Preflight Summary Report for: chapters_1-6.pdf

Profile: Fonts_and_RGB (Processed pages 1 to 278)

Processed by fontainem, Date: 8/21/2015 2:25 PM

Results (Summary)

Error

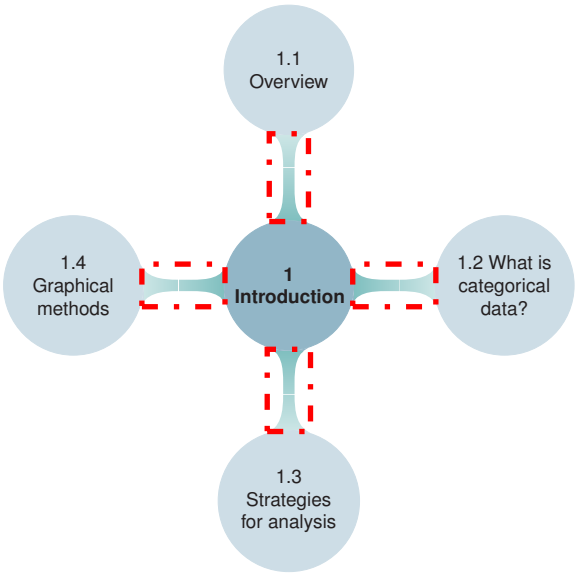
- ✖ Object uses RGB (150 matches on 18 pages)
- ✖ Transparency used (filled object with ca value smaller than 1.0) (7 matches on 2 pages)
- ✖ Transparency used (soft mask in image) (3 matches on 2 pages)
- ✖ Transparency used (stroked object with CA value smaller than 1.0) (9 matches on 1 page)
- ✖ Transparency used (transparency group) (63 matches on 1 page)

Document information

File name: "chapters_1-6.pdf"
Path: "C:\Users\fontainem\Desktop"
PDF version number: "1.6"
File size (MB): 4.05
Creator: "TeX"
Producer: "Adobe Acrobat Pro 10.1.15"
Created: "8/21/2015 1:14 PM"
Modified: "8/21/2015 1:14 PM"
Trapping: "False"
Additional actions
  Page open action
    Type: Go to page in current document
  Page open action
    Type: Go to page in current document
  Page open action
    Type: Go to page in current document
  Page open action
    Type: Go to page in current document
  Page open action
    Type: Go to page in current document
  Page open action
    Type: Go to page in current document
Number of plates: 4
Names of plates: "(Cyan) (Magenta) (Yellow) (Black) "

Environment

Preflight, 10.1.3 (090)
Acrobat version: 10.115
Operating system: Microsoft Windows Service Pack 1 (Build 7601)



1

Introduction

{ch:intro}

Categorical data consist of variables whose values comprise a set of discrete categories. Such data require different statistical and graphical methods than commonly used for quantitative data. The focus of this book is on visualization techniques and graphical methods designed to reveal patterns of relationships among categorical variables. This chapter outlines the basic orientation of the book and some key distinctions regarding the analysis and visualization of categorical data.

1.1 Data visualization and categorical data: Overview

{sec:viscat}

> Graphs carry the message home. A universal language, graphs convey information directly to the mind. Without complexity there is imaged to the eye a magnitude to be remembered. Words have wings, but graphs interpret. Graphs are pure quantity, stripped of verbal sham, reduced to dimension, vivid, unescapable.
>
> Henry D. Hubbard, in Foreword to Brinton (1939), *Graphic Presentation*

"Data visualization" can mean many things, from popular press infographics, to maps of voter turnout or party choice. Here we use this term in the narrower context of statistical analysis. As such, we refer to an approach to data analysis that focuses on *insightful* graphical display in the service of both *understanding* our data and *communicating* our results to others.

We may display the raw data, some summary statistics, or some indicators of the quality or adequacy of a fitted model. The word "insightful" suggests that the goal is (hopefully) to reveal some aspects of the data that might not be perceived, appreciated, or absorbed by other means. As

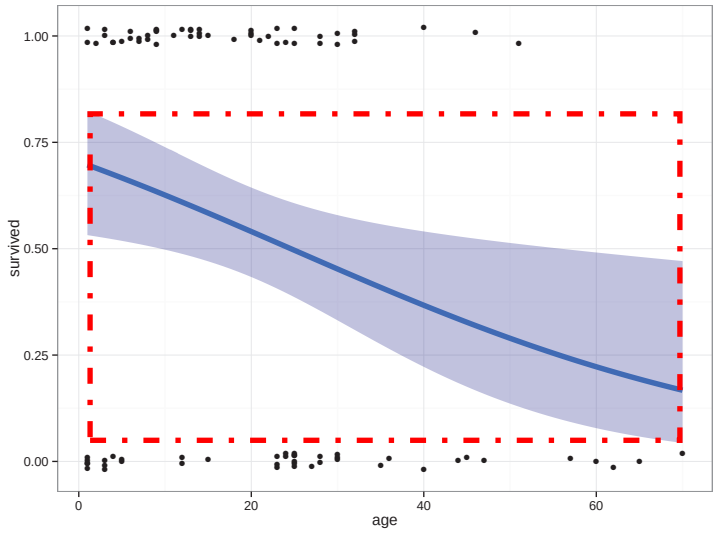


Figure 1.3: Donner party data, showing the relationship between age and survival. The blue curve and confidence band give the predicted probability of survival from a linear logistic regression model. {fig:donner0}

linear logistic regression model for these data with a 95% confidence band for the predictions. The prediction equation for this model can be given as:

$$\text{logit}(\text{survived}) = \underset{(0.372)}{0.868} - \underset{(0.015)}{0.0353}\ \text{age}$$

ignoretrue

The equation above implies that the log odds of survival decreases by 0.0352 with each additional year of age or by $10 \times 0.0352 = 0.352$ for an additional decade. Another way to say this is that the odds of survival is multiplied by $\exp(0.353) = .702$ with each 10 years of age, a 30% decrease.

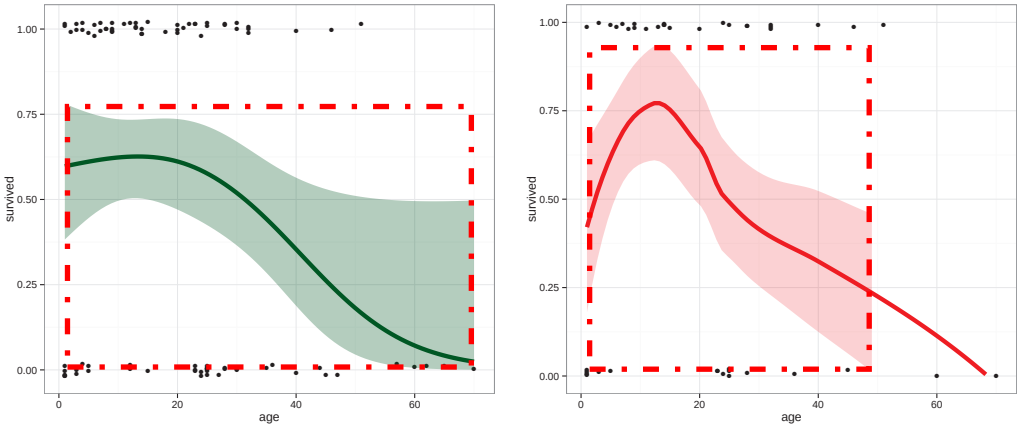


Figure 1.4: Donner party data, showing other model-based smoothers for the relationship between age and survival. Left: using a natural spline; right: using a non-parametric loess smoother. {fig:donner0-other}

Of course, these visual and statistical summaries depend on the validity of the fitted model. For contrast, Figure 1.4 shows two other model-based smoothers that relax the assumption of the linear logistic regression model. The left panel shows the result of fitting a semi-parametric model with a natural cubic spline with one more degree of freedom than the linear logistic model. The right panel shows the fitted curve for a non-parametric, locally weighted scatterplot smoothing (loess) model. Both of these hint that the relationship of survival to age is more complex than what is captured in the linear logistic regression model. We return to these data in Chapter 7.

△

1.4 Graphical methods for categorical data

{sec:methods}

> You can see a lot, just by looking
>
> Yogi Berra

The graphical methods for categorical data described in this book are in some cases straightforward adaptations of more familiar visualization techniques developed for quantitative data. Graphical principles and strategies, and the relations between the visualization approach and traditional statistical methods, are described in a number of sources, including Chambers et al. (1983), Cleveland (1993b) and several influential books by Tufte (Tufte, 1983, 1990, 1997, 2006).

The fundamental idea of statistical graphics as a comprehensive system of visual signs and symbols with a grammar and semantics was first proposed in Jacques Bertin's *Semiology of Graphics* (1983). These ideas were later extended to a computational theory in Wilkinson's *Grammar of Graphics* (2005), and implemented in R in Hadley Wickham's ggplot2 (Wickham and Chang, 2015) package (Wickham, 2009, Wickham and Chang, 2015).

Another perspective on visual data display is presented in Section 1.4.1 focusing on the communication goals of statistical graphics. However, the discrete nature of categorical data implies that some familiar graphic methods need to be adapted, while in other cases we require a new graphic metaphor for data display. These issues are illustrated in Section 1.4.2. Section 1.4.3 discusses the principle of effect ordering for categorical variables in graphs and tables.

1.4.1 Goals and design principles for visual data display

{sec:intro-goals}

Designing good graphics is surely an art, but as surely, it is one that ought to be informed by science. In constructing a graph, quantitative and qualitative information is encoded by visual features, such as position, size, texture, symbols, and color. This translation is reversed when a person studies a graph. The representation of numerical magnitude and categorical grouping, and the apperception of patterns and their *meaning* must be extracted from the visual display.

There are many views of graphs, of graphical perception, and of the roles of data visualization in discovering and communicating information. On the one hand, one may regard a graphical display as a *stimulus*—a package of information to be conveyed to an idealized observer. From this perspective certain questions are of interest: which form or graphic aspect promotes greater accuracy or speed of judgment (for a particular task or question)? What aspects lead to greatest memorability or impact? Cleveland (Cleveland and McGill, 1984, 1985, Cleveland, 1993a), Spence and Lewandowsky (Lewandowsky and Spence, 1989, Spence, 1990, Spence and Lewandowsky, 1990) have made important contributions to our understanding of these aspects of graphical display.

An alternative view regards a graphical display as an act of *communication*—like a narrative, or even a poetic text or work of art. This perspective places the greatest emphasis on the desired communication goal, and judges the effectiveness of a graphical display in how well that goal is

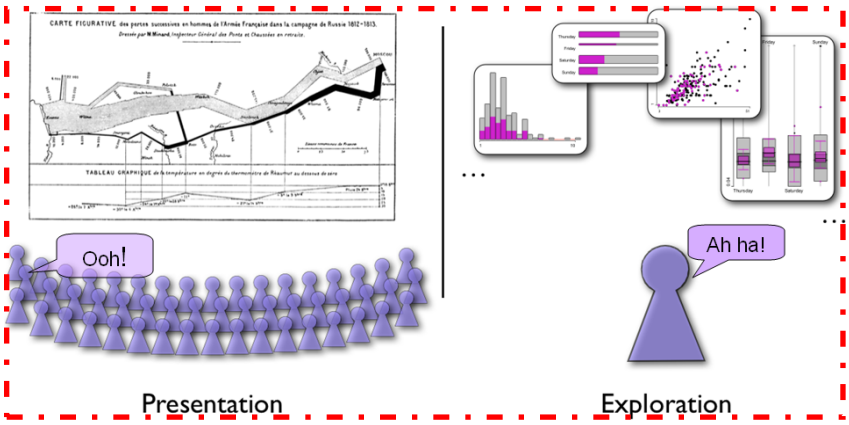


Figure 1.5: Different communication purposes require different graphs. For presentations, a single, carefully crafted graph may appeal best to a large audience; for exploratory analysis, many related images from different perspectives for a narrow audience (often you!). *Source*: Adapted from a blog entry by Martin Theus, `http://www.theusrus.de/blog/presentation-vs-exploration/`.

{fig:presentation-exploration}

achieved (Friendly and Kwan, 2011). Kosslyn (1985, 1989) and Tufte (1983, 1990, 1997) have articulated this perspective most clearly.

In this view, an effective graphical display, like good writing, requires an understanding of its *purpose*—what aspects of the data are to be communicated to the viewer. In writing we communicate most effectively when we know our audience and tailor the message appropriately. So too, we may construct a graph in different ways to: (a) use ourselves, (b) present at a conference or meeting of our colleagues, (c) publish in a research report, or (d) communicate to a general audience (Friendly (1991, Ch. 1), Friendly and Kwan (2011)). Figure 1.5 illustrates a basic contrast between graphs for presentation purposes, designed to appeal persuasively to a large audience (one-to-many) and the use of perhaps many graphs we might make for ourselves for exploratory data analysis (many-to-one).

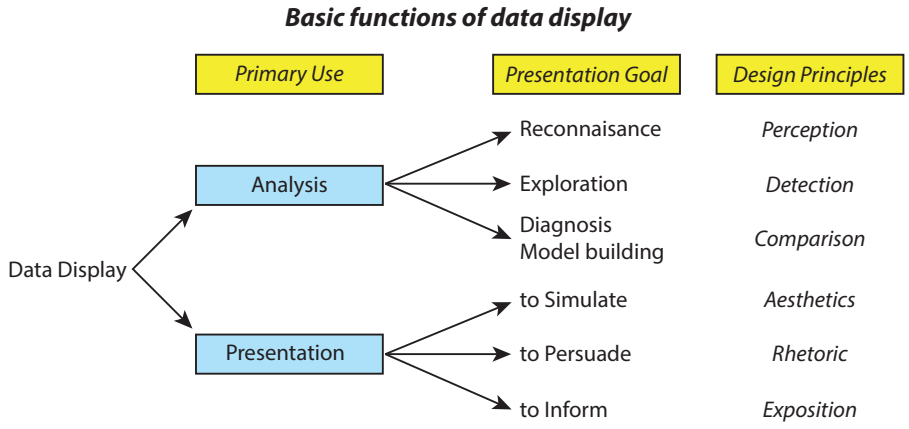


Figure 1.6: A taxonomy of the basic functions of data display by intended use, presentation goal, and design principles.

{fig:datadisp}

Figure 1.6 shows one organization of visualization methods in terms of the *primary* use or intended communication goal, the functional *presentation goal*, and suggested corresponding *design principles*.

We illustrate these ideas and distinctions in the examples below, most of which are treated again in later chapters. {ex:arrests0}

EXAMPLE 1.4: Racial profiling: Arrests for marijuana possession

In a case study that will be examined in detail in Chapter 7 (Example 7.10), the *Toronto Star* newspaper studied a huge data base of arrest records by Toronto police for indications of possible racial profiling, i.e., differential treatment of those arrested on the basis of skin color. They focused on the charge of simple possession of a small amount of marijuana, for which enforcement procedures allowed police discretion. An officer could release an arrestee with a summons ("Form 9") to appear in court, or take the person to a police station for questioning ("Form 10") or booking ("Form 11.1"), or order the person held in jail for a bail hearing ("Show cause").

The statistical issue was whether the data on these arrests showed evidence of differential treatment in relation to skin color, particularly in the treatment of blacks vs. whites, controlling, of course, for other factors. Statistical tests on these data (χ^2 tests, loglinear models, logistic regression) showed overwhelming evidence of differential treatment of blacks and whites. However, tables of these results do not reveal the nature of this association.

Figure 1.7 is an example of a graph designed for *analysis*—a mosaic display (Chapter 5) showing the frequencies of those arrested on this charge by skin color and release type. The size of each rectangle shows the frequency and these are shaded in relation to the association between skin color and release—blue for positive associations (more than expected if they were independent) to red for negative associations.

Once you know how to read such graphs, the pattern is clear: blacks were indeed more likely

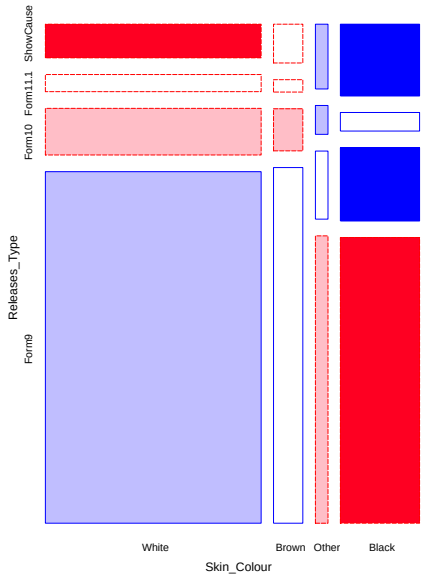


Figure 1.7: Mosaic display showing the relationship between skin color and release type for those arrested on a charge of simple possession of marijuana in Toronto, 1996–2002. {fig:arrests0-mosaic}

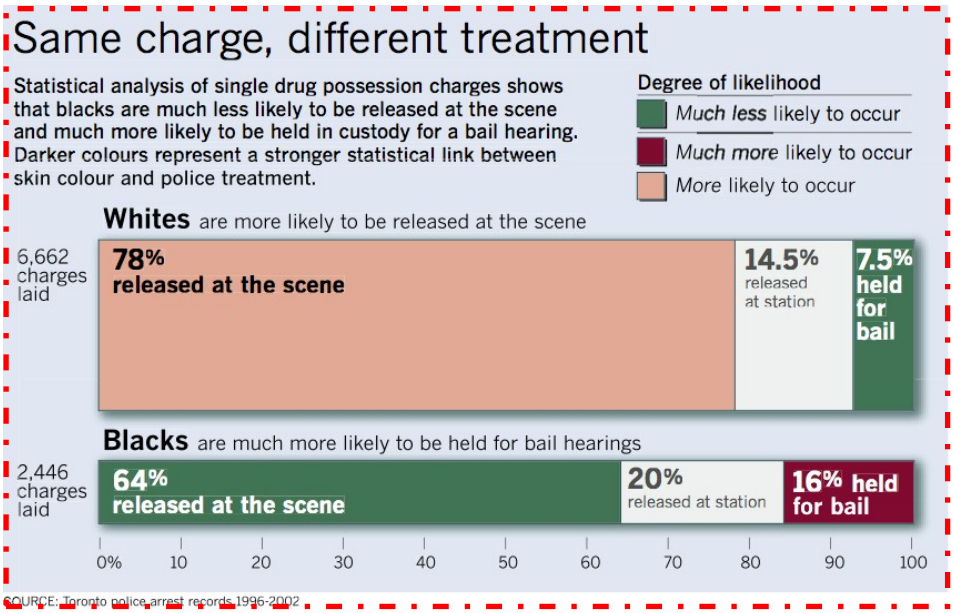


Figure 1.8: Redesign of Figure 1.7 as a presentation graphic. *Source*: Graphics department, *The Toronto Star*, December 11, 2002. Used by permission.

{fig:arrests0-star}

to be held for more severe treatment, whites were more likely to be released with a summons. But this is hardly a graph that would be clear to a general audience, and would require a good deal of explanation.

In contrast, Figure 1.8 shows a redesign of this as a *presentation graphic* prepared by the *Star* and published on December 11, 2002 in conjunction with a meeting between the newspaper and the Toronto Police Services Board to consider the issue of racial profiling. The police vehemently denied that racial profiling was taking place. The revision makes the point immediately obvious and compelling in the following ways:

- It announces the conclusion in the figure title: "Same charge, different treatment."
- The text box at the top provides the context for this conclusion
- Skin colors "Brown" and "Other," which appeared less frequently, were removed, and the release categories "Form 10" and "Form 11.1" were combined as "released at station."
- The graphic is still a mosaic display, however, it now shows explicitly the number of charges laid against whites and blacks and the percentage of each treatment.
- The labels for whites and blacks were enhanced by indicating what a reader should see for each.
- The legend for color is titled non-technically as "degree of likelihood."

Clear communication is not achieved without effort. The revised graph required several iterations and emails between the graphic designer and the statistical consultant (the first author of this book) in the few hours available before the newspaper went to press. The main question was, "what are we trying to show here?" Starting with the original Figure 1.7 mosaic, we asked, "what can we remove?" and "what can we add?" to make the message clearer.

△

1.4.2 Categorical data require different graphical methods

{sec:intro-catdata}

We mentioned earlier, and will see in greater detail in Chapter 7 and Chapter 9, that statistical models for discrete response data and for frequency data are close analogs of the linear regression and ANOVA models used for quantitative data. These analogies suggest that the graphical methods commonly used for quantitative data may be adapted directly to categorical data.

Happily, it turns out that many of the analysis graphs and diagnostic displays (e.g., effect plots, influence plots, added variable and partial residual plots, etc.) that have become common adjuncts in the analysis of quantitative data have been extended to generalized linear models including logistic regression (Section 7.5) and loglinear models (Section 11.6).

Unhappily, the familiar techniques for displaying raw data are often disappointing when applied to categorical data. The simple scatterplot, for example, widely used to show the relation between quantitative response and predictors, when applied to discrete variables, gives a display of the category combinations, with all identical values overplotted, and no representation of their frequency.

Instead, frequencies of categorical variables are often best represented graphically using *areas* rather than as position along a scale. Friendly (1995) describes conceptual and statistical models that give a rationale for this graphic representation. Figure 1.7 does this in the form of a modified bar chart (mosaic plot), where the widths of the horizontal bars show the proportions of whites and blacks in the data, and the divisions of each group give the percents of each release type. Consequently, the areas of each bar are proportional to the frequency in the cells of this 2×3 table.

As we describe later in this book, using the visual attribute

$$\textbf{area} \sim \textbf{frequency}$$

also allows creating novel graphical displays of frequency data for special circumstances.

Figure 1.9 shows two examples. The left panel gives a ***fourfold display*** of the frequencies of admission and gender in the Berkeley data shown in Table 1.1. What should be seen at a glance is that males are more often admitted and females more often rejected (shaded blue); see Section 4.4 for details.

The right panel shows another specialized display, an ***agreement chart*** designed to show the

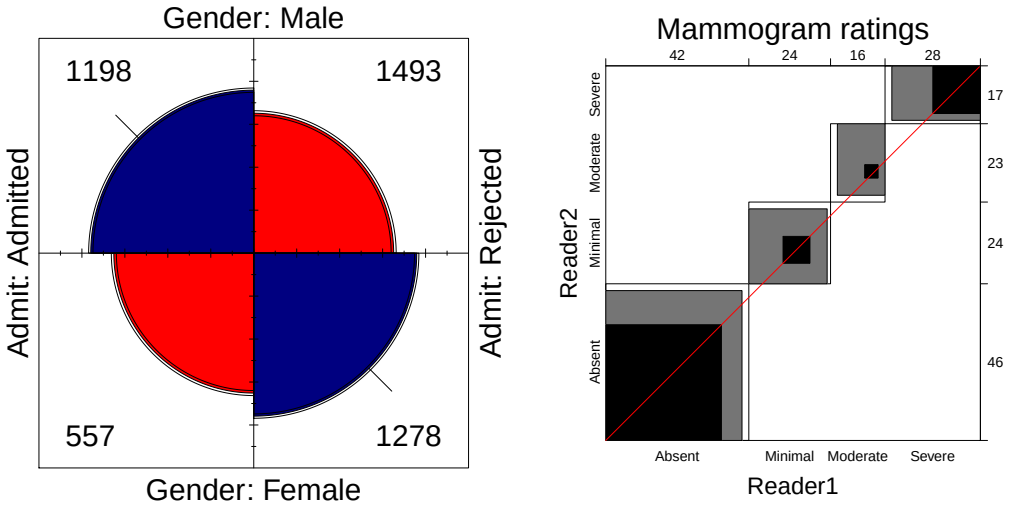


Figure 1.9: Frequencies of categorical variables shown as areas. Left: fourfold display of the relation between gender and admission in the Berkeley data; right: agreement plot for two raters assessing mammograms.

{fig:area-diagrams}

Table 1.4: Barley data, yield differences, 1931-1932, sorted by mean difference, and shaded by value {tab:barley2c}

Variety	Site						*Mean*
	Morris	Duluth	University Farm	Grand Rapids	Waseca	Crookston	
No. 475	-22	6	-5	4	6	12	0.1
Wisconsin No. 38	-18	2	1	14	1	14	2.4
Velvet	-13	4	13	-9	13	9	2.9
Peatland	-13	1	5	8	13	16	4.8
Manchuria	-7	6	0	11	15	7	5.5
Trebi	-3	3	7	9	15	5	6.1
Svansota	-9	3	8	13	9	20	7.3
No. 462	-17	6	11	5	21	18	7.4
Glabron	-6	4	6	15	17	12	8.0
No. 457	-15	11	17	13	16	11	8.8
Mean	-12.2	4.6	6.3	8.2	12.5	12.5	5.3

difference increasing with both row and column means. Against this background, one other cell, for Velvet grown at Grand Rapids, stands out with an anomalous negative value.

Although the use of color for graphs is now more common in some journals, color and other rendering details in tables are still difficult. The published version of Table 1.4 (Friendly and Kwan, 2003, Table 3) was forced to use only font shape (normal, italics) to distinguish positive and negative values.

△

Finally, effect ordering is also usefully applied to the variables in multivariate data sets, which by default, are often ordered in data displays according to their position in a data frame or alphabetically. {ex:1.7}

EXAMPLE 1.7: Iris data

The classic `iris` data set (Anderson, 1935, Fisher, 1936) gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris, *Iris setosa*, *versicolor*, and *virginica*. Such multivariate data are often displayed in ***parallel coordinate plot***s, using a separate vertical axis for each variable, scaled from its minimum to maximum.

The default plot, with variables shown in their data frame order, is shown in the left panel of Figure 1.11, and gives rise to the epithet *spaghetti plot* for such displays because of the large number of line crossings. This feature arises because one variable, sepal width, has negative relations in the species means with the other variables. Simple rearrangement of the variables to put sepal width last (or first) makes the relations among the species and the variables more apparent, as shown in the right panel of Figure 1.11. This plot has also been enhanced by using ***alpha-blending*** (partial transparency) of thicker lines, so that the density of lines is more apparent.

Parallel coordinate plots for categorical data are discussed in an online supplement on the web site for the book. A general method for reordering variables in multivariate data visualizations based on cluster analysis was proposed by Hurley (2004).

△

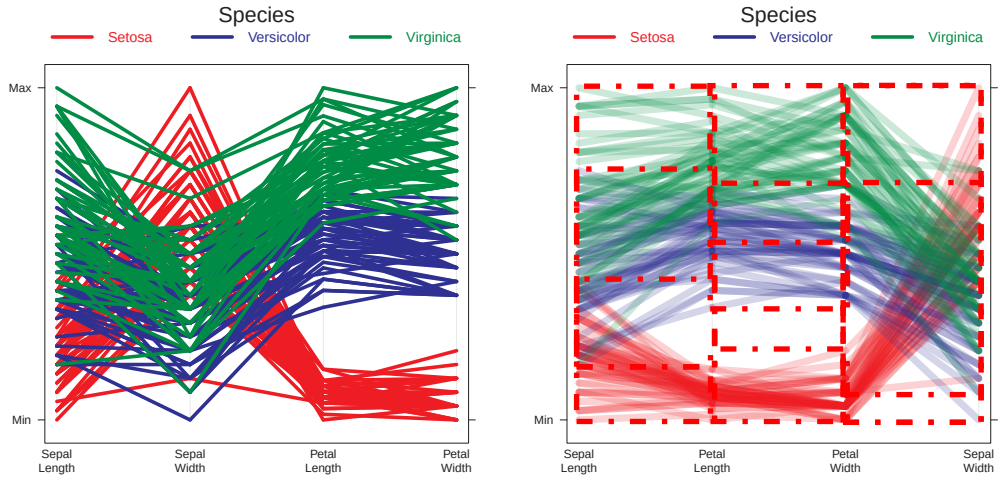


Figure 1.11: Parallel coordinates plots of the Iris data. Left: Default variable order; right: Variables ordered to make the pattern of correlations more coherent.

{fig:iris-parallel}

1.4.4 Interactive and dynamic graphics

{sec:intro-interactive}

Graphics displayed in print form, such as this book, are necessarily static and fixed at the time they are designed and rendered as an image. Yet, recent developments in software, web technology and media alternative to print have created the possibility to extend graphics in far more useful and interesting ways, for both presentation and analysis purposes.

Interactive graphics allow the viewer to directly manipulate the statistical and visual components of graphical display. These range from

- graphical controls (sliders, selection boxes, and other widgets) to control details of an analysis (e.g., a smoothing parameter) or graph (colors and other graphic details), to
- higher-level interaction including zooming in or out, drilling down to a data subset, linking multiple displays, selecting terms in a model, and so forth.

The important effect is that the analysis and/or display is immediately re-computed and updated visually.

In addition, ***dynamic graphics*** use animation to show a series of views, as frames in a movie. Adding time as an additional dimension allows far more possibilities, for example showing a rotating view of a 3D graph or showing smooth transitions or interpolations from one view to another.

There are now many packages in R providing interactive and dynamic plots (e.g., rggobi (Temple Lang et al., 2014), iplots (Urbanek and Wichtrey, 2013)) as well as capabilities to incorporate these into interactive documents, presentations, and web pages (e.g., rCharts (Vaidyanathan, 2013), googleVis (Gesmann and de Castillo, 2015), ggvis (Chang and Wickham, 2015)). The animation (Xie, 2014) package facilitates creating animated graphics and movies in a variety of formats. The RStudio editor and development environment[5] provides its own manipulate (RStudio, Inc., 2011) package, as well as the shiny (RStudio, Inc., 2015) framework for developing interactive R web applications.

{ex:512paths}

EXAMPLE 1.8: 512 paths to the White House

Shortly before the 2012 U.S. presidential election (November 2, 2012) *The New York Times*

[5] http://www.rstudio.com.

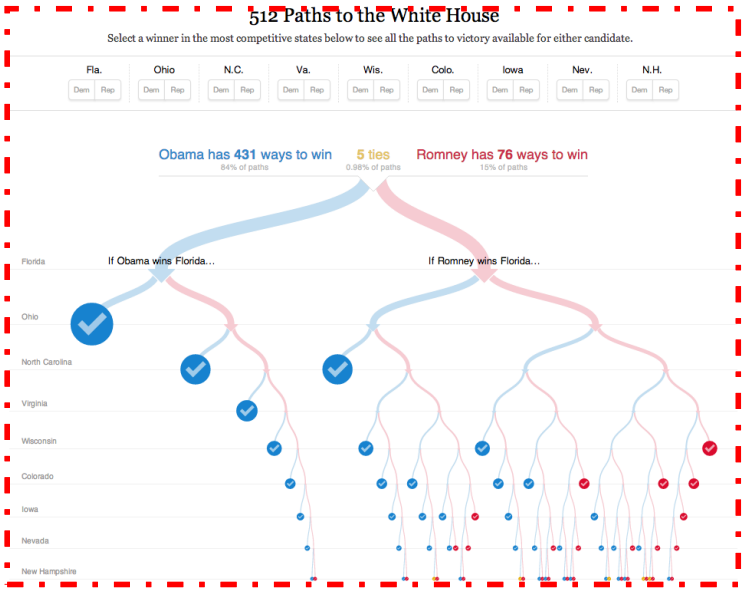


Figure 1.12: 512 paths to the White House. This interactive graphic allows the viewer to select a winner in any one or more of the nine most highly contested U.S. states and highlights the number of paths leading to a win by Obama or Romney, sorted and weighted by the number of Electoral College votes. *Source*: Mike Bostock & Shan Carter, *New York Times* interactive, November 2, 2012. Used by permission.

{fig:nyt_512paths}

published an interactive graphic, [6] designed by Mike Bostock and Shan Carter,[7] showing the effect that a win for Barack Obama or Mitt Romney in the nine most highly contested states would have on the chances that either candidate would win the presidency.

With these nine states in play there are $2^9 = 512$ possible outcomes, each with a different number of votes in the Electorial College. In Figure 1.12, a win for Obama in Florida and Virginia was selected, with wins for Romney in Ohio and North Carolina. Most other selections also lead to a win by Obama, but those with the most votes are made most visible at the top. An R version of this chart was created using the rCharts package.[8] The design of this graphic as a ***binary tree*** was chosen here, but another possibility would be a ***treemap*** graphic (Shneiderman, 1992) or a mosaic plot.

△

1.4.5 Visualization = Graphing + Fitting + Graphing . . .

{sec:vis}

> Look here, upon this picture, and on this.
>
> Shakespeare, Hamlet

Statistical summaries, hypothesis tests, and the numerical parameters derived in fitted models

[6] http://www.nytimes.com/interactive/2012/11/02/us/politics/paths-to-the-white-house.html.

[7] see: https://source.opennews.org/en-US/articles/nyts-512-paths-white-house/. for a description of their design process.

[8] http://timelyportfolio.github.io/rCharts_512paths/

are designed to capture a particular feature of the data. A quick analysis of the data from Table 1.1, for example, shows that 1198/2691 = 44.5% of male applicants were admitted, compared to 557/1835 = 30.4% of female applicants.

Statistical tests give a Pearson χ^2 of 92.2 with 1 degree of freedom for association between admission and gender ($p < 0.001$), and various measures for the strength of association. Expressed in terms of the ***odds ratio***, males were apparently 1.84 times as likely to be admitted as females, with 99% confidence bounds $(1.56, 2.17)$. Each of these numbers expresses some part of the relationship between gender and admission in the Berkeley data. Numerical summaries such as these are each designed to compress the information in the data, focusing on some particular feature.

In contrast, the visualization approach to data analysis is designed to (a) expose information and structure in the data, (b) supplement the information available from numerical summaries, and (c) suggest more adequate models. In general, the visualization approach seeks to serve the needs of both summarization and exposure.

This approach recognizes that both data analysis and graphing are *iterative* processes. You should not expect that any one model captures all features of the data, any more than we should expect that a single graph shows all that may be seen. In most cases, your initial steps should include some graphical display guided by understanding of the subject matter of the data. What you learn from a graph may then help suggest features of the data to be incorporated into a fitted model. Your desire to ensure that the fitted model is an adequate summary may then lead to additional graphs.

The precept here is that

Visualization = Graphing + Fitting + Graphing . . .

where the ellipsis indicates the often iterative nature of this process. Even for descriptive purposes, an initial fit of salient features can be removed from the data, giving residuals (departures from a model). Displaying the residuals may then suggest additional features to account for.

Simple examples of this idea include detrending time series graphs to remove overall and seasonal effects and plots of residuals from main-effect models for ANOVA designs. For categorical data, mosaic plots (Chapter 5) display the unaccounted-for association between variables by shading, as in Figure 1.10. Additional models and plots considered in Section 10.2 can reveal additional structure in square tables beyond the obvious effect that sons tend most often to follow in their fathers' footsteps.

{ex:donner0a}

EXAMPLE 1.9: Donner Party

The graphs in Figure 1.3 and Figure 1.4 suggest three different initial descriptions for survival in the Donner party. Yet they ignore all other influences, of which gender and family structure might also be important. A more complete understanding of this data can be achieved by taking these effects into account, both in fitted models and graphs. See Example 7.9 for a continuation of this story. △

{ex:nasa}

EXAMPLE 1.10: Space shuttle disaster

The space shuttle *Challenger* mentioned in Example 1.2 exploded 73 seconds after take-off on January 28, 1986. Subsequent investigation presented to the presidential commission headed by William Rogers determined that the cause was failure of the O-ring seals used to isolate the fuel supply from burning gases. The story behind the *Challenger* disaster is perhaps the most poignant missed opportunity in the history of statistical graphics. See Tufte (1997) for a complete exposition. It may be heartbreaking to find out that some important information was there, but the graphmaker missed it.

Engineers from Morton Thiokol, manufacturers of the rocket motors, had been worried about the effects of unseasonably cold weather on the O-ring seals and recommended aborting the flight.

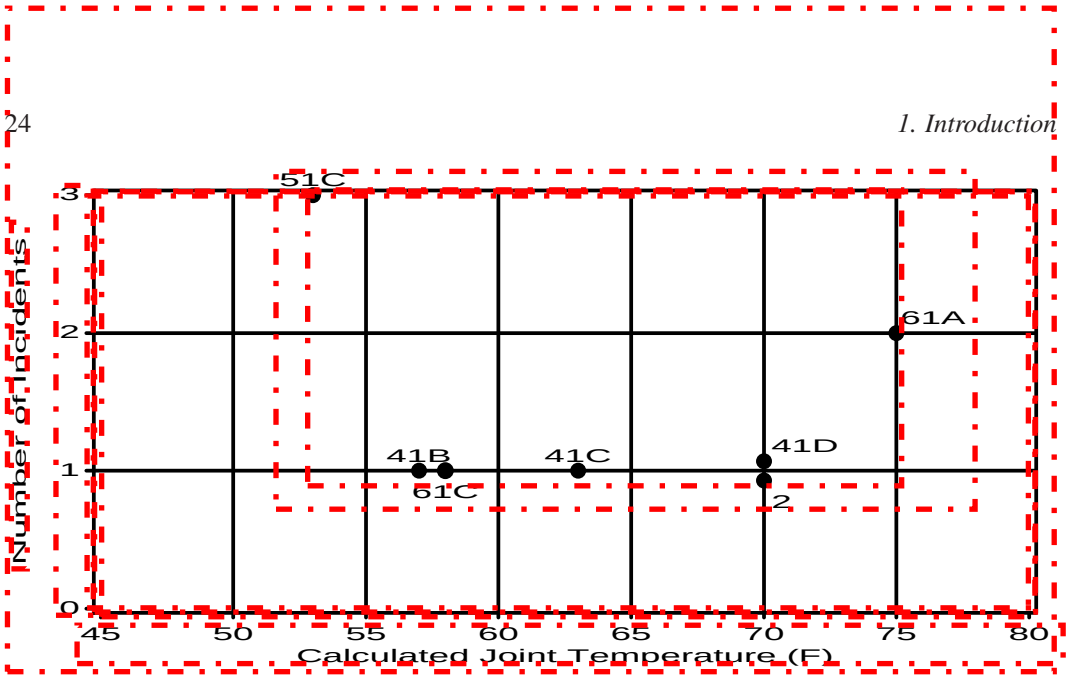


Figure 1.13: NASA Space Shuttle pre-launch graph prepared by the engineers at Morton Thiokol. {fig:nasa0}

NASA staff analysed the data, tables, and charts submitted by the engineers and concluded that there was insufficient evidence to cancel the flight.

The data relating O-ring failures to temperature were depicted as in Figure 1.13, our candidate for the most misleading graph in history. There had been 23 previous launches of these rockets giving data on the number of O-rings (out of 6) that were seen to have suffered some damage or failure. However, the engineers omitted the observations where no O-rings failed or showed signs of damage, believing that they were uninformative.

Examination of this graph seemed to indicate that there was no relation between ambient temperature and failure. Thus, the decision to launch the *Challenger* was made, in spite of the initial concerns of the Morton Thiokol engineers. Unfortunately, those observations had occurred when the launch temperature was relatively warm ($65 - -80^\circ$F.) and were indeed informative. The coldest temperature at any previous launch was 53°; when *Challenger* was launched on January 28, the temperature was a frigid 31°.

These data have been analyzed extensively (Dalal et al., 1989, Lavine, 1991). Tufte (1997) gives a thorough and convincing visual analysis of the evidence available prior to the launch. We consider statistical analysis of these data in Chapter 7, Example 7.4.

But, what if the engineers had simply made a better graph? At the very least, that would entail (a) drawing a smoothed curve to fit the points (to show the trend), and (b) removing the background grid lines (which obscure the data). Figure 1.14 shows a revised version of the same graph, highlighting the non-zero observations and adding a simple quadratic curve to allow for a possible non-linear relationship. For comparison, the excluded zero observations are also shown in grey. This plot, even showing only the non-zero points, should have caused any engineer to conclude that either: (a) the data were wrong, or (b) there were excessive risks associated with both high and low temperatures. But it is well-known that brittleness of the rubber used in the O-rings is inversely proportional to temperature cubed, so prudent interest might have focussed on the first possibility.[9]

[9] A coda to this story shows the role of visual explanation in practice as well (Tufte, 1997, pp. 50–53). The Rogers Commission contracted the reknowned theoretical physicist Richard Feynman to contribute to their investigation. He determined that the most probable cause of the shuttle failure was the lack of resiliancy of the rubber O-rings at low temperature. But how could he make this point convincingly? At a televised public hearing, he took a piece of the O-ring material, squeezed it in a C-clamp, and plunged it into a glass of ice water. After a few minutes, he released the clamp, and the rubber did not spring back to shape. He mildly said, "... there is no resilience in this particular material when it is at a temperature of 32 degrees. I believe this has some significance for our problem" (Feynman, 1988).

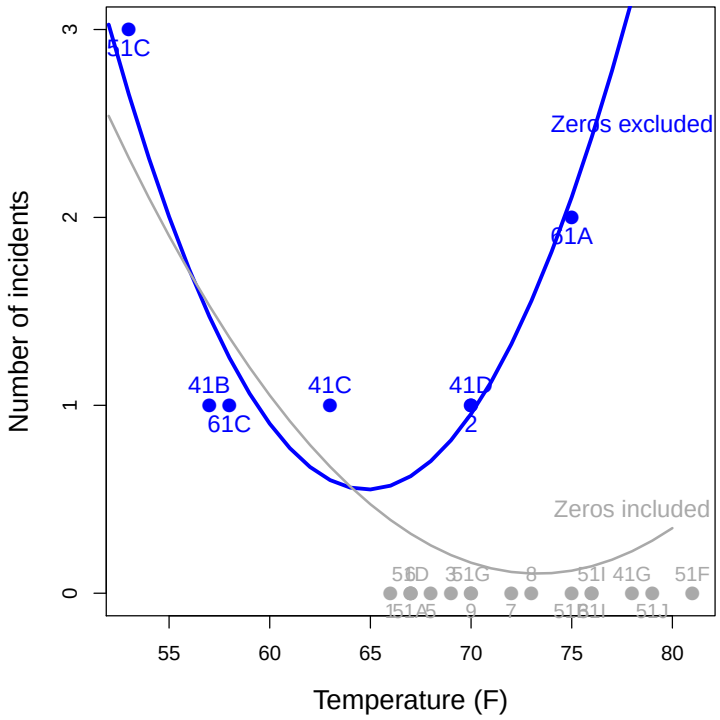


Figure 1.14: Re-drawn version of the NASA pre-launch graph, showing the locations of the excluded observations and with fitted quadratics for both sets of observations.

{fig:nasa}

△

1.4.6 The 80–20 rule

The Italian economist Vilfredo Pareto observed in 1906 that 80% of the land in Italy was owned by 20% of the population and this ratio also applied in other countries. It also applied to the yield of peas from peapods in his garden (Pareto, 1971). This idea became known as the ***Pareto principle*** or the ***80–20 rule***. The particular 80/20 ratio is not as important as the more general idea of the uneven distribution of results and causes in a variety of areas.

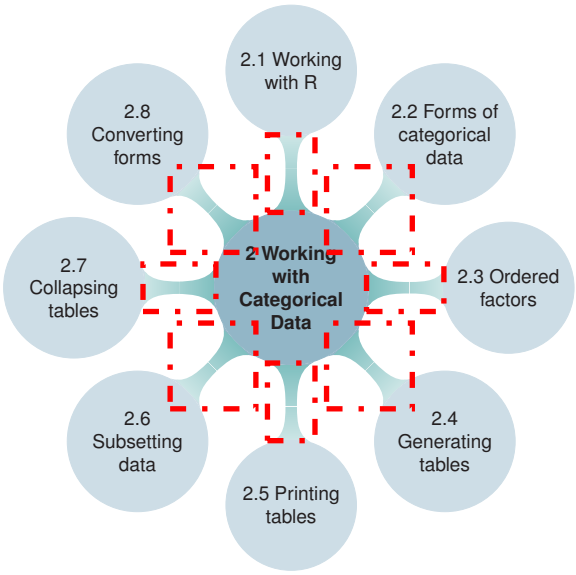
Common applications are the rules of thumb that: (a) in business 80% of sales come from 20% of clients; (b) in criminology 80% of crimes are said to be committed by 20% of the population. (c) In software development, it is said that 80% of errors and (d) crashes can be eliminated by fixing the top 20% most-reported bugs or that 80% of errors reside in 20% of the code.

The ***Pareto chart*** was designed to display the frequency distribution of a variable with a histogram or bar chart together with a cumulative line graph to highlight the most frequent category, and the ***Pareto distribution*** gives a mathematical form to such distributions with a parameter α (the *Pareto index*) reflecting the degree of inequality.

Applied to statistical graphics, the precept is that

20% of your effort can generate 80% of your desired result in producing a given plot.

This is good news for exploratory graphs you produce for yourself. Very often, the default settings will give a reasonable result, or you will see immediately something simple to add or change to make the plot easier to understand.



2

Working with Categorical Data

{ch:working}

Creating and manipulating categorical data sets requires some skills and techniques in R beyond those ordinarily used for quantitative data. This chapter illustrates these for the main formats for categorical data: case form, frequency form and table form.

> I'm a tidy sort of bloke. I don't like chaos. I kept records in the record rack, tea in the tea caddy, and pot in the pot box
>
> George Harrison, from
> `http://www.brainyquote.com/quotes/keywords/tidy.html`

Categorical data can be represented as data sets in various formats: case form, frequency form, and table form. This chapter describes and illustrates the skills and techniques in R needed to input, create, and manipulate R data objects to represent categorical data. More importantly, you also need to be able to convert these from one form to another for the purposes of statistical analysis and visualization, which are the subject of the remainder of the book.

As mentioned earlier, this book assumes that you have at least a basic knowledge of the R language and environment, including interacting with the R console (Rgui for Windows, R.app for Mac OS X) or some other editor/environment (e.g., R Studio), loading and using R functions in packages (e.g., `library(vcd)`) getting help for these from R (e.g., `help(matrix)`), etc. This chapter is therefore devoted to covering those topics needed in the book beyond such basic skills.[1]

[1] Some excellent introductory treatments of R are: Fox and Weisberg (2011, Chapter 2), Maindonald and Braun (2007), and Dalgaard (2008). Tom Short's *R Reference Card*, `http://cran.us.r-project.org/doc/contrib/Short-refcard.pdf`, is a handy 4-page summary of the main functions. The web sites Quick-R `http://www.statmethods.net/` and Cookbook for R `http://www.cookbook-r.com/` provide very helpful examples, organized by topics and tasks.

2.8.4 Publishing tables to LaTeX or HTML

OK, you've read your data into R, done some analysis, and now want to include some tables in a LaTeX document or in a web page in HTML format. Formatting tables for these purposes is often tedious and error-prone.

There are a great many packages in R that provide for nicely formatted, publishable tables for a wide variety of purposes; indeed, most of the tables in this book are generated using these tools. See Leifeld (2013) for a description of the texreg (Leifeld, 2014) package and a comparison with some of the other packages.

Here, we simply illustrate the xtable (Dahl, 2014) package, which, along with capabilities for statistical model summaries, time-series data, and so forth, has a `xtable.table` method for one-way and two-way table objects.

The *HorseKicks* data is a small one-way frequency table described in Example 3.4 and contains the frequencies of 0, 1, 2, 3, 4 deaths per corps-year by horse-kick among soldiers in 20 corps in the Prussian army.

```
> data("HorseKicks", package = "vcd")
> HorseKicks

nDeaths
  0   1   2   3   4
109  65  22   3   1
```

By default, `xtable()` formats this in LaTeX as a vertical table, and prints the LaTeX markup to the R console. This output is shown below.

```
> library(xtable)
> xtable(HorseKicks)

% latex table generated in R 3.2.1 by xtable 1.7-4 package
% Thu Aug 20 15:33:57 2015
\begin{table}[ht]
\centering
\begin{tabular}{rr}
  \hline
 & nDeaths \\
  \hline
0 & 109 \\
  1 &  65 \\
  2 &  22 \\
  3 &   3 \\
  4 &   1 \\
   \hline
\end{tabular}
\end{table}
```

When this is rendered in a LaTeX document, the result of `xtable()` appears as shown in the table below.

```
> xtable(HorseKicks)
```

The table above isn't quite right, because the column label "nDeaths" belongs to the first column, and the second column should be labeled "Freq." To correct that, we convert the *HorseKicks* table to a data frame (see Section 2.8 for details), add the appropriate `colnames`, and use the `print.xtable` method to supply some other options.

	nDeaths
0	109
1	65
2	22
3	3
4	1

```
> tab <- as.data.frame(HorseKicks)
> colnames(tab) <- c("nDeaths", "Freq")
> print(xtable(tab), include.rownames = FALSE,
+       include.colnames = TRUE)
```

nDeaths	Freq
0	109
1	65
2	22
3	3
4	1

There are many more options to control the LaTeX details and polish the appearance of the table; see `help(xtable)` and `vignette("xtableGallery", package = "xtable")`.

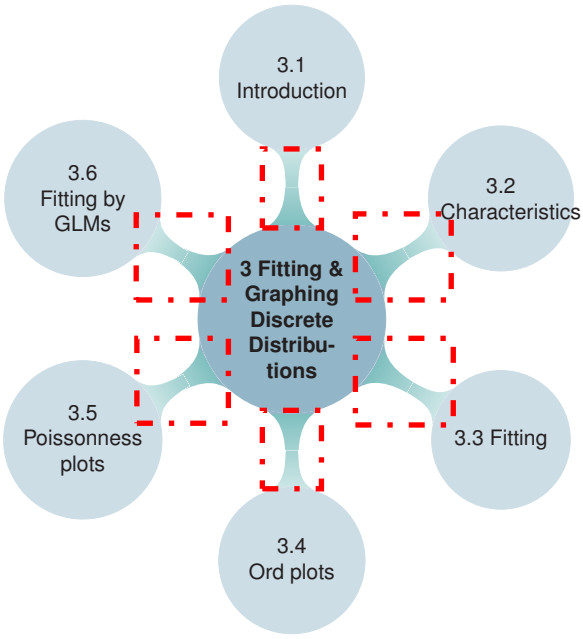
Finally, in Chapter 3, we display a number of similar one-way frequency tables in a transposed form to save display space. Table 3.3 is the finished version we show there. The code below uses the following techniques: (a) `addmargins()` is used to show the sum of all the frequency values; (b) `t()` transposes the data frame to have 2 rows; (c) `rownames()` assigns the labels we want for the rows; (d) using the `caption` argument provides a table caption, and a numbered table in LaTeX; (d) column alignment (`"r"` or `"l"`) for the table columns is computed as a character string used for the `align` argument.

```
> horsetab <- t(as.data.frame(addmargins(HorseKicks)))
> rownames(horsetab) <- c( "Number of deaths", "Frequency" )
> horsetab <- xtable(horsetab, digits = 0,
+       caption = "von Bortkiewicz's data on deaths by horse kicks",
+       align = paste0("l|", paste(rep("r", ncol(horsetab)),
+                                  collapse = ""))
+       )
> print(horsetab, include.colnames = FALSE, caption.placement="top")
```

Table 2.2: von Bortkiewicz's data on deaths by horse kicks

Number of deaths	0	1	2	3	4	Sum
Frequency	109	65	22	3	1	200

For use in a web page, blog, or Word document, you can use `type="HTML"` in the call to `print()` for "xtable" objects.



3

Fitting and Graphing Discrete Distributions

{ch:discrete}

Discrete data often follow various theoretical probability models. Graphic displays are used to visualize goodness of fit, to diagnose an appropriate model, and determine the impact of individual observations on estimated parameters.

> Not everything that counts can be counted, and not everything that can be counted counts.
>
> Albert Einstein

Discrete frequency distributions often involve counts of occurrences of events, such as accident fatalities, incidents of terrorism or suicide, words in passages of text, or blood cells with some characteristic. Often interest is focused on how closely such data follow a particular probability distribution, such as the binomial, Poisson, or geometric distribution, which provide the basis for generating mechanisms that might give rise to the data. Understanding and visualizing such distributions in the simplest case of an unstructured sample provides a building block for generalized linear models (Chapter 11) where they serve as one component. They also provide the basis for a variety of recent extensions of regression models for count data (Chapter 11), some of which account for the excess counts of zeros (zero-inflated models) caused by left- or right-truncation often encountered in statistical practice.

This chapter describes the well-known discrete frequency distributions: the binomial, Poisson, negative binomial, geometric, and logarithmic series distributions in the simplest case of an unstructured sample. The chapter begins with simple graphical displays (line graphs and bar charts) to view the distributions of empirical data and theoretical frequencies from a specified discrete distribution.

The chapter then describes methods for fitting data to a distribution of a given form, and presents simple but effective graphical methods that can be used to visualize goodness of fit, to diagnose an appropriate model (e.g., does a given data set follow the Poisson or negative binomial?) and to determine the impact of individual observations on estimated parameters.

3.1 Introduction to discrete distributions

{sec:discrete-intro}

Discrete data analysis is concerned with the study of the tabulation of one or more types of events, often categorized into mutually exclusive and exhaustive categories. ***Binary events*** having two outcome categories include the toss of a coin (head/tails), sex of a child (male/female), survival of a patient following surgery (lived/died), and so forth. ***Polytomous events*** have more outcome categories, which may be *ordered* (rating of impairment: low/medium/high, by a physician) and possibly numerically-valued (number of dots (pips), 1–6 on the toss of a die) or *unordered* (political party supported: Liberal, Conservative, Greens, Socialist).

In this chapter, we focus largely on one-way frequency tables for a single numerically-valued variable. Probability models for such data provide the opportunity to describe or explain the *structure* in such data, in that they entail some data-generating mechanism and provide the basis for testing scientific hypotheses and predicting future results. If a given probability model does not fit the data, this can often be a further opportunity to extend understanding of the data, or the underlying substantive theory, or both.

The remainder of this section gives a few substantive examples of situations where the well-known discrete frequency distributions (binomial, Poisson, negative binomial, geometric, and logarithmic series) might reasonably apply, at least approximately. The mathematical characteristics and properties of these theoretical distributions are postponed to Section 3.2.

In many cases, the data at hand pertain to two types of variables in a one-way frequency table. There is a basic outcome variable, k, taking integer values, $k = 0, 1, \ldots$, and called a ***count***. For each value of k, we also have a ***frequency***, n_k that the count k was observed in some sample. For example, in the study of children in families, the count variable k could be the total number of children or the number of male children; the frequency variable, n_k, would then give the number of families with that basic count k.

3.1.1 Binomial data

{sec:binom-data}

Binomial-type data arise as the discrete distribution of the number of "success" events in n independent binary trials, each of which yields a success (yes/no, head/tail, lives/dies, male/female) with a constant probability p.

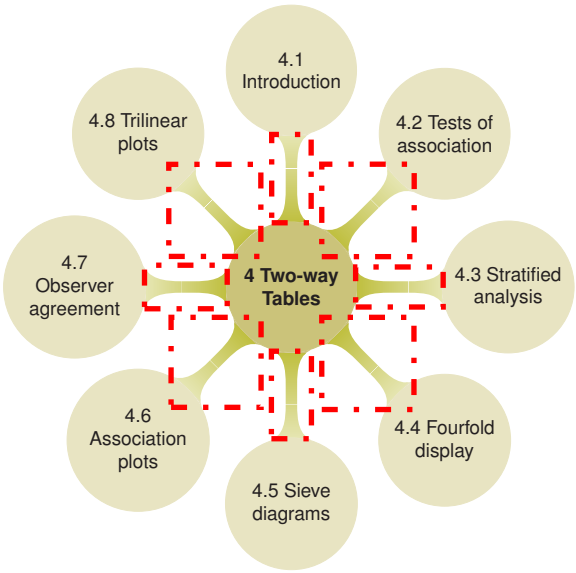
Sometimes, as in Example 3.1 below, the available data record only the number of successes in n trials, with separate such observations recorded over time or space. More commonly, as in Example 3.2 and Example 3.3, we have available data on the frequency n_k of $k = 0, 1, 2, \ldots n$ successes in the n trials.

{ex:arbuthnot1}

EXAMPLE 3.1: Arbuthnot data

Sex ratios, such as births of male to female children, have long been of interest in population studies and demography. Indeed, in 1710, John Arbuthnot (Arbuthnot, 1710) used data on the ratios of male to female christenings in London from 1629–1710 to carry out the first known significance test. The data for these 82 years showed that in *every* year there were more boys than girls. He calculated that, under the assumption that male and female births were equally likely, the probability of 82 years of more males than females was vanishingly small, ($\Pr \approx 4.14 \times 10^{-25}$). He used this to argue that a nearly constant birth ratio > 1 (or $\Pr(\text{Male}) > 0.5$) could be interpreted to show the guiding hand of a divine being.

Arbuthnot's data, along with some other related variables, are available in `Arbuthnot` in the



4

Two-Way Contingency Tables

{ch:twoway}

The analysis of two-way frequency tables concerns the association between two variables. A variety of specialized graphical displays help us to visualize the pattern of association, using area of some region to represent the frequency in a cell. Some of these methods are focused on visualizing an odds ratio (for 2×2 tables), or the general pattern of association, or the agreement between row and column categories in square tables.

4.1 Introduction

{sec:twoway-intro}

> Tables are like cobwebs, like the sieve of Danaides; beautifully reticulated, orderly to look upon, but which will hold no conclusion. Tables are abstractions, and the object a most concrete one, so difficult to read the essence of.
>
> From *Chartism* by Thomas Carlyle (1840), Chapter II, Statistics

Most methods of statistical analysis are concerned with understanding relationships or dependence among variables. With categorical variables, these relationships are often studied from data that has been summarized by a ***contingency table*** in table form or frequency form, giving the frequencies of observations cross-classified by two or more such variables. As Thomas Carlyle said, it is often difficult to appreciate the message conveyed in numerical tables.

This chapter is concerned with simple graphical methods for understanding the association between two categorical variables. Some examples are also presented that involve a third, ***stratifying variable***, where we wish to determine if the relationship between two primary variables is the same or different for all levels of the stratifying variable. More general methods for fitting models and displaying associations for three-way and larger tables are described in Chapter 5.

In Section 4.2, we describe briefly some numerical and statistical methods for testing whether an association exists between two variables, and measures for quantifying the strength of this association. In Section 4.3 we extend these ideas to situations where the relation between two variables is of primary interest, but there are one or more background variables to be controlled.

The main emphasis, however, is on graphical methods that help to describe the *pattern* of an association between variables. Section 4.4 presents the fourfold display, designed to portray the odds ratio in 2×2 tables or a set of k such tables. ***Sieve diagrams*** (Section 4.5) and ***association plot***s (Section 4.6) are more general methods for depicting the pattern of associations in any two-way table. When the row and column variables represent the classifications of different raters, specialized measures and visual displays for ***inter-rater agreement*** (Section 4.7) are particularly useful. Another specialized display, a ***trilinear plot*** or ***ternary plot***, described in Section 4.8, is designed for three-column frequency tables or compositional data. In order to make clear some of the distinctions that occur in contingency table analysis, we begin with several examples. {ex:berkeley1}

EXAMPLE 4.1: Berkeley admissions

Table 4.1 shows aggregate data on applicants to graduate school at Berkeley for the six largest departments in 1973 classified by admission and gender (Bickel *et al.*, 1975). See `UCBAdmissions` (in package datasets) for the complete data set. For such data we might wish to study whether there is an association between admission and gender. Are male (or female) applicants more likely to be admitted? The presence of an association might be considered as evidence of sex bias in admission practices.

Table 4.1 is an example of the simplest kind of contingency table, a 2×2 classification of individuals according to two dichotomous (binary) variables. For such a table, the question of whether there is an association between admission and gender is equivalent to asking if the proportions of males and females who are admitted to graduate school are different, or whether the difference in proportions admitted is not zero. △

{tab:berk22}

Table 4.1: Admissions to Berkeley graduate programs

	Admitted	Rejected	Total	% Admit
Males	1198	1493	2691	44.52
Females	557	1278	1835	30.35
Total	1755	2771	4526	38.78

Although the methods for quantifying association in larger tables can be used for 2×2 tables, there are specialized measures (described in Section 4.2) and graphical methods for these simpler tables.

As we mentioned in Section 1.2.4 it is often useful to make a distinction between ***response***, or outcome variables, on the one hand, and possible ***explanatory*** or predictor variables on the other. In Table 4.1, it is natural to consider `admission` as the outcome, and `gender` as the explanatory variable. In other tables, no variable may be clearly identified as *the* outcome, or there may be several response variables, giving a multivariate problem. {ex:haireye1}

EXAMPLE 4.2: Hair color and eye color

Table 4.2 shows data collected by Snee (1974) on the relation between hair color and eye color among 592 students in a statistics course (a two-way margin of `HairEyeColor`).

Neither hair color nor eye color is considered a response in relation to the other; our interest concerns whether an association exists between them. Hair color and eye color have both been classified into four categories. Although the categories used are among the most common, they are

```
>
> AES <- aes(x = women, y = men, z = crew, colour = side, shape = side,
+            label = id)
> ggtern(data = Lifeboats, mapping = AES) +
+       geom_text() +
+       theme_rgbw() +
+       geom_smooth(method = "lm", alpha = 0.2)
```

```
Removing Layer 2. 'smooth' is not an approved proto (for ternary plots) under the present ggter
```

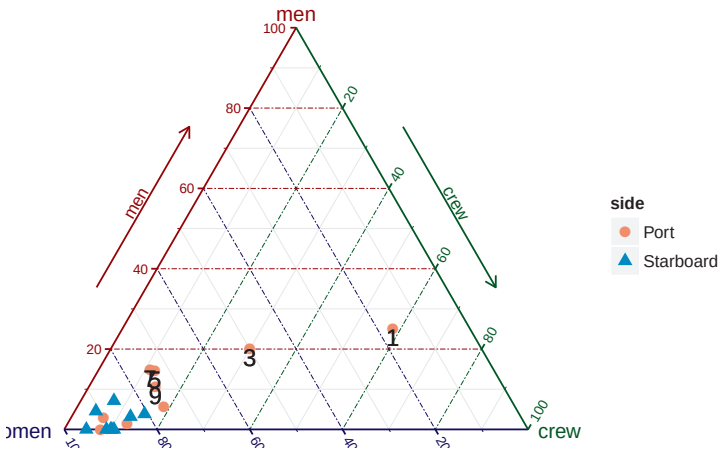


Figure 4.22: Lifeboats on the *Titanic*, showing the composition of each boat. Boats with more than 10% male passengers are labeled.

{fig:lifeboats1}

The resulting plot in Figure 4.22 (for which some more cosmetic parameters than shown in the code above have been used) makes it immediately apparent that many of the boats launched from the port side differ substantially from the starboard boats, whose passengers were almost entirely women and children. Boat 1 had only 20% (2 out of 10) women and children, while the percentage for boat 3 was only 50% (25 out of 50). We highlight the difference in composition of the boats launched from the two sides by adding seperate linear regression lines for the relation `men ~ women`.

The trilinear plot scales the numbers for each observation to sum to 1.0, so differences in the total number of people on each boat cannot be seen in Figure 4.22. The `total` number reported loaded is plotted against `launch` time in Figure 4.23, with a separate regression line and loess smooth fit to the data for the port and starboard sides (code again simplified for clarity):

```
> AES <- aes(x = launch, y = total, colour = side, label = boat)
> ggplot(data = Lifeboats, mapping = AES) +
+       geom_text() +
+       geom_smooth(method = "lm", aes(fill = side), size = 1.5) +
+       geom_smooth(method = "loess", aes(fill = side), se = FALSE,
+                   size = 1.2)
```

From the linear regression lines in Figure 4.23, it seems that the rescue effort began in panic on the port side, with relatively small numbers loaded, and (from Figure 4.22), small proportions of women and children. But the loading regime on that side improved steadily over time. The procedures began more efficiently on the starboard side but the numbers loaded increased only slightly. The smoothed loess curves indicate that over time, for each side, there was still a large variability from boat to boat.

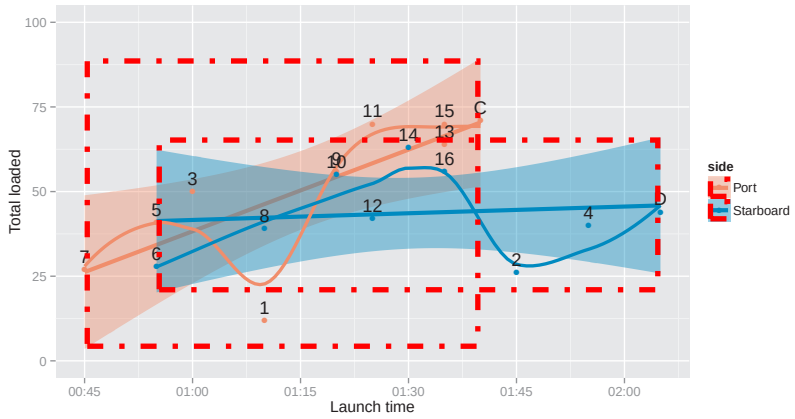


Figure 4.23: Number of people loaded on lifeboats on the Titanic vs. time of launch, by side of boat. The plot annotations show the linear regression and loess smooth.

{fig:lifeboats2}

△

4.9 Chapter summary

{sec:twoway-summary}

- A contingency table gives the frequencies of observations cross-classified by two or more categorical variables. With such data we are typically interested in testing whether associations exist, quantifying the strength of association, and understanding the nature of the association among these variables.

- For 2×2 tables, association is easily summarized in terms of the odds ratio or its logarithm. This measure can be extended to stratified $2 \times 2 \times k$ tables, where we can also assess whether the odds ratios are equal across strata or how they vary.

- For $R \times C$ tables, measures and tests of general association between two categorical variables are most typically carried out using the Pearson's chi-squared or likelihood-ratio tests provided by `assocstats()`. Stratified tests controlling for one or more background variables, and tests for ordinal categories, are provided by the Cochran–Mantel–Haenszel tests given by `CMHtest()`.

- For 2×2 tables, the fourfold display provides a visualization of the association between variables in terms of the odds ratio. Confidence rings provide a visual test of whether the odds ratio differs significantly from 1. Stratified plots for $2 \times 2 \times k$ tables are also provided by `fourfold()`.

- Sieve diagrams and association plots provide other useful displays of the pattern of association in $R \times C$ tables. These also extend to higher-way tables as part of the strucplot framework.

- When the row and column variables represent different observers rating the same subjects, interest is focused on agreement rather than mere association. Cohen's κ is one measure of strength of agreement. The observer agreement chart provides a visual display of how the observers agree and disagree.

- Another specialized display, the trilinear plot, is useful for three-column frequency tables or compositional data.

4.10 Lab exercises

{lab:4.1}
{sec:twoway-lab}

Exercise 4.1 The data set `fat`, created below, gives a 2×2 table recording the level of cholesterol in diet and the presence of symptoms of heart disease for a sample of 23 people.

```
> fat <- matrix(c(6, 4, 2, 11), 2, 2)
> dimnames(fat) <- list(diet = c("LoChol", "HiChol"),
+                       disease = c("No", "Yes"))
```

(a) Use `chisq.test(fat)` to test for association between diet and disease. Is there any indication that this test may not be appropriate here?
(b) Use a fourfold display to test this association visually. Experiment with the different options for standardizing the margins, using the `margin` argument to `fourfold()`. What evidence is shown in different displays regarding whether the odds ratio differs significantly from 1?
(c) `oddsratio(fat, log = FALSE)` will give you a numerical answer. How does this compare to your visual impression from fourfold displays?
(d) With such a small sample, Fisher's exact test may be more reliable for statistical inference. Use `fisher.test(fat)`, and compare these results to what you have observed before.
(e) Write a one-paragraph summary of your findings and conclusions for this data set.

{lab:4.2}

Exercise 4.2 The data set *Abortion* in vcdExtra gives a $2 \times 2 \times 2$ table of opinions regarding abortion in relation to sex and status of the respondent. This table has the following structure:

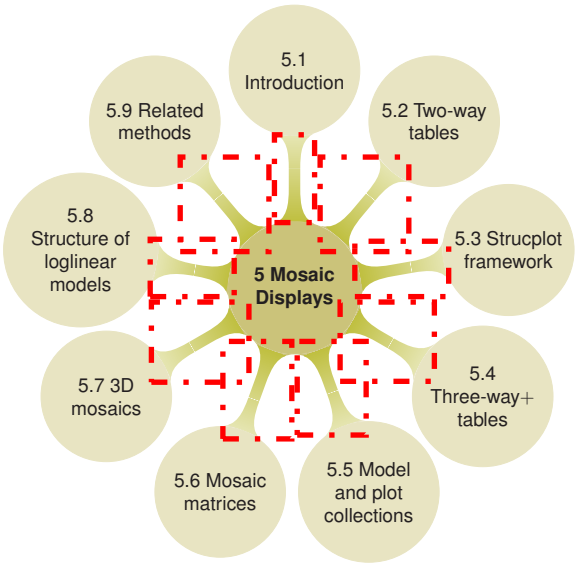
```
> data("Abortion", package = "vcdExtra")
> str(Abortion)

 table [1:2, 1:2, 1:2] 171 152 138 167 79 148 112 133
 - attr(*, "dimnames")=List of 3
  ..$ Sex             : chr [1:2] "Female" "Male"
  ..$ Status          : chr [1:2] "Lo" "Hi"
  ..$ Support_Abortion: chr [1:2] "Yes" "No"
```

(a) Taking support for abortion as the outcome variable, produce fourfold displays showing the association with sex, stratified by status.
(b) Do the same for the association of support for abortion with status, stratified by sex.
(c) For each of the problems above, use `oddsratio()` to calculate the numerical values of the odds ratio, as stratified in the question.
(d) Write a brief summary of how support for abortion depends on sex and status.

{lab:4.3}

Exercise 4.3 The *JobSat* table on income and job satisfaction created in Example 2.5 is contained in the vcdExtra package.

(a) Carry out a standard χ^2 test for association between income and job satisfaction. Is there any indication that this test might not be appropriate? Repeat this test using `simulate.p.value = TRUE` to obtain a Monte Carlo test that does not depend on large sample size. Does this change your conclusion?
(b) Both variables are ordinal, so CMH tests may be more powerful here. Carry out that analysis. What do you conclude?



5

Mosaic Displays for n-way Tables

{ch:mosaic}

Mosaic displays help to visualize the pattern of associations among variables in two-way and larger tables. Extensions of this technique can reveal partial associations, marginal associations, and shed light on the structure of loglinear models themselves.

5.1 Introduction

{sec:mosaic-intro}

> Little boxes on the hillside, Little boxes made of ticky tacky,
> Little boxes on the hillside, Little boxes all the same.
> There's a green one and a pink one And a blue one and a yellow one,
> And they're all made out of ticky tacky, And they all look just the same.
>
> Words and music by Malvina Reynolds, ©Schroder Music Company 1962, 1990;
> recorded by Pete Seeger

In Chapter 4, we described a variety of graphical techniques for visualizing the pattern of association in simple contingency tables. These methods are somewhat specialized for particular sizes and shapes of tables: 2×2 tables (fourfold display), $R \times C$ tables (tile plot, sieve diagram), square tables (agreement charts), $R \times 3$ tables (trilinear plots), and so forth.

This chapter describes the ***mosaic display*** and related graphical methods for n-way frequency tables, designed to show various aspects of high-dimensional contingency tables in a hierarchical way. These methods portray the frequencies in an n-way contingency table by a collection of rectangular "tiles" whose size (area) is proportional to the cell frequency. In this respect, the mosaic

display is similar to the sieve diagram (Section 4.5). However, mosaic plots and related methods described here:

- generalize more readily to *n*-way tables. One can usefully examine 3-way, 4-way, and even larger tables, subject to the limitations of resolution in any graph;
- are intimately connected to loglinear models, generalized linear models, and generalized non-linear models for frequency data;
- provide a method for fitting a series of sequential loglinear models to the various marginal totals of an *n*-way table; and
- can be used to illustrate the relations among variables that are fitted by various loglinear models.

The basic ideas behind these graphical methods are explained for two-way tables in Section 5.2; the ***strucplot framework*** on which these are based is described in Section 5.3. The graphical extension of mosaic plots to three-way and large tables (Section 5.4) is quite direct. However, the details require a brief introduction to loglinear models and some terminology for different types of "independence" in such tables, also described in this section. Mosaic methods are further extended to all-pairwise plots in Section 5.6 and 3D plots in Section 5.7.

5.2 Two-way tables

{sec:mosaic-twoway}

The mosaic display (Friendly, 1992, 1994, 1997, Hartigan and Kleiner, 1981, 1984) is like a grouped barchart, where the heights (or widths) of the bars show the relative frequencies of one variable, and widths (heights) of the sections in each bar show the conditional frequencies of the second variable, given the first. This gives an area-proportional visualization of the frequencies composed of tiles corresponding to the cells created by successive vertical and horizontal splits of rectangle, representing the total frequency in the table. The construction of the mosaic display, and what it reveals, are most easily understood for two-way tables.

{ex:haireye2a}

EXAMPLE 5.1: Hair color and eye color

Consider the data shown earlier in Table 4.2, showing the relation between hair color and eye color among students in a statistics course. The basic mosaic display for this 4×4 table is shown in Figure 5.1.

```
> data("HairEyeColor", package = "datasets")
> haireye <- margin.table(HairEyeColor, 1 : 2)
> mosaic(haireye, labeling = labeling_values)
```

For such a two-way table, the mosaic in Figure 5.1 is constructed by first dividing a unit square in proportion to the marginal totals of one variable, say, hair color.

For these data, the marginal frequencies and proportions of hair color are calculated below:

```
> (hair <- margin.table(haireye, 1))

Hair
Black Brown   Red Blond
  108   286    71   127

> prop.table(hair)

Hair
  Black   Brown     Red   Blond
0.18243 0.48311 0.11993 0.21453
```

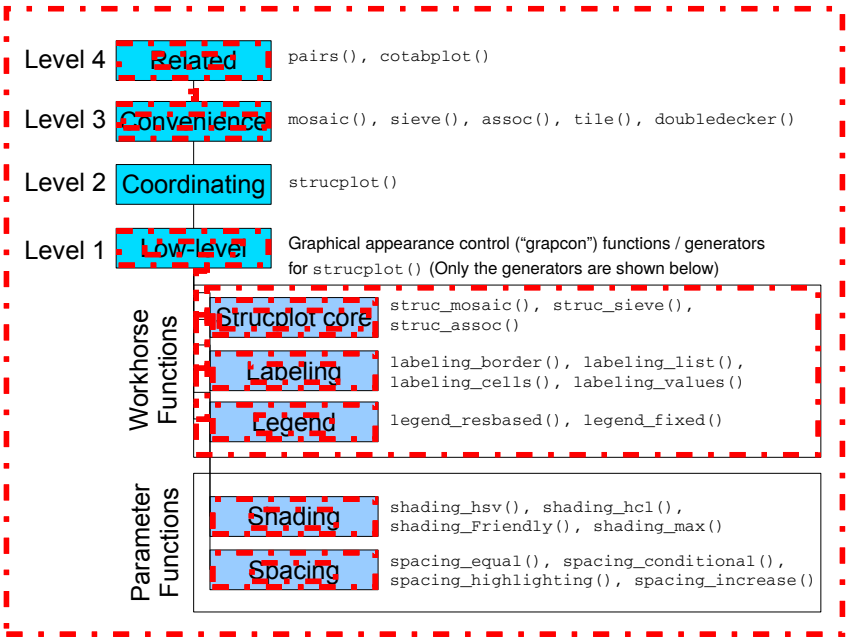


Figure 5.5: Components of the strucplot framework. High-level functions use those at lower levels to provide a general system for tile-based plots of frequency tables. {fig:struc}

functions at the top of the diagram, but it is more convenient to describe the framework from the bottom up.

1. On the lowest level, there are several groups of workhorse and parameter functions that directly or indirectly influence the final appearance of the plot (see Table 5.1 for an overview). These are examples of *graphical appearance control* functions (called ***grapcon functions***). They are created by generating functions (*grapcon generators*), allowing flexible parameterization and extensibility (Figure 5.5 only shows the generators). The generator names follow the naming convention `group_foo()`, where `group` reflects the group the generators belong to (strucplot core, labeling, legend, shading, or spacing).

   - The workhorse functions (created by `struc_foo()`) are `labeling_foo()`, and `legend_foo()`. These functions directly produce graphical output (i.e., "add ink to the canvas"), for labels and legends respectively.
   - The parameter functions (created by `spacing_foo()` and `shading_foo()`) compute graphical parameters used by the others. The grapcon functions returned by `struc_foo()` implement the core functionality, creating the tiles and their content.

2. On the second level of the framework, a suitable combination of the low-level grapcon functions (or, alternatively, corresponding generating functions) is passed as "hyperparameters" to `strucplot()`. This central function sets up the graphical layout using grid viewports, and coordinates the specified core, labeling, shading, and spacing functions to produce the plot.

3. On the third level, vcd provides several convenience functions such as `mosaic()`, `sieve()`, `assoc()`, `tile()`, and `doubledecker()` which interface to `strucplot()` through sensible parameter defaults and support for model formulae.

4. Finally, on the fourth level, there are "related" vcd functions (such as `cotabplot()` and the

Table 5.1: Available graphical appearance control (grapcon) generators in the strucplot framework {tab:grapcons}

Group	Grapcon generator	Description
strucplot core	`struc_assoc()`	core function for association plots
	`struc_mosaic()`	core function for mosaic plots (also used for tile plots)
	`struc_sieve()`	core function for sieve plots
labeling	`labeling_border()`	border labels
	`labeling_cboxed()`	centered labels with boxes, all labels clipped, and on top and left border
	`labeling_cells()`	cell labels
	`labeling_conditional()`	border labels for conditioning variables and cell labels for conditioned variables
	`labeling_doubledecker()`	draws labels for doubledecker plot
	`labeling_lboxed()`	left-aligned labels with boxes
	`labeling_left()`	left-aligned border labels
	`labeling_left2()`	left-aligned border labels, all labels on top and left border
	`labeling_list()`	draws a list of labels under the plot
	`labeling_residuals()`	show residuals in cells
	`labeling_value()`	show values (observed, expected) in cells
shading	`shading_binary()`	visualizes the sign of the residuals
	`shading_Friendly()`	implements Friendly shading (based on HSV colors)
	`shading_hcl()`	shading based on HCL colors
	`shading_hsv()`	shading based on HSV colors
	`shading_max()`	shading visualizing the maximum test statistic (based on HCL colors)
	`shading_sieve()`	implements Friendly shading customized for sieve plots (based on HCL colors)
spacing	`spacing_conditional()`	increasing spacing for conditioning variables, equal spacing for conditioned variables
	`spacing_dimequal()`	equal spacing for each dimension
	`spacing_equal()`	equal spacing for all dimensions
	`spacing_highlighting()`	increasing spacing, last dimension set to zero
	`spacing_increase()`	increasing spacing
legend	`legend_fixed()`	creates a fixed number of bins (similar to `mosaicplot()`)
	`legend_resbased()`	suitable for an arbitrary number of bins (also for continuous shadings)

`pairs()` methods for table objects) arranging collections of plots of the strucplot framework into more complex displays (e.g., by means of panel functions).

5.3.2 Shading schemes

{sec:mosaic-shading}

Unlike other graphics functions in base R, the strucplot framework allows almost full control over the graphical parameters of all plot elements. In particular, in association plots, mosaic plots, and sieve plots, you can modify the graphical appearance of each tile individually.

Built on top of this functionality, the framework supplies a set of shading functions choosing colors appropriate for the visualization of loglinear models. The tiles' graphical parameters are set using the `gp` argument of the functions of the strucplot framework. This argument basically expects an object of class "gpar" whose components are arrays of the same shape (length and dimensionality) as the data table.

For added generality, however, you can also supply a `grapcon` function that computes such an object given a vector of residuals, or, alternatively, a *generating function* that takes certain argu-

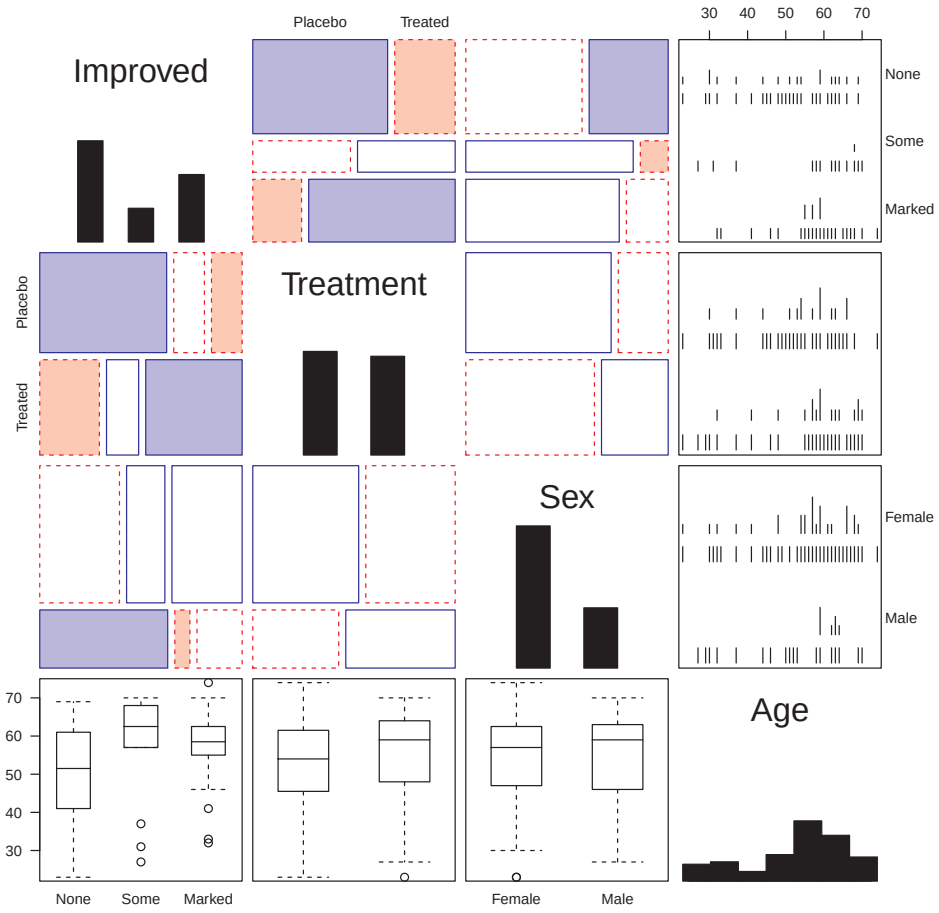


Figure 5.27: Generalized pairs plot of the Arthritis data. Combinations of categorical and quantitative variables can be rendered in various ways. {fig:arth-gpairs}

improvement. The other panels in the first row (and column) show that improvement is more likely in the treated condition and greater among women than men. △

5.7 3D mosaics

{sec:3D}

Mosaic-like displays use the idea of recursive partitioning of a unit square to portray the frequencies in an n-way table by the area of rectangular tiles with (x, y) coordinates. The same idea extends naturally to a 3D graphic. This starts with a unit cube, which is successively subdivided into 3D cuboids along (x, y, z) dimensions, and the frequency in a table cell is then represented by volume.

As in the 2D versions, each cuboid can be shaded to represent some other feature of the data, typically the residual from some model of independence. In principle, the display can accommodate more than 3 variables by using a sequence of split directions along the (x, y, z) axes.

One difficulty in implementing this method is that, short of using a 3D printer, the canvas for a 3D plot on a screen or printer is still projected on a two-dimensional surface, and graphical elements (volumes, lines, text) toward the front of the view will obscure those in the back. In R, a major advance in 3D graphics is available in the rgl (Adler and Murdoch, 2014) package, that mitigates

these problems by: (a) providing an interactive graphic window that can be zoomed and rotated manually with the mouse; (b) allowing dynamic graphics under program control, for example to animate a plot or make a movie; (c) providing control of the details of 3D rendering, including transparency of shapes, surface shading, lighting, and perspective.

The vcdExtra package implements 3D mosaics using rgl graphics. `mosaic3d()` provides methods for "loglm" as well as "table" (or "structable") objects. At the time of writing, only some features of 2D mosaics are available. {ex:bartlett-3d}

EXAMPLE 5.17: Bartlett data on plum-root cuttings

In Example 5.12 we showed the mosaic matrix for the *Bartlett*, fitting the model of mutual independence to show all associations among the table variables, `Alive`, `Time` of planting, and `Length` of cutting. Figure 5.28 shows the 3D version, produced using `mosaic3d()`:

```
> mosaic3d(Bartlett)
```

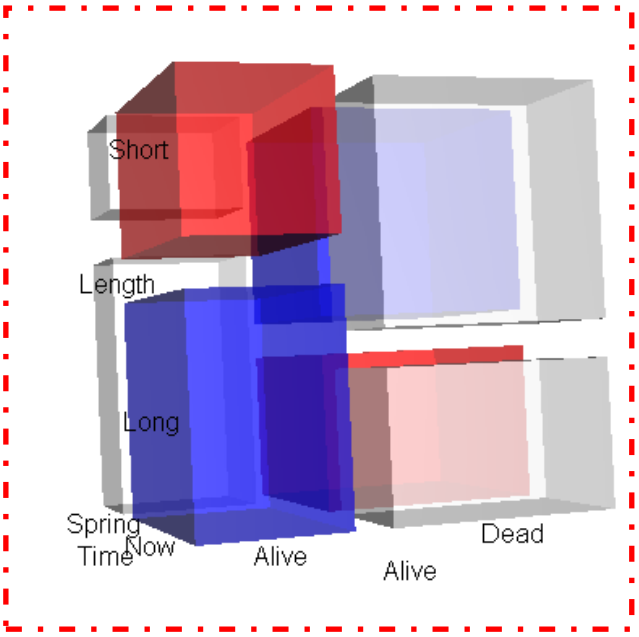


Figure 5.28: 3D mosaic plot of the Bartlett data, according to the model of mutual independence. {fig:mos3d-bartlett}

In the view of this figure, it can be seen that cuttings are more likely to be alive when planted Now and when cut Long. These relations can more easily be appreciated by rotating the 3D display. △

5.8 Visualizing the structure of loglinear models

{sec:mosaic-struc}

For quantitative response data, it is easy to visualize a fitted model—for linear regression, this is just a plot of the fitted line; for multiple regression or non-linear regression with two predictors, this is a plot of the fitted response surface. For a categorical response variable, an analog of such plots is provided by effect plots, described later in this book.

For contingency table data, mosaic displays can be used in a similar manner to illuminate the relations among variables in a contingency table represented in various loglinear models, a point

described by Theus and Lauer (1999). In fact, each of the model types depicted in Table 5.2 has a characteristic shape and structure in a mosaic display. This, in turn, leads to a clearer understanding of the structure that appears in real data when a given model fits, the relations among the models, and the use of mosaic displays. The essential idea is a simple extension of what we do for more traditional models: show the *expected* (fitted) frequencies under a given model rather than observed frequencies in a mosaic-like display.

To illustrate, we use some artificial data on the relations among age, sex, and symptoms of some disease shown in the $2 \times 2 \times 2$ table `struc` below.

```
> struc <- array(c(6, 10, 312, 44,
+                  37, 31, 192, 76),
+  dim = c(2, 2, 2),
+  dimnames = list(Age = c("Young", "Old"),
+                  Sex = c("F", "M"),
+                  Disease = c("No", "Yes"))
+  )
> struc <- as.table(struc)
> structable(struc)

              Sex   F   M
Age   Disease
Young No            6 312
      Yes          37 192
Old   No           10  44
      Yes          31  76
```

First, note that there are substantial associations in this table, as shown in Figure 5.29, fitting the (default) mutual independence model.

```
> mosaic(struc, shade = TRUE)
```

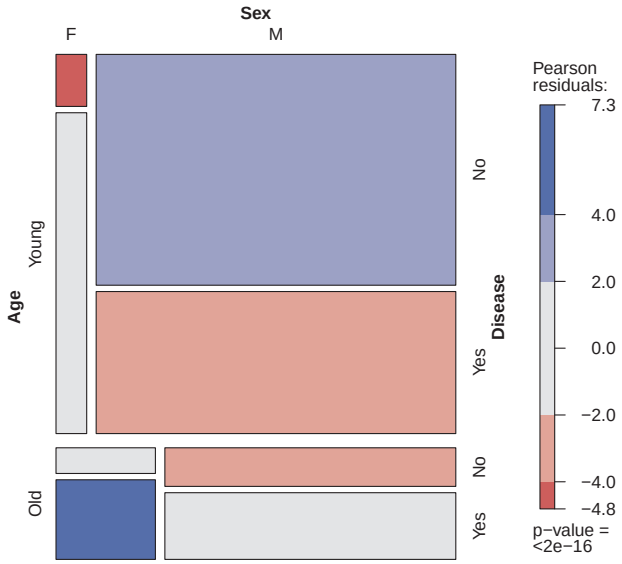


Figure 5.29: Mosaic display for the data on age, sex, and disease. Observed frequencies are shown in the plot, and residuals reflect departure from the model of mutual independence.

{fig:struc-mos1}

The first split by `Age` shows strong partial associations between `Sex` and `Disease` for both young and old. However, the residuals have an opposite pattern for young and old, suggesting a more complex relationship among these variables.

In this section we are asking a different question: what would mosaic displays look like if the data were in accord with simpler models? One way to do this is simply to use the expected frequencies to construct the tiles, as in sieve diagrams. The result, in Figure 5.30, shows that the tiles for sex and disease align for each of the age groups, but it is harder to see the relations among all three variables in this plot.

```
> mosaic(struc, type = "expected")
```

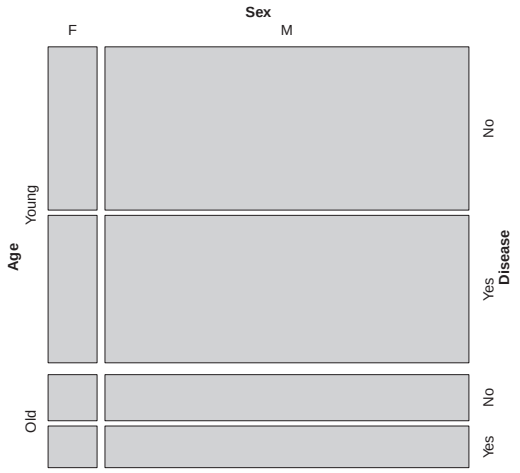


Figure 5.30: Mosaic display for the data on age, sex, and disease, using expected frequencies under mutual independence. {fig:struc-mos2}

We can visualize the model-implied relations among all variables together more easily using mosaic matrices.

5.8.1 Mutual independence

For example, to show the structure of a table that exactly fits the model of mutual independence, H_1, use the `loglm()` to find the fitted values, `fit`, as shown below. The function `fitted()` extracts these from the "loglm" object.

```
> mutual <- loglm(~ Age + Sex + Disease, data = struc, fitted = TRUE)
> fit <- as.table(fitted(mutual))
> structable(fit)

              Sex        F        M
Age   Disease
Young No           34.0991 253.3077
      Yes          30.7992 228.7940
Old   No           10.0365  74.5567
      Yes           9.0652  67.3416
```

These fitted frequencies then have the same one-way margins as the data in *`struc`*, but have no

two-way or higher associations. Then `pairs()` for this table, using `type="total"`, shows the three-way mosaic for each pair of variables, giving the result in Figure 5.30. We use `gp=shading_Friendly` to explicitly indicate the zero residuals in the display,

```
> pairs(fit, gp = shading_Friendly2, type = "total")
```

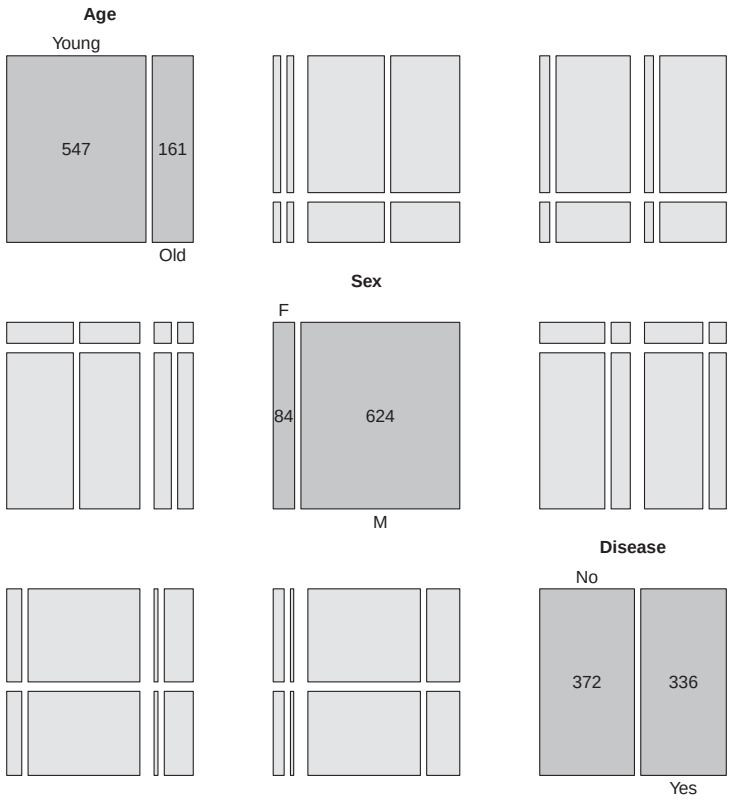


Figure 5.31: Mosaic matrix for fitted values under mutual independence. In all panels the joint frequencies conform to the one-way margins.

{fig:struc-mos3}

In this figure the same data are shown in all the off-diagonal panels and the mutual independence model was fitted in each case, but with the table variables permuted. All residuals are exactly zero in all cells, by construction. We see that in each view, the four large tiles corresponding to the first two variables align, indicating that these two variables are marginally independent. For example, in the (1, 2) panel, age and sex are independent, collapsed over disease.

Moreover, comparing the top half to the bottom half in any panel we see that the divisions by the third variable are the same for both levels of the second variable. In the (1, 2) panel, for example, age and disease are independent for both males and females. This means that age and sex are conditionally independent given disease ($\text{age} \perp \text{sex} \,|\, \text{disease}$).

Because this holds in all six panels, we see that mutual independence implies that *all pairs* of variables are conditionally independent, given the remaining one, ($X \perp Y \,|\, Z$) for all permutations of variables. A similar argument can be used to show that joint independence also holds, i.e., ($(X, Y) \perp Z$) for all permutations of variables.

Alternatively, you can also visualize these relationships interactively in a 3D mosaic using

`mosaic3d()` that allows you to rotate the mosaic to see all views. In Figure 5.32, all of the 3D tiles are unshaded and you can see that the 3D unit cube has been sliced according to the marginal frequencies.

```
> mosaic3d(fit)
```

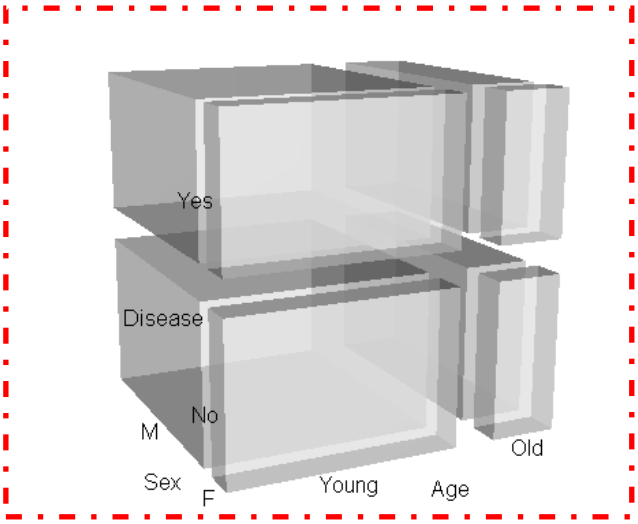


Figure 5.32: 3D mosaic plot of frequencies according to the model of mutual independence. The one-way margins are slices through the unit cube. {fig:struct-mos3d1}

5.8.2 Joint independence

The model of joint independence, $H_2 : (A, B) \perp C$, or equivalently, the loglinear model $[AB][C]$ may be visualized similarly by a mosaic matrix in which the data are replaced by fitted values under this model. We illustrate this for the model [Age Sex][Disease], calculating the fitted values in a similar way as before.

```
> joint <- loglm(~ Age * Sex + Disease, data = struc, fitted = TRUE)
> fit <- as.table(fitted(joint))
> structable(fit)

             Sex       F       M
Age   Disease
Young No           22.593 264.814
      Yes          20.407 239.186
Old   No           21.542  63.051
      Yes          19.458  56.949
```

The `pairs.table()` plot, now using simpler pairwise plots (`type="pairwise"`), is shown in Figure 5.33.

```
> pairs(fit, gp = shading_Friendly2)
```

This shows, in row 3 and column 3, the anticipated independence of both age and sex with disease, collapsing over the remaining variable. The (1, 2) and (2, 1) panels show that age and sex are still associated when disease is ignored.

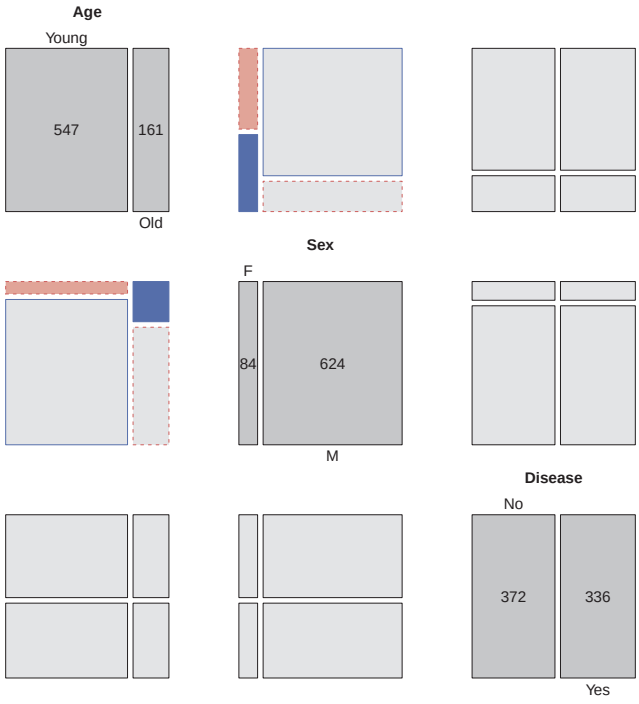


Figure 5.33: Mosaic matrix for fitted values under joint independence for the model [Age Sex][Disease]. {fig:struc-mos4}

5.9 Related visualization methods

{sec:related}

A variety of other graphical methods provide the means for visualizing relationships in multiway frequency tables. We briefly describe a few of these here, without much detail, to give a sense of some alternatives.

5.9.1 Doubledecker plots

{sec:doubledecker}

Doubledecker plots visualize the dependence of one categorical (typically binary) variable on further categorical variables. Formally, they are mosaic plots with vertical splits for all dimensions (predictors) except the last one, which represents the dependent variable (outcome). The last variable is visualized by horizontal splits, no space between the tiles, and separate colors for the levels.

They have the advantage of making it easier to "read" the differences among the conditional response proportions in relation to combinations of the explanatory variables. Moreover, for a binary response, the difference in these conditional proportions for any two columns has a direct relation to the odds ratio for a positive response in relation to those predictor levels (Hofmann, 2001).

The `doubledecker()` function in vcd takes a formula argument of the form `R ~ E1 + E2 + ...` where `R` is the response variable and `E1`, `E2`, ... are the predictors in the contingency table in array form. The shorthand notation, `R ~ .` means that all variables other than `R` are taken as predictors, in their order in the array.

{ex:berkeley-ddecker}

EXAMPLE 5.18: Berkeley admissions

Figure 5.34 shows the doubledecker plot for the *UCBAdmissions* data. By default, the levels

of the response (`Admit`) are taken in their order in the array and shaded to highlight the *last* level (Rejected). We want to highlight Admitted, so we reverse this dimension in the call below.

```
> doubledecker(Admit ~ Dept + Gender, data = UCBAdmissions[2:1, , ])
```

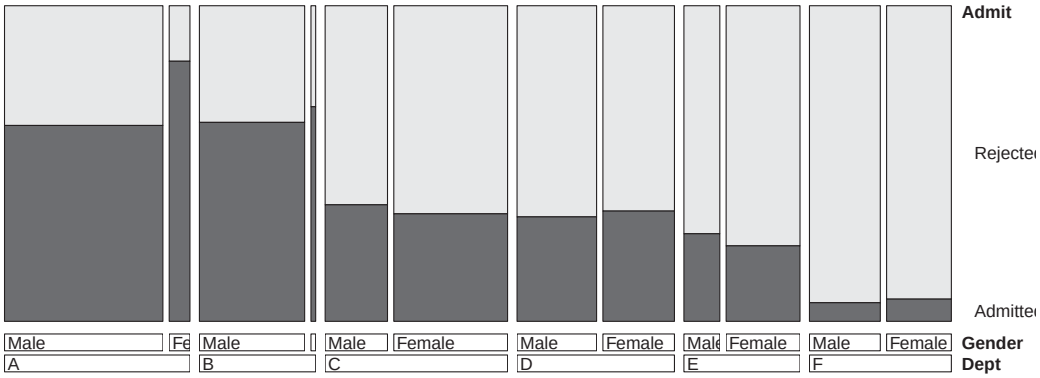


Figure 5.34: Doubledecker plot for the UCBAdmissions data. {fig:berkeley-doubledecker}

In Figure 5.34, it is easy to see the effects of both `Dept` and `Gender` on `Admit`. Admission rate declines across departments A–E, and within departments, the proportion admitted is roughly the same, except for department A, where more female applicants are admitted. △ {ex:titanic-ddecker}

EXAMPLE 5.19: Titanic data

Figure 5.35 shows the doubledecker plot for the *`Titanic`* data. The levels of the response (`Survived`) are shaded in increasing grey levels, highlighting the proportions of survival.

```
> doubledecker(Survived ~ Class + Age + Sex, Titanic)
```

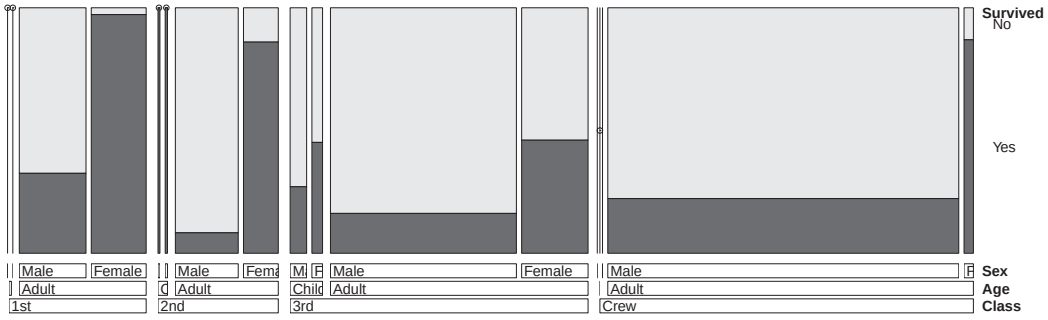


Figure 5.35: Doubledecker plot for the Titanic data. {fig:titanic-doubledecker}

This order of variables makes it easiest to compare survival of men and women within each age–class combination, but you can also see that survival of adult women decreases with class, and survival among men was greatest in first class. Some additional visualizations of these relationships are illustrated using the next topic in Example 5.21.

△

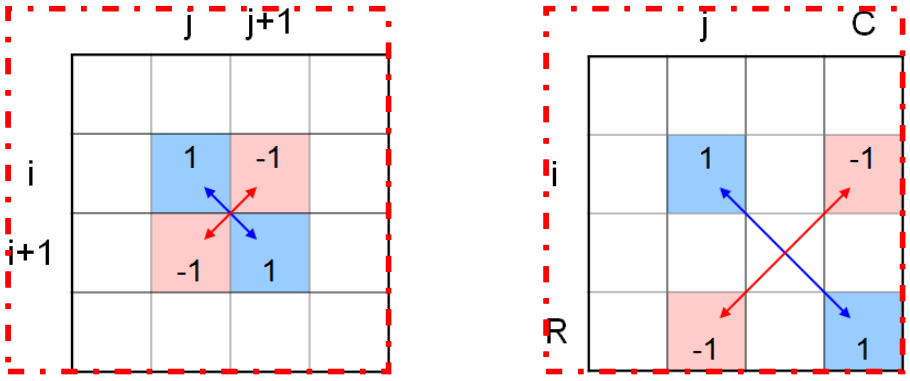


Figure 5.36: Generalized odds ratios for an $R \times C$ table. Left: local odds ratios for adjacent categories. Right: odds ratios with respect to a reference category (the last). Each log odds ratio is a contrast of the log frequencies, shown by the cell weights.

{fig:lor}

5.9.2 Generalized odds ratios*

{sec:oddsratio}

In Example 4.12, we used fourfold displays (Figure 4.7) to analyze the odds ratio between breathlessness and wheeze in coal miners as a function of age. Figure 4.8 showed that a plot of the odds ratio directly against age gave a simplified description of this three-way relationship.

Odds ratios for 2×2 tables can be generalized to $R \times C$ tables in a variety of ways, and these can also be calculated for *n*-way tables by treating all but the first two dimensions as strata. Plots of these generalized odds ratios can be quite informative, perhaps more so than in the $2 \times 2 \times k$ case.

Consider an $R \times C$ table with frequencies n_{ij}. Then a set of $(R-1) \times (C-1)$ ***local odds ratio***s, $\theta_{i,j}$, can be calculated as the odds ratios for adjacent pairs of rows and columns as shown in the left panel of Figure 5.36.

$$\theta_{ij} = \frac{n_{ij}/n_{i+1,j}}{n_{i,j+1}/n_{i+1,j+1}} = \frac{n_{ij} \times n_{i+1,j+1}}{n_{i+1,j} \times n_{i,j+1}}, \quad \begin{array}{l} i = 1, 2, \ldots, R-1 \\ j = 1, 2, \ldots, C-1 \end{array}.$$

These odds ratios correspond to "profile contrasts" (or sequential contrasts or successive differences) for ordered categories. Similarly, if one row category and one column category (say, the last) are considered baseline or reference categories, odds ratios with respect to contrasts with those categories (Figure 5.36, right panel) are defined as

$$\theta_{ij} = \frac{n_{i,j} \times n_{R,C}}{n_{i,C} \times n_{R,j}}, \quad \begin{array}{l} i = 1, 2, \ldots, R-1 \\ j = 1, 2, \ldots, C-1 \end{array}.$$

Note that all such parameterizations are equivalent, in that one can derive all other possible odds ratios from any non-redundant set, but substance-driven contrasts will be easier to interpret.

This calculation is simple in terms of log odds ratios, because it corresponds to a contrast among the log frequencies, with weights ± 1 for the four relevant cells. For local odds ratios, these are

$$\log(\theta_{ij}) = \begin{pmatrix} 1 & -1 & -1 & 1 \end{pmatrix} \log \begin{pmatrix} n_{ij} & n_{i+1,j} & n_{i,j+1} & n_{i+1,j+1} \end{pmatrix}^{\mathsf{T}}.$$

Consider an $R \times C \times K_1 \times K_2 \times \ldots$ frequency table $n_{ij\ldots}$, with factors $K_1, K_2 \ldots$ taken as strata. Let $\boldsymbol{n} = \text{vec}(n_{ij\ldots})$ be the $N \times 1$ vectorization of the frequency table. Then, all log odds ratios and their asymptotic covariance matrix can be calculated as:

$$\begin{aligned} \log(\widehat{\boldsymbol{\theta}}) &= \boldsymbol{C} \log(\boldsymbol{n}) \\ \boldsymbol{S} \equiv \mathcal{V}[\log(\boldsymbol{\theta})] &= \boldsymbol{C}\, \text{diag}\,(\boldsymbol{n})^{-1}\, \boldsymbol{C}^{\mathsf{T}} \end{aligned}$$

where $\boldsymbol{C}$ is an N-column matrix containing all zeros, except for two $+1$ elements and two -1 elements in each row that select the four cells involved in each log lodds ratio.[15]

The function `loddsratio()` in vcd calculates these values for the categories of the first two dimensions of an n-way table, together with their asymptotic covariance matrix. Additional dimensions are treated as strata. The `as.array()` and `as.data.frame()` methods can be used to convert a `loddsratio` object to a form suitable for plotting or further analysis.

{ex:punish2}

EXAMPLE 5.20: Corporal punishment data

Example 5.11 used mosaic displays to describe the relationship between attitude toward corporal punishment of children in relationship to memory of having experienced that as a child and education and age of the respondent. Given that `attitude` is the response, we could examine the odds ratios among this variable and any one predictor, treating the other variables as strata. Continuing the analysis of Example 5.11, we calculate log odds ratios for the association of `attitude` and `memory`, stratified by `age` and `education`.

```
> data("Punishment", package = "vcd")
> pun_lor <- loddsratio(Freq ~ memory + attitude | age + education,
+                       data = Punishment)
```

The `as.data.frame()` method converts this to a data frame, and adds standard errors (ASE).

```
> pun_lor_df <- as.data.frame(pun_lor)
```

The plot method for `loddsratio` objects conveniently plots the log odds ratio (LOR) against the strata variables, `age` or `education`, and by default also adds error bars. The result is shown in Figure 5.37.

```
> plot(pun_lor)
```

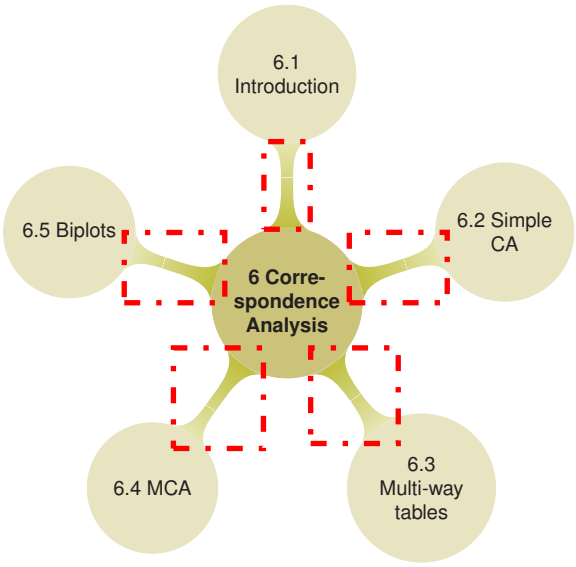
Compared to Figure 5.20, the differences among the age and education groups are now clear. For respondents less than age 40, increasing education increases the association (log odds ratio) between attitude and memory: those who remembered corporal punishment as a child are more likely to approve of it as their education increases. This result is reversed for those over 40, where all log odds ratios are negative: memory of corporal punishment makes it *less* likely to approve, and this effect becomes stronger with increased education.

Because log odds ratios have an approximate normal distribution under the null hypothesis that all $\log \theta_{ij} = 0$, you can treat these values as data, and carry out a rough analysis of the effects of the stratifying variables using ANOVA, with weights inversely proportional to the estimated sampling variances.[16] In the analysis shown below, we have treated age and education as ordered (numeric) variables.

```
> pun_mod <- lm(LOR ~ age * education, data = pun_lor_df,
+               weights = 1 / ASE^2)
> anova(pun_mod)

Analysis of Variance Table

Response: LOR
              Df Sum Sq Mean Sq F value Pr(>F)
```

[15] Some additional theory and applications of generalized odds ratios for ordered variables is given by Goodman (1983). Hofmann (2001) describes some connections between odds ratios, loglinear models, and visual modeling using doubledecker plots and mosaic plots.

[16] This ignores the covariances among the log odds ratios, which are not independent. A proper analysis uses generalized least squares with a weight matrix $\boldsymbol{S}^{-1}$, where $\boldsymbol{S} = \mathcal{V}[\log(\boldsymbol{\theta})]$ is the covariance matrix.



6

Correspondence Analysis

{ch:corresp}

Correspondence analysis provides visualizations of associations in a two-way contingency table in a small number of dimensions. Multiple correspondence analysis extends this technique to *n*-way tables. Other graphical methods, including mosaic matrices and biplots provide complementary views of loglinear models for two-way and *n*-way contingency tables, but correspondence analysis methods are particularly useful for a simple visual analysis.

6.1 Introduction

> Whenever a large sample of chaotic elements is taken in hand and marshalled in the order of their magnitude, an unsuspected and most beautiful form of regularity proves to have been latent all along.
>
> Sir Francis Galton, *Natural Inheritance*, London: Macmillan, 1889.

Correspondence analysis (CA) is an exploratory technique that displays the row and column categories in a two-way contingency table as points in a graph, so that the positions of the points represent the associations in the table. Mathematically, correspondence analysis is related to the ***biplot***, to ***canonical correlation***, and to ***principal component analysis***.

This technique finds scores for the row and column categories on a small number of dimensions that account for the greatest proportion of the χ^2 for association between the row and column categories, just as principal components account for maximum variance of quantitative variables. But CA does more—the scores provide a quantification of the categories, and have the property that

they maximize the correlation between the row and column variables. For graphical display two or three dimensions are typically used to give a reduced rank approximation to the data.

Correspondence analysis has a very large, multi-national literature and was rediscovered several times in different fields and different countries. The method, in slightly different forms, is also discussed under the names ***dual scaling***, ***optimal scaling***, ***reciprocal averaging***, ***homogeneity analysis***, and ***canonical analysis of categorical data***.

See Greenacre (1984) and Greenacre (2007) for an accessible introduction to CA methodology, or Gifi (1981) and Lebart et al. (1984) for a detailed treatment of the method and its applications from the Dutch and French perspectives. Greenacre and Hastie (1987) provide an excellent discussion of the geometric interpretation, while van der Heijden and de Leeuw (1985) and van der Heijden et al. (1989) develop some of the relations between correspondence analysis and log-linear methods for three-way and larger tables. Correspondence analysis is usually carried out in an exploratory, graphical way. Goodman (1981, 1985, 1986) has developed related inferential models, the RC model (see Section 10.1.3) and the canonical correlation model, with close links to CA.

One simple development of CA is as follows: For a two-way table the scores for the row categories, namely $\boldsymbol{X} = \{x_{im}\}$, and column categories, $\boldsymbol{Y} = \{y_{jm}\}$, on dimension $m = 1, \ldots, M$ are derived from a (generalized) ***singular value decomposition*** of (Pearson) residuals from independence, expressed as $d_{ij}/\sqrt{n}$, to account for the largest proportion of the χ^2 in a small number of dimensions. This decomposition may be expressed as

{eq:cadij}

$$\frac{d_{ij}}{\sqrt{n}} = \frac{n_{ij} - m_{ij}}{\sqrt{n\, m_{ij}}} = \boldsymbol{X}\, \boldsymbol{D}_\lambda\, \boldsymbol{Y}^{\mathsf{T}} = \sum_{m=1}^{M} \lambda_m\, x_{im}\, y_{jm}\ , \tag{6.1}$$

where m_{ij} is the expected frequency and where $\boldsymbol{D}_\lambda$ is a diagonal matrix with elements $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M$, and $M = \min(I-1,\, J-1)$. In M dimensions, the decomposition Eqn. (6.1) is exact. For example, an $I \times 3$ table can be depicted exactly in two dimensions when $I \geq 3$. The useful result for visualization purposes is that a rank-d approximation in d dimensions is obtained from the first d terms on the right side of Eqn. (6.1). The proportion of the Pearson χ^2 accounted for by this approximation is

$$n \sum_{m}^{d} \lambda_m^2 / \chi^2\ .$$

ignoretrue The quantity $\chi^2/n = \sum_i \sum_j d_{ij}^2/n$ is called the total ***inertia*** and is identical to the measure of association known as Pearson's mean-square contingency, the square of the ϕ coefficient.

Thus, correspondence analysis is designed to show how the data deviate from expectation when the row and column variables are independent, as in the sieve diagram, association plot, and mosaic display. However, the sieve, association, and mosaic plots depict every *cell* in the table, and for large tables it may be difficult to see patterns. Correspondence analysis shows only row and column *categories* as points in the two (or three) dimensions that account for the greatest proportion of deviation from independence. The pattern of the associations can then be inferred from the positions of the row and column points.

6.2 Simple correspondence analysis

{sec:ca-simple}

6.2.1 Notation and terminology

{sec:ca-notation}

Because Correspondence analysis grew up in so many homes, the notation, formulae, and terms used to describe the method vary considerably. The notation used here generally follows Greenacre (1984, 1997, 2007).

The descriptions here employ the following matrix and vector definitions:

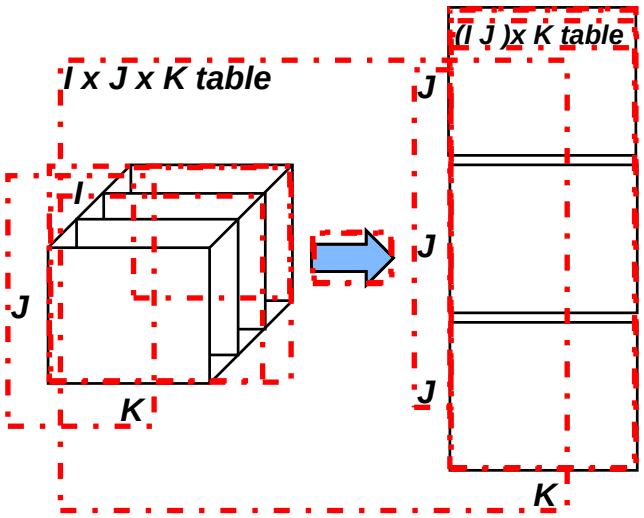


Figure 6.5: Stacking approach for a three-way table. Two of the table variables are combined interactively to form the rows of a two-way table. {fig:stacking}

to force a multiway table into a two-way table for a standard correspondence analysis, but it is a useful one.

A three-way table of size $I \times J \times K$ can be sliced into I two-way tables, each $J \times K$. If the slices are concatenated vertically, the result is one two-way table, of size $(I \times J) \times K$, as illustrated in Figure 6.5. In effect, the first two variables are treated as a single composite variable with IJ levels, which represents the main effects and interaction between the original variables that were combined. Van der Heijden and de Leeuw (1985) discuss this use of correspondence analysis for multi-way tables and show how *each* way of slicing and stacking a contingency table corresponds to the analysis of a specified loglinear model. Like the mosaic display, this provides another way to visualize the relations in a loglinear model.

In particular, for the three-way table with variables A, B, C that is reshaped as a table of size $(I \times J) \times K$, the correspondence analysis solution analyzes residuals from the log-linear model $[AB][C]$. That is, for such a table, the $I \times J$ rows represent the joint combinations of variables A and B. The expected frequencies under independence for this table are

$$m_{[ij]k} = \frac{n_{ij+}\, n_{++k}}{n} \,, \tag{6.6}$$ {eq:mij-k}

which are the ML estimates of expected frequencies for the log-linear model $[AB][C]$. The χ^2 that is decomposed by correspondence analysis is the Pearson χ^2 for this log-linear model. When the table is stacked as $I \times (J \times K)$ or $J \times (I \times K)$, correspondence analysis decomposes the residuals from the log-linear models $[A][BC]$ and $[B][AC]$, respectively, as shown in Table 6.1. In this approach, only the associations in separate $[\,]$ terms are analysed and displayed in the correspondence analysis maps. Van der Heijden and de Leeuw (1985) show how a generalized form of correspondence analysis can be interpreted as decomposing the difference between two specific loglinear models, so their approach is more general than is illustrated here.

6.3.1 Interactive coding in R

{sec:ca-interactiveR}

In the general case of an *n*-way table, the stacking approach is similar to that used by `ftable()` and `structable()` in vcd as described in Section 2.5 to flatten multiway tables to a two-way,

Table 6.1: Each way of stacking a three-way table corresponds to a loglinear model {tab:stacking}

Stacking structure	Loglinear model
$(I \times J) \times K$	$[AB][C]$
$I \times (J \times K)$	$[A][BC]$
$J \times (I \times K)$	$[B][AC]$

printable form, where some variables are assigned to the rows and the others to the columns. Both `ftable()` and `structable()` have `as.matrix()` methods[2] that convert their result into a matrix suitable as input to `ca()`.

With data in the form of a frequency data frame, you can easily create interactive coding using `interaction()` or simply use `paste()` to join the levels of stacked variables together.

To illustrate, create a 4-way table of random Poisson counts (with constant mean, $\lambda = 15$) of types of Pet, classified by Age, Color and Sex.

```
> set.seed(1234)
> dim <- c(3, 2, 2, 2)
> tab <- array(rpois(prod(dim), 15), dim = dim)
> dimnames(tab) <- list(Pet = c("dog", "cat", "bird"),
+                       Age = c("young", "old"),
+                       Color = c("black", "white"),
+                       Sex = c("male", "female"))
```

You can use `ftable()` to print this, with a formula that assigns `Pet` and `Age` to the columns and `Color` and `Sex` to the rows.

```
> ftable(Pet + Age ~ Color + Sex, tab)

             Pet   dog       cat      bird
             Age young old young old young old
Color Sex
black male          10  12    16  16    16  12
      female         8  12    13  15    11  13
white male          18  11    12  18    13  20
      female        13  13    16  15    12  15
```

Then, `as.matrix()` creates a matrix with the levels of the stacked variables combined with some separator character. Using `ca(pet.mat)` would then calculate the CA solution for the stacked table, analyzing only the associations in the loglinear model [Pet Age][Color Sex].[3]

```
> (pet.mat <- as.matrix(ftable(Pet + Age ~ Color + Sex, tab), sep = '.'))

              Pet.Age
Color.Sex      dog.young dog.old cat.young cat.old bird.young bird.old
  black.male          10      12        16      16         16       12
  black.female         8      12        13      15         11       13
  white.male          18      11        12      18         13       20
  white.female        13      13        16      15         12       15
```

With data in a frequency data frame, a similar result (as a frequency table) can be obtained using `interaction()` as shown below. The result of `xtabs()` looks the same as `pet.mat`.

[2] This requires at least R version 3.1.0 or vcd 1.3-2 or later.

[3] The result would not be at all interesting here. Why?

Greenacre, 2007, Chapter 19), an iterative method that replaces the diagonal blocks of the Burt matrix with values that minimize their impact on inertia. Unlike MCA, solutions in JCA are not nested, however.

{ex:titanic2}

EXAMPLE 6.9: Survival on the *Titanic*

An MCA analysis of the `Titanic` data is carried out using `mjca()` as shown below.

```
> titanic.mca <- mjca(Titanic)
```

`mjca()` allows different scaling methods for the contributions to inertia of the different dimensions. The default (`lambda="adjusted"`), used here, is the adjusted inertias as in Eqn. (6.8).

```
> summary(titanic.mca)


Principal inertias (eigenvalues):

 dim    value      %   cum%   scree plot
 1      0.067655  76.8  76.8  ************************
 2      0.005386   6.1  82.9  **
 3      00000000   0.0  82.9
        -------- -----
 Total: 0.088118
...
```

Using similar code to that used in Example 6.8, Figure 6.11 shows an enhanced version of the default plot that connects the category points for each factor by lines using the result returned by the `plot()` function.

In this plot, the points for each factor have the property that the sum of coordinates on each dimension, weighted inversely by the marginal proportions, equals zero. Thus high-frequency categories (e.g., Adult and Male) are close to the origin.

The first dimension is perfectly aligned with gender, and also strongly aligned with Survival. The second dimension pertains mainly to Class and Age effects. Consider those points that differ from the origin most similarly (in distance and direction) to the point for Survived ("Yes"); this gives the interpretation that survival was associated with being female or upper class or (to a lesser degree) being a child.

△

6.5 Biplots for contingency tables

{sec:biplot}

Like correspondence analysis, the ***biplot*** (Bradu and Gabriel, 1978, Gabriel, 1971, 1980, 1981, Gower et al., 2011) is a visualization method that uses the SVD to display a matrix in a low-dimensional (usually 2-dimensional) space. They differ in the relationships in the data that are portrayed, however:

- In correspondence analysis the (weighted, χ^2) *distances* between row points and distances between column points are designed to reflect *differences* between the row profiles and column profiles.
- In the biplot, on the other hand, row and column points are represented by *vectors* from the origin such that the projection (inner product) of the vector $\boldsymbol{a}_i$ for row i on $\boldsymbol{b}_j$ for column j approximates the data element y_{ij},

$$\boldsymbol{Y} \approx \boldsymbol{A}\boldsymbol{B}^{\mathsf{T}} \iff y_{ij} \approx \boldsymbol{a}_i^{\mathsf{T}} \boldsymbol{b}_j \; . \tag{6.9}$$

{eq:biplot1}

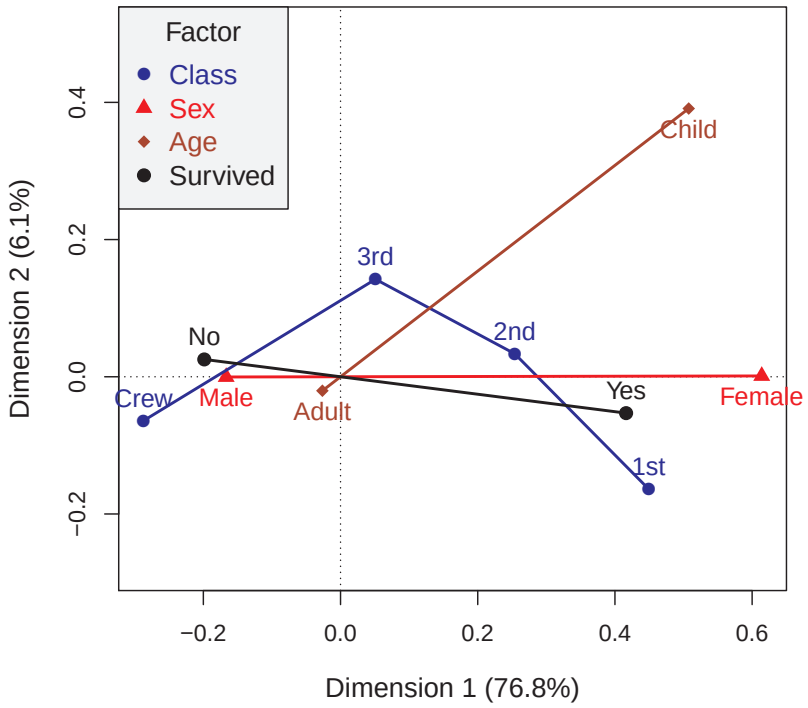


Figure 6.11: MCA plot of the Titanic data. The category points are joined separately by lines for the factor variables. {fig:titanic-mca-plot}

Geometrically, Eqn. (6.9) may be described as approximating the data value y_{ij} by the projection of the end point of vector $\boldsymbol{a}_i$ on $\boldsymbol{b}_j$ (and vice-versa), as shown in Figure 6.12.

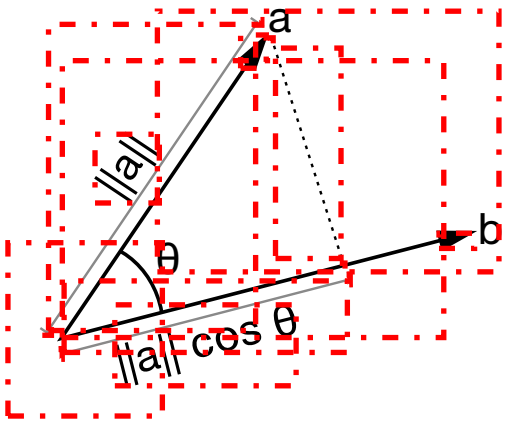


Figure 6.12: The scalar product of vectors of two points from the origin is the length of the projection of one vector on the other. {fig:Scalarproduct}

6.5.1 CA bilinear biplots

As in CA, there are a number of different representations of coordinates for row and column points for a contingency table within a biplot framework. One set of connections between CA and the biplot can be seen through the *reconstitution formula*, giving the decomposition of the correspondence matrix $\boldsymbol{P} = \boldsymbol{N}/n$ in terms of the standard coordinates $\boldsymbol{\Phi}$ and $\boldsymbol{\Gamma}$, defined in Eqn. (6.4) and Eqn. (6.5) as:

$$p_{ij} = r_i c_j \left(1 + \sum_{m=1}^{M} \sqrt{\lambda_m} \phi_{im} \gamma_{jm} \right) , \tag{6.10}$$ {eq:reconstitution1}

or, in matrix terms,

$$\boldsymbol{P} = \boldsymbol{D}_r (\mathbf{1}\mathbf{1}^\mathsf{T} + \boldsymbol{\Phi} \boldsymbol{D}_\lambda^{1/2} \Gamma^\mathsf{T}) \boldsymbol{D}_c \, . \tag{6.11}$$ {eq:reconstitution2}

The CA solution approximates this by a sum over $d \ll M$ dimensions, or by using only the first d (usually 2) columns of $\boldsymbol{\Phi}$ and $\boldsymbol{\Gamma}$.

Eqn. (6.10) can be re-written in biplot scalar form as

$$\left(\frac{p_{ij}}{r_i c_j} \right) - 1 \approx \sum_{m=1}^{d} (\sqrt{\lambda_m} \phi_{im}) \gamma_{jm} = \sum_{m=1}^{d} f_{im} \gamma_{jm} \tag{6.12}$$ {eq:rowprincipal}

where $f_{im} = (\sqrt{\lambda_m} \phi_{im})$ gives the principal coordinates of the row points. The left-hand side of Eqn. (6.12) contains the ***contingency ratios***, $p_{ij}/r_i c_j$, of the observed cell probabilities to their expected values under independence. This shows that an ***asymmetric CA plot*** of row principal coordinates $\boldsymbol{F}$ and the column standard coordinates $\boldsymbol{\Gamma}$ is a biplot that approximates the deviations of the contingency ratios from their values under independence.

In the ca package, this plot is obtained by specifying `map="rowprincipal"` in the call to `plot()`, or `map="colprincipal"` to plot the column points in principal coordinates. It is typical in such biplots to display one set of coordinates as points and the other as vectors from the origin, as controlled by the `arrows` argument, so that one can interpret the data values represented as approximated by the projections of the points on the vectors.

Two other types of asymmetric "maps" are also defined with different scalings that turn out to have better visual properties in terms of representing the relations between the row and column categories, particularly when the strength of association (inertia) in the data is low.

- The option `map="rowgab"` (or `map="colgab"`) gives a biplot form proposed by Gabriel and Odoroff (1990) with the rows (columns) shown in principal coordinates and the columns (rows) in standard coordinates multiplied by the mass c_j (r_i) of the corresponding point.
- The *contribution biplot* for CA (Greenacre, 2013), with the option `map="rowgreen"` (or `map="colgreen"`) provides a reconstruction of the standardized residuals from independence, using the points in standard coordinates multiplied by the square root of the corresponding masses. This has the nice visual property of showing more directly the contributions of the vectors to the low-dimensional solution.

{ex:suicide3}

EXAMPLE 6.10: Suicide rates in Germany — biplot

To illustrate the biplot representation, we continue with the data on suicide rates in Germany from Example 6.5, using the stacked table `suicide.tab` comprised of the age–sex combinations as rows and methods of suicide as columns.

```
> suicide.tab <- xtabs(Freq ~ age_sex + method2, data = Suicide)
> suicide.ca <- ca(suicide.tab)
```