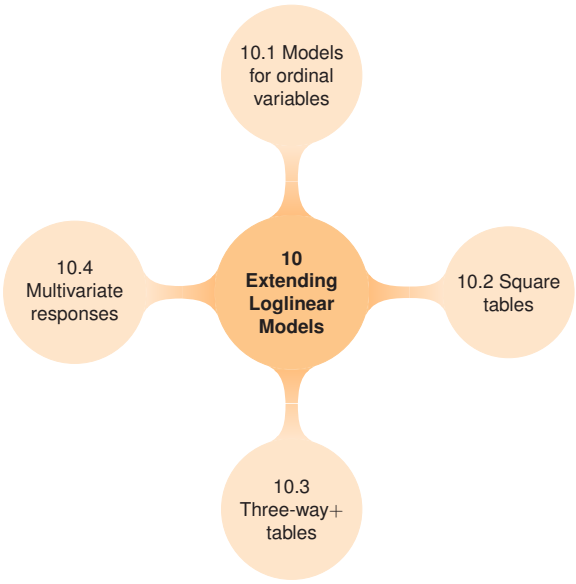


10

Extending Loglinear Models

{ch:loglin2}

Loglinear models have special forms to represent additional structure in the variables in contingency tables. Models for ordinal factors allow a more parsimonious description of associations. Models for square tables allow a wide range of specific models for the relationship between variables with the same categories. Another extended class of models arise when there are two or more response variables.

> The universe is built on a plan the profound symmetry of which is somehow present in the inner structure of our intellect.
>
> Paul Valery, 1871–1945

This chapter extends the analysis of loglinear models to some important special cases allowing us to represent additional structure in the variables in contingency tables in a way that provides a more parsimonious description of associations than available from models for general association. One class of such simplified models (Section 10.1) occurs when one or more of the explanatory variables are ordinal, and discrete levels might be replaced by numerical values.

Models for square tables (Section 10.2), with the same row and column categories comprise another special case giving simpler descriptions than the saturated model of general association. These important special cases are extended to three-way and higher-dimensional tables in Section 10.3.

Finally, Section 10.4 describes some methods for dealing with situations where there are several response variables, and it is useful to understand both the marginal relations of the responses with the predictors as well as how their association varies with the predictors.

10.1 Models for ordinal variables

{sec:loglin-ordinal}

Standard loglinear models treat all classification variables as nominal, unordered factors. In these models, all statistical tests are identical and parameter estimates are equivalent if the categories of any of the table variable are reordered. Yet we have seen that the ordering of categories often provides important information about the nature of associations and we showed (Section 4.2.4) that non-parametric tests which take into account the ordered nature of a factor are more powerful.

Correspondence analysis plots (Chapter 6) make it easy to see the relationships between ordinal variables, because the method assigns quantitative scores to the table variables which maximally account for their association. As we saw for the hair-eye color data (Figure 6.1) and the mental impairment data (Figure 6.2), an association can be interpreted in terms of ordered categories when the points for two factors are ordered similarly, usually along the first CA dimension.

Similarly, in a mosaic display, an ordered associative effect is seen when the residuals have an opposite-corner pattern of positive and negative signs and magnitudes (e.g., for the hair-eye color data, Figure 5.4). In these cases loglinear and logit models which use the ordered nature of the factors offer several advantages:

- Because they are more focused, tests which use the ordinal structure of the table variables are more powerful when the association varies systematically with the ordered values of a factor.
- Because they consume fewer degrees of freedom, we can fit unsaturated models where the corresponding model for nominal factors would be saturated. In a two-way table, for example, a variety of models for ordinal factors may be proposed which are intermediate between the independence model and the saturated model.
- Parameter estimates from these models are fewer in number, are easier to interpret, and quantify the nature of effects better than corresponding quantities in models for nominal factors. Estimating fewer parameters typically gives smaller standard errors.

These advantages are analogous to the use of tests for trends or polynomial contrasts in ANOVA models. More importantly, in some research areas in the social sciences (where categorical data is commonplace), models for ordinal variables have proved crucial in theory construction and debates, giving more precise tests of hypotheses than available from less focused or descriptive methods (Agresti, 1984).

10.1.1 Loglinear models for ordinal variables

{sec:loglin-ordlog}

For a two-way table, when either the row variable or the column variable, or both, are ordinal, one simplification comes from assigning ordered scores, $\boldsymbol{a} = (a_i), a_1 \leq a_2 \leq \cdots a_I$, and/or $\boldsymbol{b} = (b_j), b_1 \leq b_2 \leq \cdots b_J$ to the categories so that the ordinal relations are necessarily included in the model. Typically, equally spaced scores are used, for example, integer scores, $a_i = i$, or the zero-sum equivalent, $a_i = i - (I+1)/2$ (e.g., $(a_i) = (-1, 0, 1)$ for $I = 3$).

Using such scores gives simple interpretations of the association parameters in terms of *local odds ratios* for adjacent 2×2 subtables,

{eq:loddsratios}

$$\theta_{ij} = \frac{m_{ij}\, m_{i+1,j+1}}{m_{i,j+1}\, m_{i+1,j}} \ , \tag{10.1}$$

which is the odds ratio for pairs of adjacent rows and adjacent columns.

When both variables are assigned scores, this gives the ***linear-by-linear model*** $(L \times L)$

{eq:linlin}

$$\log(m_{ij}) = \mu + \lambda_i^A + \lambda_j^B + \gamma\, a_i b_j \ . \tag{10.2}$$

Because the scores $\boldsymbol{a}$ and $\boldsymbol{b}$ are fixed, this model has only one extra parameter, γ, compared to the

independence model, which is the special case, $\gamma = 0$. In contrast, the saturated model, allowing general association λ_{ij}^{AB} uses $(I-1)(J-1)$ additional parameters.

The terms $\gamma a_i b_j$ in Eqn. (10.2) describe a pattern of association where deviations from independence increase linearly with a_i and b_j in opposite directions towards the opposite corners of the table, as we have often observed in mosaic displays.

In the linear-by-linear association model, the local log odds ratios are

$$\log(\theta_{ij}) = \gamma(a_{i+1} - a_i)(b_{j+1} - b_j) \ ,$$

which reduces to

$$\log(\theta_{ij}) = \gamma$$

for integer-spaced scores, so γ is the common local log odds ratio. As a result, the linear-by-linear model is sometimes called the ***uniform association model*** (Goodman, 1979).

Generalizations of the linear-by-linear model result when only one variable is assigned scores. In the ***row effects model*** (R), the row variable, A, is treated as nominal, while the column variable, B, is assigned ordered scores (b_j). The loglinear model is then

$$\log(m_{ij}) = \mu + \lambda_i^A + \lambda_j^B + \alpha_i b_j \ , \tag{10.3}$$ {eq:roweff}

where the α_i parameters are the *row effects*. An additional constraint, $\sum_i \alpha_i = 0$ or $\alpha_1 = 0$ is imposed, so that model Eqn. (10.3) has only $(I-1)$ more parameters than the independence model. The linear-by-linear model is the special case where the row effects are equally spaced, and the independence model is the special case where all $\alpha_i = 0$.

The row-effects model Eqn. (10.3) also has a simple odds ratio interpretation. The local log odds ratio for adjacent pairs of rows and columns is

$$\log(\theta_{ij}) = \alpha_{i+1} - \alpha_i \ ,$$

which is constant for all pairs of adjacent columns. Plots of the local log odds ratio against i would appear as a set of coincident curves.

In the analogous ***column effects model*** (C), $(J-1)$ linearly independent column effect parameters β_j are estimated for the column variable, while fixed scores $\{a_i\}$ are assigned to the row variable. It is also possible to fit a ***row plus column effects model*** (R+C), that assigns specified scores to both the rows and column variables. Plots of the local odds ratios for the R+C model appear as parallel curves.

Nesting relationships among these models and others (RC(1) and RC(2)) described in Section 10.1.3 are shown in Figure 10.1. Any set of models connected by a path can be directly compared with likelihood-ratio tests of the form $G^2(M_2|M_1)$.

In R, the $L \times L$, row effects and column effects models can all be fit using `glm()` simply by replacing the appropriate table factor variable(s) with their `as.numeric()` equivalents. {ex:mental4}

EXAMPLE 10.1: Mental impairment and parents' SES

The `Mental` data on the mental health status of young New York residents in relation to their parents' socioeconomic status was examined in Example 4.7 using CMH tests for ordinal association and in Example 6.2 using correspondence analysis. Figure 6.2 showed that nearly all of the association in the table was accounted for by a single dimension along which both factors were ordered, consistent with the view that mental health increased in relation to parents' SES.

Because these models provide their interpretations in terms of local odds ratios, Eqn. (10.1), it is helpful to see these values for the observed data, corresponding to the saturated model. The values $\log(\theta_{ij})$ are calculated by `loddsratio()` in vcd (Meyer *et al.*, 2015), with the data in table form.

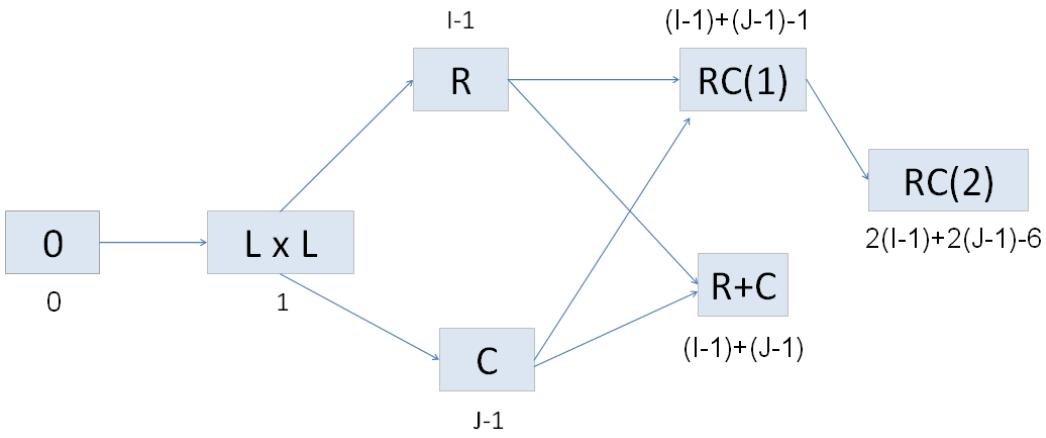


Figure 10.1: Nesting relationships among some association models for an $I \times J$ table specifying the association parameters, λij^{AB}. Model **0** is the independence model. Formulas near the boxes give the number of identifiable association parameters. Arrows point from one nested model to another that is a more general version. {fig:assoc-models}

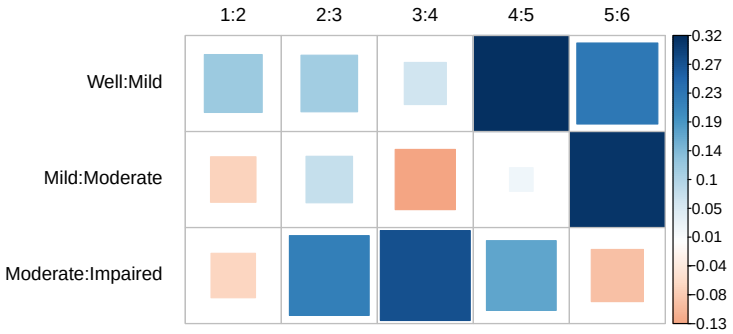
```
> library(vcd)
> data("Mental", package = "vcdExtra")
> (mental.tab <- xtabs(Freq ~ mental + ses, data=Mental))

          ses
mental      1   2   3   4   5   6
  Well     64  57  57  72  36  21
  Mild     94  94 105 141  97  71
  Moderate 58  54  65  77  54  54
  Impaired 46  40  60  94  78  71

> (LMT <- loddsratio(mental.tab))

log odds ratios for mental and ses

                   ses
mental                 1:2    2:3     3:4    4:5    5:6
  Well:Mild          0.1158 0.1107  0.0612 0.3191  0.227
  Mild:Moderate     -0.0715 0.0747 -0.1254 0.0192  0.312
  Moderate:Impaired -0.0683 0.2201  0.2795 0.1682 -0.094
```

A simple plot of these values, using area- and color-proportional shaded squares is shown in Figure 10.2. This plot is drawn using the corrplot (Wei, 2013) package. It is easy to see that most of the local odds ratios are mildly positive.[1]

```
> library(corrplot)
> corrplot(as.matrix(LMT), method = "square", is.corr = FALSE,
+          tl.col = "black", tl.srt = 0, tl.offset = 1)
```

For comparison with the $L \times L$ model fitted below, the mean local log odds ratio is 0.103.

[1] Using `plot(loddsratio(mental.tab))` would give a line plot of the odds ratios as illustrated in Section 5.9.2 (e.g., Figure 5.37).



{fig:mental-lorplot}

Figure 10.2: Shaded-square plot of the local log odds ratios in the *Mental* data.

```
> mean(LMT$coefficients)

[1] 0.10323
```

As a baseline, we first fit the independence model (testing $H_0 : \log(\theta_{ij}) = 0$) with `glm()`. As expected, this model fits quite badly, with $G^2\ (15) = 47.418$.

```
> indep <- glm(Freq ~ mental + ses, data = Mental, family = poisson)
> vcdExtra::LRstats(indep)

Likelihood summary table:
      AIC BIC LR Chisq Df Pr(>Chisq)
indep 210 220     47.4 15    3.2e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The mosaic display of standardized residuals from this model is shown in Figure 10.3. The argument `labeling=labeling_residuals` is used to show the numerical values in the cells with absolute values greater than `suppress=1`.

```
> long.labels <- list(set_varnames = c(mental="Mental Health Status",
+                                      ses="Parent SES"))
> mosaic(indep,
+        gp=shading_Friendly,
+        residuals_type="rstandard",
+        labeling_args = long.labels,
+        labeling=labeling_residuals, suppress=1,
+        main="Mental health data: Independence")
```

This figure shows the classic opposite-corner pattern of the signs and magnitudes of the residuals that would arise if the association between mental health and SES could be explained by the ordinal relation of these factors using one of the $L \times L$, R or C models.

To fit such ordinal models, you can use `as.numeric()` on a factor variable to assign integer scores, or assign other values if integer spacing is not appropriate.

```
> Cscore <- as.numeric(Mental$ses)
> Rscore <- as.numeric(Mental$mental)
```

Then, the $L \times L$, R, C and $R + C$ models can be fit as follows, using `update()`, where beyond the main effects of `mental` and `ses`, their association is represented as the interaction of the numeric score(s) or factor(s), as appropriate in each case.

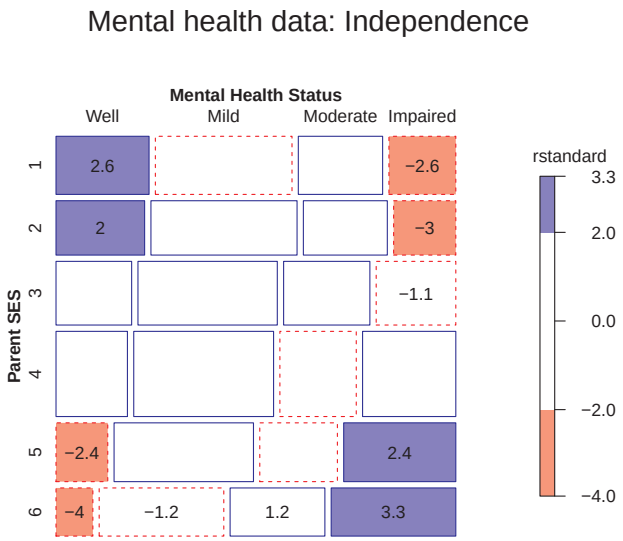


Figure 10.3: Mosaic display of the independence model for the mental health data. fig:mental-indep

```
> linlin <- update(indep, . ~ . + Rscore:Cscore)
> roweff <- update(indep, . ~ . + mental:Cscore)
> coleff <- update(indep, . ~ . + Rscore:ses)
> rowcol <- update(indep, . ~ . + Rscore:ses + mental:Cscore)
```

Goodness-of-fit tests for these models are shown below. They show that all of the $L \times L$, R and C models are acceptable in terms of the likelihood-ratio G^2. The $L \times L$ model, with only one more parameter than the independence model is judged the best by both AIC and BIC.

```
> vcdExtra::LRstats(indep, linlin, roweff, coleff, rowcol)

Likelihood summary table:
         AIC   BIC LR Chisq Df Pr(>Chisq)
indep  209.6 220.2    47.42 15   3.16e-05 ***
linlin 174.1 185.8     9.90 14      0.770
roweff 174.4 188.6     6.28 12      0.901
coleff 179.0 195.5     6.83 10      0.741
rowcol 179.2 198.1     3.05  8      0.931
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In cases where such overall tests are unclear, you can carry out tests of nested sets of models using `anova()`, giving tests of ΔG^2.

```
> anova(indep, linlin, roweff, test = "Chisq")

Analysis of Deviance Table

Model 1: Freq ~ mental + ses
Model 2: Freq ~ mental + ses + Rscore:Cscore
Model 3: Freq ~ mental + ses + mental:Cscore
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
```

```
1         15       47.4
2         14        9.9  1     37.5    9e-10 ***
3         12        6.3  2      3.6     0.16
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> anova(indep, linlin, coleff, test = "Chisq")

Analysis of Deviance Table

Model 1: Freq ~ mental + ses
Model 2: Freq ~ mental + ses + Rscore:Cscore
Model 3: Freq ~ mental + ses + ses:Rscore
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         15       47.4
2         14        9.9  1     37.5    9e-10 ***
3         10        6.8  4      3.1     0.55
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Under the $L \times L$ model, the estimate of the coefficient of `Rscore:Cscore` is $\widehat{\gamma} = 0.0907$ (s.e.=0.015) with unit-spaced scores, as shown below.

```
> # interpret linlin association parameter
> coef(linlin)[["Rscore:Cscore"]]

[1] 0.090687

> exp(coef(linlin)[["Rscore:Cscore"]])

[1] 1.0949
```

This corresponds to a local odds ratio, $\hat{\theta}_{ij} = \exp(0.0907) = 1.095$. This single number describes the association succinctly: each step down the socioeconomic scale increases the odds of being classified one step poorer in mental health by 9.5%.

△

10.1.2 Visualizing model structure

{sec:loglin-visord}

In Section 5.8 we illustrated how to use mosaic displays to visualize the *structure* of loglinear models. The basic idea was just to use mosaic plots or mosaic matrices to show the *fitted* values implied by a given model. As just described, loglinear models for ordinal variables have very simple structures in terms of log odds ratios, and you can similarly understand their structure by calculating or plotting the local odds ratios from the fitted frequencies for a given model.

{ex:mental4a}

EXAMPLE 10.2: Mental impairment and parents' SES

We illustrate this idea numerically here, for the row effects (R) model, `roweff`, fit to the `Mental` data. `fitted()` gets the fitted frequencies for this model, and using `loddsratio()` on the result show that these are constant in each row.

```
> roweff.fit <- matrix(fitted(roweff), 4, 6,
+                      dimnames=dimnames(mental.tab))
> round(as.matrix(loddsratio(roweff.fit)), 3)

                    ses
mental                 1:2   2:3   3:4   4:5   5:6
  Well:Mild          0.145 0.145 0.145 0.145 0.145
  Mild:Moderate      0.018 0.018 0.018 0.018 0.018
  Moderate:Impaired 0.143 0.143 0.143 0.143 0.143
```

Similarly, the column effects (C) model, `coleff`, shows these values to be constant in each column.

```
> coleff.fit <- matrix(fitted(coleff), 4, 6,
+                      dimnames = dimnames(mental.tab))
> round(as.matrix(loddsratio(coleff.fit)), 3)

                    ses
mental                  1:2   2:3   3:4   4:5   5:6
  Well:Mild          -0.013 0.125 0.053 0.142 0.139
  Mild:Moderate      -0.013 0.125 0.053 0.142 0.139
  Moderate:Impaired -0.013 0.125 0.053 0.142 0.139
```

Using `plot(loddsratio(model))` for these cases and the R+C model gives the plots in Figure 10.4.

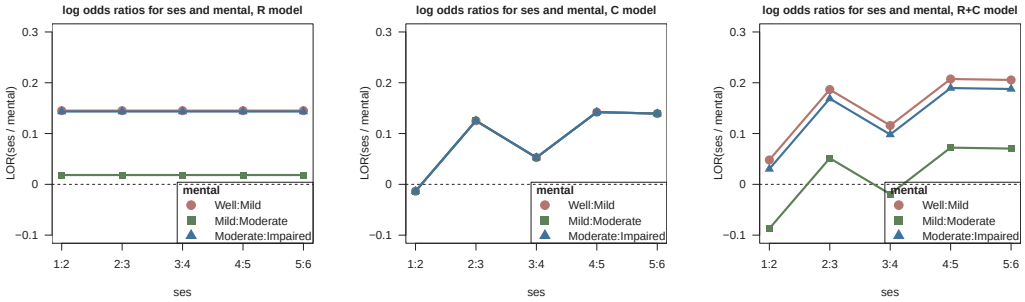


Figure 10.4: Log odds ratio plots for the R (left), C (middle) and R+C (right) models fit to the mental health data fig:mental-loddsplots

In contrast, the (more parsimonious) $L \times L$ model has a constant log odds ratio, $\widehat{\gamma} = 0.0907$. $\triangle$

10.1.3 Log-multiplicative (RC) models

{sec:RCmodels}

The association models described above are all more parsimonious and easier to interpret than the saturated model. However, they depend on assigning fixed and possibly arbitrary scores to the variable categories. A generalization of the $L \times L$ model that treats *both* row and column scores as parameters is the ***row-and-column effects model*** (RC(1)) suggested by Goodman (1979),

{eq:RC1}

$$\log(m_{ij}) = \mu + \lambda_i^A + \lambda_j^B + \gamma\, \alpha_i \beta_j \ , \tag{10.4}$$

where γ, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ comprise additional parameters to be estimated beyond the independence model.[2] This model has a close connection with correspondence analysis (Goodman, 1985), where the estimated scores $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are analogous to correspondence analysis scores on a first dimension.[3] γ, called the *intrinsic association coefficient* is analogous to the same parameter in the $L \times L$ model.

For identifiability and interpretation it is necessary to impose some normalization constraints on the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. An *unweighted, unit standardized* solution forces $\sum_i \alpha_i = \sum_j \beta_j = 0$ and $\sum_i \alpha_i^2 = \sum_j \beta_j^2 = 1$. Alternatively, and more akin to correspondence analysis solutions, the

[2] In contrast to the R, C and R+C models, RC models do not assume that the categories are appropriately ordered because the category scores are estimated from the data.

[3] However, when estimated by maximum likelihood, the RC(1) model allows likelihood-ratio tests of parameters and model fit, AIC and BIC statistics, and methods for estimating standard errors of the parameters. Such model-based methods are not available for correspondence analysis.

marginally weighted solution uses the marginal probabilities π_{i+} of the row variable and π_{+j} of the columns as weights.

{eq:RC-constraints}

$$\begin{aligned} \sum_i \alpha_i \pi_{i+} &= \sum_j \beta_j \pi_{+j} = 0 \\ \sum_i \alpha_i^2 \pi_{i+} &= \sum_j \beta_j^2 \pi_{+j} = 1 \end{aligned} \tag{10.5}$$

Goodman (1986) generalized this to multiple bilinear terms of the form $\gamma_k \; \alpha_{ik}\beta_{jk}$, with M terms (the RC(M) model) and showed that *all* associations in the saturated model could be expressed exactly as

{eq:RCm}

$$\lambda_{ij}^{AB} = \sum_{k=1}^{M} \gamma_k \; \alpha_{ik}\beta_{jk} \qquad M = \min\left(I-1,\, J-1\right) \; . \tag{10.6}$$

In practice, models with fewer terms usually suffice. For example, an RC(2) model with two multiplicative terms is analogous to a two-dimensional correspondence analysis solution. In addition to the normalization constraints for the RC(1) model, parameters in an RC(M) model must satisfy the additional constraints that the (possibly weighted) scores for distinct dimensions are orthogonal (uncorrelated), similar to correspondence analysis solutions.

The RC model is *not* a loglinear model because it contains a multiplicative term in the parameters. This model and a wide variety of other nonlinear models for categorical data can be fit using `gnm()` in the gnm (Turner and Firth, 2014) package. This provides the basic machinery for extending `glm()` models to nonlinear terms, quite generally. The function `rc()` in the logmult (Bouchet-Valat, 2015) package uses `gnm()` for fitting, and offers greater convenience in normalizing the category scores, calculating standard errors and plotting.

{ex:mental5}

EXAMPLE 10.3: Mental impairment and parents' SES

The gnm package provides a number of functions that can be used in model formulas for nonlinear association terms. Among these, `Mult()` expresses a multiplicative association in terms of two (or more) factors. The RC(1) model for factors `A, B` uses `Mult(A,B)` for the association term in Eqn. (10.4). Multiple multiplicative RC terms, as in Eqn. (10.6) can be expressed using `instances(Mult(A,B), m)`.

To illustrate, we fit the RC(1) and RC(2) models to the *Mental* data using `gnm()`. In this table, both factors are ordered, but we don't want to use the default polynomial contrasts, so we set their contrast attributes to `treatment`.

```
> library(gnm)
> contrasts(Mental$mental) <- contr.treatment
> contrasts(Mental$ses) <- contr.treatment
> indep <- gnm(Freq ~ mental + ses, data = Mental, family = poisson)
> RC1 <- update(indep, . ~ . + Mult(mental, ses), verbose = FALSE)
> RC2 <- update(indep, . ~ . + instances(Mult(mental, ses),2), verbose = FALSE)
```

For comparison with the loglinear association models fit in Example 10.1 we show the G^2 goodness of fit tests for all these models. The ordinal loglinear models and the RC models all fit well, with the $L \times L$ model preferred on the basis of parsimony by AIC and BIC.

```
> vcdExtra::LRstats(indep, linlin, roweff, coleff, RC1, RC2)

Likelihood summary table:
         AIC   BIC LR Chisq Df Pr(>Chisq)
indep  209.6 220.2    47.42 15   3.16e-05 ***
linlin 174.1 185.8     9.90 14      0.770
roweff 174.4 188.6     6.28 12      0.901
```

```
coleff 179.0 195.5     6.83 10      0.741
RC1    179.7 198.6     3.57  8      0.894
RC2    186.7 211.4     0.52  3      0.914
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The substantive difference between the $L \times L$ model and the RC(1) model is whether the categories of mental health status and SES can be interpreted as equally spaced along some latent continua, versus the alternative that category spacing is unequal. We can test this directly using the likelihood-ratio test, $G^2(L \times L \,|\, RC(1))$ Similarly, model `RC1` is nested within model `RC2`, so $G^2(RC(1) \,|\, RC(2))$ gives a direct test of the need for a second dimension.

```
> anova(linlin, RC1, RC2, test = "Chisq")

Analysis of Deviance Table

Model 1: Freq ~ mental + ses + Rscore:Cscore
Model 2: Freq ~ mental + ses + Mult(mental, ses)
Model 3: Freq ~ mental + ses + Mult(mental, ses, inst = 1) + Mult(mental,
    ses, inst = 2)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        14       9.90
2         8       3.57  6     6.32     0.39
3         3       0.52  5     3.05     0.69
```

We see that estimated scores for the categories in the model `RC1` do not provide a significantly better fit, and there is even less evidence for a second dimension of category parameters in the `RC2` model.

Nevertheless, for cases where RC models *do* provide some advantage, it is useful to know how to visualize the estimated category parameters. The key to this is the function `getContrasts()` which computes contrasts or scaled contrasts for a set of (non-eliminated) parameters from a "gnm" model, together with standard errors for the estimated contrasts following the methods of Firth (2003), Firth and Menezes (2004). The details are explained in `help(getContrasts)` and in `vignette("gnmOverview")` that comes with the gnm package.

The coefficients in the marginally-weighted solution Eqn. (10.5) can be obtained as follows.

```
> rowProbs <- with(Mental, tapply(Freq, mental, sum) / sum(Freq))
> colProbs <- with(Mental, tapply(Freq, ses, sum) / sum(Freq))
> mu <- getContrasts(RC1, pickCoef(RC1, "[.]mental"),
+                    ref = rowProbs, scaleWeights = rowProbs)
> nu <- getContrasts(RC1, pickCoef(RC1, "[.]ses"),
+                    ref = colProbs, scaleWeights = colProbs)
```

In our notation, the coefficients α and β can be extracted as the `qvframe` component of the "qv" object returned by `getContrasts()`.

```
> (alpha <- mu$qvframe)

                            Estimate Std. Error
Mult(., ses).mentalWell      1.67378    0.19043
Mult(., ses).mentalMild      0.14009    0.20018
Mult(., ses).mentalModerate -0.13669    0.27948
Mult(., ses).mentalImpaired -1.41055    0.17418

> (beta  <- nu$qvframe)

                      Estimate Std. Error
Mult(mental, .).ses1  1.111361    0.29921
Mult(mental, .).ses2  1.120459    0.31422
```

```
Mult(mental, .).ses3  0.370752    0.31915
Mult(mental, .).ses4 -0.027006    0.27328
Mult(mental, .).ses5 -1.009480    0.31470
Mult(mental, .).ses6 -1.816647    0.28095
```

For plotting this RC(1) solution for the scaled category scores together with their estimated standard errors, a `dotchart()`, shown in Figure 10.5 provides a reasonable visualization.

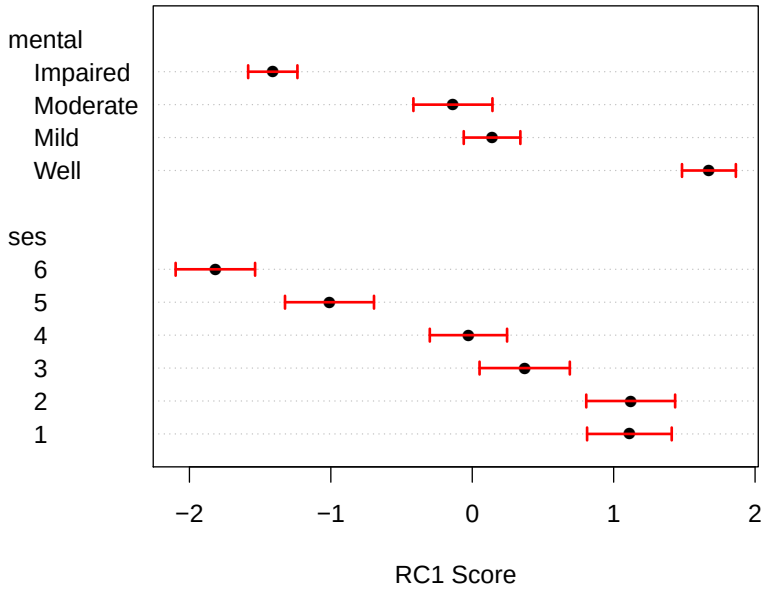


Figure 10.5: Dotchart of the scaled category scores for the RC(1) model fit the mental health data. Error bars show ±1 standard error. {fig:mental-RC1}

To create this plot, first combine the row and column scores in a data frame, and add columns `lower, upper` corresponding to ±1 standard error (or some other multiple).

```
> scores <- rbind(alpha, beta)
> scores <- cbind(scores,
+                 factor = c(rep("mental", 4), rep("ses", 6)) )
> rownames(scores) <- c(levels(Mental$mental), levels(Mental$ses))
> scores$lower <- scores[,1] - scores[,2]
> scores$upper <- scores[,1] + scores[,2]
> scores

          Estimate Std. Error factor   lower  upper
Well         1.674      0.190 mental  1.4834  1.864
Mild         0.140      0.200 mental -0.0601  0.340
Moderate    -0.137      0.279 mental -0.4162  0.143
Impaired    -1.411      0.174 mental -1.5847 -1.236
1            1.111      0.299    ses  0.8121  1.411
2            1.120      0.314    ses  0.8062  1.435
3            0.371      0.319    ses  0.0516  0.690
4           -0.027      0.273    ses -0.3003  0.246
5           -1.009      0.315    ses -1.3242 -0.695
6           -1.817      0.281    ses -2.0976 -1.536
```

The dotchart shown in Figure 10.5 is then a plot of `Estimate`, grouped by `factor`, with arrows showing the range of `lower` to `upper` for each parameter.

```
> with(scores, {
+   dotchart(Estimate, groups = factor, labels = rownames(scores),
+            cex = 1.2, pch = 16, xlab = "RC1 Score",
+            xlim = c(min(lower), max(upper)))
+   arrows(lower, c(8 + (1 : 4), 1 : 6), upper, c(8 + (1 : 4), 1 : 6),
+          col = "red", angle = 90, length = .05, code = 3, lwd = 2)
+   })
```

In this plot, the main substantive difference from the $L \times L$ model is in the spacing of the lowest two categories of `ses` and the middle two categories of `mental` which are not seen to differ in the RC1 model.

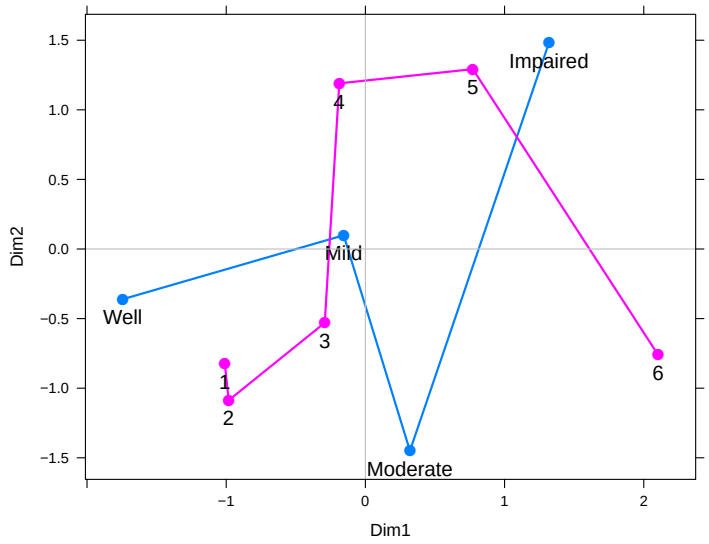
The coefficients in the `RC2` model can also be plotted (in a 2D plot) by extracting the coefficients from the "gnm" object and reshaping them to 2-column matrices. The function `pickCoef()` is handy here to get the indices of a subset of parameters by matching a pattern in their names.

```
> alpha <- coef(RC2)[pickCoef(RC2, "[.]mental")]
> alpha <- matrix(alpha, ncol=2)
> rownames(alpha) <- levels(Mental$mental)
> colnames(alpha) <- c("Dim1", "Dim2")
> alpha

              Dim1      Dim2
Well     -0.465689 -0.178558
Mild     -0.017207 -0.049355
Moderate -0.188652  0.491120
Impaired  0.573231 -0.213600

> beta <- coef(RC2)[pickCoef(RC2, "[.]ses")]
> beta <- matrix(beta, ncol=2)
> rownames(beta) <- levels(Mental$ses)
> colnames(beta) <- c("Dim1", "Dim2")
> beta

        Dim1      Dim2
1 -0.5799059 -0.294408
2 -0.5894086 -0.254062
3 -0.2011249 -0.114769
4  0.0021248 -0.292360
5  0.4826189 -0.017145
6  0.9553865  0.631284
```

For plotting and interpretation, these dimension scores need to be standardized as described at the start of this section (e.g., Eqn. (10.5)). We don't show the steps for doing this or producing a plot, because it is much simpler to use the logmult package, as described next. A basic plot using the marginal-weighted scaling is shown in Figure 10.6.

The patterns of the row and column category scores in Figure 10.6 are quite similar to the 2D correspondence analysis solution shown in Figure 6.2. The main difference is in the relative scaling of the axes. In Figure 10.6, the variances of the two dimensions are equated; in the correspondence analysis plot, the axes are scaled in relation to their contributions to Pearson χ^2, allowing an interpretation of distance between points in terms of χ^2-distance.

△



{fig:mental-RC2} **Figure 10.6:** Scaled category scores for the RC(2) model fit the mental health data.

10.1.3.1 Using logmult

It takes a fair bit of work to extract the coefficients from "gnm" objects and carry out the scaling necessary for informative plots. Much of this effort is now performed by the logmult package with several convenience functions that do the heavy lifting.

`rc()` fits the class of RC(M) models, allowing an argument `nd` to specify the number of dimensions, and also providing for standard errors estimated using jackknife and bootstrap methods (Milan and Whittaker, 1995), which are computationally intensive. For square tables, a `symmetric` argument constrains the row and column scores to be equal, and a `diagonal` option fits parameters for each diagonal cell, providing for models of quasi-independence and quasi-symmetry (see Section 10.2).

It returns an object of class "rc" with the components of the "gnm" object. An `assoc` component is also returned, containing the normalized association parameters for the categories.

`rcL()` fits extensions of RC models to tables with multiple layers, called RC(M)-L models by Wong (2010).

`plot.rc()` is a plot method for visualizing scores for RC(M) models in two selected dimensions. Among other options, it can plot confidence ellipses for the category scores, using the estimated covariance matrix (assuming a normal distribution of the category scores). The plot method returns (invisibly) the coordinates of the scores as plotted, facilitating additional plot annotation.

{ex:mental6}

EXAMPLE 10.4: Mental impairment and parents' SES

Here we use `rc()` to estimate the RC(1) and RC(2) models for the *`Mental`* data. In contrast to `gnm()`, which has a formula interface for a `data` argument, `rc()` requires the input in the form of a two-way table, given here as `mental.tab`.

```
> library(logmult)
> rc1 <- rc(mental.tab, verbose = FALSE, weighting = "marginal",
+           se = "jackknife")
> rc2 <- rc(mental.tab, verbose = FALSE, weighting = "marginal", nd = 2,
+           se = "jackknife")
```

The option `weighting="marginal"` gives the marginally-weighted solution and `se = "jackknife"` estimates the covariance matrix using the leave-one-out jackknife.[4]

A plot of the scaled category scores similar to Figure 10.6, with 1 standard error confidence ellipses (making them comparable to the 1D solution shown in Figure 10.5) but no connecting lines can then be easily produced with the `plot()` method for "rc" objects.

```
> coords  <- plot(rc2, conf.ellipses = 0.68, cex = 1.5, rev.axes = c(TRUE, FALSE))
```

The orientation of the axes is arbitrary in RC(M) models, so the horizontal axis is reversed here to conform with Figure 10.6.

This produces (in Figure 10.7) a symmetric biplot in which the scaled coordinates of points for rows (α_{ik}) and columns (β_{jk}) on both axes are the product of normalized scores and the square root of the intrinsic association coefficient (γ_k) corresponding to each dimension.

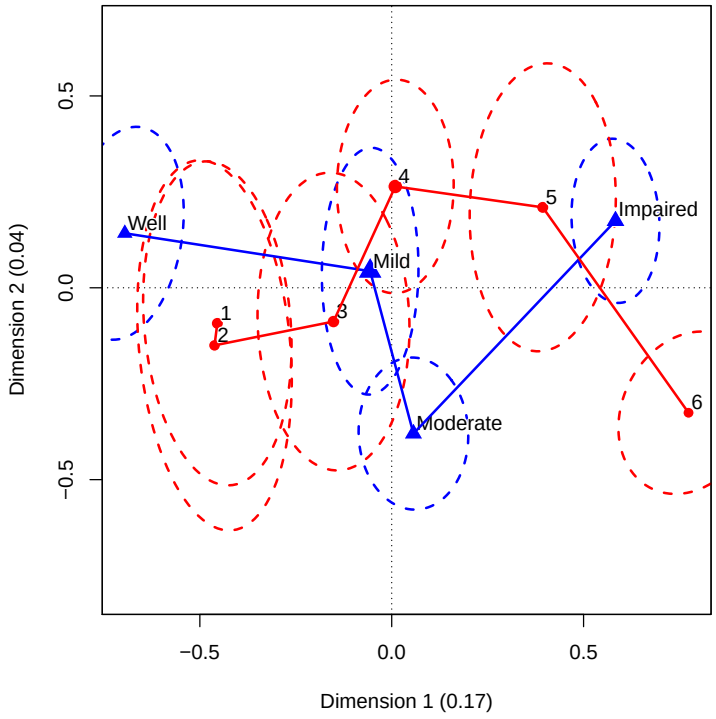


Figure 10.7: Scaled category scores for the RC(2) model fit and plotted using the logmult package. The 68% confidence ellipses correspond to bivariate ± 1 confidence intervals for the category parameters.

{fig:mental-logmult-rc2}

[4] Becker and Clogg (1989) recommend using unweighted solutions, `weighting="none"` (they call them "uniformly weighted") to preserve independence of inferences about association and marginal effects and estimates of the intrinsic association parameters, γ_k. That choice makes very little difference in the plots for this example, but the γ_k parameters are affected considerably.

Such plots can be customized using the category coordinates (`coords`) returned by the `plot()` method. As in other biplots, joining the row and column points by lines (sorted by the first dimension) makes it easier to see their relationships across the two dimensions. The following code draws the lines shown in Figure 10.7.

```
> scores <- rbind(coords$row, coords$col)
> lines(scores[1 : 4,], col = "blue", lwd = 2)
> lines(scores[-(1 : 4),], col = "red", lwd = 2)
```

We saw earlier that there was not strong evidence supporting the need for a second RC dimension to describe the relationship between mental health and SES. This is apparent in the sizes of the confidence ellipses, which overlap much more along Dimension 2 than Dimension 1. △

10.2 Square tables

{sec:loglin-square}

Square tables, where the row and column variables have the same categories comprise an important special case for loglinear models that can account for associations more parsimoniously than the saturated model. Some examples are the data on visual acuity in Example 4.14, categorical ratings of therapy clients by two observers, and mobility tables, tracking the occupational categories between generations in the same families, or migration tables, giving movement of people between regions. The latter topics has been important in sociological and geographic research and has spurred the development of a wide range of specialized loglinear models for this purpose.

10.2.1 Quasi-independence, symmetry, quasi-symmetry and topological models

{sec:sq-quasi}

In many square tables, such as the `Vision` data, independence is not a credible hypothesis because the diagonal cells, representing equal values of the row and column variables, tend to be very large and often contribute most of the lack of fit. A substantively more interesting hypothesis is whether the table exhibits independence, ignoring the diagonal cells. This leads to what is called the ***quasi-independence model***, that specifies independence only in the off-diagonal cells.

For a two-way table, quasi-independence can be expressed as

$$\pi_{ij} = \pi_{i+}\pi_{+j} \qquad \text{for } i \neq j$$

or in loglinear form as

$$\log m_{ij} = \mu + \lambda_i^A + \lambda_j^B + \delta_i I(i = j) \ .$$

This model effectively adds one parameter, δ_i, for each main diagonal cell which fits those frequencies perfectly.

Another hypothesis of substantive interest for square tables, particularly those concerning occupational and geographical mobility is that the joint distribution of row and column variables is symmetric, that is, $\pi_{ij} = \pi_{ji}$ for all $i \neq j$. For example, this ***symmetry model*** (S) asserts that sons are as likely to move from their father's occupation i to another, j, as the reverse. This form of symmetry is quite strong, because it also implies ***marginal homogeneity*** (MH), that the marginal probabilities of the row and column variables are equal, $\pi_{i+} = \sum_j \pi_{ij} = \sum_j \pi_{ji} = \pi_{+i}$ for all i.

To separate marginal homogeneity from symmetry of the association terms per se, the model of ***quasi-symmetry*** (QS) uses the standard main-effect terms in the loglinear model,

$$\log m_{ij} = \mu + \lambda_i^A + \lambda_j^B + \lambda_{ij} \ , \tag{10.7}$$

{eq:quasi-symm}

where $\lambda_{ij} = \lambda_{ji}$. It can be shown (Caussinus, 1966) that

$$\begin{aligned} \text{symmetry} &= \text{quasi-symmetry} + \text{marginal homogeneity} \\ G^2(S) &= G^2(QS) + G^2(MH) \end{aligned}$$

where $G^2(MH)$ is defined by the likelihood-ratio test of the difference between the S and QS models,

{eq:mh}
$$G^2(MH) \equiv G^2(S \,|\, QS) = G^2(S) - G^2(QS) \ . \tag{10.8}$$

The gnm package provides several model building convenience functions that facilitate fitting these and related models:

- `Diag(row, col, ...)` constructs a diagonals association factor for two (or more) factors with integer levels where the original factors are equal, and `"."` otherwise.
- `Symm(row, col, ...)` constructs an association factor giving equal levels to sets of symmetric cells. The QS model is specified using `Diag() + Symm()`.
- `Topo(row, col, ..., spec)` creates an association factor for two or more factors, as specified by an array of levels, which may be arbitrarily structured. Both `Diag()` and `Symm()` factors are special cases of `Topo()`.

The factor levels representing these association effects for a 4×4 table are shown below by their unique values in each array.

$$\text{Diag}_{4\times4} = \begin{bmatrix} 1 & . & . & . \\ . & 2 & . & . \\ . & . & 3 & . \\ . & . & . & 4 \end{bmatrix} \quad \text{Symm}_{4\times4} = \begin{bmatrix} 11 & 12 & 13 & 14 \\ 12 & 22 & 23 & 24 \\ 13 & 23 & 33 & 34 \\ 14 & 24 & 34 & 44 \end{bmatrix} \quad \text{Topo}_{4\times4} = \begin{bmatrix} 2 & 3 & 4 & 4 \\ 3 & 3 & 4 & 4 \\ 4 & 4 & 5 & 5 \\ 4 & 4 & 5 & 1 \end{bmatrix}$$

{ex:vision-glm}

EXAMPLE 10.5: Visual acuity

Example 4.14 presented the data on tests of visual acuity in the left and right eyes of a large sample of women working in the Royal Ordnance factories in World War II. A sieve diagram (Figure 4.10) showed that, as expected, most women had the same acuity in both eyes, but the off-diagonal cells had a pattern suggesting some form of symmetry.

The data set *VisualAcuity* contains data for both men and women in frequency form and for this example we subset this to include only the 4×4 table for women.

```
> data("VisualAcuity", package="vcd")
> women <- subset(VisualAcuity, gender=="female", select=-gender)
```

The four basic models of independence, quasi-independence, symmetry and quasi-symmetry for square tables are fit as shown below. We use `update()` to highlight the relations among these models in two pairs.

```
> #library(vcdExtra)
> indep <- glm(Freq ~ right + left,  data = women, family = poisson)
> quasi <- update(indep, . ~ . + Diag(right, left))
>
> symm <- glm(Freq ~ Symm(right, left), data = women, family = poisson)
> qsymm <- update(symm, . ~ right + left + .)
```

The brief summary of goodness of fit of these models below shows that the QS model fits reasonably well, but none of the others do by likelihood-ratio tests or AIC or BIC.

```
> vcdExtra::LRstats(indep, quasi, symm, qsymm)

Likelihood summary table:
       AIC  BIC LR Chisq Df Pr(>Chisq)
indep 6803 6808     6672  9     <2e-16 ***
quasi  338  347      199  5     <2e-16 ***
symm   157  164       19  6     0.0038 **
qsymm  151  161        7  3     0.0638 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Beyond just saying that the QS model fits best, the reasons *why* it does can be seen in mosaic displays. Figure 10.8 compares the mosaics for the models of quasi-independence (accounting only for the diagonal cells) and quasi-symmetry (also accounting for symmetry). It can be seen in the left panel that the non-diagonal associations are largely symmetric, and also that when they differ, visual acuity in the two eyes are most likely to differ by only one eye grade.

```
> labs <- c("High", "2", "3", "Low")
> largs <- list(set_varnames = c(right = "Right eye grade",
+                                left = "Left eye grade"),
+               set_labels=list(right = labs, left = labs))
> mosaic(quasi, ~ right + left, residuals_type = "rstandard",
+        gp = shading_Friendly,
+        labeling_args = largs,
+        main = "Quasi-Independence (women)")
> mosaic(qsymm, ~ right + left, residuals_type = "rstandard",
+        gp = shading_Friendly,
+        labeling_args = largs,
+        main = "Quasi-Symmetry (women)")
```

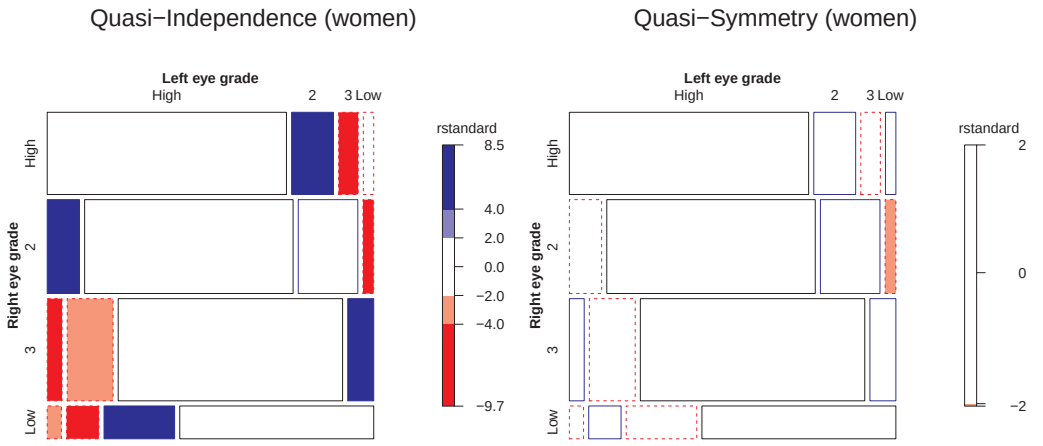


Figure 10.8: Mosaic displays comparing the models of quasi-independence and quasi-symmetry for visual acuity in women. fig:vision-mosaics

Finally, as usual, `anova()` can be used to carry out specific tests of nested models. For example, the test of marginal homogeneity Eqn. (10.8) compares models S and QS and shows here that the marginal probabilities for the left and right eyes differ.

```
> anova(symm, qsymm, test = "Chisq")

Analysis of Deviance Table

Model 1: Freq ~ Symm(right, left)
Model 2: Freq ~ right + left + Symm(right, left)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         6      19.25
2         3       7.27  3       12   0.0075 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

△

{ex:hauser1}

EXAMPLE 10.6: Hauser's occupational mobility table

The data *Hauser79* in vcdExtra (Friendly, 2015), from Hauser (1979), gives a 5×5 table in frequency form cross-classifying 19,912 individuals in the United States by father's occupation and son's first occupation. The occupational categories are represented by abbreviations, of Upper Non-Manual (UpNM), Lower Non-Manual (LoNM), Upper Manual (UpM), Lower Manual (LoM) and Farm. These data were also analysed by Powers and Xie (2008, Chapter 4).

```
> data("Hauser79", package = "vcdExtra")
> structable(~ Father + Son, data = Hauser79)

       Son UpNM LoNM  UpM  LoM Farm
Father
UpNM      1414  521  302  643   40
LoNM       724  524  254  703   48
UpM        798  648  856 1676  108
LoM        756  914  771 3325  237
Farm       409  357  441 1611 1832
```

Before fitting any models, it is useful to calculate and plot the observed local log odds ratios, as we did in Example 10.1 to see the patterns in the data that need to be accounted for. These are calculated using `loddsratio()`.

```
> hauser.tab <- xtabs(Freq ~ Father + Son, data = Hauser79)
> (lor.hauser <- loddsratio(hauser.tab))

log odds ratios for Father and Son

           Son
Father      UpNM:LoNM LoNM:UpM UpM:LoM LoM:Farm
  UpNM:LoNM     0.675   -0.179   0.262   0.0931
  LoNM:UpM      0.115    1.003  -0.346  -0.0579
  UpM:LoM       0.398   -0.449   0.790   0.1009
  LoM:Farm     -0.326    0.381  -0.166   2.7697
```

This 4×4 table is graphed using `plot(lor.hauser)`, giving Figure 10.9.

```
> plot(lor.hauser, confidence = FALSE, legend_pos = "topleft",
+      xlab = "Father's status comparisons")
> m <- mean(lor.hauser$coefficients)        # mean LOR
> grid.lines(x = unit(c(0, 1), "npc"),
+            y = unit(c(m, m), "native"))
```

Amongst the features here, you can see that there is a tendency for the odds ratio contrasting sons in the non-manual categories (`UpNM:LoNM`) to decline with the adjacent comparisons of their

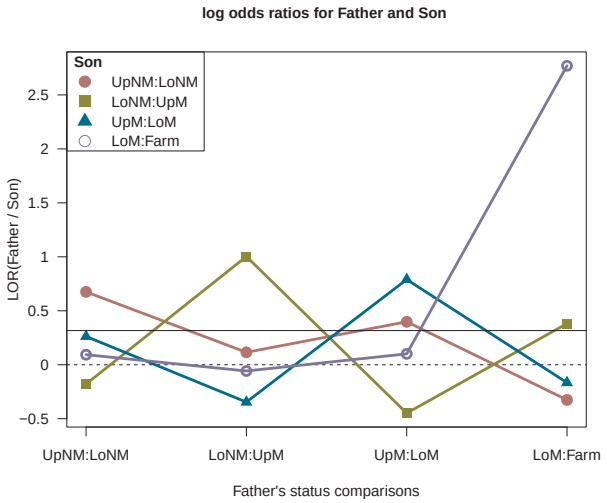


Figure 10.9: Plot of observed local log odds ratios in the Hauser79 data. The dotted horizontal line at zero shows local independence; the solid black horizontal line shows the mean. fig:hauser-lor-plot

fathers' occupations. As well, the 2×2 table for fathers and sons in the `LoM:Farm` stands out as deserving some attention. These observed features will be smoothed by fitting models, as described below. For additional interpretation, you can always construct similar plots of the log odds ratios using the `fitted()` values (see: Section 10.1.2) from any of the models described below.

We begin by fitting the independence model and the quasi-independence model, where the diagonal parameters in the latter are specified as `Diag(Father,Son)`. As expected, given the large frequencies in the diagonal cells, the quasi-independence model is a considerable improvement, but the fit is still very poor.

```
> hauser.indep <- gnm(Freq ~ Father + Son, data = Hauser79, family = poisson)
> hauser.quasi <-  update(hauser.indep, ~ . + Diag(Father, Son))
> vcdExtra::LRstats(hauser.indep, hauser.quasi)

Likelihood summary table:
             AIC  BIC LR Chisq Df Pr(>Chisq)
hauser.indep 6391 6402     6170 16     <2e-16 ***
hauser.quasi  914  931      683 11     <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The pattern of associations can be seen in the mosaic displays for both models, shown in Figure 10.10.

```
> mosaic(hauser.indep, ~ Father + Son, main = "Independence model",
+        gp = shading_Friendly)
> mosaic(hauser.quasi, ~ Father + Son, main = "Quasi-independence model",
+        gp = shading_Friendly)
```

The mosaic for quasi-independence shows an approximately symmetric pattern of residuals, so we proceed to add `Symm(Father,Son)` to the model to specify quasi-symmetry.

```
> hauser.qsymm <-  update(hauser.indep,
+                         ~ . + Diag(Father, Son) + Symm(Father, Son))
> vcdExtra::LRstats(hauser.qsymm)
```

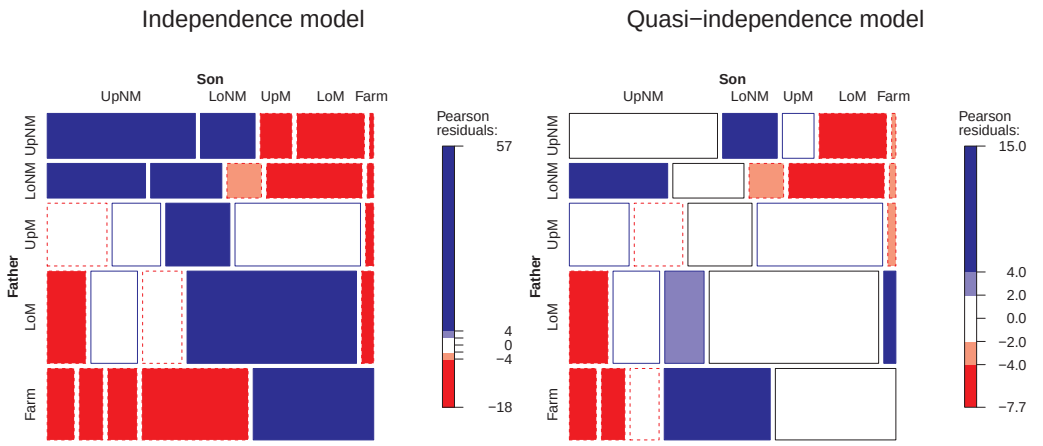


Figure 10.10: Mosaic displays for the Hauser79 data. Left: independence model; right:quasi-independence model. fig:hauser-mosaic1

```
Likelihood summary table:
             AIC BIC LR Chisq Df Pr(>Chisq)
hauser.qsymm 268 291     27.4  6    0.00012 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This model represents a huge improvement in goodness of fit. With such a large sample size, it might be considered an acceptable fit. The remaining lack of fit is shown in the mosaic for this model, Figure 10.11.

```
> mosaic(hauser.qsymm, ~ Father + Son, main = "Quasi-symmetry model",
+        gp = shading_Friendly, residuals_type = "rstandard")
```

The cells with the largest lack of symmetry (using standardized residuals) are those for the upper and lower non-manual occupations, where the son of an upper manual worker is less likely to move to lower non-manual work than the reverse.

For cases like this involving structured associations in square tables, Hauser (1979) developed the more general idea of grouping the row and column categories into levels of an association factor based on similar values of residuals or local odds ratios observed from the independence model. Such models are called ***topological model***s or ***levels model***s, which are implemented in the `Topo()` function.

To illustrate, Hauser suggested the following matrix of levels to account for the pattern of associations seen in Figure 10.10. The coding here takes the diagonal cell for the Farm category as the reference cell. Four other parameters are assigned by the numbers 2–5 to account for lack of independence.

```
> levels <- matrix(c(
+   2,  4,  5,  5,  5,
+   3,  4,  5,  5,  5,
+   5,  5,  5,  5,  5,
+   5,  5,  5,  4,  4,
+   5,  5,  5,  4,  1
+   ),
+   5, 5, byrow = TRUE)
```

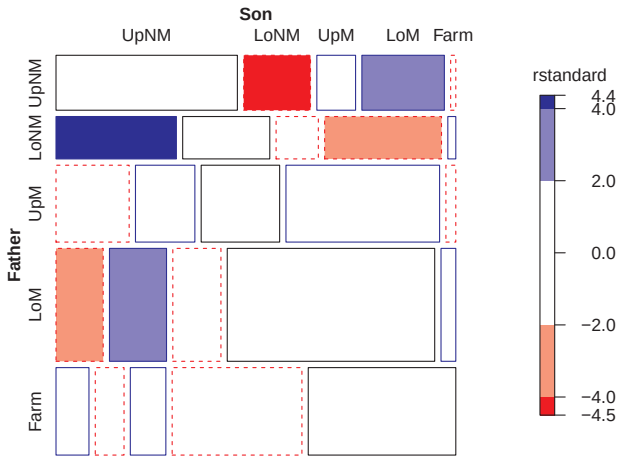


Figure 10.11: Mosaic display for the model of quasi-symmetry fit to the Hauser79 data. fig:hauser-mosaic2

This models is fit using `Topo()` as shown below. It also provides a huge improvement over the independence model, with 4 additional parameters.

```
> hauser.topo <- update(hauser.indep, ~ . + Topo(Father, Son, spec = levels))
> vcdExtra::LRstats(hauser.topo)

Likelihood summary table:
            AIC BIC LR Chisq Df Pr(>Chisq)
hauser.topo 295 311     66.6 12    1.4e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As with other models fit using `gnm()`, you can extract the coefficients for particular terms using `pickCoef()`.

```
> as.vector((coef(hauser.topo)[pickCoef(hauser.topo, "Topo")]))

[1] -1.8128 -2.4973 -2.8035 -3.4026
```

The models fit in this example are summarized below. Both AIC and BIC prefer the quasi-symmetry model, `hauser.quasi`.

```
> LRstats(hauser.indep, hauser.quasi, hauser.qsymm, hauser.topo)

Likelihood summary table:
              AIC  BIC LR Chisq Df Pr(>Chisq)
hauser.indep 6391 6402     6170 16    < 2e-16 ***
hauser.quasi  914  931      683 11    < 2e-16 ***
hauser.qsymm  268  291       27  6    0.00012 ***
hauser.topo   295  311       67 12    1.4e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

△

10.2.2 Ordinal square tables

{sec:sq-ordinal}

The theory presented in Section 10.2.1 treats the row and column variables as nominal. In many instances, such as Example 10.6, the variable categories are also ordered, yet these models do not exploit their ordinal nature. In such cases, the models such as uniform association ($L \times L$), row effects, RC and others discussed in Section 10.1 can be combined with terms for quasi-independence and symmetry of the remaining associations.

For example, the $L \times L$ model Eqn. (10.2) of uniform association applies directly to square tables, and, for square tables, can also be amended to include a diagonals term, `Diag()`, giving a model of *quasi-uniform association*. In this model, all adjacent 2×2 sub-tables not involving diagonal cells have a common local odds ratio.

A related model is the ***crossings model*** (Goodman, 1972). This hypothesizes that there are different difficulty parameters for crossing from one category to the next, and that the associations between categories decreases with their separation. In the crossings model for an $I \times I$ table, there are $I - 1$ crossings parameters, $\nu_1, \nu_2, \ldots, \nu_{I-1}$. The association parameters, λ_{ij}^{AB} have the form of the product of the intervening ν parameters,

$$\lambda_{ij}^{AB} = \begin{cases} \prod_{k=j}^{k=i-1} \nu_k & : \quad i > j \\ \prod_{k=i}^{k=j-1} \nu_k & : \quad i < j \end{cases}$$

This model can also be cast in *quasi* form, by addition of a `Diag` term to fit the main diagonal cells. See Powers and Xie (2008, §4.4.7) for further details of this model. The `Crossings()` function in vcdExtra implements such crossings terms.

{ex:hauser2}

EXAMPLE 10.7: Hauser's occupational mobility table

Without much comment or detail, for reference we first fit some of the ordinal models to the *Hauser79* data: Uniform association ($L \times L$), row effects, and the RC(1) model.

```
> Fscore <- as.numeric(Hauser79$Father)    # numeric scores
> Sscore <- as.numeric(Hauser79$Son)       # numeric scores
>
> # uniform association
> hauser.UA <- update(hauser.indep, ~ . + Fscore * Sscore)
> # row effects model
> hauser.roweff <- update(hauser.indep, ~ . + Father * Sscore)
> # RC model
> hauser.RC <- update(hauser.indep, ~ . + Mult(Father, Son), verbose = FALSE)
```

All of these fit very poorly, yet they are all substantial improvements over the independence model.

```
> vcdExtra::LRstats(hauser.indep, hauser.UA, hauser.roweff, hauser.RC)

Likelihood summary table:
               AIC  BIC LR Chisq Df Pr(>Chisq)
hauser.indep  6391 6402     6170 16     <2e-16 ***
hauser.UA     2503 2516     2281 15     <2e-16 ***
hauser.roweff 2309 2325     2080 12     <2e-16 ***
hauser.RC      920  940      685  9     <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The $L \times L$ model, `hauser.UA` might be improved by ignoring the diagonals, and, indeed it is.

```
> hauser.UAdiag <- update(hauser.UA, ~ . + Diag(Father, Son))
> anova(hauser.UA, hauser.UAdiag, test = "Chisq")

Analysis of Deviance Table

Model 1: Freq ~ Father + Son + Fscore + Sscore + Fscore:Sscore
Model 2: Freq ~ Father + Son + Fscore + Sscore + Fscore:Sscore + Diag(Father,
    Son)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        15       2281
2        10         73  5     2208   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In this model, the estimated common local log odds ratio—the coefficient for the linear-by-linear term `Fscore:Sscore` is

```
> coef(hauser.UAdiag)[["Fscore:Sscore"]]

[1] 0.1584
```

For comparisons not involving the diagonal cells, each step down the scale of occupational categories for the father multiplies the odds that the son will also be in one lower category by $\exp(0.158) = 1.172$, an increase of 17%.

The crossings model, with and without the diagonal cells can be fit as follows:

```
> hauser.CR <- update(hauser.indep, ~ . + Crossings(Father, Son))
> hauser.CRdiag <- update(hauser.CR, ~ . + Diag(Father, Son))
> vcdExtra::LRstats(hauser.CR, hauser.CRdiag)

Likelihood summary table:
             AIC BIC LR Chisq Df Pr(>Chisq)
hauser.CR     319 334     89.9 12    5.1e-14 ***
hauser.CRdiag 299 318     64.2  9    2.0e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The quasi-crossings model `hauser.CRdiag` has a reasonable G^2 fit statistic, and its interpretation and lack of fit is worth exploring further. The crossings coefficients ν can be extracted as follows.

```
> nu <- coef(hauser.CRdiag)[pickCoef(hauser.CRdiag, "Crossings")]
> names(nu) <- gsub("Crossings(Father, Son)C", "nu", names(nu), fixed = TRUE)
> nu

     nu1      nu2      nu3      nu4
-0.42275 -0.38768 -0.27500 -1.40244
```

They indicate the steps between adjacent categories in terms of the barriers for a son moving to a lower occupational category. The numerically largest gap separates the lower non-manual category from farming.

In contrast to the `UAdiag` model, the quasi-crossing model with diagonal terms implies that all 2×2 off-diagonal sub-tables are independent, i.e., the local odds ratios are all equal to 1.0. The reasons for lack of fit of this model can be seen in the corresponding mosaic display, shown in Figure 10.12

```
> mosaic(hauser.CRdiag, ~ Father + Son,
+        gp = shading_Friendly, residuals_type = "rstandard",
+        main = "Crossings() + Diag()")
```

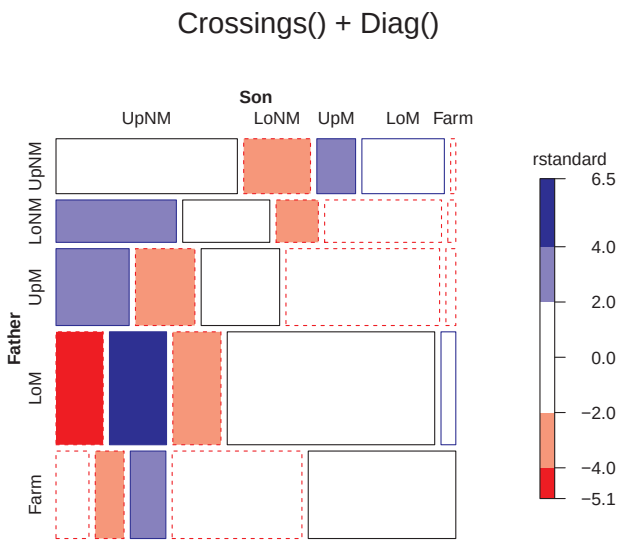


Figure 10.12: Mosaic display for the quasi-crossings model fit to the Hauser79 data. fig:hauser-mosaic3

It can be seen that lack of fit for this model is largely concentrated in the lower triangle, where the father's occupation is lower than that of his son.

In this example and the last, we have fit quite a few different models to the Hauser (1979) data. In presentations, articles and books it is common to summarize such a collection in a table, sorted by G^2, degrees of freedom, AIC or BIC, to show their ordering along some metric. For instance, here we collect all the models fit in Example 10.6 and this example in a `glmlist()` and sort in decreasing order of BIC to show model fit by this measure.

```
> modlist <- glmlist(hauser.indep, hauser.roweff, hauser.UA, hauser.UAdiag,
+                    hauser.quasi, hauser.qsymm,  hauser.topo,
+                    hauser.RC, hauser.CR, hauser.CRdiag)
> LRstats(modlist, sortby = "BIC")

Likelihood summary table:
               AIC  BIC LR Chisq Df Pr(>Chisq)
hauser.indep  6391 6402     6170 16    < 2e-16 ***
hauser.UA     2503 2516     2281 15    < 2e-16 ***
hauser.roweff 2309 2325     2080 12    < 2e-16 ***
hauser.RC      920  940      685  9    < 2e-16 ***
hauser.quasi   914  931      683 11    < 2e-16 ***
hauser.CR      319  334       90 12    5.1e-14 ***
hauser.UAdiag  306  324       73 10    1.2e-11 ***
hauser.CRdiag  299  318       64  9    2.0e-10 ***
hauser.topo    295  311       67 12    1.4e-09 ***
hauser.qsymm   268  291       27  6    0.00012 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

When there are more than just a few models, a more useful display is a ***model comparison plot*** of measures like G^2/df, AIC or BIC against degrees of freedom. For example, Figure 10.13 plots `BIC` against `Df` from the result of `LRstats()`. Because interest is focused on the smallest values of BIC and these values span a large range, BIC is shown on the log scale using `log="y"`.

```
> sumry <- LRstats(modlist)
> mods <- substring(rownames(sumry), 8)
> with(sumry, {
+   plot(Df, BIC, cex = 1.3, pch = 19,
+        xlab = "Degrees of freedom", ylab = "BIC (log scale)",
+        log = "y", cex.lab = 1.2)
+   pos <- ifelse(mods == "UAdiag", 1, 3)
+   text(Df, BIC + 55, mods, pos = pos, col = "red", xpd = TRUE, cex = 1.2)
+   })
```

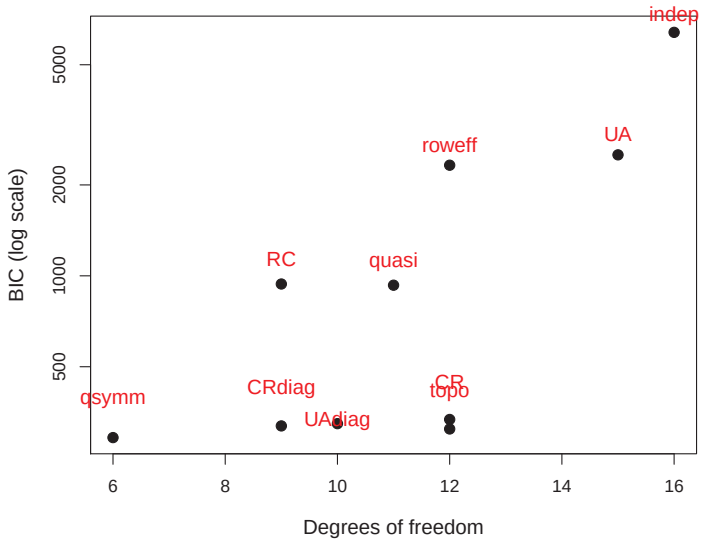


Figure 10.13: Model comparison plot for the models fit to the Hauser79 data fig:hauser-sumry-plot

Compared with the sorted tabular display shown above, such a plot sorts the models *both* by a measure of fit and by model complexity (degrees of freedom). Figure 10.13 shows that the quasi-symmetry model is best by BIC, but also shows that the next four best models by this measure are quite similar in terms of BIC. Similar plots for AIC and G^2/df show that the model of quasi-symmetry is favored by these measures.

△

10.3 Three-way and higher-dimensional tables

{sec:loglin-ord3way}

The models and methods for ordinal factors and square tables described in Section 10.1 and Section 10.2 extend readily to multidimensional tables with these properties for some of the factors. In three-way tables, these models provide a more parsimonious account than the saturated

model, $[ABC]$, and also allow simpler models than the general model of homogeneous association, $[AB][AC][BC]$ using scores for ordinal factors or terms for symmetry and diagonal factors in square layers.

For example, consider the case where all three factors are ordinal and the model of homogeneous association $[AB][AC][BC]$ fits poorly. In this case we can generalize the model of uniform association by assigning scores $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ and model the three-way association, λ_{ijk}^{ABC} as

$$\lambda_{ijk}^{ABC} = \gamma a_i b_j c_k$$

with only one more parameter. This gives the model of ***uniform interaction*** (or *homogeneous uniform association*)

{eq:uni-inter}
$$\log(m_{ijk}) = \mu + \lambda_i^A + \lambda_j^B + \lambda_k^C + \lambda_{ij}^{AB} + \lambda_{ik}^{AC} + \lambda_{jk}^{BC} + \gamma a_i b_j c_k \ . \tag{10.9}$$

This model posits that (with equally spaced scores) all local odds ratios θ_{ijk} in adjacent rows, columns and layers are constant,

$$\log(\theta_{ijk}) = \gamma \qquad \forall \quad i, j, k$$

The homogeneous association model is the special case of $\log \theta_{ijk} = \gamma = 0$.

A less restricted model of ***heterogeneous uniform association*** retains the linear-by-linear form of association for factors A and B, but allows the strength of this association to vary over layers, C, representing λ_{ijk}^{ABC} as

$$\lambda_{ijk}^{ABC} = (\gamma + \gamma_k) a_i b_j$$

with the constraint $\sum_k \gamma_k = 0$. This model is equivalent to fitting separate models of uniform association at each level k of factor C and gives estimates of the conditional local log odds ratios, $\log \theta_{ij(k)} = \gamma + \gamma_k$.

Following the development in Section 10.1 there is a large class of other models for ordinal factors (see Figure 10.1), where not all factors are assigned scores. For three-way tables, these can be represented in homogeneous form when the two-way association of A and B is the same for all levels of C, or in a heterogeneous form, when it varies over C.

Similarly, the models for square tables described in Section 10.2 extend to three-way tables with several layers (strata), allowing both homogeneous and heterogeneous terms for diagonals and symmetry describing the AB association over levels of C.

{ex:vision-glm2}

EXAMPLE 10.8: Visual acuity

We continue the analysis of the `VisualAcuity` data, but now consider the three-way, $4 \times 4 \times 2$ table comprising both men and women. The main questions here are whether the pattern of quasi-symmetry observed in the analysis for women also pertains to men and whether there is heterogeneity of the association between right, left acuity across gender.

A useful first step for n-dimensional tables is to consider the models composed of all 1-way, 2-way, ... n-way terms as a quick overview. The function `Kway()` in vcdExtra package does this automatically, returning a "glmlist" object containing the fitted models. That is, for this problem, `Kway()` generates (and fits) the following model formulae, also including the 0-way model, corresponding to $\log m_{ijk\ldots} = \mu$.

```
> Freq ~ 1
> Freq ~ right + left + gender
> Freq ~ (right + left + gender)^2
> Freq ~ (right + left + gender)^3
```

We use `Kway()` as follows:

```
> vis.kway <- Kway(Freq ~ right + left + gender, data = VisualAcuity)
> vcdExtra::LRstats(vis.kway)

Likelihood summary table:
         AIC   BIC LR Chisq Df Pr(>Chisq)
kway.0 13857 13858    13631 31    < 2e-16 ***
kway.1  9925  9937     9686 24    < 2e-16 ***
kway.2   298   332       28  9    0.00079 ***
kway.3   287   334        0  0    < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This shows that the model of homogeneous association `kway.2` ($[RL][RG][LG]$) does not fit well, but it doesn't account for diagonal agreement or symmetry to simplify the associations.

As a basis for comparison, we first fit the simple models of quasi-independence and quasi-symmetry that do not involve `gender`, asserting the same pattern of diagonal and off-diagonal cells for males and females.

```
> vis.indep <- glm(Freq ~ right + left + gender,  data = VisualAcuity,
+                  family = poisson)
> vis.quasi <- update(vis.indep, . ~ . + Diag(right, left))
> vis.qsymm <- update(vis.indep, . ~ . + Diag(right, left) + Symm(right, left))
>
> LRstats(vis.indep, vis.quasi, vis.qsymm)

Likelihood summary table:
           AIC  BIC LR Chisq Df Pr(>Chisq)
vis.indep 9925 9937     9686 24     <2e-16 ***
vis.quasi  696  714      449 20     <2e-16 ***
vis.qsymm  435  456      184 18     <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The model of homogeneous quasi-symmetry fits quite badly, even worse than the all two-way association model. We can see why in the mosaic for this model, shown in Figure 10.14.

```
> mosaic(vis.qsymm, ~ gender + right + left, condvars = "gender",
+        residuals_type = "rstandard", gp = shading_Friendly,
+        labeling_args = largs,
+        main = "Homogeneous quasi-symmetry")
```

It can be seen in Figure 10.14 that the pattern of residuals for men and women are nearly completely opposite in the upper and lower portions of the plot: men have positive residuals in the same `right`, `left` cells where women have negative residuals, and vice-versa. In particular, the diagonal cells of both tables have large absolute residuals, because the term `Diag(right, left)` fits a common set of diagonals for both men and women.

We can correct for this by allowing separate diagonal and symmetry terms, given as interactions of `gender` with `Diag()` and `Symm()`.

```
> vis.hetdiag <- update(vis.indep, . ~ . + gender * Diag(right, left) +
+                       Symm(right, left))
> vis.hetqsymm <- update(vis.indep, . ~ . + gender * Diag(right, left) +
+                        gender * Symm(right, left))
> LRstats(vis.qsymm, vis.hetdiag, vis.hetqsymm)

Likelihood summary table:
             AIC BIC LR Chisq Df Pr(>Chisq)
vis.qsymm    435 456    183.7 18    < 2e-16 ***
vis.hetdiag  312 338     52.3 14    2.5e-06 ***
```

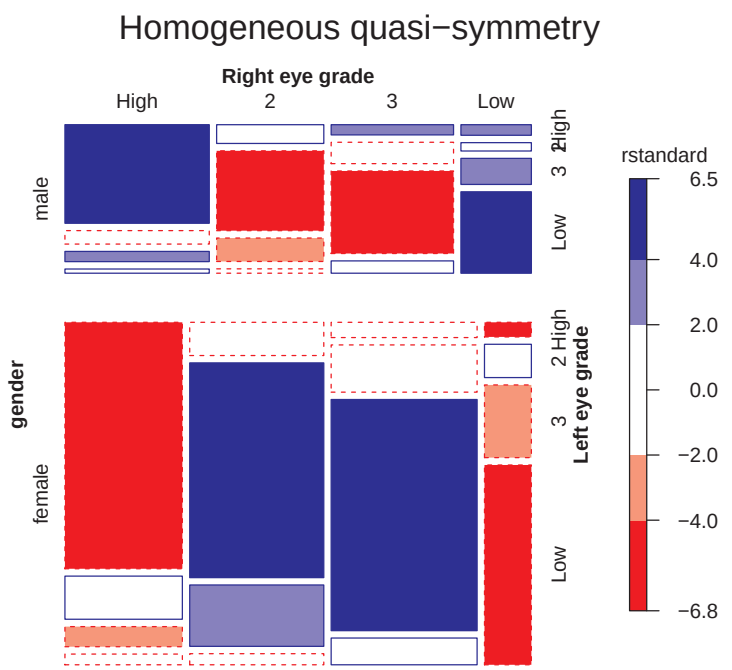


Figure 10.14: Mosaic display for the model of homogeneous quasi-symmetry fit to the VisualAcuity data. fig:vision2-qsymm

```
vis.hetqsymm 287 321     17.7  9      0.038 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that the model `vis.hetqsymm` fits better than the model `vis.hetdiag` in absolute terms, but the latter, with fewer parameters, fits better by AIC and BIC. The mosaic for the model `vis.hetqsymm` is shown in Figure 10.15.

```
> mosaic(vis.hetqsymm, ~ gender + right + left, condvars="gender",
+        residuals_type="rstandard", gp=shading_Friendly,
+        labeling_args=largs,
+        main="Heterogeneous quasi-symmetry")
```

As in the two-way case, this model now fits the diagonal cells in each table exactly, effectively ignoring this part of the association between right and left eye acuity. All remaining residuals are relatively small in magnitude, except for the two opposite off-diagonal cells `(Low, High)` and `(High, Low)` in the table for women.

The substantive interpretation of this example is that visual acuity is largely the same (diagonal cells) in the right and left eyes of both men and women. Ignoring the diagonal cells, when visual acuity differs, both men and women exhibit approximately symmetric associations. However, deviations from symmetry (Figure 10.14) are such that men are slightly more likely to have a lower grade in the right eye, while women are slightly more likely to have a higher grade in the right eye.

△

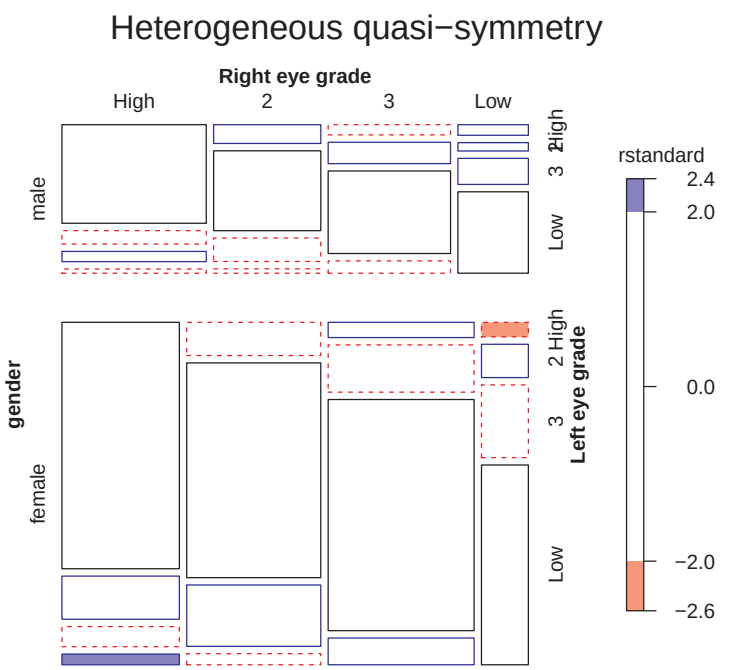


Figure 10.15: Mosaic display for the model of heterogeneous quasi-symmetry fit to the VisualAcuity data. fig:vision2-hetqsymm

10.4 Multivariate responses⋆

{sec:loglin-multiv}

In many studies, there may be *several* categorical responses observed along with one or more explanatory variables. In a clinical trial, for example, the efficacy of a drug might be the primary response, but the occurrence of side-effects might give rise to additional response variables of substantive interest. Or, in a study of occupational health, the occurrence of two or more distinct symptoms might be treated as response variables.

If there are *no* explanatory variables, then the problem is simply to understand the joint distribution of the response categories, and the loglinear models and graphical displays described earlier are sufficient. Otherwise, in these cases we usually wish to understand how the various responses are affected by the explanatory variables. Moreover, it may also be important to understand how the association between the categorical responses depends on the explanatory variables. That is, we would like to study how *both* the marginal distributions of the responses, and their joint distribution depends on the predictors. In the occupational health example, the goal might be to understand both how the prevalence of several symptoms varies with one or more predictors, and how the association (loosely, "correlation") among those symptoms varies with those predictors.

Although the general loglinear model is often used in these situations, there are special reparameterizations that may be used to separate the *marginal* dependence of each response on the explanatory variables from the relationship of the *association* among the responses on the explanatory variables.

Let us say that categorical responses, Y_1, $Y_2, \ldots$ have been observed, together with possible explanatory variables, X_1, $X_2, \ldots$, and let $\pi_{ij\ldots}$ be the joint probability of all the responses and explanatory variables; we also use $\boldsymbol{x}$ to refer to the values of X_1, $X_2, \ldots$.

Note that the minimal model of independence of all responses from each other and from the explanatory variables is the loglinear model $[Y_1][Y_2]\cdots[X_1X_2\cdots]$ (i.e., all associations among the X_i must be included). A no-effect model, in which the responses do not depend on the explanatory variables, but may be associated among themselves is $[Y_1Y_2\cdots][X_1X_2\cdots]$. However, these models do not separate the individual (marginal) effects of $X_1, X_2\ldots$ on each Y_i from their associative effects on the joint relationships among the Y_i.

There are three useful general approaches which *do* separate these effects:

1. Model the marginal dependence of each response, Y_i separately on X_1, $X_2, \ldots$, and, in addition, model the interdependence among the responses, $Y_1, Y_2, \ldots$.[5]

2. Model the joint dependence of all responses on X_1, $X_2, \ldots$, but parameterized so that marginal and associative effects are delineated.

3. Construct simultaneous models, estimated together, for the marginal and joint dependence of the responses on the explanatory variables.

The first approach is the simplest, an informative starting place, and is satisfactory in the (often unlikely) case that the responses are not associated, or if the associations among responses do not vary much over the explanatory variables (i.e., no terms like $[Y_1Y_2X_j]$ are required). In the clinical trial example, we would construct separate loglinear or logit models for efficacy of the drug, and for occurrence of side-effects, and supplement these analyses with mosaic or other displays showing the relations between efficacy and side-effects and a model for their joint association. If those who improve with the drug also show more serious side effects, the worth of the treatment would be questioned. A limitation of this method is that it does not provide an overall model comprising these effects.

In the second approach, the joint probabilities, $\pi_{ij\ldots}$, are recast to give separate information regarding the dependence of the univariate marginal probabilities $\pi_{i\bullet}, \pi_{\bullet j}, \ldots$, on the explanatory variables and the dependence of the intra-response associations on the explanatory variables. The VGAM (Yee, 2015) package provides several versions of this approach with the function `vglm()` (for *vector generalized linear model*).

The third approach, developed, for example, by Lang and Agresti (1994), is the most general, and provides a scheme to represent a model $\mathcal{J}(\bullet)$ for the joint distributions of the X, Y variables together with a model $\mathcal{M}(\bullet)$ for their first-order marginal distributions. The joint models are typically loglinear models, ranging from the mutual independence model, $\mathcal{J}(I) = [Y_1][Y_2][\cdots][X_1][X_2][\cdots]$ to the saturated model, $\mathcal{J}(S) = [Y_1Y_2\cdots X_1X_2\cdots]$, while the marginal models are logit models for the response variables. The combined model, denoted $\mathcal{J}(\bullet) \cap \mathcal{M}(\bullet)$, is estimated simultaneously by maximum likelihood. This approach is implemented in R in the hmmm (Roberto *et al.*, 2014) package (hierarchical multinomial marginal models). However, model specification in this implementation is complicated, and it will not be considered further here.

10.4.1 Bivariate, binary response models

We focus here on two related models reflecting the second approach, as discussed by McCullagh and Nelder (1989, Section 6.5). We consider here only the case of two binary responses, though the general approach can be applied to $R > 2$ responses $Y_1, Y_2, \ldots, Y_R$, and these may be polytomous or ordinal.

Let $\boldsymbol{x}$ refer to the values of all the explanatory variables and let $\pi_{ij}(\boldsymbol{x})$ be the joint probabilities in cell $Y_1 = i, Y_2 = j$. The essential idea of the ***bivariate logistic model*** arises from a linear

[5] For quantitative responses, this is roughly analogous to fitting univariate response models for each Y_i, followed by something like a principal component analysis of the relationships among the Y_i. But in this case, the multivariate linear model, $\boldsymbol{Y} = \boldsymbol{XB} + \boldsymbol{E}$ provides a general solution.

transformation of the cell probabilities $\boldsymbol{\pi}$ to interpretable functions of the marginal probabilities (logits) and their association (odds ratio), a mapping of $\boldsymbol{\pi} \rightarrow \boldsymbol{\eta}$,

{eq:blogits}

$$\begin{aligned} \eta_1 &= \text{logit}(\pi_{1\bullet}) \\ \eta_2 &= \text{logit}(\pi_{\bullet 1}) \\ \eta_{12} &= \log\left(\frac{\pi_{11}\,\pi_{22}}{\pi_{12}\,\pi_{21}}\right) \end{aligned} \tag{10.10}$$

The predictors in $\boldsymbol{x}$ are then taken into account by considering models that relate $\boldsymbol{\pi}$ to $\boldsymbol{x}$ through $\boldsymbol{\eta}$,

{eq:blogits2}

$$\begin{aligned} \eta_1 &= \boldsymbol{x}_1^{\mathsf{T}}\boldsymbol{\beta}_1 \\ \eta_2 &= \boldsymbol{x}_2^{\mathsf{T}}\boldsymbol{\beta}_2 \\ \eta_{12} &= \boldsymbol{x}_{12}^{\mathsf{T}}\boldsymbol{\beta}_{12} \end{aligned} \tag{10.11}$$

where $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and $\boldsymbol{x}_{12}$ are subsets of the predictors in $\boldsymbol{x}$ for each sub-model, and $\boldsymbol{\beta}_1$, $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_{12}$ are the corresponding parameters to be estimated.

McCullagh and Nelder (1989) arrive at this joint bivariate model in two steps. First, transform the cell probabilities $\boldsymbol{\pi}$ to a vector of probabilities $\boldsymbol{\gamma}$ which also includes the univariate margins, given by

{eq:gamma1}

$$\boldsymbol{\gamma} = \boldsymbol{L}\boldsymbol{\pi} \tag{10.12}$$

where $\boldsymbol{L}$ is a matrix of 0s and 1s of the form of a factorial design matrix. In the 2×2 case,

{eq:gamma2}

$$\boldsymbol{\gamma} = \begin{pmatrix} \pi_{1\bullet} \\ \pi_{2\bullet} \\ \pi_{\bullet 1} \\ \pi_{\bullet 2} \\ \pi_{11} \\ \pi_{12} \\ \pi_{21} \\ \pi_{22} \end{pmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} \pi_{11} \\ \pi_{12} \\ \pi_{21} \\ \pi_{22} \end{pmatrix} \quad . \tag{10.13}$$

There are of course only three linearly independent probabilities, because $\sum\sum \pi_{ij} = 1$. In the second step, the bivariate logistic model is formulated in terms of factorial contrasts on the elements of $\boldsymbol{\gamma}$ which express separate models for the two logits and the log odds. The model is expressed as

{eq:eta1}

$$\boldsymbol{\eta} = \boldsymbol{C}\log\boldsymbol{\gamma} = \boldsymbol{C}\log(\boldsymbol{L}\boldsymbol{\pi}) \ , \tag{10.14}$$

where $\boldsymbol{C}$ is a matrix of contrasts. In the 2×2 case, the usual contrasts may be defined by

{eq:eta2}

$$\boldsymbol{\eta} = \begin{pmatrix} \eta_1 \\ \eta_2 \\ \eta_{12} \end{pmatrix} = \begin{pmatrix} \text{logit}\ \pi_{1\bullet} \\ \text{logit}\ \pi_{\bullet 1} \\ \log(\theta_{12}) \end{pmatrix} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 \end{bmatrix} \begin{pmatrix} \log\ \pi_{1\bullet} \\ \log\ \pi_{2\bullet} \\ \log\ \pi_{\bullet 1} \\ \log\ \pi_{\bullet 2} \\ \log\ \pi_{11} \\ \log\ \pi_{12} \\ \log\ \pi_{21} \\ \log\ \pi_{22} \end{pmatrix} \tag{10.15}$$

Thus, we are modeling the marginal log odds of each response, together with the log odds ratio $\log(\theta_{12})$ simultaneously.

Specific models are then formulated for the dependence of $\eta_1(\boldsymbol{x}), \eta_2(\boldsymbol{x})$ and $\eta_{12}(\boldsymbol{x})$ on some

or all of the explanatory variables. For example, with one quantitative explanatory variable, x, the model

$$\begin{pmatrix} \eta_1 \\ \eta_2 \\ \eta_{12} \end{pmatrix} = \begin{pmatrix} \alpha_1 + \beta_1 x \\ \alpha_2 + \beta_2 x \\ \log(\theta) \end{pmatrix} \tag{10.16}$$

{eq:bilogit1}

asserts that the log odds of each response changes linearly with x, while the odds ratio between the responses remains constant. In the general form given by McCullagh and Nelder (1989) the submodels in Eqn. (10.16) may each depend on the explanatory variables in different ways. For example, the logits could both depend quadratically on x, while an intercept-only model could be posited for the log odds ratio.

The second model is the ***bivariate loglinear model***, the special case obtained by taking $\boldsymbol{L} = \boldsymbol{I}$ in Eqn. (10.12) and Eqn. (10.14) so that $\boldsymbol{\gamma} = \boldsymbol{\pi}$. Then a loglinear model of the form

$$\boldsymbol{\eta}(\boldsymbol{x}) = \boldsymbol{C} \log \boldsymbol{\pi}$$

expresses contrasts among log probabilities as linear functions of the explanatory variables. For the 2×2 case, we take the contrasts $\boldsymbol{C}$ as shown below

$$\boldsymbol{\eta} = \begin{pmatrix} l_1 \\ l_2 \\ \eta_{12} \end{pmatrix} = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{pmatrix} \log\, \pi_{11} \\ \log\, \pi_{12} \\ \log\, \pi_{21} \\ \log\, \pi_{22} \end{pmatrix} \tag{10.17}$$

{eq:eta3}

and models for the dependence of $l_1(\boldsymbol{x})$, $l_2(\boldsymbol{x})$ and $\eta_{12}(\boldsymbol{x})$ are expressed in the same way as in Eqn. (10.16). The estimates of the odds ratio, η_{12} are the same under both models. The marginal functions are parameterized differently, however, but lead to similar predicted probabilities.

In R, bivariate logistic models of the form Eqn. (10.10) and Eqn. (10.11) can be fit using `vglm()` with the `binom2.or()` family in the VGAM package.[6] The fitting and graphing of these models is illustrated in the next example.

{ex:coalminers}

EXAMPLE 10.9: Breathlessness and wheeze in coal miners

In Example 4.12 we examined the association between the occurrence of two pulmonary conditions, breathlessness and wheeze, among coal miners classified by age (Ashford and Sowden, 1970). Figure 4.7 showed fourfold displays focused on the odds ratio for the co-occurrence of these symptoms, and Figure 4.8 plotted these odds ratios against age directly. Here, we consider models which examine the changes in prevalence of the two symptoms over age, together with the changes in their association.

Plotting bivariate response data

As a starting point and overview of what is necessary for bivariate response models, we calculate the empirical log odds for breathlessness and for wheeze, and the log odds ratio for their association in each 2×2 table. The log odds ratios are the same values plotted in Figure 4.8 (but the youngest age group was not included in the earlier analysis).

The *CoalMiners* data is $2 \times 2 \times 9$ table. For convenience in this analysis (and for use with VGAM) we convert it to a 4×9 data frame, and relabel the columns to use the combinations of `("B", "b")` and `("W", "w")` to represent the conditions of breathlessness and wheeze, where the upper case letter indicates presence of the condition. A variable `age` is also created, using the midpoints of the age categories.

[6] This package also provides for bivariate and trivariate loglinear models with `loglinb2()` and `loglinb2`.

```
> data("CoalMiners", package = "vcd")
> coalminers <- data.frame(t(matrix(aperm(CoalMiners, c(2, 1, 3)),
+                                   4, 9)))
> colnames(coalminers) <- c("BW", "Bw", "bW", "bw")
> coalminers$age <- c(22, 27, 32, 37, 42, 47, 52, 57, 62)
> coalminers

   BW  Bw  bW   bw age
1   9   7  95 1841  22
2  23   9 105 1654  27
3  54  19 177 1863  32
4 121  48 257 2357  37
5 169  54 273 1778  42
6 269  88 324 1712  47
7 404 117 245 1324  52
8 406 152 225  967  57
9 372 106 132  526  62
```

With the data in this form, a simple function `blogits()` in vcdExtra calculates the logits and log odds ratios corresponding to Eqn. (10.10). The `add` argument accommodates cases where there are very small, or 0 frequencies in some cells, and it is common to add a small constant, such as 0.5 to each cell in calculating ***empirical logits***. This function is used to calculate the empirical logits and log odds as follows:

```
> logitsCM <- vcdExtra::blogits(coalminers[, 1 : 4], add = 0.5)
> colnames(logitsCM)[1:2] <- c("logitB", "logitW")
> logitsCM

        logitB   logitW  logOR
 [1,] -4.73568 -2.86844 3.1956
 [2,] -3.97656 -2.55717 3.6583
 [3,] -3.31713 -2.09388 3.3790
 [4,] -2.73322 -1.84818 3.1327
 [5,] -2.21492 -1.42014 3.0069
 [6,] -1.73870 -1.10922 2.7770
 [7,] -1.10116 -0.79681 2.9217
 [8,] -0.75808 -0.57219 2.4368
 [9,] -0.31902 -0.22591 2.6318
```

We plot these as shown below, using `matplot()`, which is convenient for plotting multiple columns against a given horizontal variable, `age` here.[7] For ease of interpretation of the log odds, we also use right vertical axis showing the equivalent probabilities for breathlessness and wheeze.

```
> col <- c("blue", "red", "black")
> pch <- c(15, 17, 16)
> age <- coalminers$age
>
> op <- par(mar = c(4, 4, 1, 4)+.2)
> matplot(age, logitsCM, type = "p",
+   col = col, pch = pch, cex = 1.2, cex.lab = 1.25,
+   xlab = "Age", ylab = "Log Odds or Odds Ratio")
> abline(lm(logitsCM[,1] ~ age), col = col[1], lwd = 2)
> abline(lm(logitsCM[,2] ~ age), col = col[2], lwd = 2)
> abline(lm(logitsCM[,3] ~ age), col = col[3], lwd = 2)
>
> # right probability axis
> probs <- c(.01, .05, .10, .25, .5)
> axis(4, at = qlogis(probs), labels = probs)
```

[7] It is actually a small graphical misdemeanor to plot logits and odds ratios on the same vertical axis because they are not strictly commensurable. We plead guilty with the explanation that this graph shows what we want to see here and does not distort the data.

```
> mtext("Probability", side = 4, cex = 1.2, at = -2, line = 2.5)
> # curve labels
> text(age[2], logitsCM[2, 1] + .5, "Breathlessness",
+      col = col[1], pos = NULL, cex = 1.2)
> text(age[2], logitsCM[2, 2] + .5, "Wheeze",
+      col = col[2], pos = NULL, cex = 1.2)
> text(age[2], logitsCM[2, 3] - .5, "log OR\n(B|W)/(B|w)",
+      col = col[3], pos = 1, cex = 1.2)
> par(op)
```

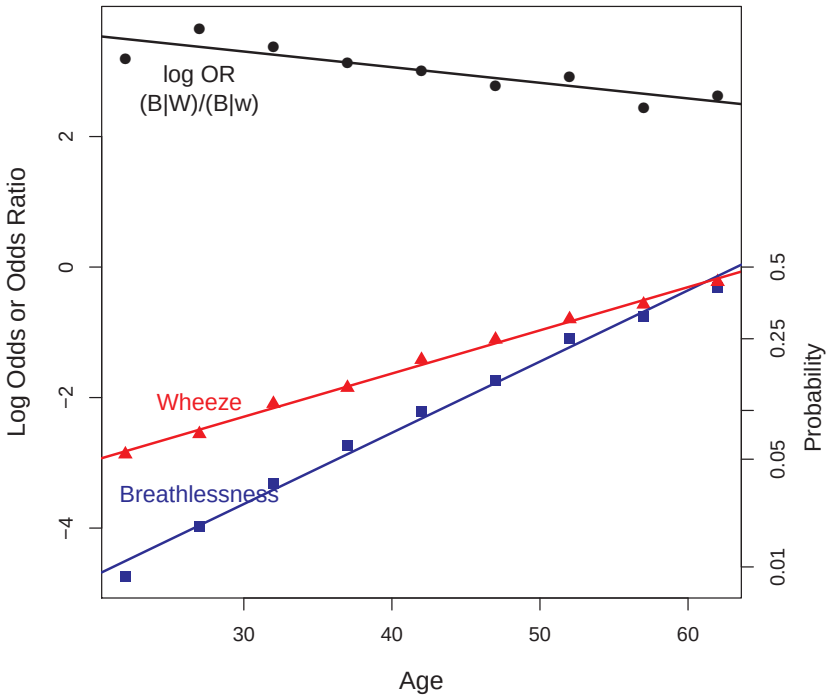


Figure 10.16: Empirical logits and log odds ratio for breathlessness and wheeze in the CoalMiners data. The lines show separate linear regressions for each function. The right vertical axis shows equivalent probabilities for the logits. fig:enn-blogits

In Figure 10.16 we see that both symptoms, while quite rare among young miners, increase steadily with age (or years working in the mine). By age 60, the probability is nearly 0.5 of having either condition. There is a hint of curvilinearity, particularly in the logit for breathlessness. The decline in the odds ratio with age may reflect selection, as miners who had retired for health or other reasons were excluded from the study.

Fitting `glm` models

Next, we illustrate what can easily be achieved using the standard `glm()` approach for loglinear models and why the bivariate models we described are more useful in this situation. `glm()` requires a data frame as input, so first reshape *`CoalMiners`* to a frequency data frame. For convenience, we simplify the variable names to `B` and `W`.

```
> CM <- as.data.frame(CoalMiners)
> colnames(CM)[1:2] <- c("B", "W")
> head(CM)

    B   W   Age Freq
1   B   W 20-24    9
2 NoB   W 20-24   95
3   B NoW 20-24    7
4 NoB NoW 20-24 1841
5   B   W 25-29   23
6 NoB   W 25-29  105
```

As a point of comparison, we fit the mutual independence model, [B][W][Age] and the baseline model for associated responses, [BW][Age] which asserts that the association between `B` and W is independent of `Age`.

```
> cm.glm0 <- glm(Freq ~ B + W + Age, data = CM, family = poisson)
> cm.glm1 <- glm(Freq ~ B * W + Age, data = CM, family = poisson)
> vcdExtra::LRstats(cm.glm0, cm.glm1)

Likelihood summary table:
         AIC  BIC LR Chisq Df Pr(>Chisq)
cm.glm0 7217 7234     6939 25     <2e-16 ***
cm.glm1 2981 3000     2702 24     <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The baseline model `cm.glm1` fits very badly. We can see the pattern of the residual association in a mosaic display for this model shown in Figure 10.17. The formula argument here specifies the order of the variables in the mosaic.

```
> vnames <- list(set_varnames = c(B = "Breathlessness", W = "Wheeze"))
> lnames <- list(B=c("B", "b"), W = c("W", "w"))
> mosaic(cm.glm1, ~ Age + B + W,
+        labeling_args = vnames, set_labels = lnames)
```

As structured here, it is easy to see the increase in the prevalence of breathlessness and wheeze with age and the changing pattern of their association with age.

From Figure 10.16 and Figure 10.17, it is apparent that both breathlessness and wheeze increase with age, so we can model this by adding terms [B Age][W Age] to the baseline model. This is the no-three-way interaction model, which could also be specified as `Freq ~ (B + W + Age)^2`.

```
> cm.glm2 <- glm(Freq ~ B * W + (B + W) * Age, data = CM, family = poisson)
> vcdExtra::LRstats(cm.glm1, cm.glm2)

Likelihood summary table:
         AIC  BIC LR Chisq Df Pr(>Chisq)
cm.glm1 2981 3000     2702 24     <2e-16 ***
cm.glm2  338  383       27  8      8e-04 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The improvement in fit is substantial, and all terms are highly significant, yet, the residual $G^2(8)$ indicates there is still lack of fit.

```
> library(car)
> Anova(cm.glm2)
```

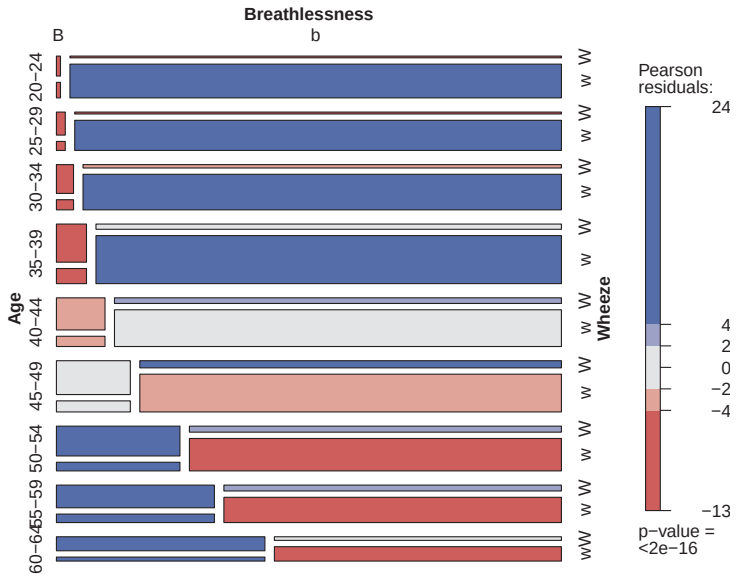


Figure 10.17: Mosaic display for the baseline model, [BW][Age], fit to the CoalMiners data fig:cm-mosaic1

```
Analysis of Deviance Table (Type II tests)

Response: Freq
      LR Chisq Df Pr(>Chisq)
B        11026  1     <2e-16 ***
W         7038  1     <2e-16 ***
Age        887  8     <2e-16 ***
B:W       3025  1     <2e-16 ***
B:Age     1130  8     <2e-16 ***
W:Age      333  8     <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

One way to improve the model using the `glm()` framework is to make use of `Age` as a quantitative variable and add a term to allow the odds ratio for the [BW] association to vary linearly with age. Here, we construct the variable `age` using the midpoints of the Age intervals.

```
> CM$age <- rep(seq(22, 62, 5), each = 4)
```

In the `glm()` approach, the odds ratio cannot be modeled directly, but we can use the following trick: For each 2×2 subtable, the odds ratio can be parameterized in terms of the frequency in any one cell, say, n_{11k}, given that the marginal total n_{++k} is included in the model. We do this by adding a new interaction variable, `ageOR` having the value of `age` for the $(1, 1, k)$ cells and 0 otherwise.

```
> CM$ageOR <- (CM$B == "B") * (CM$W == "W") * CM$age
> cm.glm3 <- update(cm.glm2, . ~ . + ageOR)
> vcdExtra::LRstats(cm.glm0, cm.glm1, cm.glm2, cm.glm3)

Likelihood summary table:
         AIC  BIC LR Chisq Df Pr(>Chisq)
cm.glm0 7217 7234     6939 25     <2e-16 ***
```

```
cm.glm1 2981 3000     2702 24     <2e-16 ***
cm.glm2  338  383       27  8     0.0008 ***
cm.glm3  320  366        7  7     0.4498
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The model `cm.glm3`, with one more parameter, now fits reasonably well, having residual $G^2(7) = 6.80$. The likelihood ratio test of model `cm.glm3` against `cm.glm2`, which assumes equal odds ratios over age, can be regarded as a test of the hypothesis of homogeneity of odds ratios, against the alternative that the [BW] association changes linearly with age. The `glm()` models fit in this example are summarized above. As usual, `anova()` can be used to compare competing nested models.

```
> anova(cm.glm2, cm.glm3, test = "Chisq")

Analysis of Deviance Table

Model 1: Freq ~ B * W + (B + W) * Age
Model 2: Freq ~ B + W + Age + ageOR + B:W + B:Age + W:Age
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         8       26.7
2         7        6.8  1     19.9  8.2e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This analysis, while useful, also shows the limitations of the `glm()` approach: (a) It doesn't easily allow us to represent and test the substantively interesting hypotheses regarding *how* the prevalence of the binary responses, `B` and `W` vary with `Age`, such as seen in Figure 10.16. (b) It doesn't represent the odds ratio for the [BW] association directly, but only through the coding trick we used here. Thus, it is difficult to interpret the coefficient for `ageOR` = -0.02613 in a substantively meaningful way, except that is shows that the odds ratio is decreasing.[8]

Fitting `vglm` models

The `vglm()` function in the VGAM package provides a very general implementation of these and other models for discrete multivariate responses. The family function, `binom2.or()` for binary logistic models allows some or all of the logits or odds ratio submodels to be constrained to be intercept-only (e.g., as in Eqn. (10.16)) and the two marginal distributions can be constrained to be equal.

Quantitative predictors (such as age, here), can be modeled linearly or nonlinearly, using `poly()` for a parametric fit, or smooth regression splines, as provided by the functions `ns()`, `bs()` and others in model formulas. In this illustration, we fit bivariate linear and quadratic models in age.

`vglm()` takes its input data in the wide form we called `coalminers` at the beginning of this example. We could use the 9-level factor, `Age` as we did with `glm()`, but we plan to use `age` as a numeric variable in all three submodels. The coefficients in these models will be more easily interpreted if we center age and express it as `agec` in units of five years, as shown below.

```
> coalminers <- transform(coalminers, agec = (age - 42) / 5)
> coalminers$Age <- dimnames(CoalMiners)[[3]]
> coalminers

   BW  Bw  bW   bw age agec   Age
1   9   7  95 1841  22   -4 20-24
```

[8] Actually, the interpretability of the coefficient for the log odds ratio can be enhanced here by centering age, and representing its units in steps of 5 years, as we do below.

```
2  23   9 105 1654  27   -3 25-29
3  54  19 177 1863  32   -2 30-34
4 121  48 257 2357  37   -1 35-39
5 169  54 273 1778  42    0 40-44
6 269  88 324 1712  47    1 45-49
7 404 117 245 1324  52    2 50-54
8 406 152 225  967  57    3 55-59
9 372 106 132  526  62    4 60-64
```

`vglm()` takes the 2×2 response frequencies as a 4-column matrix on the left hand side of the model formula. However, denoting the responses of failure and success by 0 and 1 respectively, it takes these in the order $y_{00}, y_{01}, y_{10}, y_{11}$. We specify the order below so that the logits are calculated for the occurrence of breathlessness or wheeze, rather than their absence.

```
> library(VGAM)
> #                          00  01  10  11
> cm.vglm1 <- vglm(cbind(bw, bW, Bw, BW) ~ agec,
+                  binom2.or(zero = NULL), data = coalminers)
> cm.vglm1

Call:
vglm(formula = cbind(bw, bW, Bw, BW) ~ agec, family = binom2.or(zero = NULL),
    data = coalminers)

Coefficients:
(Intercept):1 (Intercept):2 (Intercept):3        agec:1
     -2.26247      -1.48776       3.02191       0.51451
       agec:2        agec:3
      0.32545      -0.13136

Degrees of Freedom: 27 Total; 21 Residual
Residual deviance: 30.394
Log-likelihood: -100.53
```

In this call, the argument `zero = NULL` indicates that none of the linear predictors, $\eta_1, \eta_2, \eta_{12}$ are modeled as constants.[9]

At this writing, there is no `anova()` method for the "vgam" objects produced by `vglm()`, but we can test the residual deviance of the model (against the saturated model) as follows, showing that this model has an acceptable fit.

```
> (G2 <- deviance(cm.vglm1))

[1] 30.394

> # test residual deviance
> 1-pchisq(deviance(cm.vglm1), cm.vglm1@df.residual)

[1] 0.084355
```

The estimated coefficients in this model are usefully shown as below, using the argument `matrix=TRUE` in `coef()`. Using `exp()` on the result gives values of odds that can be easily interpreted:

```
> coef(cm.vglm1, matrix = TRUE)

            logit(mu1) logit(mu2) log(oratio)
(Intercept)   -2.26247   -1.48776     3.02191
agec           0.51451    0.32545    -0.13136

> exp(coef(cm.vglm1, matrix = TRUE))
```

[9] The default, `zero=3` gives the model shown in Eqn. (10.16), with the odds ratio constant.

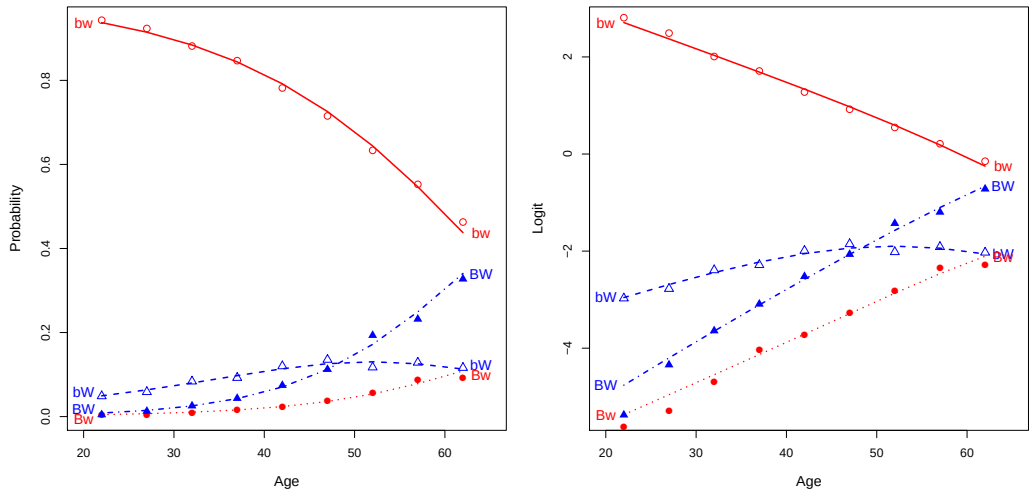


Figure 10.18: Observed and fitted values for the combinations of breathlessness and wheeze in the binary logistic regression model `cm.vglm1`. Left: probabilities; right: on the log odds scale. {fig:cm-vglm1}

```
             logit(mu1) logit(mu2) log(oratio)
(Intercept)     0.10409    0.22588     20.5304
agec            1.67282    1.38465      0.8769
```

Thus, the odds of a miner showing breathlessness are multiplied by 1.67, a 67% increase, for each 5 years increase in age; similarly, the odds of wheeze are multiplied by 1.38, a 38% increase. The odds ratio for the association between the two symptoms are multiplied by 0.88, a 12% decrease over each 5 year interval.

The VGAM package has no special plot methods for "vglm" objects, but it is not hard to construct these using the methods we showed earlier in this example. First, we can obtain the fitted probabilities for the 4 response combinations using `fitted()` and the corresponding observed probabilities using `depvar()`.

```
> age <- coalminers$age
> P <- fitted(cm.vglm1)
> colnames(P) <- c("bw", "bW", "Bw", "BW")
> head(P)

       bw       bW        Bw        BW
1 0.93747 0.049409 0.0046356 0.0084831
2 0.91461 0.063636 0.0069757 0.0147776
3 0.88411 0.080029 0.0104965 0.0253679
4 0.84394 0.097484 0.0158138 0.0427671
5 0.79188 0.113839 0.0238598 0.0704196
6 0.72578 0.125910 0.0359684 0.1123366

> Y <- depvar(cm.vglm1)
```

In the left panel of Figure 10.18, we plot the fitted probabilities in the matrix `P` using `matplot()` and the observed probabilities in `Y` using `matpoints()`.

```
> col <- c("red", "blue", "red", "blue")
> pch <- c(1, 2, 16, 17)
>
> op <- par(mar = c(5, 4, 1, 1) + .1)
```

```
> matplot(age, P, type = "l",
+   col = col,
+   lwd = 2, cex = 1.2, cex.lab = 1.2,
+   xlab = "Age", ylab = "Probability",
+   xlim = c(20,65))
> matpoints(age, Y, pch = pch, cex = 1.2, col = col)
> # legend
> text(64, P[9,]+ c(0,.01, -.01, 0), labels = colnames(P), col = col, cex = 1.2)
> text(20, P[1,]+ c(0,.01, -.01, .01), labels = colnames(P), col = col, cex = 1.2)
> par(op)
```

The right panel of Figure 10.18 shows these on the log odds scale, produced using the same code as above, applied to the probabilities transformed using `qlogis()`, the quantile function for the logistic distribution.

```
> lP <- qlogis(P)
> lY <- qlogis(Y)
```

In Figure 10.16 we plotted the empirical logits and log odds using the function `blogits()` to transform frequencies to these values. An essentially identical plot can be produced by transforming the fitted and observed probabilities, as calculated below.

```
> # blogits, but for B and W
> logitsP <- blogits(P[, 4 : 1])
> logitsY <- blogits(Y[, 4 : 1])
```

To test for nonlinearity in the prevalence of the symptoms or their odds ratio with age, we can fit a similar model using `poly()` or a smoothing spline, such as `ns()`. We illustrate this here using a bivariate model allowing quadratic effects of age on all three components.

```
> cm.vglm2 <- vglm(cbind(bw, bW, Bw, BW) ~ poly(agec, 2),
+                  binom2.or(zero = NULL), data = coalminers)
```

This model has a residual $G^2 = 16.963$ with 18 df. Compared to the linear model `cm.vglm1`, this represents a significant improvement in goodness of fit.

```
> (LR <- deviance(cm.vglm1) - deviance(cm.vglm2))

[1] 13.43

> 1 - pchisq(LR, cm.vglm1@df.residual - cm.vglm2@df.residual)

[1] 0.0037925
```

A plot of the fitted logits and log odds ratios under this model is shown in Figure 10.19. You can interpret this plot as showing that the statistical evidence for the quadratic model indicates some slight tendency for the prevalence of breathlessness and wheeze levels off slightly with age, particularly the former.

△

10.4.2 More complex models

When there is more than one explanatory variable and several responses, the methods described above using `glm()` and `vglm()` still apply. However, it is useful to begin with a more thorough visual examination of the relations within and between these sets. Some useful graphical displays include:

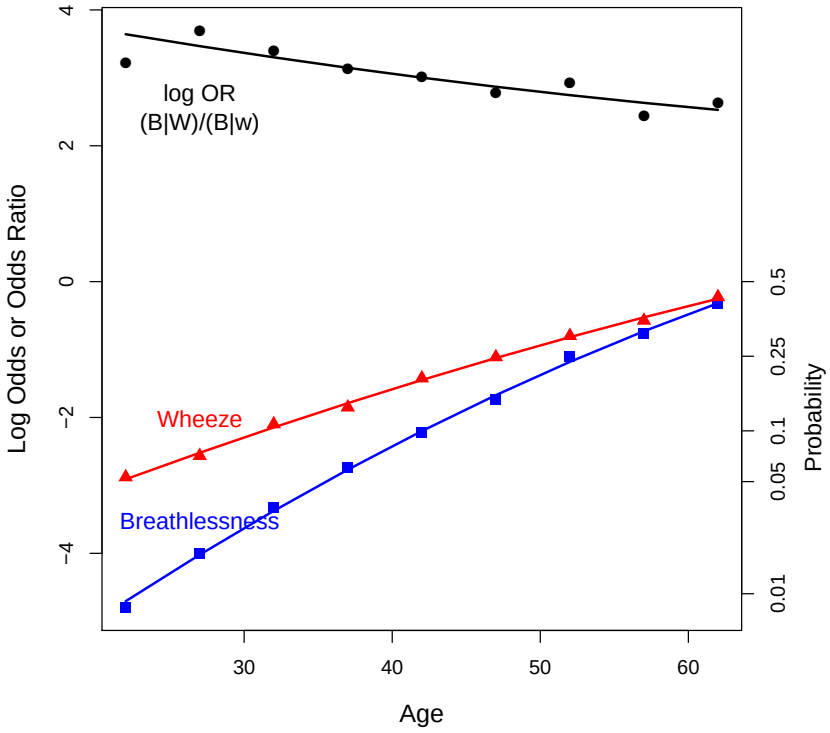


Figure 10.19: Observed (points) and fitted (lines) logits and log odds ratios for the quadratic binary logistic regression model `cm.vglm2`.

{fig:cm-vglm2-blogit}

- mosaic displays showing the marginal relations among the response variables and of the explanatory variables, each collapsed over the other set;
- conditional mosaics or fourfold displays of the associations among the responses, stratified by one or more of the explanatory variables;
- plots of empirical logits and log odds ratios, as in Figure 10.16 or model-based plots, such as Figure 10.19, showing a model-smoothed summary.

These displays can, and should, inform our search for an adequate descriptive or explanatory model. Some of these ideas are illustrated in the following example.

{ex:toxaemia}

EXAMPLE 10.10: Toxaemic symptoms in pregnancy

Brown *et al.* (1983) gave the data used here on two signs of *toxaemia*, an abnormal condition during pregnancy characterized by high blood pressure (hypertension) and high levels of protein in the urine. If untreated, both the mother and baby are at risk of complications or death. The data frame *Toxaemia* in vcdExtra represents 13,384 expectant mothers in Bradford, England in their first pregnancy, who were also classified according to social class and the number of cigarettes smoked per day.

There are thus two response variables, and two explanatory variables in this data set in frequency form. For convenience, we also convert it to a $2 \times 2 \times 5 \times 3$ table.

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

```
'data.frame': 60 obs. of  5 variables:
 $ class: Factor w/ 5 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ smoke: Factor w/ 3 levels "0","1-19","20+": 1 1 1 1 2 2 2 2 3 3 ...
 $ hyper: Factor w/ 2 levels "Low","High": 2 2 1 1 2 2 1 1 2 2 ...
 $ urea : Factor w/ 2 levels "Low","High": 2 1 2 1 2 1 2 1 2 1 ...
 $ Freq : int  28 82 21 286 5 24 5 71 1 3 ...

> tox.tab <- xtabs(Freq ~ class + smoke + hyper + urea, Toxaemia)
> ftable(tox.tab, row.vars = 1)

      smoke    0                1-19                20+
      hyper  Low      High       Low      High       Low      High
      urea   Low High  Low High  Low High  Low High  Low High  Low High
class
1            286   21   82   28   71    5   24    5   13    0    3    1
2            785   34  266   50  284   17   92   13   34    3   15    0
3           3160  164 1101  278 2300  142  492  120  383   32   92   16
4            656   52  213   63  649   46  129   35  163   12   40    7
5            245   23   78   20  321   34   74   22   65    4   14    7
```

The questions of main interest are how the occurrence of each symptom varies with social class and smoking, and how the association between these symptoms varies. It is useful, however, to examine first the marginal relationship between the two responses, and between the two predictors. The calls to `mosaic()` below produce the two panels in Figure 10.20.

```
> mosaic(~ smoke + class, data = tox.tab, shade = TRUE,
+        main = "Predictors", legend = FALSE)
> mosaic(~ hyper + urea, data = tox.tab, shade = TRUE,
+        main = "Responses", legend = FALSE)
```

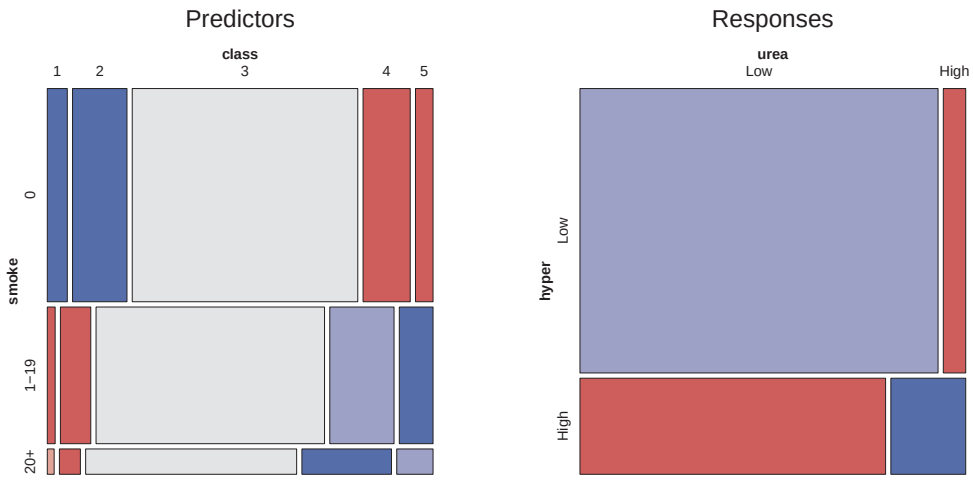


Figure 10.20: Mosaic displays for Toxaemia data: Predictor and response associations fig:tox-mosaic1

We see in Figure 10.20 that the majority of the mothers are in the third social class, and that smoking is negatively related to social class, with the highest levels of smoking in classes 4 and 5. (Social class 1 is the highest in status here.) More than 50% are non-smokers. Within the responses, the great majority of women exhibit neither symptom, but showing one symptom makes it much more likely to show the other. Marginally, hypertension is somewhat more prevalent than protein urea.

We next examine how the association between responses varies with social class and with smok-

ing. Figure 10.21 shows a collection of conditional mosaic plots using `cotabplot()` of the association between hypertension and urea, for each level of smoking, collapsed over social class.

```
> cotabplot(~ hyper + urea | smoke, tox.tab, shade = TRUE,
+           legend = FALSE, layout = c(1, 3))
```

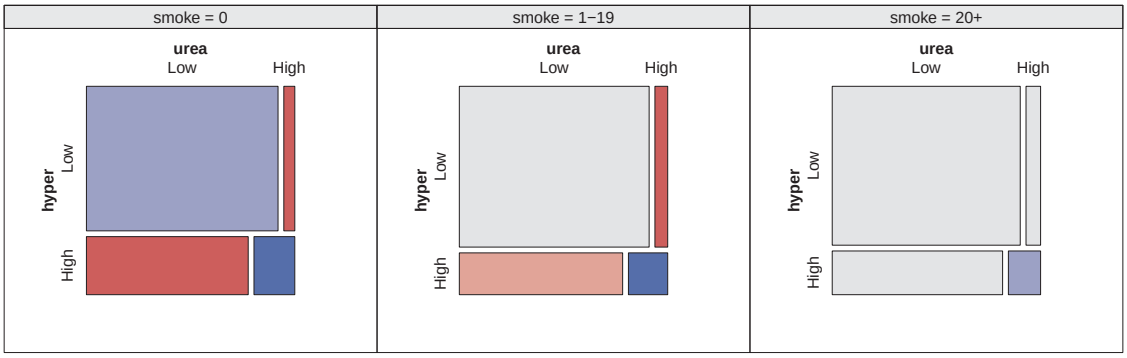


Figure 10.21: Toxaemia data: Response association conditioned on smoking level fig:tox-mosaic2

Figure 10.22 is similar, but stratified by social class. The marginal frequencies of the conditioning variable is not represented in these plots. (For example, as can be seen in Figure 10.20, the greatest number of women are in class 3.)

```
> cotabplot(~ hyper + urea | class, tox.tab, shade = TRUE,
+           legend = FALSE, layout = c(1, 5))
```

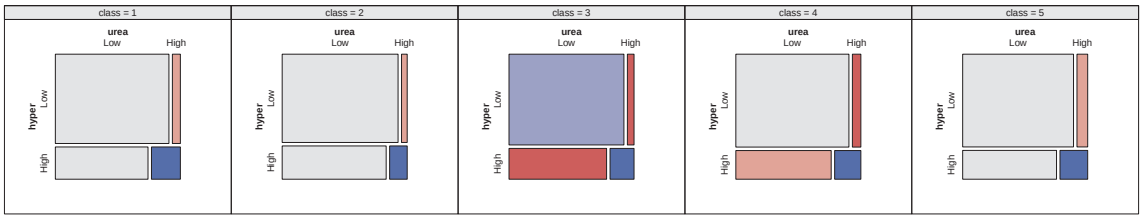


Figure 10.22: Toxaemia data: Response association conditioned on social class fig:tox-mosaic3

Ignoring social class, the association between hypertension and protein urea decreases with smoking. Ignoring smoking, the association is greatest in social class 3. However, these displays don't show directly how the two symptoms are associated in the combinations of social class and smoking. The fourfold display in Figure 10.23, does that.

```
> fourfold(aperm(tox.tab), fontsize = 16)
```

It can be seen in Figure 10.23 that the odds ratio appears to increase with both smoking and social class number and these two symptoms are positively associated in nearly all cases. In only two cases the odds ratio is not significantly different from 1: mothers in classes 1 and 2, who smoke more than 20 cigarettes a day, but the frequency in this cell is quite small.

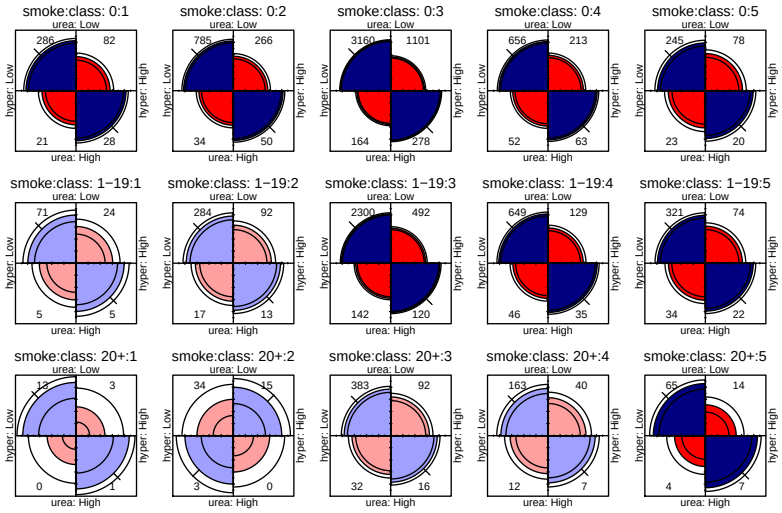


Figure 10.23: Fourfold display for Toxaemia data. Smoking levels vary in the rows and social class in the columns. {fig:tox-fourfold}

```
> margin.table(tox.tab, 2 : 1)

      class
smoke     1    2    3    4    5
  0     417 1135 4703  984  366
  1-19  105  406 3054  859  451
  20+    17   52  523  222   90
```

From these plots, it is useful to examine the association between hypertension and urea more directly, by calculating and plotting the odds ratios. For a $2 \times 2 \times K \times L \times \cdots$ table, the function `loddsratio()` in vcd calculates these for each 2×2 subtable, and returns an array of dimension $K \times L \times \cdots$, together with similar array of standard errors.

```
> (LOR <- loddsratio(urea ~ hyper | smoke + class, data = tox.tab))

log odds ratios for urea and hyper by smoke, class

      class
smoke       1        2       3       4      5
  0    1.5268  1.46196 1.58056 1.31351 1.0036
  1-19 1.0710  0.86401 1.37370 1.34260 1.0348
  20+  2.4485 -1.14579 0.74425 0.88469 2.0187
```

The `plot()` method for the resulting "logoddsratio" object treats the conditioning variables in the formula argument as strata, and plots the log odds ratios for the first such variable on the horizontal axis with curves for the subsequent strata variables. The lines below produce Figure 10.24.

```
> plot(t(LOR), confidence = FALSE, legend_pos = "bottomright",
+      xlab = "Social class of mother")
```

The association between the response symptoms, shown in Figure 10.24 is clearer, once we take the variation in sample sizes into account. Except for the heavy smokers, particularly in social classes 1 and 2, the log odds ratio appears to range only between 1–1.5, meaning that, given one symptom, the odds of also having the other range between $\exp(1) = 2.72$ and $\exp(1.5) = 4.48$.

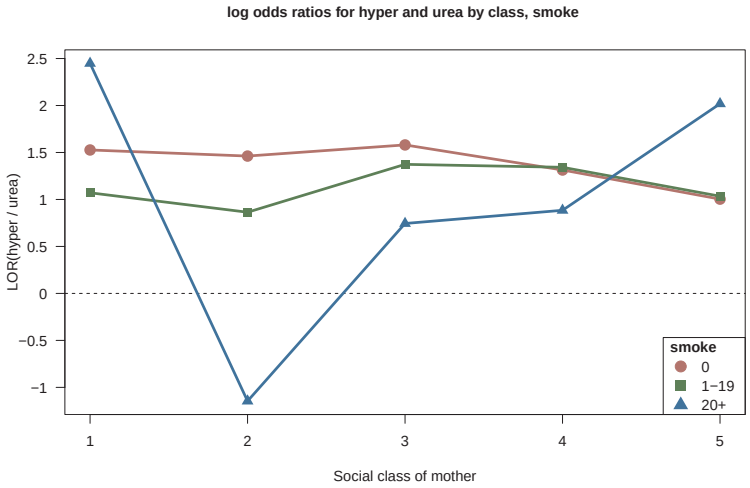


Figure 10.24: Log odds ratios for protein urea given hypertension, by social class and level of maternal smoking. fig:tox-LOR

This initial overview of the data is completed by calculating and plotting the log odds for each symptom within each class-smoke population. This could be done in the same way as in Example 10.9 (except that there are now two explanatory factors). The steps used there were: (a) Reshape the $2 \times 2 \times K \cdots$ table to a matrix with four columns corresponding to the binary response combinations. (b) Calculate the logits (and log odds ratio) using `blogits()`.

Here, it is more useful to make separate plots for each of the logits, and we illustrate a more general approach that applies to two or more binary responses, with two or more predictor variables. The essential idea is to fit a separate logit model for each response separately, using the *highest-order interaction* of all predictors (the saturated model). The fitted logits in these models then match those in the data.

```
> tox.hyper <- glm(hyper == "High" ~ class * smoke, weights = Freq,
+                  data = Toxaemia, family = binomial)
> tox.urea <- glm(urea == "High" ~ class * smoke, weights = Freq,
+                 data = Toxaemia, family = binomial)
```

It is then simple to plot these results using the effects (Fox *et al.*, 2015) package as shown in Figure 10.25. Each plot shows the logit for the response measure against class, with separate curves for the levels of smoking.[10]

```
> library(effects)
> plot(allEffects(tox.hyper),
+   ylab = "Probability (hypertension)",
+   xlab = "Social class of mother",
+   main = "Hypertension: class*smoke effect plot",
+   colors = c("blue", "black", "red"),
+   lwd=3,  multiline = TRUE,
+   key.args = list(x = 0.05, y = 0.2, cex = 1.2, columns = 1)
+   )
>
> plot(allEffects(tox.urea),
```

[10] As is usual for effect plots of binary response `glm()` models, the vertical axis is plotted on the scale of log odds, but labeled in terms of probabilities.

```
+    ylab = "Probability (Urea)",
+    xlab = "Social class of mother",
+    main = "Urea: class*smoke effect plot",
+    colors = c("blue", "black", "red"),
+    lwd=3,  multiline = TRUE,
+    key.args = list(x = 0.65, y = 0.2, cex = 1.2, columns = 1)
+    )
```

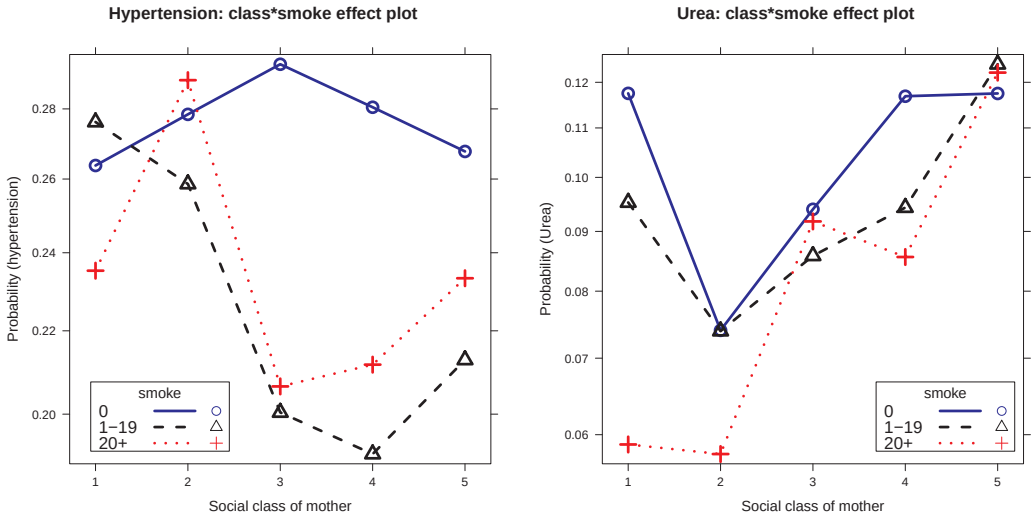


Figure 10.25: Effect plots for hypertension and urea, by social class of mother and smoking. fig:tox-effplots

From Figure 10.25, it can be seen that the prevalence of these symptoms has a possibly complex relation to social class and smoking. However, the mosaic for these predictors in Figure 10.20 has shown us that several of the class-smoking categories are quite small (particularly heavy smokers in Classes 1 and 2) so the response effects for these classes will be poorly estimated. Taking this into account, we suspect that protein urea varies with social class, but not with smoking, while the prevalence of hypertension may truly vary with neither, just one, or both of these predictors.

Fitting models

The plots shown so far in this example are all essentially *data-based*, in that they use the observed frequencies or transformations of them and don't allow for a simpler view, based on a reasonable model. That is, abbreviating the table variables by their initial letters, the plots in Figure 10.24 and Figure 10.25 are plots of the saturated model, [CSHU] that fits perfectly, but with the data transformed for each 2×2 subtable to the log odds ratio and the two log odds for `hyper` and `urea`.

The bivariate logistic model fit by `vglm()` still applies when there are two or more predictors; however, like other multivariate response models, it doesn't easily allow the logits to depend on *different* predictor terms. To illustrate this, we first transform the *Toxaemia* to a 15×4 data frame in the form required by `vglm()`.

```
> tox.tab <- xtabs(Freq~class + smoke + hyper + urea, Toxaemia)
> toxaemia <- t(matrix(aperm(tox.tab), 4, 15))
> colnames(toxaemia) <- c("hu", "hU", "Hu", "HU")
> rowlabs <- expand.grid(smoke = c("0", "1-19", "20+"), class = factor(1:5))
> toxaemia <- cbind(toxaemia, rowlabs)
```

```
> head(toxaemia)

   hu hU  Hu HU smoke class
1 286 21  82 28     0     1
2  71  5  24  5  1-19     1
3  13  0   3  1   20+     1
4 785 34 266 50     0     2
5 284 17  92 13  1-19     2
6  34  3  15  0   20+     2
```

In the model specification for `vglm()`, the `zero` argument in `binom.or()` allows any one or more of the two log odds and log odds ratio to be fit as a constant (intercept-only) in Eqn. (10.11). However, in that equation, the predictors $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_{12}$, must be the *same* in all three submodels. For example, the model `tox.vglm1` below uses main effects of `class` and `smoke` in both models for the logits, and `zero=3` for a constant log odds ratio.

```
> tox.vglm1 <- vglm(cbind(hu, hU, Hu, Hu) ~ class + smoke,
+                   binom2.or(zero = 3), data = toxaemia)
> coef(tox.vglm1, matrix=TRUE)

             logit(mu1) logit(mu2) log(oratio)
(Intercept) -0.50853648 -1.2214518      2.7808
class2       0.18156457  0.0382046      0.0000
class3       0.06332765 -0.0087552      0.0000
class4      -0.02227055 -0.0031541      0.0000
class5      -0.00077172  0.0821863      0.0000
smoke1-19   -0.41298650 -0.2198673      0.0000
smoke20+    -0.30562472 -0.1245019      0.0000
```

Instead, when there are no quantitative predictors, and when the odds ratio is relatively constant (as here) it is easier to fit ordinary loglinear models than to use the bivariate logit formulation of the previous example. These allow the responses H and U to depend on the class-smoking combinations separately, by including the terms $[CSH]$ or $[CSU]$, respectively.

The minimal, null model, $[CS][H][U]$ fits the marginal association of the numbers in each class-smoking category, but asserts that the responses, H and U are independent, which we have already seen is contradicted by the data. We take $[CS][HU]$ as the baseline model (Model 1), asserting no relation between response and predictor variables, but associations within each set are allowed, These models are fit as shown below.

```
> # null model
> tox.glm0 <- glm(Freq ~ class*smoke + hyper + urea,
+                 data = Toxaemia, family = poisson)
> # baseline model: no association between predictors and responses
> tox.glm1 <- glm(Freq ~ class*smoke + hyper*urea,
+                 data = Toxaemia, family = poisson)
```

We proceed to fit a collection of other models, adding terms to allow more associations between the responses and predictors. Summary measures of goodness of fit and parsimony are shown in Table 10.1.

```
> tox.glm2 <- update(tox.glm1, . ~ . + smoke*hyper + class*urea)
>
> tox.glm3 <- glm(Freq ~ (class + smoke + hyper + urea)^2,
+                 data=Toxaemia, family=poisson)
>
> tox.glm4 <- glm(Freq ~ class*smoke*hyper + hyper*urea + class*urea,
+                 data=Toxaemia, family=poisson)
>
> tox.glm5 <- update(tox.glm4, . ~ . + smoke*urea)
>
> tox.glm6 <- update(tox.glm4, . ~ . + class*smoke*urea)
```

Table 10.1: Loglinear models, `tox.glm*`, fit to the Toxaemia data {tab:toxmod}

Model	Terms	df	G^2	p-value	G^2/df	AIC	BIC	R^2
0	CS H U	43	672.85	0.0000	15.65	586.85	264.27	.
1	CS HU	42	179.03	0.0000	4.26	95.03	-220.04	0.000
2	CS HU SH CU	36	46.12	0.1203	1.28	-25.88	-295.94	0.742
3	CS CH CU HU SH CU	30	40.47	0.0960	1.35	-19.53	-244.58	0.774
4	CSH CU HU	24	26.00	0.3529	1.08	-22.00	-202.04	0.855
5	CSH CU SU HU	22	25.84	0.2588	1.17	-18.16	-183.20	0.856
6	CSH CSU HU	14	22.29	0.0729	1.59	-5.71	-110.74	0.875
7	CSH CSU SHU	12	15.65	0.2079	1.30	-8.35	-98.37	0.913
8	CSH CSU CHU SHU	8	12.68	0.1233	1.59	-3.32	-63.33	0.929
9	CSHU	0	0.00	0	0	0.00	0.00	1.000

```
>
> tox.glm7 <- update(tox.glm6, . ~ . + smoke*hyper*urea)
>
> tox.glm8 <- glm(Freq ~ (class + smoke + hyper + urea)^3,
+                 data = Toxaemia, family = poisson)
>
> tox.glm9 <- glm(Freq ~ (class + smoke + hyper + urea)^4,
+                 data = Toxaemia, family = poisson)
```

Model 2 adds the simple dependence of hypertension on smoking ($[SH]$) and that of urea on class ($[CU]$). Model 3 includes all two-way terms. In Model 4, hypertension is allowed to depend on both class and smoking jointly ($[CSH]$). In Model 5 an additional dependence of urea on smoking ($[SU]$) is included, while in Model 6 urea depends on class and smoking jointly ($[CSU]$).

None of these models contain three-way terms involving both H and U, so these models assume that the log odds ratio for hypertension given urea is constant over the explanatory variables. Recalling the conditional mosaics (Figure 10.21 and Figure 10.22), Models 7 and 8 add terms which allow the odds ratio to vary, first with smoking ($[SHU]$), then with class ($[CHU]$) as well. Finally, Model 9 is the saturated model, that fits perfectly.

How do we choose among these models? Model 2 is the smallest model whose deviance is non-significant. Models 4 and 5 both have a smaller ratio of G^2/df. For comparing nested models, we can also examine the change in deviance as terms are added (or dropped). Thus, going from Model 2 to Model 3 decreases the deviance by 5.65 on 6 df, while the step from Model 3 to Model 4 gives a decrease of 14.47, also on 6 df. These tests can be performed using `lrtest()` in the lmtest (Hothorn *et al.*, 2014) package, shown below for models `tox.glm1`–`tox.glm5`.

```
> library(lmtest)
> lmtest::lrtest(tox.glm1, tox.glm2, tox.glm3, tox.glm4, tox.glm5)

Likelihood ratio test

Model 1: Freq ~ class * smoke + hyper * urea
Model 2: Freq ~ class + smoke + hyper + urea + class:smoke + hyper:urea +
    smoke:hyper + class:urea
Model 3: Freq ~ (class + smoke + hyper + urea)^2
Model 4: Freq ~ class * smoke * hyper + hyper * urea + class * urea
Model 5: Freq ~ class + smoke + hyper + urea + class:smoke + class:hyper +
    smoke:hyper + hyper:urea + class:urea + smoke:urea + class:smoke:hyper
  #Df LogLik Df  Chisq Pr(>Chisq)
1  18   -260
2  24   -194  6 132.91     <2e-16 ***
3  30   -191  6   5.65      0.464
4  36   -184  6  14.47      0.025 *
5  38   -184  2   0.17      0.920
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The AIC and BIC statistics, balancing parsimony and goodness-of-fit, have their minimum value for Model 2, which we adopt here for this example.

Plotting model results

Whatever model is chosen, as a final step, it is important to determine what that model implies about the original research questions. Because our focus here is on the prevalence of each symptom, and their association, it is helpful to graph the fitted logits and log odds ratios implied by the model, as was done in Figure 10.18 and Figure 10.19.

The presentation goal here is to produce plots showing the observed logits and log odds ratios as in Figure 10.25 and Figure 10.24, supplemented by lines showing these values according to the fitted model. In Example 10.9 we fit the bivariate logit model, for which the response functions were the desired logits and log odds. Here, where we have fit ordinary loglinear models, the observed and fitted logits can be calculated from the observed and fitted frequencies. The calculations require a bit of R calisthenics to arrange these into forms suitable for plotting.

As we did earlier, we first reshape the *Toxaemia* to wide format, as a 15×4 table of observed frequencies. Because there are now two predictor variables, we take care to include the levels of `smoke` and `class` as additional columns.

```
> # reshape to 15 x 4 table of frequencies
> tox.tab <- xtabs(Freq ~ class + smoke + hyper + urea, Toxaemia)
> toxaemia <- t(matrix(aperm(tox.tab), 4, 15))
> colnames(toxaemia) <- c("hu", "hU", "Hu", "HU")
> rowlabs <- expand.grid(smoke = c("0", "1-19", "20+"), class = factor(1:5))
> toxaemia <- cbind(toxaemia, rowlabs)
```

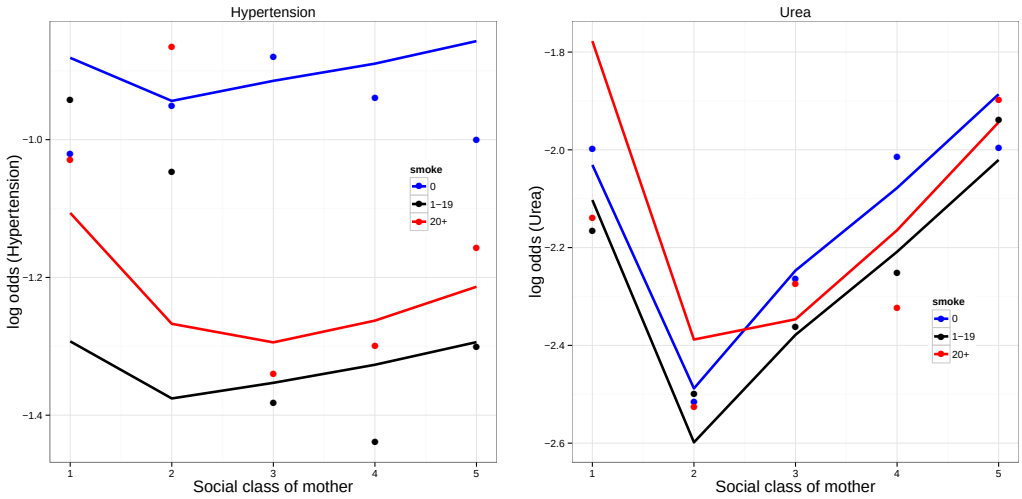
Applying `blogits()`, we get the observed logits and log odds ratios in `logitsTox`.

```
> # observed logits and log odds ratios
> logitsTox <- blogits(toxaemia[,4:1], add=0.5)
> colnames(logitsTox)[1:2] <- c("logitH", "logitU")
> logitsTox <- cbind(logitsTox, rowlabs)
> head(logitsTox)

    logitH  logitU    logOR smoke class
1 -1.02057 -1.9988  1.52679     0     1
2 -0.94261 -2.1665  1.07102  1-19     1
3 -1.02962 -2.1401  2.44854   20+     1
4 -0.95040 -2.5158  1.46196     0     2
5 -1.04699 -2.4983  0.86401  1-19     2
6 -0.86500 -2.5257 -1.14579   20+     2
```

The fitted frequencies are extracted using `predict(tox.glm2, type="response")` and then manipulated in a similar way to give `logitsFit`.

```
> # fitted frequencies, as a 15 x 4 table
> Fit <- t(matrix(predict(tox.glm2, type = "response"), 4, 15))
> colnames(Fit) <- c("HU", "Hu", "hU", "hu")
> Fit <- cbind(Fit, rowlabs)
> logitsFit <- blogits(Fit[, 1 : 4], add=0.5)
> colnames(logitsFit)[1 : 2] <- c("logitH", "logitU")
> logitsFit <- cbind(logitsFit, rowlabs)
```

In tabular form, you can examine any of these components, for example, the log odds ratios from the fitted values shown below.



{fig:tox-glm-logits1}

Figure 10.26: Observed (points) and fitted (lines) logits for the *Toxaemia* data under Model 2.

```
> matrix(logitsFit$logOR, 3, 5,
+        dimnames = list(smoke = c("0", "1-19", "20+"), class = 1 : 5))

      class
smoke       1      2      3      4      5
  0    1.3588 1.3638 1.3675 1.3643 1.3582
  1-19 1.3582 1.3678 1.3683 1.3674 1.3658
  20+  1.2799 1.3471 1.3662 1.3622 1.3511
```

Finally, we can plot the observed values in `logitsTox` (as points) and the fitted values under Model 2 in `logitsFit` (as lines), separately for the `logitH`, `logitU`, and `logOR` components. The code below uses ggplot2 (Wickham and Chang, 2015) for the log odds of hypertension, and is repeated for urea and the log odds ratio. These graphs are shown in Figure 10.26 and Figure 10.27.

```
> ggplot(logitsFit, aes(x = as.numeric(class), y = logitH, color = smoke)) +
+   theme_bw() +
+   geom_line(size = 1.2) +
+   scale_color_manual(values = c("blue", "black", "red")) +
+   ylab("log odds (Hypertension)") +
+   xlab("Social class of mother") +
+   ggtitle("Hypertension") +
+   theme(axis.title = element_text(size = 16)) +
+   geom_point(data = logitsTox,
+              aes(x = as.numeric(class), y = logitH, color = smoke), size = 3) +
+   theme(legend.position = c(0.85, .6))
```

According to this model, Figure 10.27 shows that the fitted log odds ratio is in fact nearly constant, while Figure 10.26 shows that the log odds for hypertension depends mainly on smoking (with a large difference of the non-smoking mothers from the rest) and that for protein urea depends mainly on social class.[11]

Yet, the great variability of the observed points around the fitted curves indicates that these relationships are not well-determined. Adding error bars showing the standard error around each

[11] Some possible enhancements to these graphs include (a) plotting on the scale of probabilities or including a right vertical axis showing corresponding probabilities; (b) using the same vertical axis limits for the two graphs for direct comparison.

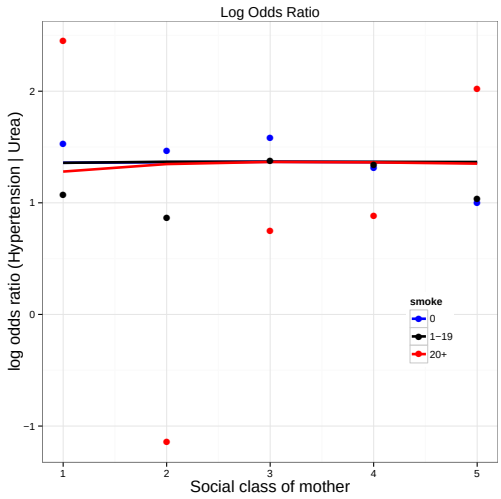


Figure 10.27: Observed (points) and fitted (lines) log odds ratios for the *Toxaemia* data under Model 2.

{fig:tox-glm-logits3}

fitted point would indicate that the data conforms as closely to the model as can be expected, given the widely different sample sizes. However, this would make the plots more complex, and so was omitted here. In addition to showing the pattern of the results according to the fitted model, such graphs also help us to appreciate the model's limitations.

△

10.5 Chapter summary

{sec:loglin2-summary}

- Standard loglinear models treat all variables as unordered factors. When one or more factors are ordinal, however, loglinear and logit models may be simplified by assigning quantitative scores to the levels of an ordered factor. Such models are often more sensitive and have greater power because they are more focused.

- Models for square tables, with the same row and column categories are an important special case. For these and other structured tables, a variety of techniques provide the opportunity to fit models more descriptive than the independence model and more parsimonious than the saturated model.

- When there are several categorical responses, along with one or more explanatory variables, some special forms of loglinear and logit models may be used to separate the marginal dependence of each response on the explanatory variables from the interdependence among the responses.

- In all these cases, the interplay between graphing and fitting is important in arriving at an understanding of the relationships among variables and an adequate descriptive model which is faithful to the details of the data.

10.6 Lab exercises

{lab:10.1}
{sec:loglin2-lab}

Exercise 10.1 Example 10.5 presented an analysis of the data on visual acuity for the subset of women in the *VisualAcuity* data. Carry out a parallel analysis of the models fit there for the men in this data set, given by:

```
> data("VisualAcuity", package="vcd")
> men <- subset(VisualAcuity, gender=="male", select=-gender)
```

{lab:10.2}

Exercise 10.2 Table 10.2 gives a 4×4 table of opinions about premarital sex and whether methods of birth control should be made available to teenagers aged 14–16 from the 1991 General Social Survey (Agresti, 2013, Table 10.3). Both variables are ordinal, and their grades are represented by the case of the row and column labels.

Table 10.2: Opinions about premarital sex and availability of teenage birth control. *Source*: Agresti (2013, Table 10.3)

{tab:birthcontrol}

Premarital sex	Birth control			
	DISAGREE	disagree	agree	AGREE
WRONG	81	68	60	38
Wrong	24	26	29	14
wrong	18	41	74	42
OK	36	57	161	157

(a) Fit the independence model to these data using `loglm()` or `glm()`.
(b) Make a mosaic display showing departure from independence and describe verbally the pattern of association.
(c) Treating the categories as equally spaced, fit the $L \times L$ model of uniform association, as in Section 10.1. Test the difference against the independence model with a likelihood-ratio test.
(d) Fit the RC(1) model with `gnm()`, and test the difference of this against the model of uniform association.
(e) Write a brief summary of these results, including plots useful for explaining the relationships in this data set.

{lab:10.3}

Exercise 10.3 For the data on attitudes toward birth control in Table 10.2,

(a) Calculate and plot the observed local log odds ratios.
(b) Also fit the R, C, and R+C models.
(c) Use the method described in Section 10.1.2 to visualize the structure of fitted local log odds ratios implied by each of these models, together with the RC(1) model.

{lab:10.4}

Exercise 10.4 The data set *gss8590* in logmult gives a $4 \times 5 \times 4$ table of education levels and occupational categories for the four combinations of gender and race from the General Social Surveys, 1985–1990 as reported by Wong (2001, Table 2). Wong (2010, Table 2.3B) later used the subset pertaining to women to illustrate RC(2) models. This data is created below as `Women.tab`, correcting an inconsistency to conform with the 2010 table.

```
> data("gss8590", package="logmult")
> Women.tab <- margin.table(gss8590[,,c("White Women", "Black Women")], 1:2)
> Women.tab[2,4] <- 49
> colnames(Women.tab)[5] <- "Farm"
```

(a) Fit the independence model, and also the RC(1) and RC(2) models using `rc()` with marginal weights, as illustrated in Example 10.4. Summarize these statistical tests in a table.
(b) Plot the solution for the RC(2) model with 68% confidence ellipses. What verbal labels would you use for the two dimensions?
(c) Is there any indication that a simpler model, using integer scores for the row (Education) or column (Occupation) categories or both might suffice? If so, fit the analogous column effects, row effects or $L \times L$ model, and compare with the models fit in part (a).

References

Agresti, A. (1984). *Analysis of Ordinal Categorical Data*. New York: Wiley.

Agresti, A. (2013). *Categorical Data Analysis*. Wiley Series in Probability and Statistics. New York: Wiley-Interscience [John Wiley & Sons], 3rd edn.

Ashford, J. R. and Sowden, R. D. (1970). Multivariate probit analysis. *Biometrics*, 26, 535–546.

Becker, M. P. and Clogg, C. C. (1989). Analysis of sets of two-way contingency tables using association models. *Journal of the American Statistical Association*, 84(405), 142–151.

Bouchet-Valat, M. (2015). *logmult: Log-Multiplicative Models, Including Association Models*. R package version 0.6.1.

Brown, P. J., Stone, J., and Ord-Smith, C. (1983). Toxaemic signs during pregnancy. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, 32, 69–72.

Caussinus, H. (1966). Contribution à l'analyse statistique des tableaux de corrélation. *Annales de la Faculté des Sciences de l'Université de Toulouse*, 39 (année 1965), 77–183.

Firth, D. (2003). Overcoming the reference category problem in the presentation of statistical models. *Sociological Methodology*, 33, 1–18.

Firth, D. and Menezes, R. X. d. (2004). Quasi-variances. *Biometrika*, 91, 65–80.

Fox, J., Weisberg, S., Friendly, M., and Hong, J. (2015). *effects: Effect Displays for Linear, Generalized Linear, and Other Models*. R package version 3.0-4/r200.

Friendly, M. (2015). *vcdExtra: vcd Extensions and Additions*. R package version 0.6-7.

Goodman, L. A. (1972). Some multiplicative models for the analysis of cross classified data. In *Proceedings of the sixth Berkeley Symposium on Mathematical Statistics and Probability*, (pp. 649–696). Berkeley, CA: University of California.

Goodman, L. A. (1979). Simple models for the analysis of association in cross-classifications having ordered categories. *Journal of the American Statistical Association*, 74, 537–552.

Goodman, L. A. (1985). The analysis of cross-classified data having ordered and/or unordered categories: Association models, correlation models, and asymmetry models for contingency tables with or without missing entries. *Annals of Statistics*, 13(1), 10–69.

Goodman, L. A. (1986). Some useful extensions of the usual correspondence analysis approach and the usual log-linear models approach in the analysis of contingency tables. *International Statistical Review*, 54(3), 243–309. With a discussion and reply by the author.

Hauser, R. M. (1979). Some exploratory methods for modeling mobility tables and other cross-classified data. In K. F. Schuessler, ed., *Sociological Methodology 1980*, (pp. 413–458). San Francisco: Jossey-Bass.

Hothorn, T., Zeileis, A., Farebrother, R. W., and Cummins, C. (2014). *lmtest: Testing Linear Regression Models*. R package version 0.9-33.

Lang, J. B. and Agresti, A. (1994). Simultaneously modeling joint and marginal distributions of multivariate categorical responses. *Journal of the American Statistical Association*, 89(426), 625–632.

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

Meyer, D., Zeileis, A., and Hornik, K. (2015). *vcd: Visualizing Categorical Data*. R package version 1.3-3.

Milan, L. and Whittaker, J. (1995). Application of the parametric bootstrap to models that incorporate a singular value decomposition. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 44(1), 31–49.

Powers, D. A. and Xie, Y. (2008). *Statistical Methods for Categorical Data Analysis*. Bingley, UK: Emerald, 2nd edn.

Roberto, C., Giordano, S., Cazzaro, M., and Lang, J. (2014). *hmmm: hierarchical multinomial marginal models*. R package version 1.0-3.

Turner, H. and Firth, D. (2014). *gnm: Generalized Nonlinear Models*. R package version 1.0-7.

Wei, T. (2013). *corrplot: Visualization of a correlation matrix*. R package version 0.73.

Wickham, H. and Chang, W. (2015). *ggplot2: An Implementation of the Grammar of Graphics*. R package version 1.0.1.

Wong, R. S.-K. (2001). Multidimensional association models: A multilinear approach. *Sociological Methods and Research*, 30(2), 197–240.

Wong, R. S.-K. (2010). *Association Models*. Quantitative Applications in the Social Sciences. Los Angeles: SAGE Publications.

Yee, T. W. (2015). *VGAM: Vector Generalized Linear and Additive Models*. R package version 0.9-7.