles. A little more can be seen in the thinned plot: there is a lot of variability from individual to individual, and there seems to be a slight upward trend.*

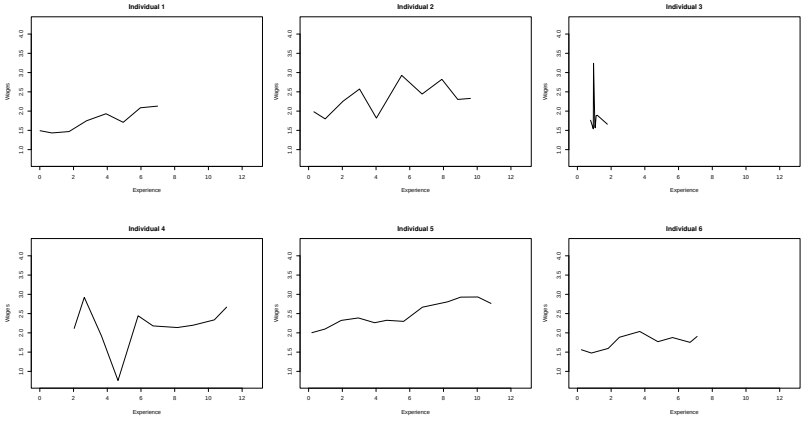

Figure 8.3. *Profiles of the first six individuals. We can make several interesting observations here: Individual 3 has had a short volatile wage history, perhaps due to hourly jobs? But can you imagine looking at 888, a hundred-fold more than the few here? Sometimes an animation is generated that consecutively shows profiles from individual 1 to n. Its simply not possible to learn much by animating 888 profiles, especially that has not natural ordering.*

to know more about whats in the data. An animation would be more digestible if we organize the individuals into similar groups or some informative order, but at this stage we don't know enough about the data to organize it.

8.4 Mean Trends

The primary question is how responses vary with time. To assess the trend with time requires some estimate of the trend to be plotted, along with enough information about the distribution of values to assess the strength of the trend. For ragged time indexed data we use a smoother to estimate the trend, otherwise we calculate the median or mean at each time. Plots and calculations are conditioned by time-independent categorical covariates. For non-ragged time indexed data we can also condition on time.

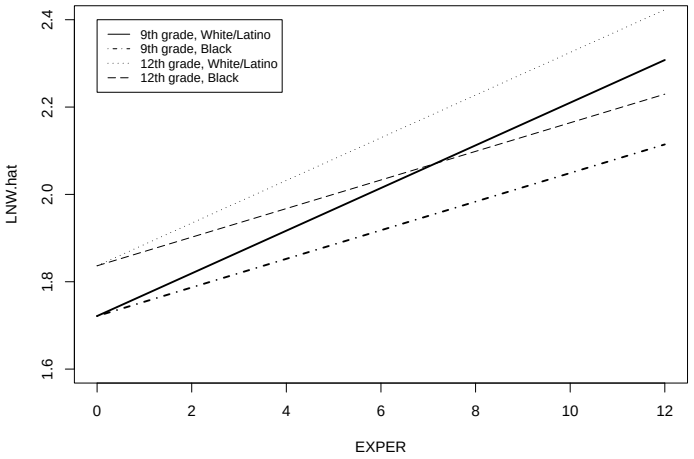

Figure 8.4. *Model for ln wages based on experience, race and highest grade achieved.*

With ragged time data it is difficult to build models for the trend. The data may be processed into regularized time, but we'd like to avoid pre-processing the data this much. Singer & Willett (2003) fit mixed linear models to the wages data. Figure 8.4 shows their model for wages based on experience, race and highest grade achieved. The model says that on average, the starting wage is higher for men achieving 12th grade as opposed to 9th grade education, but that with more experience, even with though they have achieved a higher grade, blacks experience lower wages. Controlled for highest grade achieved the rate of change for whites and hispanics is 5.0%, but for blacks it is a 3.3%. It should also be noted that the

variance components are also significant in this model, suggesting that the variance from individual to individual is substantial. This is a part of the data that we will want to explore in more detail.

Lets take a look at this data, and assess how well this model fits the trend. Figure 8.5 (top left) displays the lowess smooth (Cleveland 1979) of the wages value in relation to experience, overlaid on a scatterplot of the data values (Y_{ij}, t_i). The trend according to the smoothed line matches the model reasonably well. The purpose of showing the scatterplot underneath the curve is to assess the variation around the mean trend. The variation in this data is huge. The mean trend is quite strong, but its also clear that the variation in wages is quite large across the full range of experience. We use very small glyphs, single pixels because there are a lot of points, and a lot of ink.

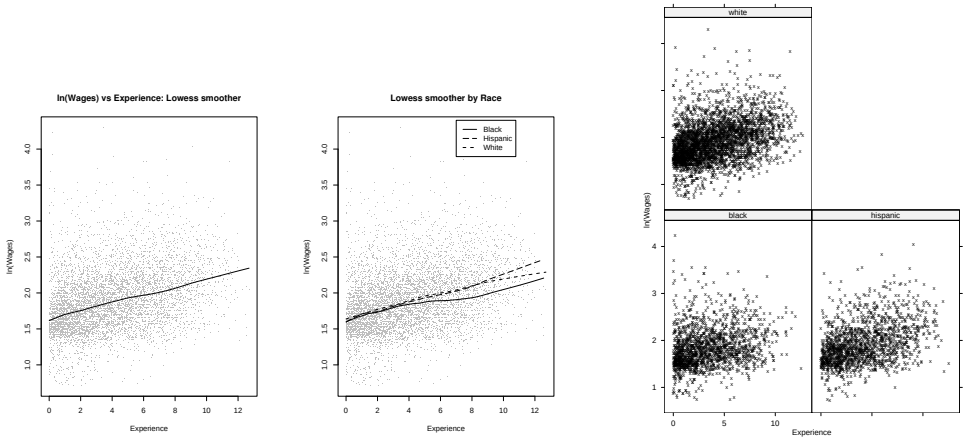

Figure 8.5. *Mean trends using lowess smoother: (Left) Overall wages increase with experience. (Middle) Race makes a difference, as more experience is gained. (Right) The scatter plot of wages against experience conditioned on race. The pattern is different for the different races, in that whites and Hispanics appear to have a more positive linear dependence than blacks, and there are less blacks with the longest experiences. This latter fact could be a major reason for the trend difference.*

Figure 8.5 (middle plot) display the lowess smoothed lines of wages based on experience conditionally on race. There appears to be a difference in the wages for men with more workforce experience according to race: people who are black have somewhat lower wages on average than Hispanic and other races when they have similarly high levels of experience. This differs from the Singer and Willett model: the difference appears to be not linear. For blacks the wages plateau out around 5-7 years of experience and then increase again. The trellis scatterplots at right, where ln(Wage) is plotted against experience conditionally on race, also show a dramatic difference. Whites and Hispanics have a clearly positive linear association between wages and experience, but the relationship is not positive linear for blacks.

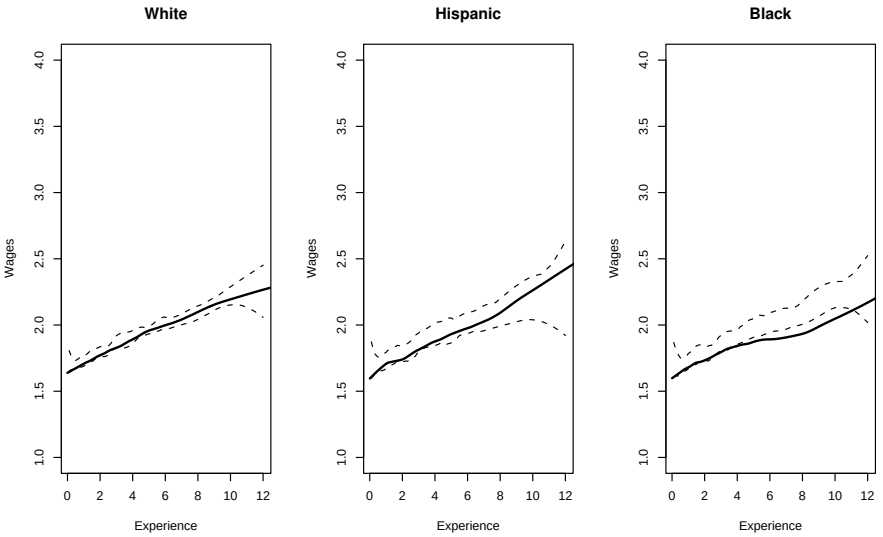

Figure 8.6. *Reference bands (dashed lines) for the smoothed curves for race, computed by permuting the race labels 100 times and recording the lowest and highest observed values at each experience value. The most important feature is that the smoothed curve for the true black label (solid line) is outside the reference region, around the middle experience values. This suggests this feature is really there in the data. Its also important to point out that the large difference between the races at the higher values of experience is not borne out to be real. The reference band is larger in this region of experience and all smoothed curves lie within the band, which says that there is difference between the curves could occur randomly. This is probably due to the few sample points at the longer workforce experiences.*

But there are also fewer blacks with 9 or more years of experience than whites and hispanics, which makes the results at the upper end of experience less reliable. How do we assess the significance of these observations? We'll use ideas similar to those discussed in Bowman & Wright (2000), and used in Prvan & Bowman (2003), and similar to the ideas described in the chapter on inference in this book. We will generate reference regions by permuting the race labels of the individuals. Take the column of race labels for 888 men in the study, and shuffle the values in the column. Associate these new (meaningless) labels to the full time profile of the individual. Compute the smoothed curve for each group, and evaluate this on a fine scale on the experience variable. Repeat this many times. We repeated it 100 times to get Figure 8.6. Record the minimum and maximum value observed for each value of the experience variable. This provides reference bands for the minimum

and maxmimum we'd expect if the race labels were irrelevant.

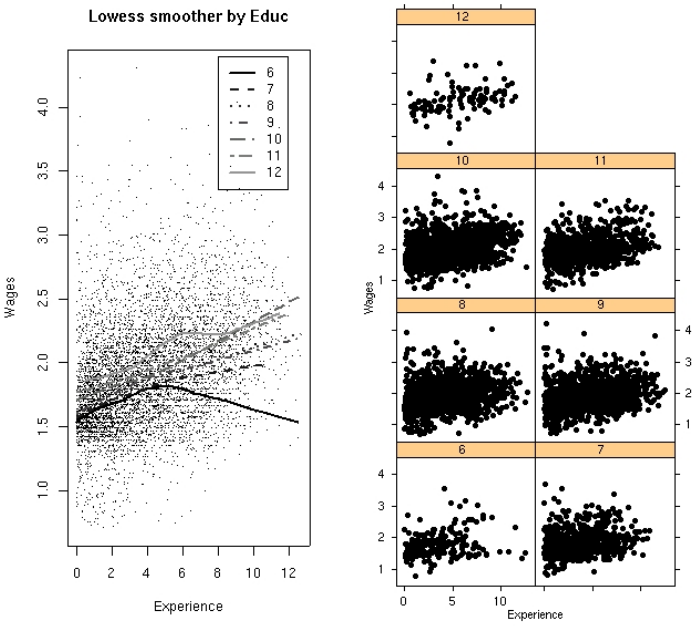

Figure 8.7. *(Left) Mean trends using lowess smoother conditioned on last year of school. (Right) The scatter plot of wages against experience conditioned on last year of school.*

Figure 8.7 shows lowess smoothed lines of wages based on experience conditionally on education. For education, there is some difference in average wages when there is little experience and the gap widens with more experience. In general, more education means higher wages, especially with more experience. The interesting contradiction is for individuals with the least education (6 years) is that with more experience the wages drop dramatically. A slightly similar pattern can be seen with people with the most education (12 years). These trends are suspicious, and really can probably be explained by lack of data. The bottom right shows the wages vs experience plot conditioned on the education, and it can be seen that there are not too many people in the 6 and 12 years categories of education. An interesting observation is that with earlier dropout there are fewer men at the longer times of workforce experience.

Some notes: Several generalizations emerge from initial exploration of longitudinal data:

- The primary intention is to understand the relationship between responses and the temporal context. Thus the basic plot is response(s) against time.
- Plotting all the individual traces on the one plot can produce an unreadable

plot. The purpose is to digest the mean trend, so that in general, it may be more useful to plot the points only.

- Along with a representation of the mean trend, a representation of the variation is important. A scatter plot of the points overlaid by the trend representation is the simplest approach to assessing the variation around the trend that works for all types of longitudinal data. In some constrained types of longitudinal data it is possible to use boxplots to display the distribution or display confidence intervals at common time points.
- Use conditional plots for assessing the trend in relation to categorical covariates, or common time points.
- Plots of model estimates are no substitute for plots of data.
- We've intentionally used a longer vertical than horizontal axis in these plots, which may seem strange at first. Generally for time series it is recommended that the horizontal axis is longer than the vertical axis (Cleveland 1993), which is appropriate when examining periodicity in time series. When the interest is focused on the overall trend, though, it is easier to assess with a longer vertical axis.

8.5 Individuals

The ability to study the individual is a defining characteristic of longitudinal data analysis. With a tangle of overplotted profiles this can be a daunting task. There are two approaches in common use: (1) sample the individuals to reduce the number of lines plotted (Diggle et al. 2002), and (2) show one individual at a time, animating over all individuals. Neither of these provide satisfying insights into individual patterns. With sampling there can be too much missing to find the interesting individuals. To make a successful animation there needs to be continuity from frame to frame, and the order in which individuals appear in a data set is unlikely to be ordered in a way that will produce continuity. Animations over individuals invariably produce quick flashes of radically differing profiles from frame to frame, allowing little chance for digesting any patterns. This section describes some alternative approaches to studying individuals.

8.5.1 Example 1: Wages

The purpose of studying the individual profiles is that we want to get a sense for the individual wage vs experience pattern as it differs from a common trend. On average more experience means more wages but is it usual that as an individual gets more experience that their wages will go up. How common is this? Or is it more typical that a persons wages will bounce around regardless of experience? Figure 8.8 displays profiles of individuals who are at the extremes in the data to some extent. We take a brush and select an observation on the extremes of the wages

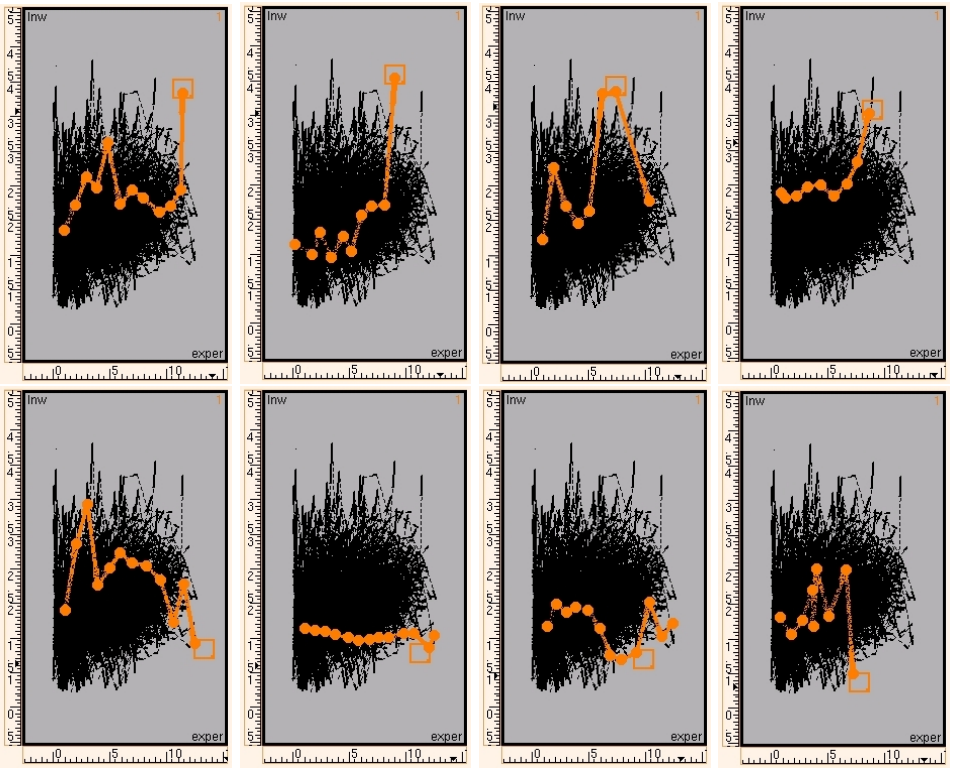

Figure 8.8. *Extreme values in wages and experience are highlighted revealing several interesting individual profiles: large jumps and dips late in experience, early peaks and then drops, constant wages.*

and experience plot, and their record is highlighted. We can observe quite a range in the individual differences. Two people (top two at left) with extremely high wages with long experience both received substantial late jumps in wages. The first person had a quick rise in wages in their early experience, and then a sharp drop, oscillated around an average wage for several years and then jumped substantially at 12 years experience. Another person (third plot) with high wage, at 7-8 years of experience, took a dramatic drop in wages in the later years of experience. The people with low wages later in experience also had quite dramatically different patterns. One person (fifth plot) began their career with relatively high wages early on, and their wages have continued to drop. Another person (sixth plot) has consistently earned low wages despite more experience. The individual wage vs experience pattern is really quite varied!

Figure 8.9 shows several typical cases we might wish to explore. *How do people with high wages early fare with more experience? How do people with lots of experience and high wages get there?* If a person starts off with a high wage how do they fare with more experience. The top early wage earners are highlighted in

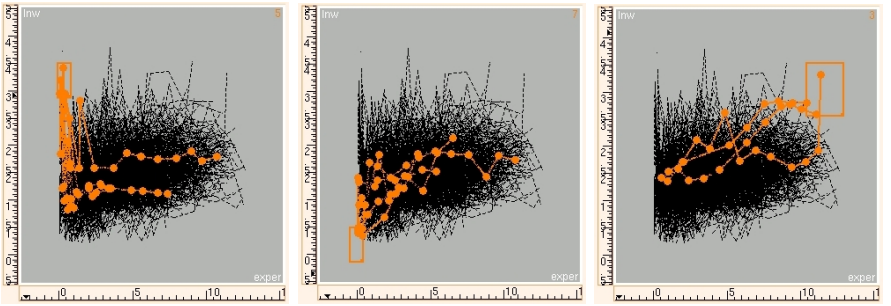

Figure 8.9. *Early high/low earners, late high earners with experience.*

the left plot, and only two of these people are retained for the full study, and their wages end up just at moderate levels. The middle plot highlights individuals who's wages started off very low. Again only two of these people were retained for the full study and their incomes did increase to be moderately high with more experience. The third plot highlights several individuals with high wages and more experience. Its interesting to see how they got there. All three started off with moderate wages. Two steadily increased their wages and the third person had a quite volatile wage history.

Searching for Special Trends: With what we've seen about the individual variability, a next stage is to search for particular types of patterns: the individuals that have the most volatility in their wages, the individuals who's wages steadily increase or decrease. We have created several new variables to measure the overall variability in wages for each individual, and the variance in the differences in wages for each individual, to extract the individuals with smoother transitions. These two new variables are called `SDWages` and `UpWages` respectively. When these two variables are incorporated into the analysis, they help identify the complex tapestry among wage history of these respondents. Figure 8.10 shows a few. The first plot shows a person who has had an extremely volatile wage history. The second plot shows two high earning people that have had dramatic increases in wages as they have gained experience. The third plot shows three people with more steady increases in wages as they have become more experienced. The fourth plot shows many individuals who have had very little change in their wages with increasing experience. The fifth plot shows a person who has had their wage steadily decline despite increasing experience. The sixth plot shows an individual who's had some dramatic changes in wages with increasing experience, some volatility early in their career, and then a period of high earning with moderate experience to be declining in wages with more experience.

So what have we learned about wages and experience? A brief summary of what we have learned about the general patterns and individual variation from this data is:

- On average wages increase with experience, but there is a lot of variation in wages depending on experience.

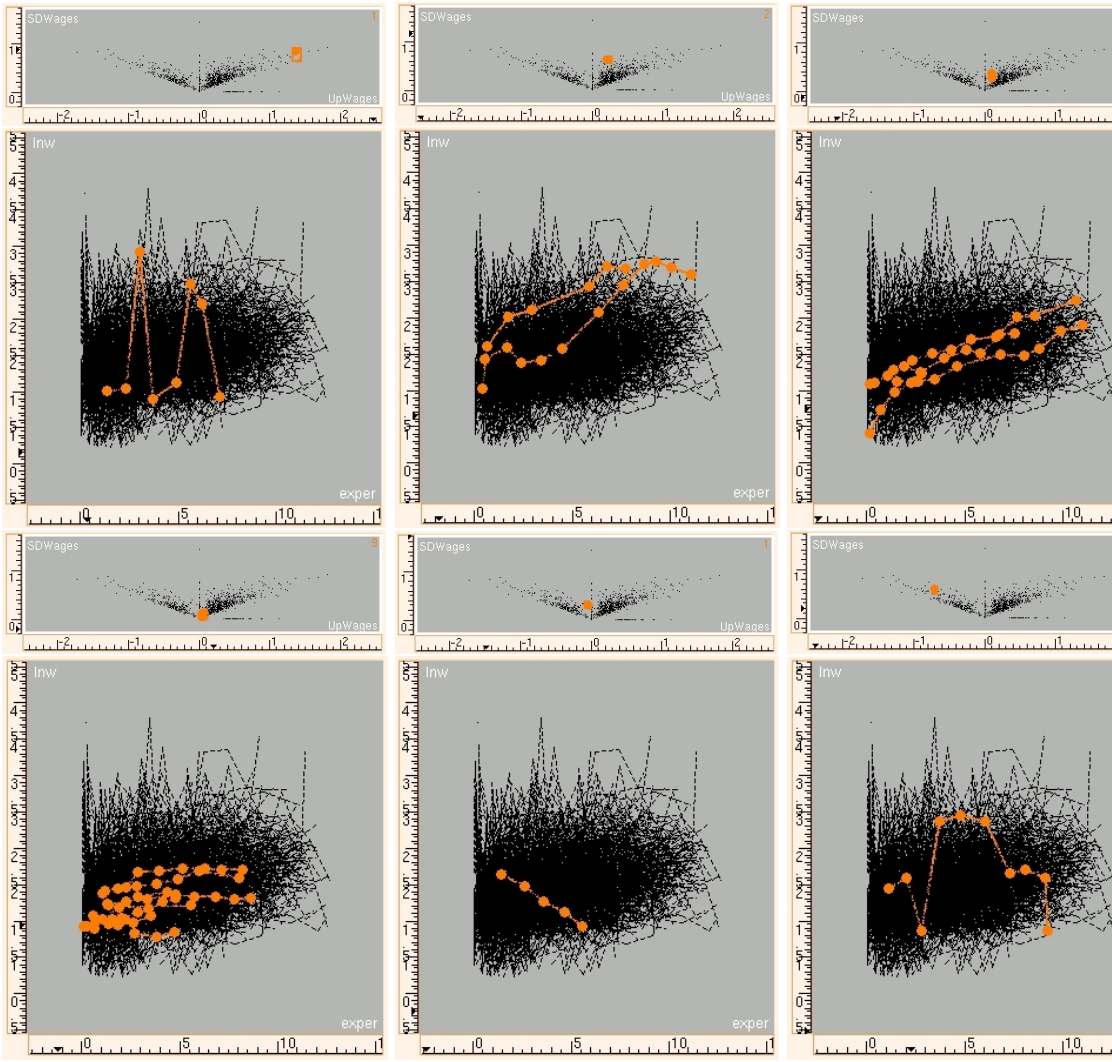

Figure 8.10. *Special patterns: with some quick calculations to create indicators for particular types of structure we can find individuals with volatile wage histories and those with steady increases or declines in wages.*

- The amount of increase differs according to race and educational experience in the later years of experience.
- The individual patterns are dramatically different. We found several individuals have extremely volatile wages in relation to experience, several who have very constant wages despite more experience, and several people who saw a decline in their wages as they gained more experience in the workforce.

8.6 Exercises

1. This question uses the panel study of income dynamics data.

   (a) How does income vary over the years? Assess the trend in income over time using histograms for each year of data. (Note that income is logged base e.)

   (b) Calculate the median ln(Income) for each year. Make a scatterplot of ln(Income) vs Year, draw the median trend using line segments connecting the medians for each year. Describe the temporal trend of income.

   (c) Calculate the median ln(Income) for each gender for each year, and draw these two trend lines on a scatterplot of the data. Also generate trellis plots of ln(Income) vs Year conditioned on gender. Is there a difference between male and female incomes over the study period?

   (d) Examine the relationship between ln(Income) and year for the interaction between gender and education, using plots similar to the previous questions. Is there an interaction effect on ln(Income) between gender and education?

   (e) Using linked brushing explore the extreme individuals. Who is the person with an extremely low income in the early nineties (male or female, high school or college educated)? Who is the top earning person? Are there any people with relatively steady incomes over the years?

2. This question uses data from the Iowa Youth and Families Project.

   (a) Prepare a side-by-side boxplot of each of the three responses (logged) against survey year. Describe the trend for each response.

   (b) Compute the medians for each survey year for the three (logged) responses. Plot these medians as trend lines on bivariate scatterplots of the (logged) responses. An extra challenge is to jitter the points to spread them out so that the distribution is slightly more readable. Describe the bivariate trends of responses in time.

   (c) Examine the medians, connected by line segments, in the 3D response space using a grand tour. We think that the main trend is simply that the kids become less stressed with time. This is seen by the string of medians falling essentially along a straight line in 3D. If it bends substantially then there is something else occurring over time, such as depression reduces more than anxiety in certain periods. Do you see any patterns like this?

   (d) Repeat the last three questions with the data conditioned on gender. What do you notice anything different about boys and girls responses on stress?

Chapter 10

Inference for Data Visualization

Good revealing plots often provoke the question "Is what we see really there?" To date, its been very difficult to address this question, but it seems that if inference is possible with numbers, why not for visual features? To begin we need to understand what "really there" really means. This chapter develops the concepts and describes approaches for making inference with pictures. It discusses ways to overcome the subjectiveness of the eye and the tendency to overinterpret structure.

10.1 Really There?

Sometimes when we see a pattern in a plot, its clear, there is no doubt that what we see is real. What is "real"? We're thinking about what patterns might be seen even if there is nothing happening, that is, arising from a null scenario. In terms of the statistical testing thinking we could consider "really there" to be:

> *Under scenarios where the underlying feature is absent, the visible feature in the data is too unlikely to have arisen by chance.*

In terms of classical hypothesis testing language, the null hypothesis would be that the "underlying" feature is absent, the alternative hypothesis would be that the underlying feature is present. The test statistic would be the visible feature itself. The problem, and advantage, in exploratory data analysis is that we don't know what feature we'll detect, so we have to include them all, which leads to:

> *Null hypothesis*: "absence of **all** features."
>
> *Alternative*: "presence of **some** features."

What are some examples of null scenarios? In a simple linear regression scenario with two variables X, Y, we are interested in the dependence between the two variables. Because we naturally are interested in dependence between X and Y the natural null hypothesis is that the *two variables are independent.* The plots in

Figure 10.1 show pairs of variables which would be considered dependent if correlation is used as the dependence. What might we learn about the departure from independence from studying these four plots. The top left plot is undoubtedled the perfect example of linear dependence between X and Y. The top right plot is a clear example where correlation is misleading, that the apparent dependence is due solely to one sample point. These two variables are not dependent. The plot at bottom right shows two variables that are clearly dependent, but the dependence is amongst sub-groups in the sample and it is negative rather than positive as indicated by the correlation. The plot are bottom right show two variables with some positive dependence but also strongly non-linear dependence. With graphics we not only detect a linear trend, but virtually any other trend (nonlinear, decreasing, discontinuous, outliers) as well. That is, we can detect many different types of dependence with visual methods easily.

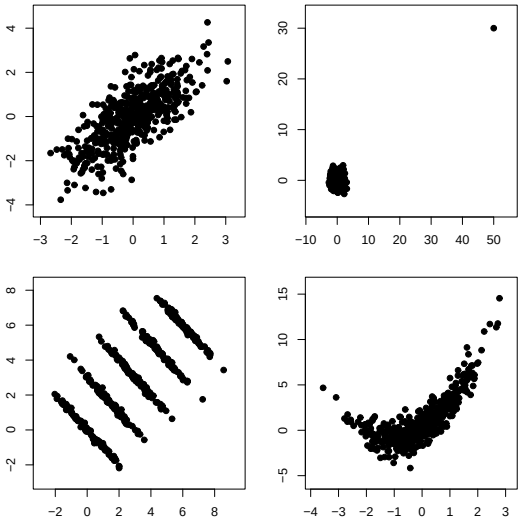

Figure 10.1. *Dependence between X and Y? All four pairs of variables have correlation approximately equal to 0.7.*

However, the eye can be easily distracted. If we are interested in dependence between X and Y we must try to ignore marginal structure. The plots in Figure 10.2 differ only in the marginal structure of X. In each plot the two variables are generated independently.

In general, it may be difficult to tailor visual detection to the structure of interest. It depends on being able to define the null scenario clearly. And it depends on human visual skills, how the structure may be perceived.

10.2 The Process of Assessing Significance

A recipe to establish a visual significance level is as follows:

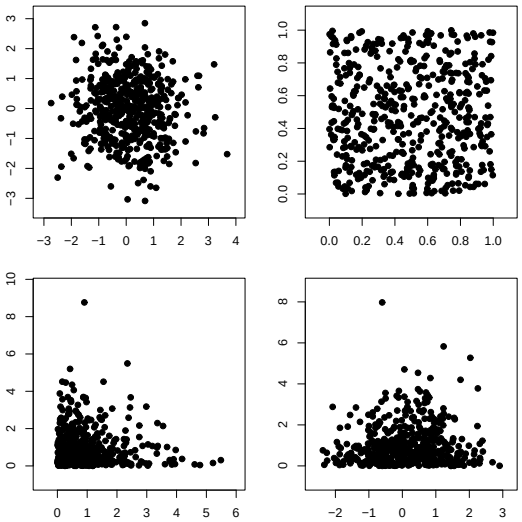

Figure 10.2. *Different forms of independence between X and Y.*

1. Identify the null hypothesis, and a mechanism for generating data consistent with this null.

2. Create a large number $(N-1)$ of plots of simulated null data.

3. Randomly insert the plot of the actual data, to give N plots.

4. Ask an uninvolved person to select the most special looking plot, and their reason for selecting it.

5. If the selected plot shows the actual data, and if the person's reason to select the plot is consistent with the structure in the plot, then the existence of a feature is significant at the level $\alpha = 1/N$.

10.3 Types of Null Hypotheses

There are several easy null scenarios to generate:

1. Any distributional assumption, simulate samples from the distribution having parameters estimated from the sample. If we suspect, or hope, that the data is consistent with a normal population, simulate samples from a normal distribution using the sample mean and variance-covariance matrix as the parameters.

2. For independence assumptions, using permutation test methods, shuffle the appropriate columns of data values, For two variables shuffle X-values against the Y-values, as in a permutation test.

3. In labeled data problems, when the assumption is that the labels matter, shuffle the labels. In a designed experiment, with two groups, control and treatment, randomly re-assign the control/treatment labels. In supervised classification, shuffle the class ids.

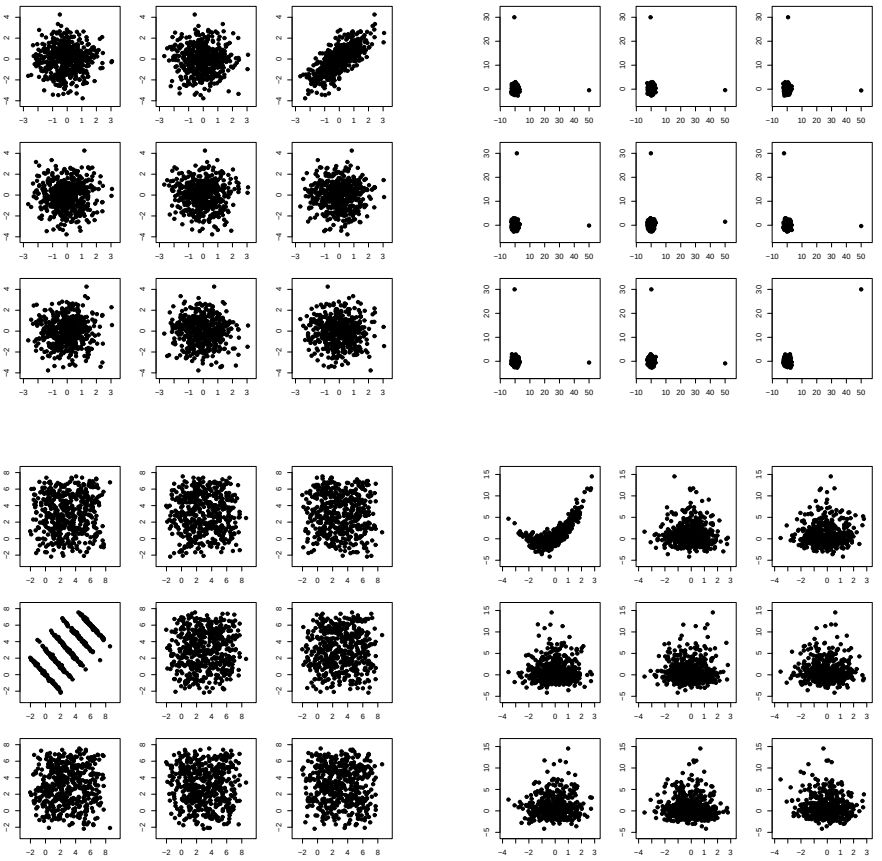

Figure 10.3. *(Top left) The plot of the original data is very different from the other plots, clearly there is dependence between the two variables. (Top right) The permuted data plots are almost all the same as the plot of the original data, except for the outlier. (Bottom Left, Right) The original data plot is very different to permuted data plots. Clearly there is dependence between the variables, but we also can see that the dependence is not so simple as positive linear association.*

Lets take a look at the simulated data examples from Figure 10.1. In each of these examples we are assessing assumptions about independence. The data in the top left plot is without a doubt linearly dependent, but lets check this observation.

We permute the X-values and plot the data again, several times. These plots are arranged along with the plot of the original data in Figure 10.3. In each of these example data sets the original data plot is clearly distinguishable from the permuted data plots, which establishes that there is dependence between two variables, though not necessarily positive linear association. The least clear of the examples is the plot of the data containing the outlier. In the permuted data the X-Y pair of high values is split, resulting in two points that locate in the top left and bottom right of the plots. The interesting feature in this data that defies an independence assumption is the outlier.

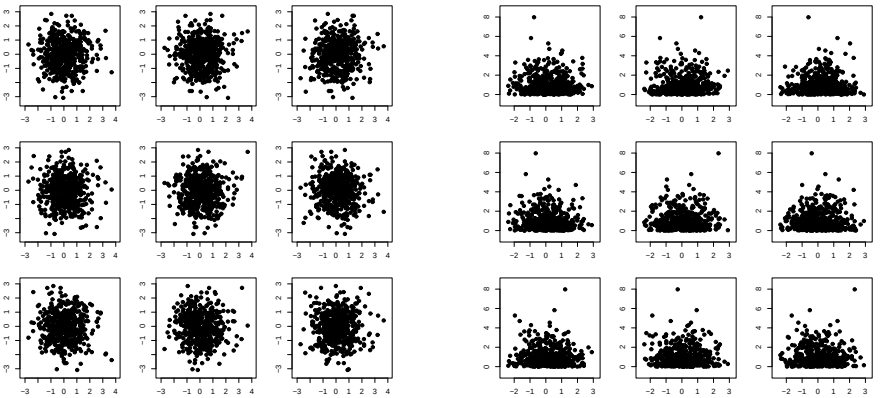

Figure 10.4. *Plots independent examples, two variables generated independently, from different distributions, embedded into plots of permuted data. The plots of the original data are indistinguishable from the permuted data: clearly there is no dependence.*

10.4 Examples

10.4.1 Tips

One of the observations that we made about tipping behavior is that for smoking parties there was very little relationship between tip and total bill. We'll assess this observation by subsetting the smoking parties from the data, and embedding the plot of this subset amongst plots of the subset where total bill is permuted. This is done in Figure 10.5. Can you tell which was the real data? It is obvious. Its the plot in the second row, third column. How can we tell? There are several reasons. The most obvious difference is clear because we've just seen a similar example in the simulated data: There is the large outlier in the upper right of the plot. In all the permuted data the outliers are not in the upper right part of the plot. The slightly less obvious but more important difference is that the concentration of

points along the diagonal is stronger than in the permuted data plots. This suggests that although the dependence is weak it is really is there.

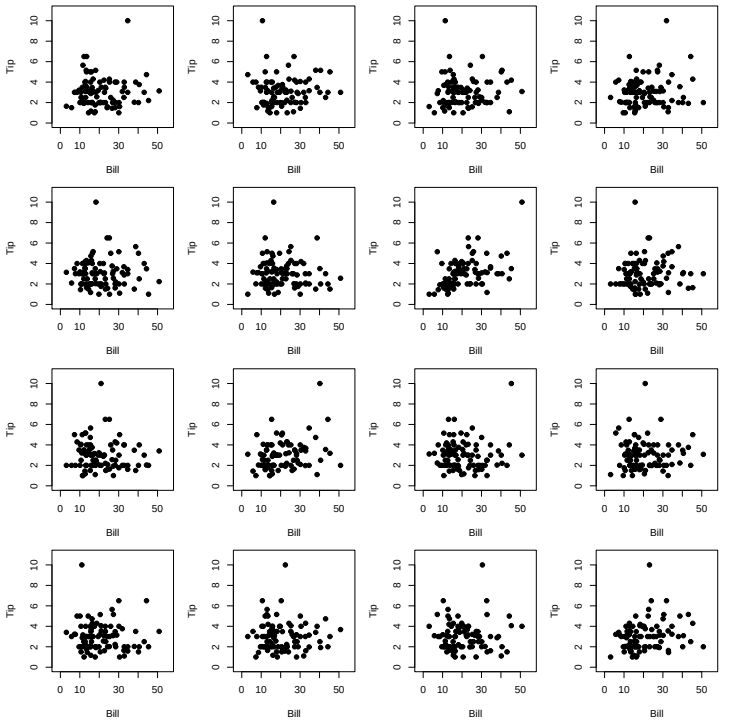

Figure 10.5. *Tip vs Bill for smoking parties: Which is the plot of the original data?*

Would it have mattered if we'd permuted Tip instead of Bill? No. Remember we commented on the horizontal bands due to rounding of tip. This is marginal structure that we'd like to ignore, but is not affected by permuting either of the two variables.

10.4.2 Particle physics

Did we just see a triangle? Recall in the particle physics example we used graphics to uncover a distinct geometric pattern in the 7D space: the points lie close to a 2D triangle, and six lines extending from the vertices of the triangle. How can we assure ourselves that this is a 2D triangle and not a 3D simplex? Simulating data according to both models, and compare these with the original data. The plots in Figure 10.6 illustrate this. We generated data uniformly in a 3D simplex with 4D of small amounts of noise, and also data uniformly in a 2D triangle with 5D of small amounts of noise. We looked at each of these data sets in a tour.

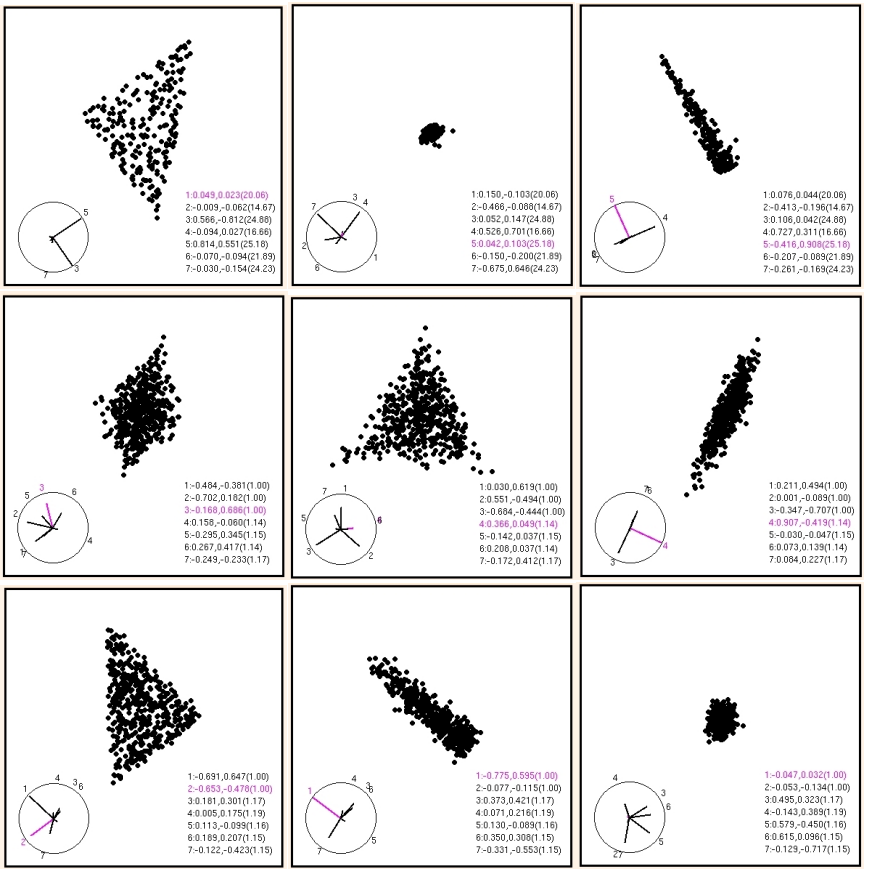

Figure 10.6. *(Top row) Three revealing tour projections - a triangle, a line, and almost collapsed to a point - of the subset of the actual data that seems to follow a 2D triangle shape. (Middle row) Plots of the 3D simplex plus noise: the most revealing plot is the first one, where four vertices are seen. This alone establishes that what we have in the actual data is not a 3D simplex. (Bottom row) Tour plots of the 2D triangle plus noise, more closely matches the original data.*

10.4.3 Baker data

This is an interesting example. The real plot of Yield against Boron is the one in the last plot in the second row. From the plot it appears that as boron concentration increases yield is consistently higher. Is this possibly true? From discussions with soil scientists, boron has an interesting dynamic with plants: it is beneficial to corn yield, but in high doses it can be toxic. In this data, most of the boron conentrations are low, with less and less larger values, that is boron concentration is skewed. Yield is also skewed. Thus the variance difference is counfounded by sample size. We definitely should expect to see a reduction in the variance as boron

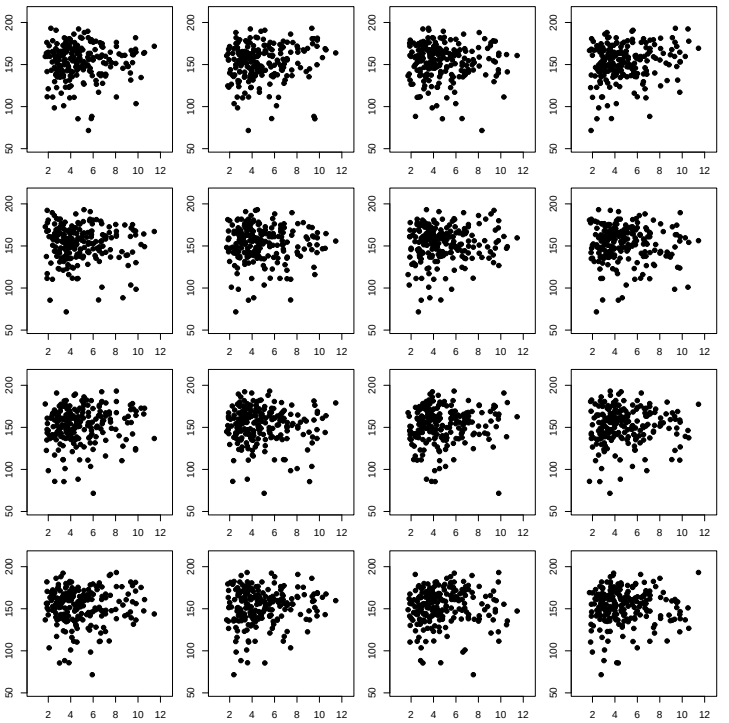

Figure 10.7. *Which is the real plot of Yield vs Boron?*

increases, and probably shouldn't be surprised to see mostly high values of yield. And this is what we see from the plots of permuted data: the pattern of higher yield for higher boron is visible in several plots not just the real data. But there is something surprising here: there is one sample point where boron concentration is very high but the yield is extraordinarily low. This outlier is not present in any of the permuted data plots, which suggests that this is potentially important. In informal tests we have found that people can pick out the plot the real data based on this outlier.

10.4.4 Wages data

In the wages data, we suspect there is an odd trend in the wages vs experience for black kids which is significantly different from whites and hispanics. To test this we take the race labels for each individual and shuffle them. All the time points for an individual are now give with the new label. We recompute the lowess smoothed curve for each group, 15 times, plot these, and embed the real data. The field of plots is shown in Figure 10.8. What can be seen? The actual data has a much lower dip in the mid-range of experience than any of the other permuted data plots. This is evidence for this difference being real. What else can be seen? Many of the plots have substantial difference between the curves at the higher values of experience.

Clearly, the difference that can be seen in the actual data, at this end of the range, is not important, because it occurs by chance. Its likely due to the smaller sample size in this range of workforce experience.

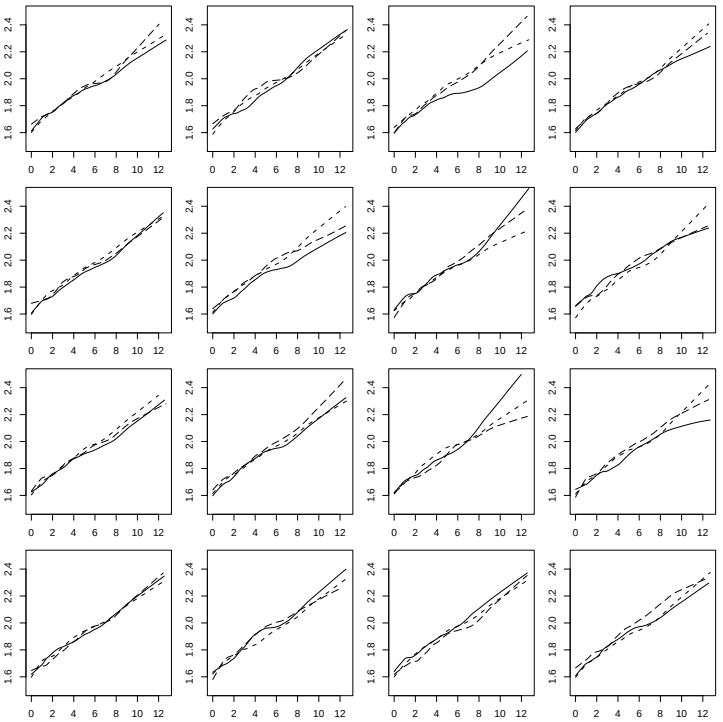

Figure 10.8. *Which of these plots is not like the others? One of these plots is the actual data, where wages and experience have lowess smooth curves conditional on race. The remaining are generated by permuting the race labels for each individual.*

10.4.5 Leukemia

This is an example of how permutations may be used in supervised classification. The data is the Leukemia gene expression data. The top 40 genes are used, from the 7129 original genes. There are 3 cancer types that constitute the class variable, and this is the column that we permute to check the validity of the class separations. We're going to use the 1D tour to check the separations between classes. The top row of plots in Figure 10.9 shows two 1D projections for the data colored using the correct class. In each of these plots we are seeing a 1D projection of the 40 variables corresponding to genes. Each point corresponds to one tissue sample labelled as one

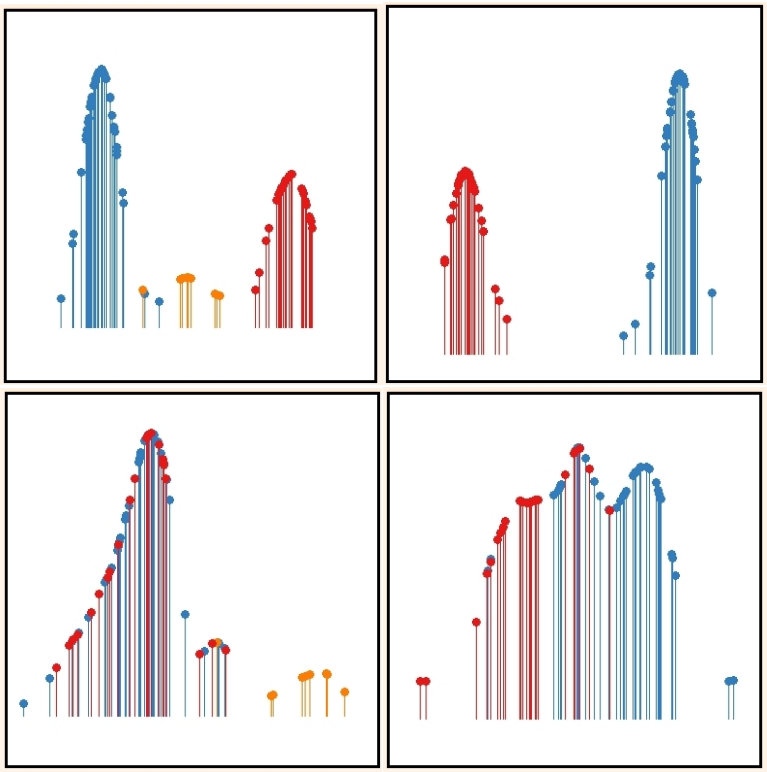

Figure 10.9. *Leukemia gene expression data: (Top row) 1D tour projections of the actual data revealing separations between the three cancer classes. (Bottom row) 1D tour projections of permuted class data shows there are still some separations but not as large as for the actual class.*

of three cancer types. We used projection pursuit with the LDA index to obtain these plots. We also removed the smallest class, and projection pursuit was run on the remaining two classes. (We re-scaled the plots so that the data fills the window.) In the actual data the two largest classes (red, blue) are very well separated, and to a lesser extent the smaller group is separable from the others also. In the plots of the data where the colors are nonsense, colored according to permuted class, there is less separation between the classes. The small group is quite separated, but the two larger groups are not. Now its important to think a little more about this. This is a situation where there a lot of variables and few sample points. There's good chance of finding separations between classes even if the class labels are randomly assigned. We see some of this here: the small group is better separated from the others in the permuted data than the actual data.

10.5 Exercises

1. These questions relate to the particle physics data.

   (a) For the particle physics data, simulate data from a 7D multivariate exponential distribution with independent variables. You'll need to generate seven samples of 500 points from univariate exponential distributions. Look at this data using the spin-and-brush method. How does this simulated data differ from the actual data? Would you say that the actual data coule be considered to be a sample from seven independent exponential distributions?

   (b) Use the permutation approach to generate data from a null distribution of independence between variables. You'll need to shuffle the data values for six of the seven variables and use the spin-and-brush analysis on the permuted data. Is the geometric structure that we suspect underlies this data simply an artifact?

2. Subset the Australia crabs data to be blue males only. Compute the mean and variance-covariance of this subset, and use this information to generate a sample from a multivariate normal distribution having population mean and variance the same as the sample mean and variance for this data. Compare this simulated data with the actual data. Do you think the blue male crabs subset is consistent with a sample from a multivariate notmal?

3. In the baker data, examine the relationship between log(copper) and yield, using the permutation approach. Do you think that yield is improved by increased copper in the soil?

4. Choose a data set from the supervised classification chapter permute the class values and search for the best separation. Is the separation as good as the separation for the true class variable?

5. This question is about sampling variability in multivariate distributions.

   (a) Generate 16 samples of size 20 from a bivariate standard normal distribution. What patterns can you see in the different plots? These patterns are purely due to sampling variability.

   (b) Generate 16 samples of size 50 from a bivariate standard normal distribution. What patterns can you see in the different plots?

   (c) Generate 16 samples of size 150 from a bivariate standard normal distribution. What patterns can you see in the different plots? Are there more or less strange patterns with the larger sample size?