

# Chapter 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, little boxes, little boxes made of ticky-tacky;  
Little boxes, little boxes, little boxes all the same.  
There are red ones, and blue ones, and green ones, and yellow ones;  
Little boxes, little boxes, and they all look just the same.

---

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:  $2 \times 2$  tables (fourfold display),  $r \times c$  tables (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 nonlinear 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 which are fitted by various loglinear models.

## 5.2 Two-way tables

{sec:mosaic

The mosaic display (Friendly, 1992, 1994, 1997, Hartigan and Kleiner, 1981, 1984) is like a grouped bar chart, 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  $4 \times 4$  table is shown in Figure 5.1.

```
## Error in eval(expr, envir, enclos): could not find function "mosaic"
## Error in eval(expr, envir, enclos): could not find function "labeling_cells"
```

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

```
## Error in eval(expr, envir, enclos): could not find function "mosaic"
```

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
```

```
## Error in eval(expr, envir, enclos): could not find function "mosaic"
## Error in eval(expr, envir, enclos): could not find function "labeling_cells"
## Error in eval(expr, envir, enclos): could not find function "mosaic"
```

These frequencies can be shown as the mosaic for the first variable (hair color), with the unit square split according to the marginal proportions as in Figure 5.2 (left). The rectangular tiles are then shaded to show the residuals (deviations) from a particular model as shown in the right panel of Figure 5.2. The details of the calculations for shading are:

- The one-way table of marginal totals can be fit to a model, in this case, the (implausible) model that all hair colors are equally probable. This model has expected frequencies  $m_i = 592/4 = 148$ :

```
expected <- rep(sum(hair)/4, 4)
names(expected) <- names(hair)
expected

## Black Brown Red Blond
## 148 148 148 148
```

- The Pearson residuals from this model,  $r_i = (n_i - m_i)/\sqrt{m_i}$ , are:

```
(residuals <- (hair - expected) / sqrt(expected))

## Hair
## Black Brown Red Blond
## -3.2880 11.3435 -6.3294 -1.7262
```

and these values are shown by color and shading as shown in the legend. The high positive value for Brown hair indicates that people with brown hair are much more frequent in this sample than the equiprobability model would predict; the large negative residual for Red hair shows that red heads are much less common. Further details of the schemes for shading are described below, but essentially we use increasing intensities of blue (red) for positive (negative) residuals.

In the next step, the rectangle for each Hair color is subdivided in proportion to the *relative* (conditional) frequencies of the second variable— Eye color, giving the following conditional row proportions:

```
round(addmargins(prop.table(haireye, 1), 2), 3)

## Eye
## Hair Brown Blue Hazel Green Sum
## Black 0.630 0.185 0.139 0.046 1.000
## Brown 0.416 0.294 0.189 0.101 1.000
## Red 0.366 0.239 0.197 0.197 1.000
## Blond 0.055 0.740 0.079 0.126 1.000
```

The proportions in each row determine the heights of the tiles in the second mosaic display in Figure 5.3.

```
mosaic(haireye, shade=TRUE, suppress=0,
       labeling=labeling_residuals, gp_text=gpar(fontface=2))

## Error in eval(expr, envir, enclos): could not find function "mosaic"
```

- Again, the cells are shaded in relation to standardized Pearson residuals,  $r_{ij} = (n_{ij} - m_{ij})/\sqrt{m_{ij}}$ , from a model. For a two-way table, the model is that Hair color and Eye color are independent in the population from which this sample was drawn. These residuals are calculated as shown below using `loglm()` to fit the independence model and `residuals()`.

```
HE.mod <- loglm(~ Hair + Eye, data=haireye)
round(resids <- residuals(HE.mod, type="pearson"), 2)

## Re-fitting to get frequencies and fitted values
##      Eye
## Hair   Brown  Blue Hazel Green
## Black  4.40 -3.07 -0.48 -1.95
## Brown  1.23 -1.95  1.35 -0.35
## Red   -0.07 -1.73  0.85  2.28
## Blond -5.85  7.05 -2.23  0.61
```

- Thus, in Figure 5.3, the two tiles shaded deep blue correspond to the two cells, (Black, Brown) and (Blond, Blue), whose residuals are greater than +4, indicating much greater frequency in those cells than would be found if Hair color and Eye color were independent. The tile shaded deep red, (Blond, Brown), corresponds to the largest negative residual = -5.85, indicating this combination is extremely rare under the hypothesis of independence.
- The overall Pearson  $\chi^2$  statistic for the independence model is just the sum of squares of the residuals, with degrees of freedom  $(r - 1) \times (c - 1)$ .

```
(chisq <- sum(resids^2))

## [1] 138.29

(df <- prod(dim(haireye)-1))

## [1] 9

chisq.test(haireye)

##
## Pearson's Chi-squared test
##
## data:  haireye
## X-squared = 138.29, df = 9, p-value < 2.2e-16
```

△

## Shading levels

A variety of schemes for shading the tiles are available in the `strucplot` framework (Section 5.3), but the simplest (and default) shading patterns for the tiles are based on the sign and magnitude of the standardized Pearson residuals, using shades of blue for positive residuals and red for negative residuals, and two threshold values for their magnitudes,  $|r_{ij}| > 2$  and  $|r_{ij}| > 4$ .

Because the standardized residuals are approximately unit-normal  $N(0, 1)$  values, this corresponds to highlighting cells whose residuals are *individually* significant at approximately the .05 and .0001 level, respectively. Other shading schemes described later provide tests of significance, but the main purpose of highlighting cells is to draw attention to the *pattern* of departures of the data from the assumed model of independence.

## Interpretation and reordering

To interpret the association between Hair color and Eye color, consider the pattern of positive (blue) and negative (red) tiles in the mosaic display. We interpret positive values as showing cells whose observed frequency is substantially greater than would be found under independence; negative values indicate cells which occur less often than under independence.

The interpretation can often be enhanced by reordering the rows or columns of the two-way table so that the residuals have an *opposite corner* pattern of signs. This usually helps us interpret any systematic patterns of association in terms of the ordering of the row and column categories.

In this example, a more direct interpretation can be achieved by reordering the Eye colors as shown in Figure 5.4. Note that in this rearrangement both hair colors and eye colors are ordered from dark to light, suggesting an overall interpretation of the association between Hair color and Eye color.

```
# re-order Eye colors from dark to light
haireye2 <- haireye[, c("Brown", "Hazel", "Green", "Blue")]
mosaic(haireye2, shade=TRUE)

## Error in eval(expr, envir, enclos): could not find function "mosaic"
```

In general, the levels of a factor in mosaic displays are often best reordered by arranging them according to their scores on the first (largest) *correspondence analysis* dimension (Friendly, 1994); see Chapter 6 for details. Friendly and Kwan (2003) use this as one example of *effect ordering* for data displays, illustrated in Chapter 1.

Thus, the mosaic in Figure 5.4 shows that the association between Hair and Eye color is essentially that:

- people with dark hair tend to have dark eyes,
- those with light hair tend to have light eyes
- people with red hair and hazel eyes do not quite fit this pattern

## 5.3 The strucplot framework

{sec:mosaic-strucplot}

Mosaic displays have much in common with sieve plots and association plots described in Chapter 4 and with related graphical methods such as *doubledecker plots* described later in this chapter. The main idea is to visualize a contingency table of frequencies by “tiles” corresponding to the table cells arranged in rectangular form. For multiway tables with more than two factors, the variables are nested into rows and columns using recursive conditional splits, given the table margins. The result is a “flat” representation that can be visualized in ways similar to a two-dimensional representation of a table. The `strutable()` function described in Section 2.5 gives the tabular version of a strucplot. The description below follows Meyer *et al.* (2006), also included as a vignette, (accessible from R as `vignette("strucplot", pkg="vcd")`), in `vcd`.

Rather than implementing each of these methods separately, the *strucplot framework* in the `vcd` package provides a general class of methods of which these are all instances. This framework defines a class of conditional displays which allows for granular control of graphical appearance aspects, including:

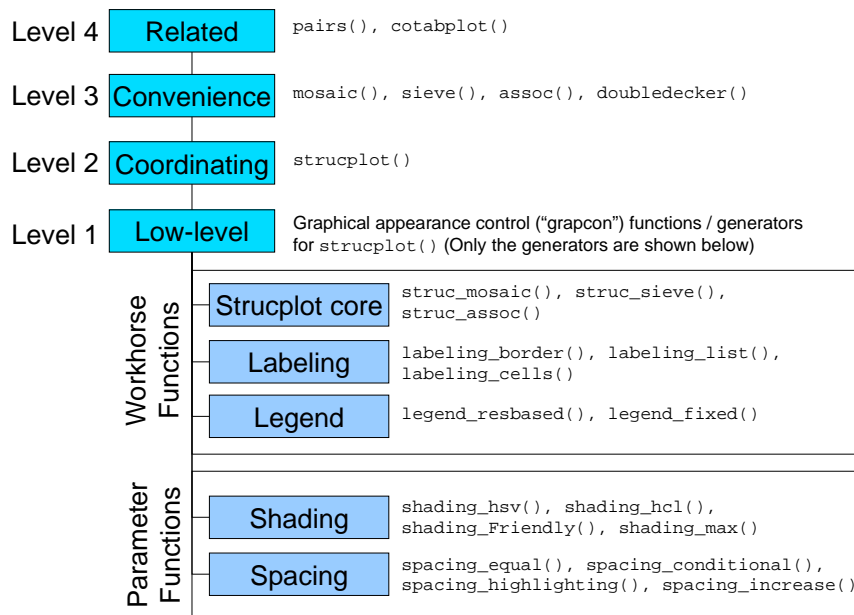


Figure 5.1: Components of the strucplot framework. High level functions use those at lower levels to provide a general system for tile-based plots of frequency tables.

{fig:struc}

- the content of the tiles, e.g., observed or expected frequencies
- the split direction for each dimension, horizontal or vertical
- the graphical parameters of the tiles' content, e.g., color or other visual attributes
- the spacing between the tiles
- the labeling of the tiles

The strucplot framework is highly modularized: Figure 5.1 shows the hierarchical relationship between the various components. For the most part, you will use directly the convenience and related 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 5.1 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.

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

Table 5.1: Available graphical appearance control (grapcon) generators in the strucplot framework

b:grapcons}

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

### 5.3.1 Shading schemes

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 arguments and returns such a `grapcon` function (see Table 5.1). `vcd` provides several shading functions, including support for both HSV and HCL colors, and the visualization of significance tests. **TODO: This points to the need for a section, probably in Chapter 1, on color spaces and color schemes for categorical data graphics.**

### Specifying graphical parameters for `strucplot` displays

`Strucplot` displays in `vcd` are built using the `grid` graphics package. There are many graphical parameters that can be set using `gp = gpar(...)` in a call to a high-level `strucplot` function. Among these, the following are often most useful to control the drawing components:

<code>col</code>	Color for lines and borders.
<code>fill</code>	Color for filling rectangles, polygons, ...
<code>alpha</code>	Alpha channel for transparency of fill color.
<code>lty</code>	Line type for lines and borders.
<code>lwd</code>	Line width for lines and borders.

In addition, a number of parameters control the display of text labels in these displays:

<code>fontsize</code>	The size of text (in points)
<code>cex</code>	Multiplier applied to <code>fontsize</code>
<code>fontfamily</code>	The font family

{sec:mosaic-shading}



fontface The font face (**bold**, *italic*, ...)

See `help(gpar)` for a complete list and further details.

We illustrate this capability below using the Hair color and Eye color data as reordered in Figure 5.4. The following example produces a *Marimekko chart*, or a “poor-man’s mosaic display” as shown in the left panel of Figure 5.2. This is essentially a divided bar chart where the eye colors within each horizontal bar for the hair color group are all given the same color. In the example, the matrix `fill_colors` is constructed to conform to the `haireye2` table, using color values that approximate the eye colors.

```
# color by hair color
fill_colors <- c("brown4", "#acba72", "green", "lightblue")
(fill_colors <- t(matrix(rep(fill_colors, 4), ncol=4)))

##      [,1]      [,2]      [,3]      [,4]
## [1,] "brown4" "#acba72" "green" "lightblue"
## [2,] "brown4" "#acba72" "green" "lightblue"
## [3,] "brown4" "#acba72" "green" "lightblue"
## [4,] "brown4" "#acba72" "green" "lightblue"

mosaic(haireye2, gp=gpar(fill=fill_colors, col=0))

## Error in eval(expr, envir, enclos): could not find function "mosaic"
```

Note that because the hair colors and eye colors are both ordered, this shows the decreasing prevalence of light hair color amongst those with brown eyes and the increasing prevalence of light hair with blue eyes.

Alternatively, for some purposes,<sup>1</sup> we might like to use color to highlight the pattern of diagonal cells, and the off-diagonals 1, 2, 3 steps removed. The R function `toeplitz()` returns such a patterned matrix, and we can use this to calculate the `fill_colors` by indexing the `palette()` function. The code below produces the right panel in Figure 5.2.

```
# toeplitz designs
toeplitz(1:4)

##      [,1] [,2] [,3] [,4]
## [1,] 1 2 3 4
## [2,] 2 1 2 3
## [3,] 3 2 1 2
## [4,] 4 3 2 1

fill_colors <- palette()[1+toeplitz(1:4)]
mosaic(haireye2, gp=gpar(fill=fill_colors, col=0))

## Error in eval(expr, envir, enclos): could not find function "mosaic"
```

More simply, to shade a mosaic according to the levels of one variable (typically a response variable), you can use the highlighting arguments of `mosaic()`. The first call below gives a result similar to the left panel of Figure 5.2. Alternatively, using the formula method for `mosaic()`, specify the response variable as the left-hand side.

<sup>1</sup>For example, this would be appropriate for a square table, showing agreement between row and column categories, as in Section 4.7.

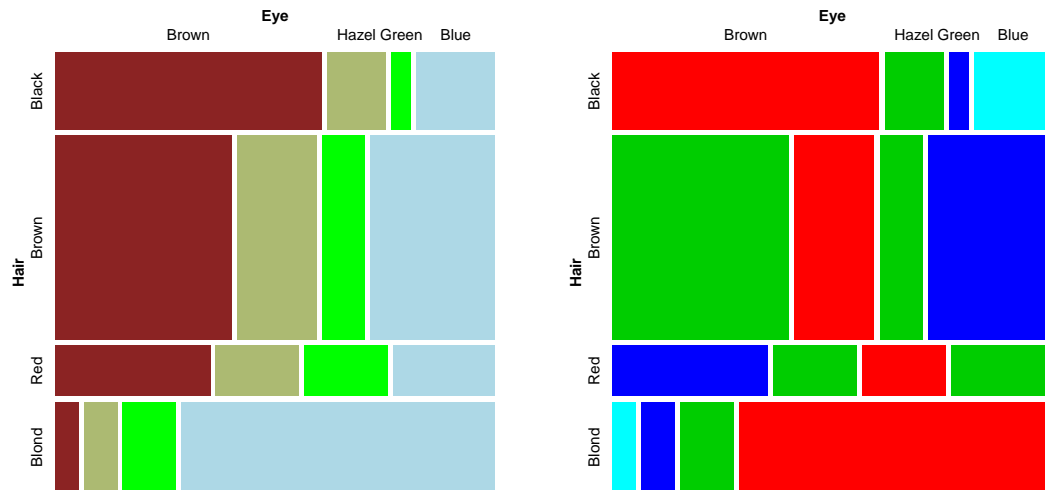


Figure 5.2: Mosaic displays for the `haireye2` data, using custom colors to fill the tiles. Left: Marimekko chart, using colors to reflect the eye colors; right: Toeplitz-based colors, reflecting the diagonal strips in a square table.

{fig:HE-fil

```
mosaic(haireye2, highlighting="Eye", highlighting_fill=fill_colors)
mosaic(Eye ~ Hair, data=as.table(haireye2))
```

### Residual-based shading

The important idea that differentiates mosaic and other strucplot displays from the “poor-man’s,” Marimekko versions (Figure 5.2) often shown in other software is that rather than just using shading color to *identify* the cells, we can use these attributes to show something more—*residuals* from some model, whose pattern helps to explain the association between the table variables.

As described above, the strucplot framework includes a variety of `shading_` functions, and these can be customized with optional arguments. Zeileis *et al.* (2007) describe a general approach to residual-based shadings for area-proportional visualizations, used in the development of the strucplot framework in `vcd`.

{ex:interp}

#### EXAMPLE 5.2: Interpolation options

One simple thing to do is to modify the `interpolate` option passed to the default `shading_hcl` function, as shown in Figure 5.7.

```
# more shading levels
mosaic(haireye2, shade=TRUE, gp_args=list(interpolate=1:4))

## Error in eval(expr, envir, enclos): could not find function "mosaic"

# continuous shading
interp <- function(x) pmin(x/6, 1)
mosaic(haireye2, shade=TRUE, gp_args=list(interpolate=interp))

## Error in eval(expr, envir, enclos): could not find function "mosaic"
```

For the left panel of Figure 5.7, a numeric vector is passed as `interpolate=1:4`, defining the boundaries of a step function mapping the absolute values of residuals to saturation levels in the HCL color scheme. For the right panel, a user-defined function, `interp()`, is created which maps the absolute residuals to saturation values in a continuous way (up to a maximum of 6).

Note that these two interpolation schemes produce quite similar results, differing mainly in the shading level of residuals within  $\pm 1$  and in the legend. In practice, the default discrete interpolation, using cutoffs of  $\pm 2$ ,  $\pm 4$  usually works quite well.  $\triangle$

{ex:shading}

### EXAMPLE 5.3: Shading functions

Alternatively, the names of shading functions can be passed as the `gp` argument, as shown below, producing Figure 5.8. Two shading function are illustrated here:

- The left panel of Figure 5.8 uses the classical Friendly (1994) shading scheme, `shading_Friendly` with HSV colors of blue and red and default cutoffs for absolute residuals,  $\pm 2$ ,  $\pm 4$ , corresponding to `interpolate = c(2, 4)`. In this shading scheme, all tiles use an outline color (`col`) corresponding to the sign of the residual. As well, the border line type (`lty`) distinguishes positive and negative residuals, which is useful if a mosaic plot is printed in black and white.
- The right panel uses the `shading_max()` function, based on the ideas of Zeileis *et al.* (2007) on residual-based shadings for area-proportional visualizations. Instead of using the cut-offs 2 and 4, it employs the critical values,  $M_\alpha$ , for the maximum absolute Pearson residual statistic,

$$M = \max_{i,j} |r_{ij}| ,$$

by default at  $\alpha = 0.10$  and  $0.01$ .<sup>2</sup> Only those residuals with  $|r_{ij}| > M_\alpha$  are colored in the plot, using two levels for Value (“lightness”) in HSV color space. Consequently, all color in the plot signals a significant departure from independence at 90% or 99% significance level, respectively.<sup>3</sup>

```
mosaic(haireye2, gp=shading_Friendly, legend=legend_fixed)

## Error in eval(expr, envir, enclos): could not find function "mosaic"

set.seed(1234)
mosaic(haireye2, gp=shading_max)

## Error in eval(expr, envir, enclos): could not find function "mosaic"
```

In this example, the difference between these two shading schemes is largely cosmetic, in that the pattern of association is similar in the two panels of Figure 5.8, and the interpretation would be the same. This is not always the case, as we will see in the next example.  $\triangle$

<sup>2</sup>These default significance levels were chosen because this leads to displays where fully colored cells are clearly significant ( $p < 0.01$ ), cells without color are clearly non-significant ( $p > 0.1$ ), and cells in between can be considered to be weakly significant ( $0.01 \leq p \leq 0.1$ ).

<sup>3</sup>This computation uses the `vcd` function `coinddep_test()` to calculate generalized tests of (conditional) independence by simulation from the marginal distribution of the input table under (conditional) independence. In these examples using `shading_max`, the function `set.seed()` is used to initialize the random number generators to a given state for reproducibility.

**EXAMPLE 5.4: Arthritis treatment**

This example uses the Arthritis data, illustrated earlier (Example ??), on the relation between treatment and outcome for rheumatoid arthritis. To confine this example to a two-way table, we use only the (larger) female patient group.

```
art <- xtabs(~ Treatment + Improved, data = Arthritis,
            subset = Sex == "Female")

## Error in terms.formula(formula, data = data): object 'Arthritis' not
found

names(dimnames(art))[2] <- "Improvement"

## Error in names(dimnames(art))[2] <- "Improvement": object 'art' not
found
```

The calls to `mosaic()` below compare `shading_Friendly` and `shading_max`, giving the plots shown in Figure 5.9.

```
mosaic(art, gp=shading_Friendly, margin = c(right = 1),
       labeling=labeling_residuals, suppress=0, digits=2)

## Error in eval(expr, envir, enclos): could not find function "mosaic"

set.seed(1234)
mosaic(art, gp=shading_max, margin = c(right = 1))

## Error in eval(expr, envir, enclos): could not find function "mosaic"
```

This data set is somewhat paradoxical, in that the standard `chisq.test()` for association with these data gives a highly significant result,  $\chi^2(2) = 11.3, p = 0.0035$ , while the shading pattern using `shading_Friendly` in the left panel of Figure 5.9 shows all residuals within  $\pm 2$ , and thus unshaded.

On the other hand, the `shading_max` shading in the right panel of Figure 5.9 shows that significant deviations from independence occur in the four corner cells, corresponding to more of the treated group showing marked improvement, and more of the placebo group showing no improvement.

Some details behind the `shading_max` method are shown below. The Pearson residuals for this table are calculated as:

```
residuals(loglm(~Improvement + Treatment, data=art), type="pearson")

## Error in loglm(~Improvement + Treatment, data = art): object 'art' not
found
```

The `shading_max()` function then calls `coindep_test(art)` to generate  $n = 1000$  random tables with the same margins, and computes the maximum residual statistic for each. This gives a non-parametric  $p$ -value for the test of independence,  $p = 0.011$  shown in the legend.

```
set.seed(1243)
art_max <- coinddep_test(art)

## Error in eval(expr, envir, enclos): could not find function "coinddep_test"

art_max

## Error in eval(expr, envir, enclos): object 'art_max' not found
```

Finally, the 0.90 and 0.99 quantiles of the simulation distribution are used as shading levels, passed as the value of the `interpolate` argument.

```
art_max$qdist(c(0.90, 0.99))

## Error in eval(expr, envir, enclos): object 'art_max' not found
```

△

The converse situation can also arise in practice. An overall test for association using Pearson's  $\chi^2$  may not be significant, but the maximum residual test may highlight one or more cells worthy of greater attention, as illustrated in the following example.

{ex:soccer2}

#### EXAMPLE 5.5: UK Soccer scores

In Example 3.9, we examined the distribution of goals scored by the home team and the away team in 380 games in the 1995/96 season by the 20 teams in the UK Football Association, Premier League. The analysis there focused on the distribution of the total goals scored, under the assumption that the number of goals scored by the home team and the away team were independent.

Here, the rows and columns of the table `UKSoccer` are both ordered, so it is convenient and compact to carry out all the CMH tests taking ordinality into account.

```
data("UKSoccer", package="vcd")
CMHtest(UKSoccer)

## Error in eval(expr, envir, enclos): could not find function "CMHtest"
```

All of these are non-significant, so that might well be the end of the story, as far as independence of goals in home and away games is concerned. Yet, one residual,  $r_{42} = 3.08$  stands out, corresponding to 4 or more goals by the home team and only 2 goals by the away team, which accounts for nearly half of the  $\chi^2(16) = 18.7$  for general association.

```
set.seed(1234)
mosaic(UKSoccer, gp=shading_max, labeling=labeling_residuals, digits=2)

## Error in eval(expr, envir, enclos): could not find function "mosaic"
```

This occurrence may or may not turn out to have some explanation, but at least the mosaic plot draws it to our attention. △

## 5.4 Three-way and larger tables

{sec:mosaic}

The mosaic displays and other graphical methods within the `strucplot` framework extend quite naturally to three-way and higher-way tables. The essential idea is that for the variables in a multiway table in a given order, each successive variable is used to subdivide the tile(s) in proportion to the relative (conditional) frequencies of that variable, given all previous variables. This process continues recursively until all table variables have been included.

For simplicity, we continue with the running example of Hair color and Eye color. Imagine that each cell of the two-way table for Hair and Eye color is further classified by one or more additional variables—sex and level of education, for example. Then each rectangle can be subdivided horizontally to show the proportion of males and females in that cell, and each of those horizontal portions can be subdivided vertically to show the proportions of people at each educational level in the hair-eye-sex group.

{ex:HEC1}

### EXAMPLE 5.6: Hair color, eye color and sex

Figure 5.11 shows the mosaic for the three-way table, with Hair and Eye color groups divided according to the proportions of Males and Females. As explained in the next section (Section 5.4.1) there are different models for “independence” we could display. Here, we show residuals for the model of joint independence,  $[\text{HairEye}][\text{Sex}]$ , which asserts that the combinations of Hair color and Eye color are independent of Sex. This model, and the corresponding mosaic plot does *not* show the (overall) association between Hair color and Eye color we explored in earlier examples (see Figure 5.3). It merely shows how where the Hair color–Eye color combinations might differ by Sex.

In the call to `mosaic()` below, the model of joint independence is specified as the argument `expected = ~ Hair*Eye + Sex`. The `strucplot` labeling function `labeling_residuals` is used to display the residuals in the highlighted cells.

```
HEC <- HairEyeColor[, c("Brown", "Hazel", "Green", "Blue"),]
mosaic(HEC, expected = ~ Hair*Eye + Sex,
       labeling=labeling_residuals, digits=2)

## Error in eval(expr, envir, enclos): could not find function "mosaic"
```

In Figure 5.11 it is easy to see that there is no systematic association between sex and the combinations of Hair and Eye color—except among blue-eyed blonds, where there are an overabundance of females.

The model of joint independence has a non-significant Pearson  $\chi^2(15) = 19.567, p = 0.189$ . Yet, the two largest residuals highlighted in the plot account for nearly half ( $-2.15^2 + 2.03^2 = 8.74$ ) of the lack of fit, and so are worthy of attention here. An easy (probably facile) interpretation is that among the blue-eyed blonds, some of the females benefited from hair products.  $\triangle$

### 5.4.1 Fitting models

{sec:mosaic-fitting}

When three or more variables are represented in a table, we can fit several different models of types of “independence” and display the residuals from each model. We treat these models as null or *baseline models*, which may not fit the data particularly well. The deviations of observed frequencies from expected ones, displayed by shading, will often suggest terms to be added to an explanatory model that achieves a better fit.

For a three-way table, with variables  $A$ ,  $B$  and  $C$ , some of the hypothesized models which can be fit are described below and summarized in Table 5.2. Here we use  $[\bullet]$  notation to list the **high-order terms** in a hierarchical loglinear model; these correspond to the margins of the table which are fitted exactly, and which translate directly into R formulas used in `loglm()` and `mosaic(..., expected=)`. **TODO: Tweak the association diagrams here to use smaller circles, allowing longer connecting lines.**

The notation  $[AB][AC]$ , for example, is shorthand for the model  $\text{loglm}(\sim A*B + A*C)$  that implies

$$\log m_{ijk} = \mu + \lambda_i^A + \lambda_j^B + \lambda_k^C + \lambda_{ij}^{AB} + \lambda_{ik}^{AC} \quad (5.1) \quad \{\text{eq:AB-AC}\}$$

(as described in Section 8.2) and reproduces the  $\{AB\}$  and  $\{AC\}$  marginal subtables.<sup>4</sup> That is, the calculated expected frequencies in these margins are always equal to the corresponding observed frequencies,  $m_{ij+} = n_{ij+}$  and  $m_{i+k} = n_{i+k}$ .

Table 5.2: Fitted margins, model symbols and interpretations for some hypotheses for a three-way table.

{tab:hyp3way}

Hypothesis	Fitted margins	Model symbol	Independence interpretation	Association graph
$H_1$	$n_{i++}, n_{+j+}, n_{++k}$	$[A][B][C]$	$A \perp B \perp C$	
$H_2$	$n_{ij+}, n_{++k}$	$[AB][C]$	$(A, B) \perp C$	
$H_3$	$n_{i+k}, n_{+jk}$	$[AC][BC]$	$A \perp B \mid C$	
$H_4$	$n_{ij+}, n_{i+k}, n_{+jk}$	$[AB][AC][BC]$	NA	

In this table,  $A \perp B$  is read, “ $A$  is independent of  $B$ .” The independence interpretation of the model Eqn. (5.1) is  $B \perp C \mid A$ , which can be read as “ $B$  is independent of  $C$ , given (conditional on)  $A$ .” Table 5.2 also depicts the relations among variables as an **association graph**, where associated variables are connected by an edge and variables that are asserted to be independent are unconnected. In mosaic-like displays, other associations present in the data will appear in the pattern of residuals.

For a three-way table, there are four general classes of independence models illustrated in

<sup>4</sup>The notation here uses curly braces,  $\{\bullet\}$  to indicate a marginal subtable summed over all other variables.

Table 5.2, as described below.<sup>5</sup> Not included here is the *saturated model*,  $[ABC]$ , which fits the observed data exactly.

**$H_1$ : Complete independence.** The model of complete (mutual) independence, symbolized  $A \perp B \perp C$ , with model formula  $\sim A + B + C$ , asserts that all joint probabilities are products of the one-way marginal probabilities:

$$\pi_{ijk} = \pi_{i++} \pi_{+j+} \pi_{++k} ,$$

for all  $i, j, k$  in a three-way table. This corresponds to the log-linear model  $[A][B][C]$ . Fitting this model puts all higher terms, and hence all association among the variables, into the residuals.

**$H_2$ : Joint independence.** Another possibility is to fit the model in which variable  $C$  is jointly independent of variables  $A$  and  $B$ , ( $\{A, B\} \perp C$ ), with model formula  $\sim A*B + C$ , where

$$\pi_{ijk} = \pi_{ij+} \pi_{++k} .$$

This corresponds to the loglinear model  $[AB][C]$ . Residuals from this model show the extent to which variable  $C$  is related to the combinations of variables  $A$  and  $B$  but they do not show any association between  $A$  and  $B$ , since that association is fitted exactly. For this model, variable  $C$  is also independent of  $A$  and  $B$  in the marginal  $\{AC\}$  table (collapsing over  $B$ ) and in the marginal  $\{BC\}$ .

**$H_3$ : Conditional independence.** Two variables, say  $A$  and  $B$  are conditionally independent given the third ( $C$ ) if  $A$  and  $B$  are independent when we control for  $C$ , symbolized as  $A \perp B | C$ , and model formula  $\sim A*C + B*C$ . This means that conditional probabilities,  $\pi_{ij|k}$  obey

$$\pi_{ij|k} = \pi_{i+|k} \pi_{+j|k} ,$$

where  $\pi_{ij|k} = \pi_{ijk}/\pi_{ij+}$ ,  $\pi_{i+|k} = \pi_{i+k}/\pi_{i++}$ , and  $\pi_{+j|k} = \pi_{+jk}/\pi_{+j+}$ . The corresponding loglinear models is denoted  $[AC][BC]$ . When this model is fit, the mosaic display shows the conditional associations between variables  $A$  and  $B$ , controlling for  $C$ , but does not show the associations between  $A$  and  $C$ , or  $B$  and  $C$ .

**$H_4$ : No three-way interaction.** For this model, no pair is marginally or conditionally independent, so there is *no* independence interpretation. Nor is there a closed-form expression for the cell probabilities. However, the association between any two variables is the same at each level of the third variable. The corresponding loglinear model formula is  $[AB][AC][BC]$ , indicating that all two-way margins are fit exactly and so only the three-way association is shown in the mosaic residuals.

**TODO:** Add a textbox or text describing the general scheme for translating among loglinear shorthand, R model formulas and independence interpretations.

{ex:HEC2}

#### EXAMPLE 5.7: Hair color, eye color and sex

We continue with the analysis of the `HairEyeColor` data from Example 5.6. Figure 5.11 showed the fit of the joint-independence model  $[HairEye][Sex]$ , testing whether the joint distribution of hair color and eye color is associated with sex.

<sup>5</sup>For  $H_2$  and  $H_3$ , permutation of the variables  $A$ ,  $B$ , and  $C$  gives other members of each class.



Any other model fit to this table will have the same size tiles in the mosaic since the areas depend on the observed frequencies; the residuals, and hence the shading of the tiles will differ. Figure 5.12 shows mosaics for two other models. Shading in the left panel shows residuals from the model of mutual independence, [Hair][Eye][Sex], and so includes all sources of association among these three variables. The right panel shows the conditional independence model, [Hair-Sex][EyeSex] testing whether, given sex, hair color and eye color are independent. Note that the pattern of residuals here is similar to that in the two-way display, Figure 5.4, that collapsed over sex.

```
abbrev <- list(abbreviate=c(FALSE, FALSE, 1))
mosaic(HEC, expected = ~ Hair + Eye + Sex, labeling_args=abbrev,
       main="Model: ~Hair + Eye + Sex")

## Error in eval(expr, envir, enclos): could not find function "mosaic"

mosaic(HEC, expected = ~ Hair*Sex + Eye*Sex, labeling_args=abbrev,
       main="Model: ~Hair*Sex + Eye*Sex")

## Error in eval(expr, envir, enclos): could not find function "mosaic"
```

Compared with Figure 5.11 for the joint independence model, [HairEye][Sex], it is easy to see that both of these models fit very poorly.

We consider loglinear models in more detail in Chapter 8, but for now note that these models are fit using `loglm()` in the `MASS` package, with the model formula given in the `expected` argument. The details of these models can be seen by fitting these models explicitly, and the fit of several models can be summarized compactly using `Summarise()` in `vcdExtra`.

```
library(MASS)
mod1 <- loglm(~ Hair + Eye + Sex, data=HEC)      # mutual independence
mod2 <- loglm(~ Hair*Sex + Eye*Sex, data=HEC)   # conditional independence
mod3 <- loglm(~ Hair*Eye + Sex, data=HEC)      # joint independence
Summarise(mod1, mod2, mod3)

## Error in eval(expr, envir, enclos): could not find function "Summarise"
```

Alternatively, you can get the Pearson and likelihood ratio (LR) tests for a given model using `anova()`, or compare a set of models using LR tests on the *difference* in LR  $\chi^2$  from one model to the next, when a list of models is supplied to `anova()`.

```
anova(mod1)

## Call:
## loglm(formula = ~Hair + Eye + Sex, data = HEC)
##
## Statistics:
##              X^2 df P(> X^2)
## Likelihood Ratio 166.30 24      0
## Pearson          164.92 24      0

anova(mod1, mod2, mod3, test="chisq")

## LR tests for hierarchical log-linear models
##
```

```
## Model 1:
## ~Hair + Eye + Sex
## Model 2:
## ~Hair * Sex + Eye * Sex
## Model 3:
## ~Hair * Eye + Sex
##
##           Deviance df Delta(Dev) Delta(df) P(> Delta(Dev))
## Model 1    166.300  24
## Model 2    156.678  18      9.6222      6      0.14149
## Model 3     19.857  15    136.8213      3      0.00000
## Saturated   0.000   0     19.8566     15      0.17750
```

△

## 5.4.2 Sequential plots and models

{sec:mosaic-seq}

As described in Section 5.2, we can think of the mosaic display for an  $n$ -way table as being constructed in stages, with the variables listed in a given order, and the unit tile decomposed recursively as each variable is entered in turn. This process turns out to have the useful property that it provides an additive (hierarchical) decomposition of the total association in a table, in a way analogous to sequential fitting with Type I sum of squares in regression models.

Typically, we just view the mosaic and fit models to the full  $n$ -way table, but it is useful to understand the connection with models for the marginal subtables, defined by summing over all variables not yet entered. For example for a three-way table with variables,  $A, B, C$ , the marginal subtables  $\{A\}$  and  $\{AB\}$  are calculated in the process of constructing the three-way mosaic. The  $\{A\}$  marginal table can be fit to a model where the categories of variable  $A$  are equiprobable as shown in Figure 5.2 (or some other discrete distribution); the independence model can be fit to the  $\{AB\}$  subtable as in Figure 5.2 and so forth.

This connection can be seen in the following formula that decomposes the joint cell probability in an  $n$ -way table with variables  $v_1, v_2, \dots, v_n$  as a sequential product of conditional probabilities,

$$\{eq:seqprod\} \quad p_{ijkl\dots} = \underbrace{p_i \times p_{j|i} \times p_{k|ij}}_{\{v_1 v_2 v_3\}} \times p_{l|ijk} \times \cdots \times p_{n|ijk\dots} \quad (5.2)$$

In Eqn. (5.2), the first term corresponds to the one-way mosaic for  $v_1$ , the first two terms to the mosaic for  $v_1$  and  $v_2$ , the first three terms to the mosaic for  $v_1, v_2$  and  $v_3$ , and so forth.

It can be shown (Friendly, 1994) that this sequential product of probabilities corresponds to a set of sequential models of *joint independence*, whose likelihood ratio  $G^2$  statistics provide an additive decomposition of the total association,  $G^2_{[v_1][v_2]\dots[v_n]}$  for the mutual independence model in the full table:

$$\{eq:seqgsq\} \quad G^2_{[v_1][v_2]\dots[v_n]} = G^2_{[v_1][v_2]} + G^2_{[v_1 v_2][v_3]} + G^2_{[v_1 v_2 v_3][v_4]} + \cdots + G^2_{[v_1 \dots v_{n-1}][v_n]} \quad (5.3)$$

For example, for the hair-eye data, the mosaic displays for the [Hair] [Eye] marginal table (Figure 5.4) and the [HairEye] [Sex] table (Figure 5.11) can be viewed as representing the partition of  $G^2$  shown as a table below:

Model	Model symbol	df	$G^2$
Marginal	[Hair] [Eye]	9	146.44
Joint	[Hair, Eye] [Sex]	15	19.86
Mutual	[Hair] [Eye] [Sex]	24	166.30

```
## Error in eval(expr, envir, enclos): could not find function "mosaic"
```

```
## Error in eval(expr, envir, enclos): could not find function "mosaic"
```

```
## Error in eval(expr, envir, enclos): could not find function "mosaic"
```

The decomposition in this table reflecting Eqn. (5.3) is shown as a visual equation in Figure 5.3. You can see from the shading how the two sequential submodels contribute to overall association in the model of mutual independence.

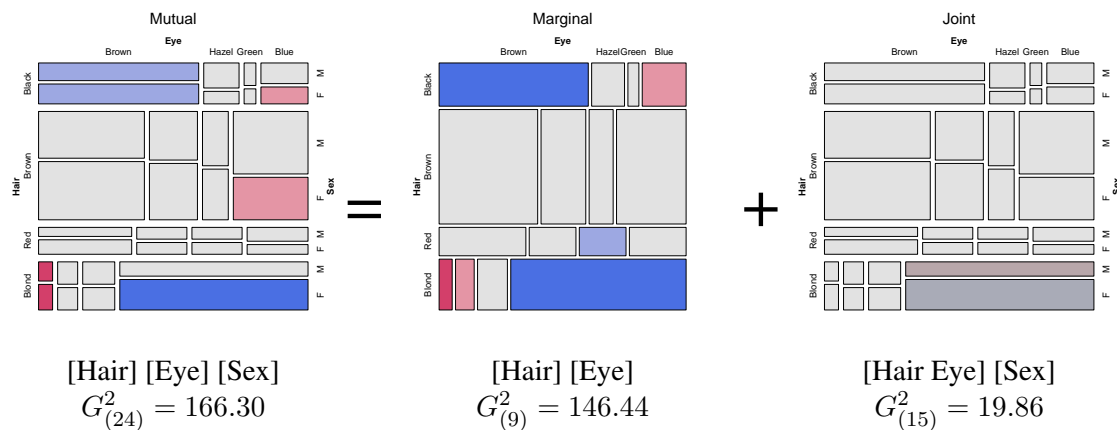


Figure 5.3: Visual representation of the decomposition of the  $G^2$  for mutual independence (total) as the sum of marginal and joint independence.

{fig:HEC-seq}

Although sequential models of joint independence have the nice additive property illustrated above, other classes of sequential models are possible, and sometimes of substantive interest. The main types of these models are illustrated in Table 5.3 for 3-, 4-, and 5- way tables, with variables A, B, ... E. In all cases, the natural model for the one-way margin is the equiprobability model, and that for the two-way margin is  $[A][B]$ .

The `vcdExtra` package provides a collection of convenience functions that generate the log-linear model formulae symbolically, as indicated in the **function** column. The functions `mutual()`, `joint()`, `conditional()`, `markov()` and so forth simply generate a list of terms suitable for a model formula for `loglin()`. See `help(loglin-utilities)` for further details.

Wrapper functions `loglin2string()` and `loglin2formula()` convert these to character strings or model formulae respectively, for use with `loglm()` and `mosaic()`-related functions in `vcdExtra`. Some examples are shown below.

```
for(nf in 2:5) {
  print(loglin2string(joint(nf, factors=LETTERS[1:5])))
}
```

Table 5.3: Classes of sequential models for *n*-way tables

{tab:seqmod

function	3-way	4-way	5-way
mutual	[A] [B] [C]	[A] [B] [C] [D]	[A] [B] [C] [D] [E]
joint	[AB] [C]	[ABC] [D]	[ABCE] [E]
joint (with=1)	[A] [BC]	[A] [BCD]	[A] [BCDE]
conditional	[AC] [BC]	[AD] [BD] [CD]	[AE] [BE] [CE] [DE]
conditional (with=1)	[AB] [AC]	[AB] [AC] [AD]	[AB] [AC] [AD] [AE]
markov (order=1)	[AB] [BC]	[AB] [BC] [CD]	[AB] [BC] [CD] [DE]
markov (order=2)	[A] [B] [C]	[ABC] [BCD]	[ABC] [BCD] [CDE]
saturated	[ABC]	[ABCD]	[ABCDE]

```
## Error in print(loglin2string(joint(nf, factors = LETTERS[1:5]))): could
not find function "loglin2string"

for(nf in 2:5) {
  print(loglin2string(conditional(nf, factors=LETTERS[1:5]), sep=""))
}

## Error in print(loglin2string(conditional(nf, factors = LETTERS[1:5]),
: could not find function "loglin2string"

for(nf in 2:5) {
  print(loglin2formula(conditional(nf, factors=LETTERS[1:5])))
}

## Error in print(loglin2formula(conditional(nf, factors = LETTERS[1:5]))):
could not find function "loglin2formula"
```

Applied to data, these functions take a `table` argument, and deliver the string or formula representation of a type of model for that table:

```
loglin2formula(joint(3, table=HEC))

## Error in eval(expr, envir, enclos): could not find function "loglin2formula"

loglin2string(joint(3, table=HEC))

## Error in eval(expr, envir, enclos): could not find function "loglin2string"
```

Their main use, however, is within higher-level functions, such as `seq_loglm()`, which fit the collection of sequential models of a given type.

```
HEC.mods <- seq_loglm(HEC, type="joint")

## Error in eval(expr, envir, enclos): could not find function "seq_loglm"

Summarise(HEC.mods)

## Error in eval(expr, envir, enclos): could not find function "Summarise"
```

In this section we have described a variety of models which can be fit to higher-way tables, some relations among those models, and the aspects of lack-of-fit which are revealed in the mosaic displays. The following examples illustrate the process of model fitting, using the mosaic as an interpretive guide to the nature of associations among the variables. In general, we start with a minimal baseline model.<sup>6</sup> The pattern of residuals in the mosaic will suggest associations to be added to an adequate explanatory model. As the model achieves better fit to the data, the degree of shading decreases, so we may think of the process of model fitting as “cleaning the mosaic.”

### 5.4.3 Causal models

{sec:causal}

The sequence of models of joint independence has another interpretation when the ordering of the variables is based on a set of ordered hypotheses involving causal relationships among variables (Goodman (1973), Fienberg (1980, §7.2)). Suppose, for example, that the causal ordering of four variables is  $A \rightarrow B \rightarrow C \rightarrow D$ , where the arrow means “is antecedent to.” Goodman suggests that the conditional joint probabilities of  $B$ ,  $C$ , and  $D$  given  $A$  can be characterized by a set of recursive logit models which treat (a)  $B$  as a response to  $A$ , (b)  $C$  as a response to  $A$  and  $B$  jointly, (c) and  $D$  as a response to  $A$ ,  $B$  and  $C$ . These are equivalent to the loglinear models which we fit as the sequential baseline models of joint independence, namely  $[A][B]$ ,  $[AB][C]$ , and  $[ABC][D]$ . The combination of these models with the marginal probabilities of  $A$  gives a characterization of the joint probabilities of all four variables, as in Eqn. (5.2). In application, residuals from each submodel show the associations that remain unexplained.

{ex:marital1}

#### EXAMPLE 5.8: Marital status and pre- and extramarital sex

A study of divorce patterns in relation to premarital and extramarital sex by Thornes and Collard (1979) reported the  $2^4$  table shown below, and included in `vcd` as `PreSex`.

```
data("PreSex", package="vcd")
structable(Gender+PremaritalSex+ExtramaritalSex ~ MaritalStatus, PreSex)

## Error in eval(expr, envir, enclos): could not find function "structable"
```

These data were analysed by Agresti (2013, §6.1.7) and by Friendly (1994, 2000), from which this account draws. A sample of about 500 people who had petitioned for divorce, and a similar number of married people were asked two questions regarding their pre- and extramarital sexual experience: (1) “Before you married your (former) husband/wife, had you ever made love with anyone else?” (2) “During your (former) marriage (did you) have you had any affairs or brief sexual encounters with another man/woman?” The table variables are thus gender ( $G$ ), reported premarital ( $P$ ) and extramarital ( $E$ ) sex, and current marital status ( $M$ ).

In this analysis we consider the variables in the order  $G$ ,  $P$ ,  $E$ , and  $M$ , and first reorder the table variables for convenience.

```
PreSex <- aperm(PreSex, 4:1) # order variables G, P, E, M
```

That is, the first stage treats  $P$  as a response to  $G$  and examines the  $[Gender][Pre]$  mosaic to assess whether gender has an effect on premarital sex. The second stage treats  $E$  as a response to

<sup>6</sup>When one variable,  $R$  is a response, this normally is the model of joint independence,  $[E_1 E_2 \dots][R]$ , where  $E_1, E_2, \dots$  are the explanatory variables. Better-fitting models will often include associations of the form  $[E_i R]$ ,  $[E_i E_j R] \dots$

$G$  and  $P$  jointly; the mosaic for [Gender, Pre] [Extra] shows whether extramarital sex is related to either gender or premarital sex. These are shown in Figure 5.14.

```
# (Gender Pre)
mosaic(margin.table(PreSex, 1:2), shade=TRUE,
       main = "Gender and Premarital Sex")

## Error in eval(expr, envir, enclos): could not find function "mosaic"

## (Gender Pre) (Extra)
mosaic(margin.table(PreSex, 1:3),
       expected = ~Gender * PremaritalSex + ExtramaritalSex ,
       main = "Gender*Pre + ExtramaritalSex")

## Error in eval(expr, envir, enclos): could not find function "mosaic"
```

Finally, the mosaic for [Gender, Pre, Extra] [Marital] is examined for evidence of the dependence of marital status on the three previous variables jointly. As noted above, these models are equivalent to the recursive logit models whose path diagram is  $G \rightarrow P \rightarrow E \rightarrow M$ .<sup>7</sup> The  $G^2$  values for these models shown below provide a decomposition of the  $G^2$  for the model of complete independence fit to the full table.

Model	df	$G^2$
[G] [P]	1	75.259
[GP] [E]	3	48.929
[GPE] [M]	7	107.956
[G] [P] [E] [M]	11	232.142

The [Gender] [Pre] mosaic in the left panel of Figure 5.14 shows that men are much more likely to report premarital sex than are women; the sample odds ratio is 3.7. We also see that women are about twice as prevalent as men in this sample. The mosaic for the model of joint independence, [Gender Pre] [Extra] in the right panel of Figure 5.14 shows that extramarital sex depends on gender and premarital sex jointly. From the pattern of residuals in Figure 5.14 we see that men and women who have reported premarital sex are far more likely to report extramarital sex than those who have not. In this three-way marginal table, the conditional odds ratio of extramarital sex given premarital sex is nearly the same for both genders (3.61 for men and 3.56 for women). Thus, extramarital sex depends on premarital sex, but not on gender.

```
oddsratio(margin.table(PreSex, 1:3), stratum=1, log=FALSE)

## Error in eval(expr, envir, enclos): could not find function "oddsratio"

## (Gender Pre Extra) (Marital)
mosaic(PreSex,
       expected = ~Gender*PremaritalSex*ExtramaritalSex
             + MaritalStatus,
       main = "Gender*Pre*Extra + MaritalStatus")

## Error in eval(expr, envir, enclos): could not find function "mosaic"
```

<sup>7</sup>Agresti (2013, Figure 6.1) considers a slightly more complex, but more realistic model in which premarital sex affects both the propensity to have extramarital sex and subsequent marital status.

```
## (GPE) (PEM)
mosaic(PreSex,
  expected = ~ Gender * PremaritalSex * ExtramaritalSex
          + MaritalStatus * PremaritalSex * ExtramaritalSex,
  main = "G*P*E + P*E*M")

## Error in eval(expr, envir, enclos): could not find function "mosaic"
```

△

**TODO:** Complete this section with other examples: Titanic

### 5.4.4 Partial association

{sec:mospart}

In a three-way (or larger) table it may be that two variables, say  $A$  and  $B$ , are associated at some levels of the third variable,  $C$ , but not at other levels of  $C$ . More generally, we may wish to explore whether and how the association among two (or more) variables in a contingency table varies over the levels of the remaining variables. The term *partial association* refers to the association among some variables within the levels of the other variables.

Partial association represents a useful “divide and conquer” statistical strategy: it allows you to refine the question you want to answer for complex relations by breaking it down to smaller, easier questions.<sup>8</sup> It is a statistically happy fact that an answer to the larger, more complex question can be expressed as an algebraic sum of the answers to the smaller questions, just as was the case with sequential models of joint independence.

For concreteness, consider the case where you want to understand the relationship between *attitude* toward corporal punishment of children by parents or teachers (Never, Moderate use OK) and *memory* that the respondent had experiences corporal punishment as a child (Yes, No). But you also have measured other variables on the respondents, including their level of *education* and *age* category. In this case, the question of association among all the table variables may be complex, but we can answer a highly relevant, specialized question precisely, “is there an association between attitude and memory, *controlling for education and age*?” The answer to this question can be thought of as the sum of the answers to the simpler question of association between attitude and memory across the education, age categories.

A simpler version of this idea is considered first below (Example 5.9): among workers who were laid off due to either the closure of a plant or business vs. replacement by another worker, the (conditional) relationship of employment status (new job vs. still unemployed) and duration of unemployment can be studied as a sum of the associations between these focal variables over the separate tables for cause of layoff.

To make this idea precise, consider for example the model of conditional independence,  $A \perp B | C$  for a three-way table. This model asserts that  $A$  and  $B$  are independent within *each* level of  $C$ . Denote the hypothesis that  $A$  and  $B$  are independent at level  $C(k)$  by  $A \perp B | C(k), k = 1, \dots, K$ . Then one can show (Andersen, 1991) that

$$G_{A \perp B | C}^2 = \sum_k^K G_{A \perp B | C(k)}^2 \quad (5.4) \quad \{\text{eq:partial1}\}$$

<sup>8</sup>This is an analog, for categorical data, of the ANOVA strategy for “probing interactions” by testing *simple effects* at the levels of one or more of the factors involved in a two- or higher-way interaction.

That is, the overall likelihood ratio  $G^2$  for the conditional independence model with  $(I-1)(J-1)K$  degrees of freedom is the sum of the values for the ordinary association between  $A$  and  $B$  over the levels of  $C$  (each with  $(I-1)(J-1)$  degrees of freedom). The same additive relationship holds for the Pearson  $\chi^2$  statistics:  $\chi^2_{A \perp B | C} = \sum_k^K \chi^2_{A \perp B | C(k)}$ .

Thus, (a) the overall  $G^2$  ( $\chi^2$ ) may be decomposed into portions attributable to the  $AB$  association in the layers of  $C$ , and (b) the collection of mosaic displays for the dependence of  $A$  and  $B$  for each of the levels of  $C$  provides a natural visualization of this decomposition. These provide an analog, for categorical data, of the conditioning plot, or *coplot*, that Cleveland (1993) has shown to be an effective display for quantitative data. See Friendly (1999a) for further details.

Mosaic and other displays in the *strucplot* framework for partial association can be produced in several different ways. One way is to use a model formula in the call to `mosaic()` which lists the conditioning variables after the "|" (given) symbol, as in `~ Memory + Attitude | Age + Education`. Another way is to use `cotabplot()`. This takes the same kind of conditioning model formula, but presents each panel for the conditioning variables in a separate frame within a trellis-like grid.<sup>9</sup>

{ex:employ}

### EXAMPLE 5.9: Employment status data

Data from a 1974 Danish study of 1314 employees who had been laid off are given in the data table `Employment` in `vcd` (from Andersen (1991, Table 5.12)). The workers are classified by: (a) their employment status, on January 1, 1975 ("NewJob" or still "Unemployed"), (b) the length of their employment at the time of layoff, (c) the cause of their layoff ("Closure", etc. or "Replaced").

```
data("Employment", package = "vcd")
structable(Employment)
```

```
## Error in eval(expr, envir, enclos): could not find function "structable"
```

In this example, it is natural to regard `EmploymentStatus` (variable  $A$ ) as the response variable, and `EmploymentLength` ( $B$ ) and `LayoffCause` ( $C$ ) as predictors. In this case, the minimal baseline model is the joint independence model,  $[A][BC]$ , which asserts that employment status is independent of both length and cause. This model fits quite poorly, as shown in the output from `loglm()` below.

```
loglm(~ EmploymentStatus + EmploymentLength*LayoffCause, data=Employment)
```

```
## Call:
## loglm(formula = ~EmploymentStatus + EmploymentLength * LayoffCause,
##       data = Employment)
##
## Statistics:
##               X^2 df P(> X^2)
## Likelihood Ratio 172.28 11      0
## Pearson          165.70 11      0
```

The residuals, shown in Figure 5.16, indicate an opposite pattern for the two categories of `LayoffCause`: those who were laid off as a result of a closure are more likely to be unemployed, regardless of length of time they were employed. Workers who were replaced, however,

<sup>9</sup>Depending on your perspective, this has the advantage of adjusting for the total frequency in each conditional panel, or the disadvantage of ignoring these differences.



apparently are more likely to be employed, particularly if they were employed for 3 months or more.

```
# baseline model [A][BC]
mosaic(Employment, shade=TRUE,
       expected = ~ EmploymentStatus + EmploymentLength*LayoffCause,
       main = "EmploymentStatus + Length * Cause")

## Error in eval(expr, envir, enclos): could not find function "mosaic"
```

Beyond this baseline model, it is substantively more meaningful to consider the conditional independence model,  $A \perp B | C$ , (or [AC][BC] in shorthand notation), which asserts that employment status is independent of length of employment, given the cause of layoff. We fit this model as shown below:

```
loglm(~ EmploymentStatus*LayoffCause + EmploymentLength*LayoffCause,
      data=Employment)

## Call:
## loglm(formula = ~EmploymentStatus * LayoffCause + EmploymentLength *
##       LayoffCause, data = Employment)
##
## Statistics:
##           X^2 df  P(> X^2)
## Likelihood Ratio 24.630 10 0.0060927
## Pearson          26.072 10 0.0036445
```

This model fits far better ( $G^2(10) = 24.63$ ), but the lack of fit is still significant. The residuals, shown in Figure 5.17, still suggest that the pattern of association between employment and length is different for replaced workers and those laid off due to closure of their workplace.

```
mosaic(Employment, shade=TRUE, gp_args=list(interpolate=1:4),
       expected = ~ EmploymentStatus*LayoffCause + EmploymentLength*LayoffCause,
       main = "EmploymentStatus * Cause + Length * Cause")

## Error in eval(expr, envir, enclos): could not find function "mosaic"
```

To explain this result better, we can fit separate models for the *partial* relationship between EmploymentStatus and EmploymentLength for the two levels of LayoffCause. In R, with the Employment data as in table form, this is easily done using apply() over the LayoffCause margin, giving a list containing the two loglm() models.

```
mods.list <- apply(Employment, "LayoffCause",
                  function(x) loglm(~EmploymentStatus + EmploymentLength, data=x))
mods.list

## $Closure
## Call:
## loglm(formula = ~EmploymentStatus + EmploymentLength, data = x)
##
## Statistics:
##           X^2 df  P(> X^2)
## Likelihood Ratio 1.4786  5 0.91553
## Pearson          1.4835  5 0.91497
##
```

```
## $Replaced
## Call:
## loglm(formula = ~EmploymentStatus + EmploymentLength, data = x)
##
## Statistics:
##
##           X^2 df    P(> X^2)
## Likelihood Ratio 23.151  5 0.00031578
## Pearson          24.589  5 0.00016727
```

Extracting the model fit statistics for these partial models and adding the fit statistics for the overall model of conditional independence,  $[AC][BC]$  gives the table below, illustrating the additive property of  $G^2$ , (Eqn. (5.4)) and  $\chi^2$ .

Model	df	$G^2$	$\chi^2$
$A \perp B   C_1$	5	1.49	1.48
$A \perp B   C_2$	5	23.15	24.59
$A \perp B   C$	10	24.63	26.07

One simple way to visualize these results is to call `mosaic()` separately for each of the layers corresponding to `LayoffCause`. The result is shown in Figure 5.18.

```
mosaic(Employment[,,"Closure"], shade=TRUE, gp_args=list(interpolate=1:4),
       margin = c(right = 1), main = "Layoff: Closure")

## Error in eval(expr, envir, enclos): could not find function "mosaic"

mosaic(Employment[,,"Replaced"], shade=TRUE, gp_args=list(interpolate=1:4),
       margin = c(right = 1), main = "Layoff: Replaced")

## Error in eval(expr, envir, enclos): could not find function "mosaic"
```

The simple summary from this example is that for workers laid off due to closure of their company, length of previous employment is unrelated to whether or not they are re-employed. However, for workers who were replaced, there is a systematic pattern: those who had been employed for three months or less are likely to remain unemployed, while those with longer job tenure are somewhat more likely to have found a new job.  $\triangle$

The statistical methods and R techniques described above for three-way tables extend naturally to higher-way tables, as can be seen in the next example.

{ex:punish}

### EXAMPLE 5.10: Corporal punishment data

Here we use the `Punishment` data from `vcd` which contains the results of a study by the Gallup Institute in Denmark in 1979 about the attitude of a random sample of 1,456 persons towards corporal punishment of children (Andersen, 1991, pp. 207-208). As shown below, this data set is a frequency data frame representing a  $2 \times 2 \times 3 \times 3$  table, with table variables (a) attitude toward use of corporal punishment (approve of “moderate” use or “no” approval) (b) memory of whether the respondent had experienced corporal punishment as a child (yes/no); (c) education level of respondent (elementary, secondary, high); (d) age category of respondent.

```
data("Punishment", package = "vcd")
str(Punishment)
```

```
## 'data.frame': 36 obs. of 5 variables:
## $ Freq : num 1 3 20 2 8 4 2 6 1 26 ...
## $ attitude : Factor w/ 2 levels "no","moderate": 1 1 1 1 1 1 1 1 1 1 ...
## $ memory : Factor w/ 2 levels "yes","no": 1 1 1 1 1 1 1 1 1 2 ...
## $ education: Factor w/ 3 levels "elementary","secondary",...: 1 1 1 2 2 2 3 3 3 1 ...
## $ age : Factor w/ 3 levels "15-24","25-39",...: 1 2 3 1 2 3 1 2 3 1 ...
```

Of main interest here is the association between attitude toward corporal punishment as an adult ( $A$ ) and memory of corporal punishment as a child ( $B$ ), controlling for age ( $C$ ) and education ( $D$ ); that is, the model  $A \perp B \mid (C, D)$ , or  $[ACD][BCD]$  in shorthand notation.

As noted above, this conditional independence hypothesis can be decomposed into the  $3 \times 3$  partial tests of  $A \perp B \mid (C_k, D_\ell)$ .

These tests and the associated graphics are somewhat easier to carry out with the data in table form (pun) constructed below. While we're at it, we recode the variable names and factor levels for nicer graphical displays.

```
pun <- xtabs(Freq ~ memory + attitude + age + education, data = Punishment)
dimnames(pun) <- list(
  Memory = c("yes", "no"),
  Attitude = c("no", "moderate"),
  Age = c("15-24", "25-39", "40+"),
  Education = c("Elementary", "Secondary", "High"))
```

Then, the overall test of conditional independence can be carried using `loglm()` out as

```
(mod.cond <- loglm(~ Memory*Age*Education + Attitude*Age*Education,
  data = pun))

## Call:
## loglm(formula = ~Memory * Age * Education + Attitude * Age *
##       Education, data = pun)
##
## Statistics:
##
##           X^2 df      P(> X^2)
## Likelihood Ratio 39.679   9 8.6851e-06
## Pearson          34.604   9 6.9964e-05
```

Alternatively, `coindep_test()` in `vcd` provides tests of conditional independence of two variables in a contingency table by simulation from the marginal permutation distribution of the input table. The version reporting a Pearson  $\chi^2$  statistic is given by

```
set.seed(1071)
coindep_test(pun, margin=c("Age", "Education"),
  indepfun = function(x) sum(x^2), aggfun=sum)

## Error in eval(expr, envir, enclos): could not find function "coindep_test"
```

These tests all show substantial association between attitude and memory of corporal punishment. How can we understand and explain this?

As in Example 5.9, we can partition the overall  $G^2$  or  $\chi^2$  to show the contributions to this association from the combinations of age and education. The call to `apply()` below returns a  $3 \times 3$  matrix, each of whose elements is the list of results returned by `loglm()`. The Pearson  $\chi^2$  statistics for each subtable can be extracted using `sapply()` as shown below.

```

mods.list <- apply(pun, c("Age", "Education"),
  function(x) loglm(~Memory + Attitude, data=x))
XSQ <- matrix(sapply(mods.list, function(x)x$spearson), 3, 3)
dimnames(XSQ) <- dimnames(mods.list)
addmargins(XSQ)

```

```

##           Education
## Age      Elementary Secondary      High      Sum
## 15-24      3.5907  0.078782 0.091429  3.7609
## 25-39      8.5844  0.934669 0.480000  9.9990
## 40+       11.6256  6.094854 3.123714 20.8442
## Sum       23.8006  7.108305 3.695143 34.6041

```

One visual analog of this table of  $\chi^2$  statistics is a `cotabplot()` of the (conditional) association of attitude and memory over the age and education cells, shown in Figure 5.19. `cotabplot()` is very general, allowing a variety of functions of the residuals to be used for shading (Zeileis *et al.*, 2007). Here we use the (Pearson) sum of squares statistic,  $\sum_{k,\ell} \chi_{k,\ell}^2$ .

```

set.seed(1071)
pun_cotab <- cotab_coindep(pun, condvars = 3:4, type = "mosaic",
  varnames = FALSE, margins = c(2, 1, 1, 2),
  test = "sumchisq", interpolate = 1:2)

## Error in eval(expr, envir, enclos): could not find function "cotab_coindep"

cotabplot(~ Memory + Attitude | Age + Education, data =
  pun, panel = pun_cotab)

## Error in eval(expr, envir, enclos): could not find function "cotabplot"

```

Alternatively, the pattern of conditional association can be shown somewhat more directly in a conditional mosaic plot (Figure 5.20), using the same model formula to condition on age and education. This simply organizes the display to split on the conditioning variables first, with larger spacings.

```

mosaic(~ Memory + Attitude | Age + Education, data = pun,
  shade=TRUE, gp_args=list(interpolate=1:4))

## Error in eval(expr, envir, enclos): could not find function "mosaic"

```

Both Figure 5.19 and Figure 5.20 reveal that the association between attitude and memory becomes stronger with increasing age among those with the lowest education (first column). Among those in the highest age group (bottom row), the strength of association *decreases* with increasing education. These two displays differ in that in the `cotabplot()` of Figure 5.19 the marginal frequencies of age and education are not shown, whereas in the `mosaic()` of Figure 5.20 they determine the relative sizes of the tiles for the combinations of age and education.

The divide-and-conquer strategy of partial association using statistical tests and visual displays now provides a simple, coherent explanation for this table: memory of experienced violence as a child tends to engender a more favorable attitude toward corporal punishment as an adult, but this association varies directly with both age and education.  $\triangle$

## 5.5 Mosaic matrices for categorical data

sec:mosmat }

One reason for the wide usefulness of graphs of quantitative data has been the development of effective, general techniques for dealing with high-dimensional data set. The *scatterplot matrix* shows all pairwise (marginal) views of a set of variables in a coherent display, whose design goal is to show the interdependence among the collection of variables as a whole. It combines multiple views of the data into a single display which allows detection of patterns which could not readily be discerned from a series of separate graphs. In effect, a multivariate data set in  $p$  dimensions (variables) is shown as a collection of  $p(p-1)$  two-dimensional scatterplots, each of which is the projection of the cloud of points on two of the variable axes. These ideas can be readily extended to categorical data.

A multiway contingency table of  $p$  categorical variables,  $A, B, C, \dots$ , contains the interdependence among the collection of variables as a whole. The saturated loglinear model,  $[ABC \dots]$  fits this interdependence perfectly, but is often too complex to describe or understand.

By summing the table over all variables except two,  $A$  and  $B$ , say, we obtain a two-variable (marginal) table, showing the bivariate relationship between  $A$  and  $B$ , which is also a projection of the  $p$ -variable relation into the space of two (categorical) variables. If we do this for all  $p(p-1)$  unordered pairs of categorical variables and display each two-variable table as a mosaic, we have a categorical analog of the scatterplot matrix, called a *mosaic matrix*. Like the scatterplot matrix, the mosaic matrix can accommodate any number of variables in principle, but in practice is limited by the resolution of our display to three or four variables.

In R, the main implementation of this idea is in the generic function `pairs()`. The `vcd` package extends this to mosaic matrices with methods for "table" and "structable" objects. The `gpairs` package provides a *generalized pairs plot*, with appropriate graphics for a mixture of quantitative and categorical variables.

{ex:bartlett }

### EXAMPLE 5.11: Bartlett data on plum root cuttings

The simplest example of what you can see in a mosaic matrix is provided by the  $2 \times 2 \times 2$  table used by Bartlett (1935) to illustrate a method for testing for no three-way interaction in a contingency table (hypothesis  $H_4$  in Table 5.2).

The data set `Bartlett` in `vcdExtra` gives the result of an agricultural experiment to investigate the survival of plum root cuttings (*Alive*) in relation to two factors: *Time* of planting and the *Length* of the cutting. In this experiment, 240 cuttings were planted for each of the  $2 \times 2$  combinations of these factors, and their survival (Alive, Dead) was later recorded.

```
pairs(Bartlett, gp=shading_Friendly)

## Error in pairs(Bartlett, gp = shading_Friendly): object 'Bartlett' not found
```

The mosaic matrix for these data, showing all twoway marginal relations, is shown in Figure 5.21. It can immediately be seen that *Time* and *Length* are independent by the design of the experiment; we use `gp=shading_Friendly` here to emphasize this.

The top row and left column show the relation of survival to each of time of planting and cutting length. It is easily seen that greater survival is associated with cuttings taken now (vs. spring) and those cut long (vs. short), and the degree of association is stronger for planting time than for cutting length. △

{ex:marital

**EXAMPLE 5.12: Marital status and pre- and extramarital sex**

In Example 5.8 we examined a series of models relating marital status to reported premarital and extramarital sexual activity and gender in the `PreSex` data. Figure 5.22 shows the mosaic matrix for these data. The diagonal panels show the labels for the category levels as well as the one-way marginal totals.

```
data("PreSex", package="vcd")
pairs(PreSex, gp=shading_Friendly, gp_args=list(interpolate=1:4), space=0.25)

## Error in pairs.table(PreSex, gp = shading_Friendly, gp_args = list(interpolate
= 1:4), : could not find function "grid.newpage"
```

If we view gender, premarital sex and extramarital sex as explanatory, and marital status (Divorced vs. still Married) as the response, then the mosaics in row 1 (and in column 1)<sup>10</sup> shows how marital status depends on each predictor marginally. The remaining panels show the relations within the set of explanatory variables.

Thus, we see in row 1, column 4, that marital status is independent of gender (all residuals equal zero, here), by design of the data collection. In the (1, 3) panel, we see that reported premarital sex is more often followed by divorce, while non-report is more prevalent among those still married. The (1, 2) panel shows a similar, but stronger relation between extramarital sex and marriage stability. These effects pertain to the associations of P and E with marital status (M)—the terms [PM] and [EM] in the loglinear model. We saw earlier that an interaction of P and E (the term [PEM]) is required to fully account for these data. This effect is not displayed in Figure 5.22.

Among the background variables (the loglinear term [GPE]), the (2, 3) panel shows a strong relation between premarital sex and subsequent extramarital sex, while the (2, 4) and (3, 4) panels show that men are far more likely to report premarital sex than women in this sample, and also more likely to report extramarital sex.

Even though the mosaic matrix shows only pairwise, bivariate associations, it provides an integrated view of all of these together in a single display.

△

{ex:berkeley4}

**EXAMPLE 5.13: Berkeley admissions**

In Chapter 4 we examined the relations among the variables `Admit`, `Gender` and `Department` in the Berkeley admissions data (Example 4.1, Example 4.10, Example 4.14) using fourfold displays (Figure 4.3 and Figure 4.4) and sieve diagrams (Figure 4.10). These displays showed either a marginal relation (e.g., `Admit`, `Gender`) or the full three-way table.

In contrast, Figure 5.23 shows all pairwise marginal relations among these variables, produced using `pairs()`. Some additional arguments are used to control the details of labels for the diagonal and off-diagonal panels.

<sup>10</sup>Rows and columns in a mosaic matrix are identified as in a table or numerical matrix, with row 1, column 1 in the upper left corner.

```

largs <- list(labeling = labeling_border(varnames = FALSE,
                                       labels = c(T, T, F, T), alternate_labels = FALSE))

## Error in eval(expr, envir, enclos): could not find function "labeling_border"

dargs <- list(gp_varnames = gpar(fontsize = 20), offset_varnames = -1,
             labeling = labeling_border(alternate_labels = FALSE))

## Error in eval(expr, envir, enclos): could not find function "gpar"

pairs(UCBAdmissions, shade = TRUE, space = 0.25,
      diag_panel_args = dargs,
      upper_panel_args = largs, lower_panel_args = largs)

## Error in pairs.table(UCBAdmissions, shade = TRUE, space = 0.25, diag_panel_args
= dargs, : could not find function "grid.newpage"

```

The panel in row 2, column 1 shows that Admission and Gender are strongly associated marginally, as we saw in Figure 4.3, and overall, males are more often admitted. The diagonally-opposite panel (row 1, column 2) shows the same relation, splitting first by gender.<sup>11</sup>

The panels in the third column (and third row) provide the explanation for the paradoxical result (see Figure 4.4) that, within all but department A, the likelihood of admission is equal for men and women, yet, overall, there appears to be a bias in favor of admitting men (see Figure ??). The (1,3) and (3, 1) panels show the marginal relation between Admission and Department, that is, how admission rate varies across departments. Departments A and B have the greatest overall admission rate, departments E and F the least. The (2, 3) and (3,2) panels show how men and women apply differentially to the various departments. It can be seen that men apply in much greater numbers to departments A and B, with higher admission rates, while women apply in greater numbers to the departments C–F, with the lowest overall rate of admission.

△

### 5.5.1 Generalized mosaic matrices and pairs plots

{sec:condmat}

We need not show only the marginal relation between each pair of variables in a mosaic matrix. (Friendly, 1999b) describes the extension of this idea to conditional, partial, and other views of a contingency table.

In `pairs.table()`, different *panel functions* can be used to specify what is displayed in the upper, lower and diagonal panels. For the off-diagonal panels, a `type` argument can be used to plot mosaics showing various kinds of independence relations:

`type="pairwise"` Shows bivariate marginal relations, collapsed over all other variables.

`type="total"` Shows mosaic plots for mutual independence.

`type="conditional"` Shows mosaic plots for conditional independence given all other variables.

`type="joint"` Shows mosaic plots for joint independence of all pairs of variables from the others.

<sup>11</sup>Note that this is different than just the transpose or interchange of horizontal and vertical dimensions as in a scatterplot matrix, because the mosaic display splits the total frequency first by the horizontal variable and then (conditionally) by the vertical variable. The areas of all corresponding tiles are the same in each diagonally opposite pair, however, as are the residuals shown by color and shading.

**EXAMPLE 5.14: Berkeley admissions**

Figure 5.24 shows the generalized mosaic matrix for the UCBA admissions data, using 3-way mosaics for all the off-diagonal cells. The observed frequencies, of course, are the same in all these cells. However, in the lower panels, the tiles are shaded according to models of joint independence, while in the upper panels, they are shaded according to models of mutual independence.

```

pairs(UCBAAdmissions,
      lower_panel = pairs_mosaic(type = "joint", shade=TRUE),
      upper_panel = pairs_mosaic(type = "total", shade=TRUE),
      space=0.2)

## Error in pairs.table(UCBAAdmissions, lower_panel = pairs_mosaic(type =
## "joint", : could not find function "grid.newpage"

```

**TODO:** Replace this with a figure using `type = "conditional"`, once we can get this to work.

In this example, it is more useful to fit and display the models of conditional independence for each pair of row, column variables given the remaining one, as shown in Figure 5.25.

```

pairs(UCBAAdmissions,
      lower_panel = pairs_mosaic(type = "conditional", shade=TRUE),
      upper_panel = pairs_mosaic(type = "conditional", shade=TRUE),
      space=0.2)

## Error in pairs.table(UCBAAdmissions, lower_panel = pairs_mosaic(type =
## "conditional", : could not find function "grid.newpage"

```

Thus, the shading in the (1,2) and (2,1) panels show the fit of the model [Admit, Dept] [Gender, Dept], which asserts that Admission and Gender are independent, given (controlling for) Department. Except for Department A, this model fits quite well, again indicating lack of gender bias. The (1,3) and (3,1) panels show the relation between admission and department controlling for gender, highlighting the differential admission rates across departments. **TODO: This isn't quite right!**

△

Beyond this, the framework of pairs plots can be further generalized to *mixtures* of quantitative and categorical variables, as first described in Friendly (2003) and then in a wider context by Emerson *et al.* (2013), Friendly (2013). The essential idea is to consider the combination of two variables, each of which can be either categorical (**C**) or quantitative (**Q**), and various ways to *render* that combination in a graphical display:

- CC:** mosaic display, sieve diagram, doubledecker plot, faceted or divided bar chart;
- CQ:** side-by-side boxplots, stripplots, faceted histograms, aligned density plots;
- QQ:** scatterplot, corrgram, data ellipses, etc.

In R some of these possibilities are provided in the `gpairs` package (using `grid` graphics and the `vcd` strucplot framework), and the `GGally` package (an extension to `ggplot2`).



**EXAMPLE 5.15: Arthritis treatment**

We illustrate these ideas with the *Arthritis* data using the `gpairs` package in Figure 5.26. In this data, the variables *Treatment*, *Sex* and *Improved* are categorical, and *Age* is quantitative. The call to `gpairs()` below reorders the variables to put the response variable *Improved* in row 1, column 1. Various options can be passed to `mosaic()` using the `mosaic.pars` argument.

```
library(gpairs)
data("Arthritis", package="vcd")
gpairs(Arthritis[, c(5, 2, 3, 4)],
       diag.pars=list(fontsize = 20),
       mosaic.pars=list(gp=shading_Friendly,
                       gp_args=list(interpolate=1:4)))

## Error in gpairs(Arthritis[, c(5, 2, 3, 4)], diag.pars = list(fontsize
## = 20), : object 'shading_Friendly' not found
```

`gpairs()` provides a variety of options for the **CQ** and **QQ** combinations, as well as the diagonal cells, but only the defaults are used here. The bottom row, corresponding to *Age* uses boxplots to show the distributions of age for each of the categorical variables. The last column shows these same variables as stripplots (or “barcodes”), which show all the individual observations. In the (1,4) and (4,1) panels, it can be seen that younger patients are more likely to report no 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. △

## 5.6 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` 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 5.16: Bartlett data on plum root cuttings**

In Example 5.11 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 5.4 shows the 3D version, produced using `mosaic3d()`:

```
mosaic3d(Bartlett)
```

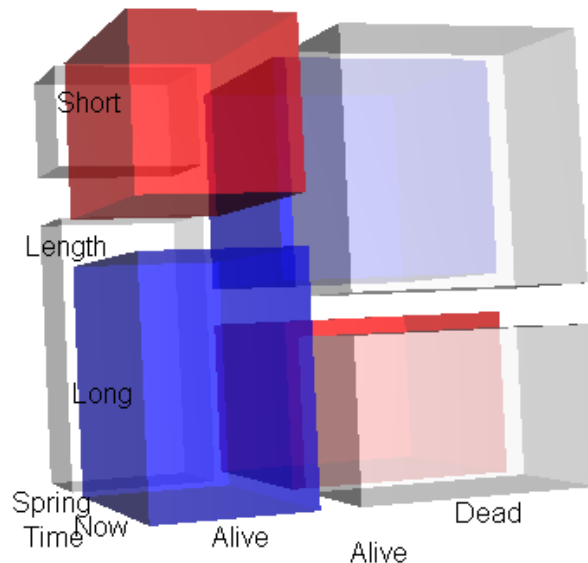


Figure 5.4: 3D mosaic plot of the Bartlett data, according to the model of mutual independence

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. △

## 5.7 Parallel coordinate plots for categorical data

Beyond pairwise plots, parallel coordinate plots (Inselberg, 1985, 1989, Wegman, 1990) provide another means to extend visualization methods beyond 2/3D. With the cartesian coordinate system, we run out of axes in 3D. Parallel coordinates overcome this limitation by plotting multiple axes in parallel. The geometry of parallel coordinates is the dual of cartesian geometry in standard plots: points in cartesian space appear as lines in parallel coordinates and vice versa. We first illustrate this visual framework for quantitative data, and then describe extensions to categorical data.

### EXAMPLE 5.17: 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*.

Standard parallel coordinate plots are easily constructed in R using `parallelplot()` from the `lattice` package.

```

library(lattice)
data("iris", package="datasets")
vnames <- gsub("\\.", "\\n", names(iris))
key = list(
  columns = 3,
  lines = list(col=c("red", "blue", "green3"), lwd=4),
  col=c("red", "blue", "green3"),
  text = list(c("Setosa", "Versicolor", "Virginica")))

parallelplot(~iris[1:4], iris, groups = Species,
  varnames = vnames[1:4], key=key,
  horizontal.axis = FALSE, lwd=2,
  col=c("red", "blue", "green3"))

```

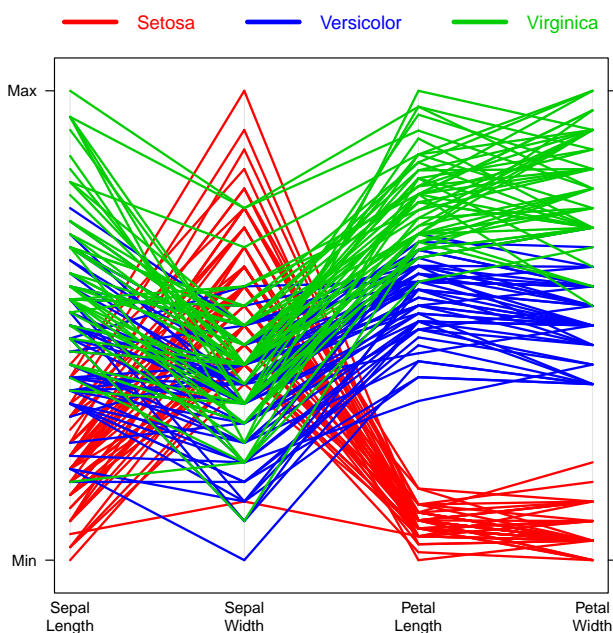


Figure 5.5: Parallel coordinates plot of the Iris data<sup>fig:iris1</sup>

Figure 5.5 shows the typical parallel coordinate plot for quantitative variables, where each variable axis represents the range of the corresponding variable, and observation values for the iris flowers are connected by lines. It can readily be seen that the flowers within each species vary systematically on the four variables, with the *setosa* flowers smaller on all except sepal width. Moreover, the patterns of the cases are positively correlated on all except sepal width, which is negatively related to the other three variables.

The transition to such plots for categorical data can be illustrated as shown in Figure 5.6. In the left panel, we have tried to show the *density* of a discrete variable *visually*, by (a) making the connecting lines thicker, but coloring them using transparent colors, so that more data “ink” corresponds to increasing frequency. (b) showing the categorical variable `Species` itself.

```

# alpha-blending, and showing species
parallelplot(~iris[1:5], data=iris, groups = Species,
  varnames = vnames, key = key,
  horizontal.axis = FALSE, lwd=8,
  col=c(rgb(1,0,0,.2), rgb(0,0,1,.2), rgb(0,1,0,.2) )

```

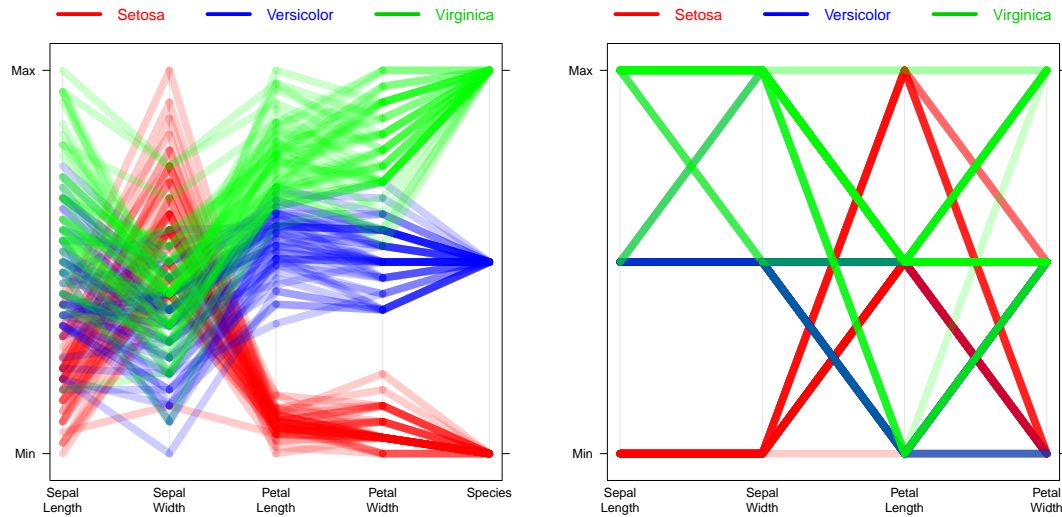


Figure 5.6: Discretized versions of parallel coordinate plots. Left: frequency shown by intensity of shading, along with a categorical variable; right: A less useful parallel coordinate plot for all categorical variables.

{fig:iris2}

)

Alternatively, we can `cut()` the quantitative variables into (ordered) categorical variables as shown in Figure 5.6 (right panel). However, although `parallelplot()` does handle discrete variables, the resulting plot is relatively uninformative.

```
# make ordered factors
iris2 <- within(iris, {
  sepalL <- cut(Sepal.Length, 3)
  sepalW <- cut(Sepal.Width, 3)
  petalL <- cut(Petal.Length, 3)
  petalW <- cut(Petal.Width, 3)
})
parallelplot(~iris2[6:9], data=iris2, groups = Species,
  varnames = vnames[1:4],
  horizontal.axis = FALSE, lwd=8, key=key,
  col=c(rgb(1,0,0,.2), rgb(0,0,1,.2), rgb(0,1,0,.2))
)
```

Note that, just as in the mosaic display, parallel coordinate plots are influenced by the order of the variable axes, because only adjacent pairs of variables are connected by lines. You can see this yourself by reordering the `iris` variables to place sepal width last (or first), as in this call. The result (another example of effect ordering for data displays) was shown in Figure 1.11, and is a more coherent display than Figure 5.5.

```
# effect of order of variables
parallelplot(~iris[c(1,3,4,2)], data=iris, groups = Species,
  varnames = vnames[c(1,3,4,2)], key=key,
  horizontal.axis = FALSE, lwd=8,
  col=c(rgb(1,0,0,.2), rgb(0,0,1,.2), rgb(0,205/255,0,.2))
)
```

### 5.7.1 Parallel set plots: Hammock plots and common angle plots

The right panel of Figure 5.6 is unsuccessful in showing the relations among the four categorical iris measures because the individual observations are shown; their discrete nature results on much overplotting, obscuring the visual interpretation of frequency, and making unusual points (outliers) more dominant. A simple way to circumvent this is to use *parallel sets* (Kosara *et al.*, 2006) representation, that shows data *frequencies* instead of the individual data points. The method is based on the same axis layout of parallel coordinates, but with boxes representing the categories and parallelograms between the axes showing the relations between categories.

In implementations, the sizes of the boxes typically represent the frequencies of the categories, and the total length of each axis is subdivided according to their relative frequencies. We illustrate these methods below using the `ggparallel` package, which provides two other varieties: *hammock plots* (Schonlau, 2003) and *common angle plots* (Hofmann and Vendettuoli, 2013). These have better perceptual properties, as we describe below.

{ex:titanic-par1}

#### EXAMPLE 5.18: Titanic data

In this example, we use the `Titanic`, converted to a frequency data frame. In the call to `ggparallel()`, we use the `Freq` variable to weight the categories. `order=0` says to keep the order of the factor levels unchanged (rather than sorting them by frequency); `method="parset"` gives the basic parallel sets version.

```
library(RColorBrewer)
library(ggparallel)
titanic <- as.data.frame(Titanic)
vars <- names(titanic)[c(1, 4, 2)]
ggparallel(vars, titanic, order=0, weight="Freq", method="parset") +
  scale_fill_brewer(palette="Paired", guide="none") +
  scale_colour_brewer(palette="Paired", guide="none")
```

In this example, we positioned the variable `Survived` between `Class` and `Sex` to focus attention on the relation of each of these with survival. The bands between the first two axes show, for each class, the number who lived and the number who died. Yet, it is hard to accurately compare the relative frequencies of the various bands, because a perceptual illusion called the *line width illusion* makes the less slanted bands appear wider than more slanted bands that represent equal frequencies.<sup>12</sup> The main reason for this is that there is a strong visual bias toward evaluating the width of lines *orthogonal* to their slopes as opposed to vertically, which is the representation of frequency in this plot.

△

One solution to this problem is the *hammock plot* (Schonlau, 2003), which adjusts the width of the line by a factor ( $\sin \theta$ ) to make the *perceived* orthogonal line proportional in width to the number of observations it represents. This works, but may overcorrect, because it assumes that everyone is governed 100% by the line width illusion.

In contrast, the *common angle plot* (Hofmann and Vendettuoli, 2013) tries to draw all lines in a plot with the same angle. To achieve this, instead of drawing straight lines between variables, it

<sup>12</sup>You can see this for yourself with the following task: From Figure 5.7, write down the *order* of the classes according to the number who survived. Don't be too cocky: in one experimental study (Hofmann and Vendettuoli, 2013), only 6% of respondents could do so correctly. You can find the correct answer by running `sort(margin.table(Titanic[,, "Yes"], 1), dec=TRUE)`.

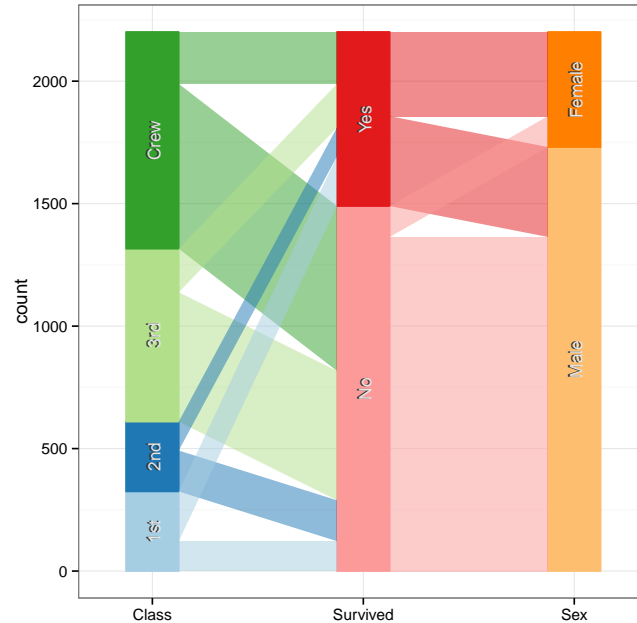


Figure 5.7: Parallel sets plot of the Titanic data. This graphic aims to show the frequencies of categorical variables in a parallel coordinates format.

{fig:titanic}

uses ribbons composed of connected line segments where at least one line segment is drawn with the same angle.

{ex:titanic}

### EXAMPLE 5.19: Titanic data

With `ggparallel()`, hammock plots are obtained using `method="hammock"` and common angle plots using `method="angle"` (the default). Figure 5.8 (left) shows a hammock plot designed to explore relations of Sex and Survived with Class, by plotting Class twice, on outside axes. The analogous common angle plot is shown in the right panel.

The labels for variables and factor levels are easier to read when the parallel axes are horizontal rather than vertical; in the `ggplot2` framework, this is done with `coord_flip()`.

```
# define colors for factor levels
cols <- c(brewer.pal(name="Blues", 6)[-c(1,2)],
         rev(brewer.pal(name="Reds", 3)[-1]),
         rev(brewer.pal(name="Greens", 3)[-1]))
# hammock plot
vars <- names(titanic)[c(1, 4, 2, 1)]
ggparallel(vars, data=titanic, weight="Freq", method="hammock",
          order=c(0,1,1,0), ratio=.25, text.angle=0) +
  scale_fill_manual(values=cols, guide="none") +
  scale_colour_manual(values=cols, guide="none") + coord_flip()
```

```
# angle plot
ggparallel(vars, data=titanic, weight="Freq", method="angle",
          order=c(0,1,1,0), text.angle=0) +
  scale_fill_manual(values=cols, guide="none") +
  scale_colour_manual(values=cols, guide="none") + coord_flip()
```

△

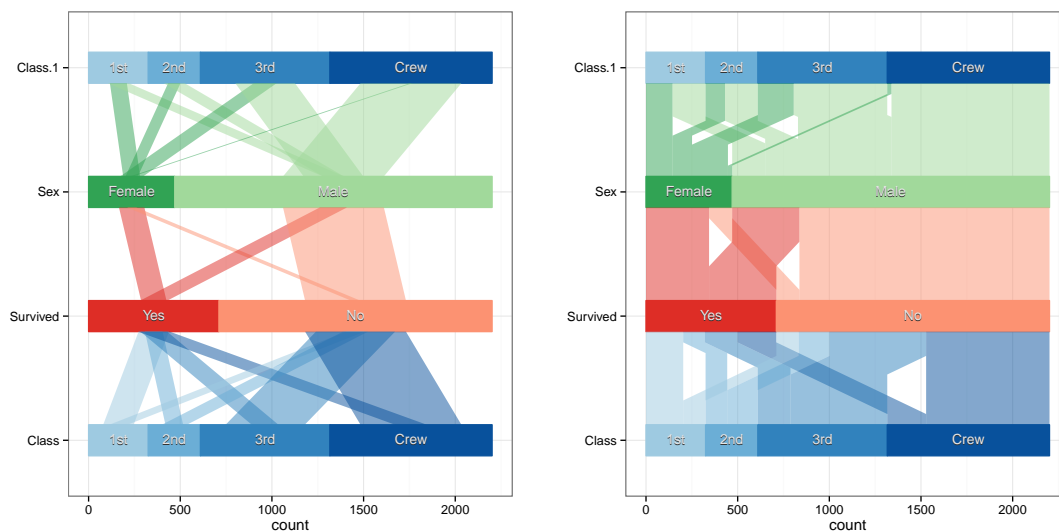


Figure 5.8: Hammock plot and common angle plot of the Titanic data, with coordinate axes flipped. The hammock plot (left) tries to adjust the width of lines by a factor designed counter the line width illusion. The common angle plot (right) uses ribbons, where at least one segment has the same angle.

## 5.8 Visualizing the structure of loglinear models

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 which 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(Sex+Age ~ Disease, struc)
```

```
## Error in eval(expr, envir, enclos): could not find function "structable"
```

First, note that there are substantial associations in this table, as shown in Figure 5.32, fitting the (default) mutual independence model.

```
mosaic(struc, shade=TRUE)

## Error in eval(expr, envir, enclos): could not find function "mosaic"
```

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 5.33, 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")

## Error in eval(expr, envir, enclos): could not find function "mosaic"
```

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 which 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(Sex+Age ~ Disease, fit)

## Error in eval(expr, envir, enclos): could not find function "structable"
```

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 5.33. We use `gp=shading_Friendly` to explicitly indicate the zero residuals in the display.

```
pairs(fit, gp=shading_Friendly, type="total")

## Error in pairs.table(fit, gp = shading_Friendly, type = "total"): could
not find function "grid.newpage"
```

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 ( $\text{age} \perp \text{sex} \mid \text{disease}$ ).

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,  $(X \perp Y \mid Z)$  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 5.9, 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)
```

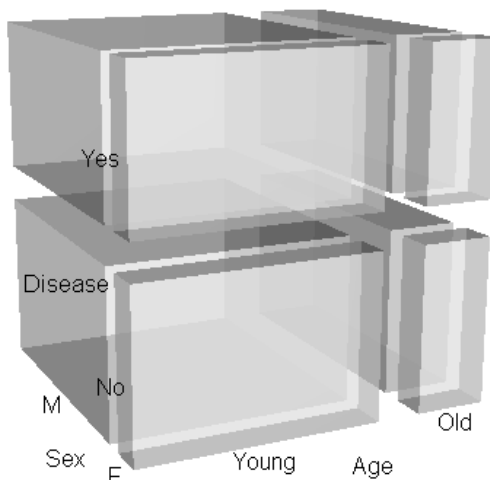


Figure 5.9: 3D mosaic plot of frequencies according to the model of mutual independence

```
{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(Sex+Age ~ Disease, fit)
```

```
## Error in eval(expr, envir, enclos): could not find function "structable"
```

The `pairs.table()` plot, now using simpler pairwise plots (`type="pairwise"`), is shown in Figure 5.36.

```

pairs (fit, gp=shading_Friendly)

## Error in pairs.table(fit, gp = shading_Friendly): could not find function
"grid.newpage"

```

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.

## 5.9 Chapter summary

{sec:mosaic-summary}

- The mosaic display depicts the frequencies in a contingency table by a collection of rectangular “tiles” whose area is proportional to the cell frequency. The residual from a specified model is portrayed by shading the tile to show the sign and magnitude of the deviation from the model.
- For two-way tables, the tiles for the second variable align at each level of the first variable when the two variables are independent (see Figure 5.10).
- The perception and understanding of *patterns of association* (deviations from independence) are enhanced by reordering the rows or columns to give the shading of the residuals a more coherent pattern. An opposite-corner pattern “explains” the association in terms of the ordering of the factor levels.
- For three-way and larger tables, a variety of models can be fit and visualized. Starting with a minimal baseline model, the pattern of residuals will often suggest additional terms which must be added to “clean the mosaic.”
- It is often useful to examine the *sequential* mosaic displays for the marginal subtables with the variables in a given order. Sequential models of joint independence provide a breakdown of the total association in the full table, and are particularly appropriate when the last variable is a response.
- Partial association, which refers to the associations among a subset of variables, within the levels of other variables, may be easily studied by constructing separate mosaics for the subset variables for the levels of the other, “given” variables. These displays provide a breakdown of a model of conditional association for the whole table, and serve as an analog of coplots for quantitative data.
- Mosaic matrices, consisting of all pairwise plots of an  $n$ -way table, provide a way to visualize all marginal, joint, or conditional relations simultaneously. Parallel set plots provide another method to visualize  $n$ -way tables.
- The structural relations among model terms in various loglinear models themselves can also be visualized by mosaic matrices showing the expected, rather than observed, frequencies under different models.

## 5.10 Further reading

### 5.11 Lab exercises

**Exercise 5.1** The data set `criminal` in the package `logmult` gives the  $4 \times 5$  table below of the number of men aged 15-19 charged with a criminal case for whom charges were dropped in Denmark from 1955–1958.

```
data("criminal", package="logmult")
criminal

##      Age
## Year  15  16  17  18  19
## 1955 141 285 320 441 427
## 1956 144 292 342 441 396
## 1957 196 380 424 462 427
## 1958 212 424 399 442 430
```

- Use `loglm()` to test whether there is an association between *Year* and *Age*. Is there evidence that dropping of charges in relation to age changed over the years recorded here?
- Use `mosaic()` with the option `shade=TRUE` to display the pattern of signs and magnitudes of the residuals. Compare this with the result of `mosaic()` using “Friendly shading,” from the option `gp=shading_Friendly`. Describe verbally what you see in each regarding the pattern of association in this table.

{lab:5.2}

**Exercise 5.2** The `Lahman` package contains comprehensive data on baseball statistics for Major League Baseball from 1871 through 2012. For all players, the `Master` table records the handedness of players, in terms of throwing (L, R) and batting (B, L, R), where B indicates “both.” The table below was generated using the following code:

```
library(Lahman)
data("Master", package="Lahman")
basehands <- with(Master, table(throws, bats))
```

	Bats		
Throws	B	L	R
L	177	2640	527
R	924	1962	10442

- Use the code above, or else enter these data into a frequency table in R.
- Construct mosaic displays showing the relation of batting and throwing handedness, split first by batting and by throwing.
- From these displays, what can be said about players who throw with their left or right hands in terms of their batting handedness?

{lab:5.3}

**Exercise 5.3** \* A related analysis concerns differences in throwing handedness among baseball players according to the fielding position they play. The following code calculates a such a frequency table.

```
library(Lahman)
MasterFielding <- data.frame(merge(Master, Fielding, by="playerID"))
throwPOS <- with(MasterFielding, table(POS, throws))
```

- Make a mosaic display of throwing hand vs. fielding position.
- Calculate the percentage of players throwing left-handed by position. Make a sensible graph of this data.
- Re-do the mosaic display with the positions sorted by percentage of left-handers.
- Is there anything you can say about positions that have very few left-handed players?

{lab:5.4}

**Exercise 5.4** For the `Bartlett` data described in Example 5.11, fit the model of no three-way association,  $H_4$  in Table 5.2.

- Summarize the goodness of fit for this model, and compare to simpler models that omit one or more of the two-way terms.
- Use a mosaic-like display to show the lack of fit for this model.

{lab:5.5}

**Exercise 5.5** Red core disease, caused by a fungus, is not something you want if you are a strawberry. The data set `jansen.strawberry` from the `agridat` package gives a frequency data frame of counts of damage from this fungus from a field experiment reported by Jansen (1990). See the help file for details. The following lines create a  $3 \times 4 \times 3$  table of crossings of 3 male parents with 4 (different) female parents, recording the number of plants in four blocks of 9 or 10 plants each showing red core disease in three ordered categories, C1, C2 or C3.

```
data("jansen.strawberry", package="agridat")

dat <- jansen.strawberry
dat <- transform(dat, category=ordered(category, levels=c('C1','C2','C3')))
levels(dat$male) <- paste0("M", 1:3)
levels(dat$female) <- paste0("F", 1:4)

jansen.tab <- xtabs(count~male + female + category, data=dat)
names(dimnames(jansen.tab)) <- c("Male parent", "Female parent",
                                "Disease category")

ftable(jansen.tab)
```

- Use `pairs(jansen.tab, shade=TRUE)` to display the pairwise associations among the three variables. Describe how disease category appears to vary with male and female parent? Why is there no apparent association between male and female parent?
- As illustrated in Figure 5.2, use `mosaic` to prepare a 3-way mosaic plot with the tiles colored in increasing shades of some color according to disease category. Describe the pattern of category C3 in relation to male and female parent. (Hint: the `highlighting` arguments are useful here.)
- With `category` as the response variable, the minimal model for association is  $[MF][C]$ , or  $\sim 1*2 + 3$ . Fit this model using `loglm()` and display the residuals from this model with `mosaic()`. Describe the pattern of lack of fit of this model.

{lab:5.6}

**Exercise 5.6** The data set `caith` in `MASS` gives another classic  $4 \times 5$  table tabulating hair color and eye color, this for people in Caithness, Scotland, originally from Fisher (1940). The data is stored as a data frame of cell frequencies, whose rows are eye colors and whose columns are hair colors.

```
data("caith", package="MASS")
caith

##          fair red medium dark black
## blue      326  38   241  110     3
## light     688 116   584  188     4
## medium    343  84   909  412    26
## dark       98  48   403  681    85
```

- The `loglm()` and `mosaic()` functions don't understand data in this format, so use `Caith <- as.matrix(caith)` to convert to array form. Examine the result, and use `names(dimnames(Caith)) <- c()` to assign appropriate names to the row and column dimensions.
- Fit the model of independence to the resulting matrix using `loglm()`.
- Calculate and display the residuals for this model.
- Create a mosaic display for this data.

```
{lab:5.7}
```

**Exercise 5.7** The `HairEyePlace` data in `vcdExtra`, gives similar data on hair color and eye color, for both Caithness and Aberdeen as a  $4 \times 5 \times 2$  table.

- Prepare separate mosaic displays, one for each of Caithness and Aberdeen. Comment on any difference in the pattern of residuals.
- Construct condition mosaic plots, using the formula `~Hair+Eye | Place` and both `mosaic()` and `cotabplot()`. It is probably more useful here to suppress the legend in these plots. Comment on the difference in what is shown in the two displays.

```
{lab:mosaic-accident}
```

**Exercise 5.8** Bertin (1983, p. 30–31) used a 4-way table of frequencies of traffic accident victims in France in 1958 to illustrate his scheme for classifying data sets by numerous variables, each of which could have various types and could be assigned to various visual attributes. His data are contained in `Accident` in `vcdExtra`, a frequency data frame representing his  $5 \times 2 \times 4 \times 2$  table of the variables `age`, `result` (died or injured), `mode` of transportation and `gender`.

```
data("Accident", package="vcdExtra")
str(Accident)

## 'data.frame': 80 obs. of 5 variables:
## $ age : Ord.factor w/ 5 levels "0-9"<"10-19"<...: 5 5 5 5 5 5 5 5 5 5 ...
## $ result: Factor w/ 2 levels "Died","Injured": 1 1 1 1 1 1 1 1 2 2 ...
## $ mode : Factor w/ 4 levels "4-Wheeled","Bicycle",...: 4 4 2 2 3 3 1 1 4 4 ...
## $ gender: Factor w/ 2 levels "Female","Male": 2 1 2 1 2 1 2 1 2 1 ...
## $ Freq : int 704 378 396 56 742 78 513 253 5206 5449 ...
```

- Use `loglm()` to fit the model of mutual independence, `Freq ~ age+mode+gender+result` to this data.
- Use `mosaic()` to produce an interpretable mosaic plot of the associations among all variables under the model of mutual independence. Try different orders of the variables in the mosaic. (*Hint*: the abbreviate component of the `labeling_args` argument to `mosaic()` will be useful to avoid some overlap of the category labels.)
- Treat `result` ("Died" vs. "Injured") as the response variable, and fit the model `Freq ~ age*mode*gender + result` that asserts independence of `result` from all others jointly.

- (d) Construct a mosaic display for the residual associations in this model. Which combinations of the predictor factors are more likely to result in death?

```
#detach(package:ggtern)
.locals$ch05 <- setdiff(ls(), .globals)
#locals$ch05
remove(list=locals$ch05[sapply(locals$ch05, function(n) {!is.function(get(n))})])
```

# References

- Agresti, A. (2013). *Categorical Data Analysis*. Wiley Series in Probability and Statistics. New York: Wiley-Interscience [John Wiley & Sons], 3rd edn.
- Andersen, E. B. (1991). *Statistical Analysis of Categorical Data*. Berlin: Springer-Verlag, 2nd edn.
- Anderson, E. (1935). The irises of the Gaspé peninsula. *Bulletin of the American Iris Society*, 35, 2–5.
- Bartlett, M. S. (1935). Contingency table interactions. *Journal of the Royal Statistical Society, Supplement*, 2, 248–252.
- Bertin, J. (1983). *Semiology of Graphics*. Madison, WI: University of Wisconsin Press. (trans. W. Berg).
- Cleveland, W. S. (1993). *Visualizing Data*. Summit, NJ: Hobart Press.
- Emerson, J. W., Green, W. A., Schloerke, B., Crowley, J., Cook, D., Hofmann, H., and Wickham, H. (2013). The generalized pairs plot. *Journal of Computational and Graphical Statistics*, 22(1), 79–91.
- Fienberg, S. E. (1980). *The Analysis of Cross-Classified Categorical Data*. Cambridge, MA: MIT Press, 2nd edn.
- Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 8, 379–388.
- Fisher, R. A. (1940). The precision of discriminant functions. *Annals of Eugenics*, 10, 422–429.
- Friendly, M. (1992). Mosaic displays for loglinear models. In *ASA, Proceedings of the Statistical Graphics Section*, (pp. 61–68). Alexandria, VA.
- Friendly, M. (1994). Mosaic displays for multi-way contingency tables. *Journal of the American Statistical Association*, 89, 190–200.
- Friendly, M. (1997). Conceptual models for visualizing contingency table data. In M. Greenacre and J. Blasius, eds., *Visualization of Categorical Data*, chap. 2, (pp. 17–35). San Diego, CA: Academic Press.
- Friendly, M. (1999a). Extending mosaic displays: Marginal, conditional, and partial views of categorical data. *Journal of Computational and Graphical Statistics*, 8(3), 373–395.
- Friendly, M. (1999b). Extending mosaic displays: Marginal, conditional, and partial views of categorical data. *Journal of Computational and Graphical Statistics*, 8(3), 373–395.

- Friendly, M. (2000). *Visualizing Categorical Data*. Cary, NC: SAS Institute.
- Friendly, M. (2003). Visions of the past, present and future of statistical graphics: An ideo-graphic view. American Psychological Association. Toronto, ON, URL: <http://datavis.ca/papers/apa-2x2.pdf>.
- Friendly, M. (2013). Comment on the generalized pairs plot. *Journal of Computational and Graphical Statistics*, 22(1), 290–291.
- Friendly, M. and Kwan, E. (2003). Effect ordering for data displays. *Computational Statistics and Data Analysis*, 43(4), 509–539.
- Goodman, L. A. (1973). The analysis of multidimensional contingency tables when some variables are posterior to others: A modified path analysis approach. *Biometrika*, 60, 179–192.
- Hartigan, J. A. and Kleiner, B. (1981). Mosaics for contingency tables. In W. F. Eddy, ed., *Computer Science and Statistics: Proceedings of the 13th Symposium on the Interface*, (pp. 268–273). New York, NY: Springer-Verlag.
- Hartigan, J. A. and Kleiner, B. (1984). A mosaic of television ratings. *The American Statistician*, 38, 32–35.
- Hofmann, H. and Vendettuoli, M. (2013). Common angle plots as perception-true visualizations of categorical associations. *IEEE Transactions on Visualization and Computer Graphics*, 19(12), 2297–2305.
- Inselberg, A. (1985). The plane with parallel coordinates. *The Visual Computer*, 1, 69–91.
- Inselberg, A. (1989). Discovering multi-dimensional structure using parallel coordinates. In *Proc. Amer. Stat. Assoc, Sec. on Stat. Graphics*, (pp. 1–16). American Statistical Association.
- Jansen, J. (1990). On the statistical analysis of ordinal data when extravariation is present. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 39(1), 75–84.
- Kosara, R., Bendix, F., and Hauser, H. (2006). Parallel sets: Interactive exploration and visual analysis of categorical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(4), 558–568.
- Meyer, D., Zeileis, A., and Hornik, K. (2006). The strucplot framework: Visualizing multi-way contingency tables with `vcd`. *Journal of Statistical Software*, 17(3), 1–48.
- Schonlau, M. (2003). Visualizing categorical data arising in the health sciences using hammock plots. In *Proceedings of the Section on Statistical Graphics*. RAND Corporation, American Statistical Association.
- Theus, M. and Lauer, S. R. W. (1999). Visualizing loglinear models. *Journal of Computational and Graphical Statistics*, 8(3), 396–412.
- Thornes, B. and Collard, J. (1979). *Who Divorces?* London: Routledge & Kegan.
- Wegman, E. J. (1990). Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, 85(411), 664–675.
- Zeileis, A., Meyer, D., and Hornik, K. (2007). Residual-based shadings for visualizing (conditional) independence. *Journal of Computational and Graphical Statistics*, 16(3), 507–525.