Chapter 9

Loglinear and Logit Models for Contingency Tables

{ch:loglin}

This chapter extends the model-building approach to loglinear and logit models. These comprise another special case of generalized linear models designed for contingency tables of frequencies. They are most easily interpreted through visualizations, including mosaic displays and effect plots of associated logit models.

9.1 Introduction

{sec:loglin-intro}

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

The chapter continues the modeling framework begun in Chapter 7, and takes up the case of loglinear models for contingency tables of frequencies, when all variables are discrete, another special case of generalized linear models. These models provide a comprehensive scheme to describe and understand the associations among two or more categorical variables. Whereas logistic regression models focus on the prediction of one response factor, loglinear models treat all variables symmetrically, and attempt to model all important associations among them.

In this sense, loglinear models are analogous to a correlation analysis of continuous variables, where the goal is to determine the patterns of dependence and independence among a set of variables. When one variable is a response and the others are explanatory, certain loglinear models are equivalent to logistic models for that response. Such models are also particularly useful when there are two or more response variables, a case that would require a multivariate version of the generalized linear model, for which the current theory and implementations are thin at best.

Chapter 5 and Chapter 6 introduced some basic aspects of loglinear models in connection with mosaic displays and correspondence analysis. In this chapter, the focus is on fitting and interpreting loglinear models. The usual analyses, with `loglm()` and `glm()` present the results in terms of tables of parameter estimates. Particularly for larger tables, it becomes difficult to understand the nature of these associations from tables of parameter estimates. Instead, we emphasize plots of observed and predicted frequencies, probabilities or log odds (when there are one or more response

variables), as well as mosaic and other displays for interpreting a given model. We also illustrate how mosaic displays and correspondence analysis plots may be used in a complementary way to the usual numerical summaries, to provide additional insights into the data.

Section 9.2 gives a brief overview of loglinear models in relation to the more familiar ANOVA and regression models for quantitative data. Methods and software for fitting these models are discussed in Section 9.3. When one variable is a response, logit models for that response provide a simpler, but equivalent means for interpreting and graphing results of loglinear models, as we describe in Section 9.4. In Section 9.5 we consider problems that arise in sparce contingency tables containing cells with frequencies of zero.

9.2 Loglinear models for frequencies

{sec:loglin-counts}

Loglinear models have been developed from two formally distinct, but related perspectives. The first is a discrete analog of familiar ANOVA models for quantitative data, where the multiplicative relations among joint and marginal probabilities are transformed into an additive one by transforming the counts to logarithms. The second is an analog of regression models, where the log of the cell frequency is modeled as a linear function of discrete predictors, with a random component often taken as the Poisson distribution and called ***Poisson regression***; this approach is treated in more detail as generalized linear models for count data in Chapter 11.

9.2.1 Loglinear models as ANOVA models for frequencies

For two discrete variables, A and B, suppose we have a multinomial sample of n_{ij} observations in each cell i, j of an $I \times J$ contingency table. To ease notation, we replace a subscript by $+$ to represent summation over that dimension, so that $n_{i+} = \Sigma_j n_{ij}$, $n_{+j} = \Sigma_i n_{ij}$, and $n_{++} = \Sigma_{ij} n_{ij}$.

Let π_{ij} be the joint probabilities in the table, and let $m_{ij} = n_{++}\pi_{ij}$ be the expected cell frequencies under any model. Conditional on the observed total count, n_{++}, each count has a Poisson distribution, with mean m_{ij}. Any loglinear model may be expressed as a linear model for the $\log m_{ij}$. For example, the hypothesis of independence means that the expected frequencies, m_{ij}, obey

$$m_{ij} = \frac{m_{i+}\, m_{+j}}{m_{++}} \ .$$

This multiplicative model can be transformed to an additive (linear) model by taking logarithms of both sides:

$$\log(m_{ij}) = \log(m_{i+}) + \log(m_{+j}) - \log(m_{++}) \ ,$$

which is usually expressed in an equivalent form in terms of model parameters,

{eq:lmain}

$$\log(m_{ij}) = \mu + \lambda_i^A + \lambda_j^B \tag{9.1}$$

where μ is a function of the total sample size, λ_i^A is the "main effect" for variable A, $\lambda_i^A = \log \pi_{i+} - \sum_k (\log \pi_{k+})/I$, and λ_j^B is the "main effect" for variable B, $\lambda_j^B = \log \pi_{+j} - \sum_k (\log \pi_{+k})/J$. Model Eqn. (9.1) is called the ***loglinear independence model*** for a two-way table.

In this model, there are $1 + I + J$ parameters, but only $(I-1)+(J-1)$ are separately estimable. Hence, the typical ANOVA sum-to-zero restrictions are usually applied to the parameters:

$$\sum_i^I \lambda_i^A = \sum_j^J \lambda_j^B = 0 \ .$$

These "main effects" in loglinear models pertain to differences among the marginal probabilities of a variable (which are usually not of direct interest).

Other restrictions to make the parameters identifiable are also used. Setting the first values, λ_1^A and λ_1^B to zero (the default in `glm()`), defines $\lambda_i^A = \log \pi_{i+} - \log \pi_{1+}$, and $\lambda_j^B = \log \pi_{+j} - \log \pi_{+1}$, as deviations from the first, reference category, but these parameterizations are otherwise identical. For modeling functions in R (`lm()`, `glm()`, etc.) the reference category parameterization is obtained using `contr.treatment()`, while the sum-to-zero constraints are obtained with `contr.sum()`.

Model Eqn. (9.1) asserts that the row and column variables are independent. For a two-way table, a model that allows an arbitrary association between the variables is the ***saturated model***, including an additional term, λ_{ij}^{AB}:

$$\log(m_{ij}) = \mu + \lambda_i^A + \lambda_j^B + \lambda_{ij}^{AB} \ , \tag{9.2}$$ {eq:lsat}

where again, restrictions must be imposed for estimation:

$$\sum_i^I \lambda_i^A = 0, \quad \sum_j^J \lambda_j^B = 0, \quad \sum_i^I \lambda_{ij}^{AB} = \sum_j^J \lambda_{ij}^{AB} = 0 \ . \tag{9.3}$$ {eq:lrestrict}

There are thus $I-1$ linearly independent λ_i^A row parameters, $J-1$ linearly independent λ_j^B column parameters, and $(I-1)(J-1)$ linearly independent λ_{ij}^{AB} association parameters. This model is called the *saturated model* because the number of parameters in μ, λ_i^A, λ_j^B, and λ_{ij}^{AB} is equal to the number of frequencies in the two-way table,

$$\underset{(\mu)}{1} + \underset{(\lambda_i^A)}{I-1} + \underset{(\lambda_j^B)}{J-1} + \underset{(\lambda_{ij}^{AB})}{(I-1)(J-1)} = \underset{(n_{ij})}{IJ}$$

The association parameters λ_{ij}^{AB} express the departures from independence, so large absolute values pertain to cells that differ from the independence model.

Except for the difference in notation, model Eqn. (9.2) is formally the same as a two-factor ANOVA model with an interaction, typically expressed as $E(y_{ij}) = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij}$. Hence, associations between variables in loglinear models are analogous to interactions in ANOVA models. The use of superscripted symbols, $\lambda_i^A, \lambda_j^B, \lambda_{ij}^{AB}$ rather than separate Greek letters is a convention in loglinear models, and useful mainly for multiway tables.

Models such as Eqn. (9.1) and Eqn. (9.2) are examples of ***hierarchical models***. This means that the model must contain all lower-order terms contained within any high-order term in the model. Thus, the saturated model, Eqn. (9.2) contains λ_{ij}^{AB}, and therefore *must* contain λ_i^A and λ_j^B. As a result, hierarchical models may be identified by the shorthand notation which lists only the high-order terms: model Eqn. (9.2) is denoted $[AB]$, while model Eqn. (9.1) is $[A][B]$.

9.2.2 Loglinear models for three-way tables

{sec:loglin-3way}

Loglinear models for three-way contingency tables were described briefly in Section 5.4.2. Each type of model allows associations among different sets of variables and each has a different independence interpretation, as illustrated in Table 5.2.

For a three-way table, the saturated model, denoted $[ABC]$ is

$$\log\, m_{ijk} = \mu + \lambda_i^A + \lambda_j^B + \lambda_k^C + \lambda_{ij}^{AB} + \lambda_{ik}^{AC} + \lambda_{jk}^{BC} + \lambda_{ijk}^{ABC} \ . \tag{9.4}$$ {eq:lsat3}

This model allows all variables to be associated; Eqn. (9.4) fits the data perfectly because the number of independent parameters equals the number of table cells. Two-way terms, such as λ_{ij}^{AB} pertain to the *conditional association* between pairs of factors, controlling for the remaining variable. The presence of the three-way term, λ_{ijk}^{ABC}, means that the partial association (conditional odds ratio) between any pair varies over the levels of the third variable.

Omitting the three-way term in Model Eqn. (9.4) gives the model $[AB][AC][BC]$,

$$\log m_{ijk} = \mu + \lambda_i^A + \lambda_j^B + \lambda_k^C + \lambda_{ij}^{AB} + \lambda_{ik}^{AC} + \lambda_{jk}^{BC} \ , \tag{9.5}$$ {eq:lno3way}

in which all pairs are conditionally dependent given the remaining one. For any pair, the conditional odds ratios are the *same* at all levels of the remaining variable, so this model is often called the ***homogeneous association model***.

The interpretation of terms in this model may be illustrated using the Berkeley admissions data (Example 4.11 and Example 4.15), for which the factors are Admit, Gender, and Department, in a $2 \times 2 \times 6$ table. In the homogeneous association model,

$$\log m_{ijk} = \mu + \lambda_i^A + \lambda_j^D + \lambda_k^G + \lambda_{ij}^{AD} + \lambda_{ik}^{AG} + \lambda_{jk}^{DG} \ , \tag{9.6}$$ {eq:berk1}

the λ-parameters have the following interpretations:

- The main effects, λ_i^A, λ_j^D and λ_k^G pertain to differences in the one-way marginal probabilities. Thus λ_j^D relates to differences in the total number of applicants to these departments, while λ_k^G relates to the differences in the overall numbers of men and women applicants.
- λ_{ij}^{AD} describes the conditional association between admission and department, that is different admission rates across departments (controlling for gender).
- λ_{ik}^{AG} relates to the conditional association between admission and gender, controlling for department. This term, if significant, might be interpreted as indicating gender-bias in admissions.
- λ_{jk}^{DG}, the association between department and gender, indicates whether males and females apply differentially across departments.

As we discussed earlier (Section 5.4), loglinear models for three-way (and larger) tables often have an interpretation in terms of various types of independence relations illustrated in Table 5.2. The model Eqn. (9.5) has no such interpretation, however the smaller model $[AC][BC]$ can be interpreted as asserting that A and B are (conditionally) independent controlling for C; this independence interpretation is symbolized as $A \perp B \,|\, C$. Similarly, the model $[AB][C]$ asserts that A and B are jointly independent of C: $(A, B) \perp C$, while the model $[A][B][C]$ is the model of mutual (complete) independence, $A \perp B \perp C$.

9.2.3 Loglinear models as GLMs for frequencies

{sec:loglin-glms}

In the GLM approach, a loglinear model may be cast in the form of a regression model for $\log \boldsymbol{m}$, where the table cells are reshaped to a column vector. One advantage is that models for tables of any size and structure may be expressed in a compact form.

For a contingency table of variables $A, B, C, \cdots$, with $N = I \times J \times K \times \cdots$ cells, let $\boldsymbol{n}$ denote a column vector of the observed counts arranged in standard order, and let $\boldsymbol{m}$ denote a similar vector of the expected frequencies under some model. Then *any* loglinear model may be expressed in the form

$$\log \boldsymbol{m} = \boldsymbol{X}\boldsymbol{\beta} \ ,$$

where $\boldsymbol{X}$ is a known design or ***model matrix*** and $\boldsymbol{\beta}$ is a column vector containing the unknown λ parameters.

For example, for a 2×2 table, the saturated model Eqn. (9.2) with the usual zero-sum constraints Eqn. (9.3) can be represented as

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

Note that only the linearly independent parameters are represented here. $\lambda_2^A = -\lambda_1^A$, because $\lambda_1^A + \lambda_2^A = 0$, and $\lambda_2^B = -\lambda_1^B$, because $\lambda_1^B + \lambda_2^B = 0$, and so forth.

An additional substantial advantage of the GLM formulation is that it makes it easier to express models with ordinal or quantitative variables. `glm()`, with a model formula of the form `Freq ~ .` involving factors $A, B, \ldots$ and quantitative variables $x_1, x_2, \ldots$, constructs the model matrix $\boldsymbol{X}$ from the terms given in the formula. A factor with K levels gives rise to $K-1$ columns for its main effect and sets of $K-1$ columns in each interaction effect. A quantitative predictor, say x_1 (with a linear effect) creates a single column with its values and interactions with other terms are calculated at the products of the columns for the main effects.

The parameterization for factors is controlled by the contrasts assigned to a given factor (if any), or by the general `contrasts` option, that gives the contrast functions used for unordered and ordered factors:

```
> options("contrasts")

$contrasts
        unordered           ordered
"contr.treatment"      "contr.poly"
```

This says that, by default, unordered factors use the baseline (first) reference-level parameterization, while ordered factors are given a parameterization based on orthogonal polynomials, allowing linear, quadratic, ... effects, assuming integer-spacing of the factor levels.

9.3 Fitting and testing loglinear models

{sec:loglin-fitting}

For a given table, possible loglinear models range from the baseline model of mutual independence, $[A][B][C][\ldots]$ to the saturated model, $[ABC\ldots]$ that fits the observed frequencies perfectly, but offers no simpler description or interpretation than the data itself.

Fitting a loglinear model is usually a process of deciding which association terms are large enough ("significantly different from zero") to warrant inclusion in a model to explain the observed frequencies. Terms which are excluded from the model go into the residual or error term, which reflects the overall badness-of-fit of the model. The usual goal of loglinear modeling is to find a small model (few association terms) which nonetheless achieves a reasonable fit (small residuals).

9.3.1 Model fitting functions

{sec:loglin-functions}

In R, the most basic function for fitting loglinear models is `loglin()` in the stats package. This uses the classical iterative proportional fitting (IPF) algorithm described in Haberman (1972) and Fienberg (1980, §3.4). It is designed to work with the frequency data in table form, and a model specified in terms of the (high-order) table margins to be fitted. For example, the model Eqn. (9.5) of homogenous association for a three-way table is specified as

```
> loglin(mytable, margin=list(c(1, 2), c(1, 3), c(2, 3)))
```

The function `loglm()` in MASS provides a more convenient front-end to `loglin()` to allow loglinear models to be specified using a model formula. With table variables `A`, `B` and `C`, the same model can be fit using `loglm()` as

```
> loglm(~ (A + B + C)^2, data=mytable)
```

When the data is a frequency data frame with frequencies in `Freq`, for example, the result of `mydf <- as.data.frame(mytable)`, you can also use a two-sided formula:

```
> loglm(Freq ~ (A + B + C)^2, data=mydf)
```

As implied in Section 9.2.3, loglinear models can also be fit using `glm()`, using `family=poisson` which constructs the model for `log(Freq)`. The same model is fit with `glm()` as:

```
> glm(Freq ~ (A + B + C)^2, data=mydf, family=poisson)
```

While all of these fit equivalent models, the details of the printed output, model objects, and available methods differ, as indicated in some of the examples that follow.

It should be noted that both the `loglin()`/`loglm()` methods based on iterative proportional fitting, and the `glm()` approach using the Poisson model for log frequency give maximum likelihood estimates, $\widehat{\boldsymbol{m}}$, of the expected frequencies, as long as all observed frequencies $\boldsymbol{n}$ are *all* positive. Some special considerations when there cells with zero frequencies are described in Section 9.5.

9.3.2 Goodness-of-fit tests

sec:loglin-goodfit}

For an *n*-way table, global goodness-of-fit tests for a loglinear model attempt to answer the question "How well does the model reproduce the observed frequencies?" That is, how close are the fitted frequencies estimated under the model to those of the saturated model or the data?

To avoid multiple subscripts for an *n*-way table, let $\boldsymbol{n} = n_1, n_2, \ldots, n_N$ denote the observed frequencies in a table with N cells, and corresponding fitted frequencies $\widehat{\boldsymbol{m}} = \widehat{m}_1, \widehat{m}_2, \ldots, \widehat{m}_N$ according to a particular loglinear model. The standard goodness-of-fit statistics are sums over the cells of measures of the difference between the $\boldsymbol{n}$ and $\widehat{\boldsymbol{m}}$.

The most commonly used are the familiar Pearson chi-square,

{eq:pchi}

$$X^2 = \sum_i^N \frac{(n_i - \widehat{m}_i)^2}{\widehat{m}_i} \ , \tag{9.7}$$

and the likelihood-ratio G^2 or ***deviance*** statistic,

{eq:pgsq}

$$G^2 = 2 \sum_i^N n_i \ \log \left(\frac{n_i}{\widehat{m}_i} \right) \ . \tag{9.8}$$

Both of these statistics have asymptotic χ^2 distributions (as $\Sigma \boldsymbol{n} \rightarrow \infty$), reasonably well-approximated when all expected frequencies are large.[1] The (residual) degrees of freedom are the number of cells (N) minus the number of estimated parameters. The likelihood-ratio test can also be expressed as twice the difference in log-likelihoods under saturated and fitted models,

$$G^2 = 2 \log \left[\frac{\mathcal{L}(\boldsymbol{n}; \boldsymbol{n})}{\mathcal{L}(\widehat{\boldsymbol{m}}; \boldsymbol{n})} \right] = 2[\log \mathcal{L}(\boldsymbol{n}; \boldsymbol{n}) - \log \mathcal{L}(\widehat{\boldsymbol{m}}; \boldsymbol{n})] \ ,$$

where $\mathcal{L}(\boldsymbol{n}; \boldsymbol{n})$ is the likelihood for the saturated model and $\mathcal{L}(\widehat{\boldsymbol{m}}; \boldsymbol{n})$ is the corresponding maximized likelihood for the fitted model.

In practice such global tests are less useful for comparing competing models. You may find that several different models have an acceptable fit or, sadly, that none do (usually because you are "blessed" with a large sample size). It is then helpful to compare competing models *directly*, and two strategies are particularly useful in these cases.

[1] Except in bizarre or borderline cases, these tests provide the same conclusions when expected frequencies are at least moderate (all $\widehat{\boldsymbol{m}} > 5$). However, G^2 approaches the theoretical chi-squared distribution more slowly than does χ^2, and the approximation may be poor when the average cell frequency is less than 5.

First, the likelihood-ratio G^2 statistic has the property in that one can compare two ***nested models*** by their difference in G^2 statistics, which has a χ^2 distribution on the difference in degrees of freedom. Two models, M_1 and M_2, are nested when one, say, M_2, is a special case of the other. That is, model M_2 (with ν_2 residual df) contains a subset of the parameters of M_1 (with ν_1 residual df), the remaining ones being effectively set to zero. Model M_2 is therefore more restrictive and cannot fit the data better than the more general model M_1, i.e., $G^2(M_2) \geq G^2(M_2)$. The least restrictive of all models, with $G^2 = 0$ and $\nu = 0$ df is the saturated model for which $\widehat{\boldsymbol{m}} = \boldsymbol{n}$.

Assuming that the less restrictive model M_1 fits, the difference in G^2,

$$\begin{aligned} \Delta G^2 \equiv G^2(M_2 \,|\, M_1) &= G^2(M_2) - G^2(M_1) && (9.9) \quad \texttt{\{eq:gsqnest1\}} \\ &= 2 \sum_i n_i \, \log(\widehat{m}_{i1} / \widehat{m}_{i2}) && (9.10) \quad \texttt{\{eq:gsqnest2\}} \end{aligned}$$

has a chi-squared distribution with df = $\nu_2 - \nu_1$. The last equality Eqn. (9.10) follows from substituting in Eqn. (9.8).

Rearranging terms in Eqn. (9.9), we see that we can partition the $G^2(M_2)$ into two terms,

$$G^2(M_2) = G^2(M_1) + G^2(M_2 \,|\, M_1) \ .$$

The first term measures the difference between the data and the more general model M_1. If this model fits, the second term measures the additional lack of fit imposed by the more restrictive model. In addition to providing a more focused test, $G^2(M_2 \,|\, M_1)$ also follows the chi-squared distribution more closely when some $\{m_i\}$ are small (Agresti, 2013, §10.6.3).

Alternatively, a second strategy uses other measures that combine goodness-of-fit with model parsimony and may also be used to compare non-nested models. The statistics described below are all cast in the form of badness-of-fit relative to degrees of freedom, so that smaller values reflect "better" models.

The simplest idea (Goodman, 1971) is to use G^2/df (or χ^2/df), which has an asymptotic expected value of 1 for a good-fitting model. This type of measure is not routinely reported by R software, but is easy to calculate from output.

The ***Akaike Information Criterion*** (AIC) statistic (Akaike, 1973) is a very general criterion for model selection with maximum likelihood estimation, based on the idea of maximizing the information provided by a fitted model. AIC is defined generally as

$$\text{AIC} = -2 \log \mathcal{L} + 2k$$

where $\log \mathcal{L}$ is the maximized log likelihood and k is the number of parameters estimated in the model. Better models correspond to *smaller* AIC. For loglinear models, minimizing AIC is equivalent to minimizing

$$\text{AIC}^\star = G^2 - 2\,\nu \ ,$$

where ν is the residual df, but the values of AIC and AIC* differ by an arbitrary constant. This form is easier to calculate by hand from the output of any modeling function if AIC is not reported, or an `AIC()` method is not available.

A third statistic of this type is the ***Bayesian Information Criterion*** (BIC) due to Schwartz (1978) and Raftery (1986),

$$\text{BIC} = G^2 - \log(n)\,\nu \ ,$$

where n is the total sample size. Both AIC and BIC penalize the fit statistic for increasing number of parameters. BIC also penalizes the fit directly with (log) sample size, and so expresses a preference for less complex models than AIC as the sample size increases.

9.3.3 Residuals for loglinear models

{sec:loglin-residuals}

Test statistics such as G^2 can determine whether a model has significant lack of fit, and model comparison tests using $\Delta G^2 = G^2(M_2 \,|\, M_1)$ can assess whether the extra term(s) in model M_1 significantly improves the model fit. Beyond these tests, the pattern of residuals for individual cells offers important clues regarding the nature of lack of fit and can help suggests associations that could be accounted for better.

As with logistic regression models (Section 7.5.1), several types of residuals are available for loglinear models. For cell i in the vector form of the contingency table, the ***raw residual*** is simply the difference between the observed and fitted frequencies, $e_i = n_i - \widehat{m}_i$.

The ***Pearson residual*** is the square root of the contribution of the cell to the Pearson χ^2,

{eq:reschi2}

$$r_i = \frac{n_i - \widehat{m}_i}{\sqrt{\widehat{m}_i}} \tag{9.11}$$

Similarly, the ***deviance residual*** can be defined as

{eq:resdev2}

$$g_i = \text{sign}(n_i - \widehat{m}_i)\sqrt{2n_i \log(n_i/\widehat{m}_i) - 2(n_i - \widehat{m}_i)} \tag{9.12}$$

Both of these attempt to standardize the distribution of the residuals to a standard normal, $N(0, 1)$ form. However, as pointed out by Haberman (1973), the asymptotic variance of these is less than one (with average value df/N) but, worse— the variance decreases with $\widehat{m}_i$. That is, residuals for cells with small expected frequencies have larger sampling variance, as might be expected.

Consequently, Haberman suggested dividing the Pearson residual by its estimated standard error, giving what are often called ***adjusted residual***s. When loglinear models are fit using the GLM approach, the adjustment may be calculated using the leverage ("hat value"), h_i to give appropriately standardized residuals,

$$\begin{aligned} r_i^\star &= r_i/\sqrt{1 - h_i} \\ g_i^\star &= g_i/\sqrt{1 - h_i} \end{aligned}$$

These standardized versions are generally preferable, particularly for visualizing model lack of fit using mosaic displays. The reason for preferring adjusted residuals is illustrated in Figure 9.1, a plot of the factors, $\sqrt{1 - h_i}$, determining the standard errors of the residuals against the fitted values, $\widehat{m_i}$, in the model for the `UCBAdmissions` data described in Example 9.2 below. The values shown in this plot are calculated as:

```
> berkeley <- as.data.frame(UCBAdmissions)
> berk.glm1 <- glm(Freq ~ Dept * (Gender+Admit), data=berkeley, family="poisson")
> fit <- fitted(berk.glm1)
> hat <- hatvalues(berk.glm1)
> stderr <- sqrt(1-hat)
```

In R, raw, Pearson and deviance residuals may be obtained using `residuals(model, type=)`, where `type` is one of `"raw"`, `"pearson"` and `"deviance"`. Standardized (adjusted) residuals can be calculated using `rstandard(model, type=)`, for `type="pearson"` and `type="deviance"` versions.

9.3.4 Using `loglm()`

{loglin-loglin}

Here we illustrate the basics of fitting loglinear models using `loglm()`. As indicated in Section 9.3.1, the model to be fitted is specified by a model formula involving the table variables. The

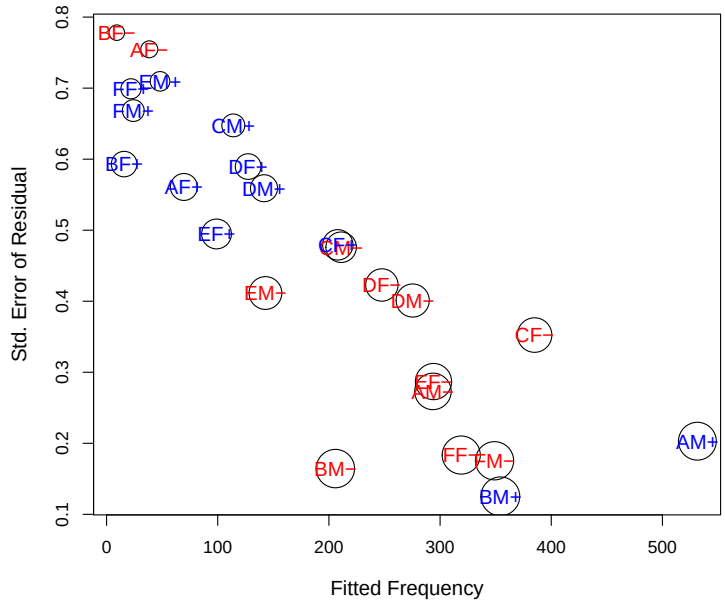

Figure 9.1: Standard errors of residuals, $\sqrt{1-h_i}$ decrease with expected frequencies. This plot shows why ordinary Pearson and deviance residuals may be misleading. The symbol size in the plot is proportional to leverage, h_i. Labels abbreviate Department, Gender and Admit, colored by Admit.

{fig:stres-plot}

MASS package provides a `coef()` method for "loglm" objects that extracts the estimated parameters and a `residuals()` method that calculates various types of residuals according to a `type` argument, one of `"deviance"`, `"pearson"`, `"response"`. vcd and vcdExtra provide a variety of plotting methods, including `assoc()`, `sieve()`, `mosaic()` and `mosaic3d()` for "loglm" objects.

{ex:berkeley5}

EXAMPLE 9.1: Berkeley admissions

The *UCBAdmissions* on admissions to the six largest graduate departments at U.C. Berkeley was examined using graphical methods in Chapter 4 (Example 4.15) and in Chapter 5 (Example 5.14). We can fit and compare several loglinear models as shown below.

The model of mutual independence, $[A][D][G]$, is not substantively reasonable here, because the association of `Dept` and `Gender` should be taken into account to control for these variables, but we show it here to illustrate the form of the printed output, giving the Pearson χ^2 and likelihood-ratio G^2 tests of goodness of fit, as well as some optional arguments for saving additional components in the result.

```
> data("UCBAdmissions")
> library(MASS)
> berk.loglm0 <- loglm(~ Dept + Gender + Admit, data=UCBAdmissions,
+                      param=TRUE, fitted=TRUE)
> berk.loglm0

Call:
loglm(formula = ~Dept + Gender + Admit, data = UCBAdmissions,
    param = TRUE, fitted = TRUE)

Statistics:
                     X^2 df P(> X^2)
```

```
Likelihood Ratio 2097.7 16        0
Pearson          2000.3 16        0
```

The argument `param=TRUE` stores the estimated parameters in the loglinear model and `fitted=TRUE` stores the fitted frequencies $\widehat{m}_{ijk}$. The fitted frequencies can be extracted from the model object using `fitted()`.

```
> structable(Dept ~ Admit+Gender, fitted(berk.loglm0))

                Dept      A      B      C      D      E      F
Admit    Gender
Admitted Male        215.10 134.87 211.64 182.59 134.64 164.61
         Female      146.68  91.97 144.32 124.51  91.81 112.25
Rejected Male        339.63 212.95 334.17 288.30 212.59 259.91
         Female      231.59 145.21 227.87 196.59 144.96 177.23
```

Similarly, you can extract the estimated parameters with `coef(berk.loglm0)`, and the Pearson residuals with `residuals(berk.loglm0, type="pearson")`.

Next, consider the model of conditional independence of gender and admission given department, $[AD][GD]$ that allows associations of admission with department and gender with department.

```
> # conditional independence in UCB admissions data
> berk.loglm1 <- loglm(~ Dept * (Gender + Admit), data=UCBAdmissions)
> berk.loglm1

Call:
loglm(formula = ~Dept * (Gender + Admit), data = UCBAdmissions)

Statistics:
                     X^2 df  P(> X^2)
Likelihood Ratio 21.736  6 0.0013520
Pearson          19.938  6 0.0028402
```

Finally for this example, the model of homogeneous association, $[AD][AG][GD]$ can be fit as follows.[2]

```
> berk.loglm2 <-loglm(~(Admit + Dept + Gender)^2, data=UCBAdmissions)
> berk.loglm2

Call:
loglm(formula = ~(Admit + Dept + Gender)^2, data = UCBAdmissions)

Statistics:
                     X^2 df  P(> X^2)
Likelihood Ratio 20.204  5 0.0011441
Pearson          18.823  5 0.0020740
```

Neither of these models fits particularly well, as judged by the goodness-of-fit Pearson χ^2 and likelihood-ratio G^2 test against the saturated model. The `anova()` method for a nested collection of "loglm" models gives a series of likelihood-ratio tests of the difference, ΔG^2 between each sequential pair of models according to Eqn. (9.9).

[2] It is useful to note here that the added term $[AG]$ allows a general association of admission with gender (controlling for department). A significance test for this term, or for model `berk.loglm2` against `berk.loglm1` is a proper test for the assertion of gender bias in admissions.

```
> anova(berk.loglm0, berk.loglm1, berk.loglm2, test="Chisq")

LR tests for hierarchical log-linear models

Model 1:
 ~Dept + Gender + Admit
Model 2:
 ~Dept * (Gender + Admit)
Model 3:
 ~(Admit + Dept + Gender)^2

          Deviance df Delta(Dev) Delta(df) P(> Delta(Dev)
Model 1   2097.671 16
Model 2     21.736  6  2075.9357        10        0.00000
Model 3     20.204  5     1.5312         1        0.21593
Saturated    0.000  0    20.2043         5        0.00114
```

The conclusion from these results is that the model `berk.loglm1` is not much worse than model `berk.loglm2`, but there is still significant lack-of-fit. The next example, using `glm()`, shows how to visualize the lack of fit and account for it.

△

9.3.5 Using `glm()`

{sec:loglin-glm}

Loglinear models fit with `glm()` require the data in a data frame in frequency form, for example as produced by `as.data.frame()` from a table. The model formula expresses the model for the frequency variable, and uses `family=poisson` to specify the error distribution. More general distributions for frequency data are discussed in Chapter 11.

{ex:berkeley6}

EXAMPLE 9.2: Berkeley admissions

For the $2 \times 2 \times 6$ *UCBAdmissions* table, first transform this to a frequency data frame:

```
> berkeley <- as.data.frame(UCBAdmissions)
> head(berkeley)

     Admit Gender Dept Freq
1 Admitted   Male    A  512
2 Rejected   Male    A  313
3 Admitted Female    A   89
4 Rejected Female    A   19
5 Admitted   Male    B  353
6 Rejected   Male    B  207
```

Then, the model of conditional independence corresponding to `berk.loglm1` can be fit using `glm()` as shown below.

```
> berk.glm1 <- glm(Freq ~ Dept * (Gender+Admit),
+                  data=berkeley, family="poisson")
```

Similarly, the all two-way model of homogeneous association is fit using

```
> berk.glm2 <- glm(Freq ~ (Dept + Gender + Admit)^2,
+                  data=berkeley, family="poisson")
```

These models are equivalent to those fit using `loglm()` in Example 9.1. We get the same residual G^2 as before, and the likelihood-ratio test of ΔG^2 given by `anova()` gives the same result, that the model `berk.glm2` offers no significant improvement over model `berk.glm1`.

```
> anova(berk.glm1, berk.glm2, test="Chisq")

Analysis of Deviance Table

Model 1: Freq ~ Dept * (Gender + Admit)
Model 2: Freq ~ (Dept + Gender + Admit)^2
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         6       21.7
2         5       20.2  1     1.53     0.22
```

Among other advantages of using `glm()` as opposed to `loglm()` is that an `anova()` method is available for *individual* "glm" models, giving significance tests of the contributions of each *term* in the model, as opposed to the tests for individual coefficients provided by `summary()`.[3]

```
> anova(berk.glm1, test="Chisq")

Analysis of Deviance Table

Model: poisson, link: log

Response: Freq

Terms added sequentially (first to last)


             Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                            23       2650
Dept          5      160        18       2491   <2e-16 ***
Gender        1      163        17       2328   <2e-16 ***
Admit         1      230        16       2098   <2e-16 ***
Dept:Gender   5     1221        11        877   <2e-16 ***
Dept:Admit    5      855         6         22   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We proceed to consider what is wrong with these models and how they can be improved. A mosaic display can help diagnose the reason(s) for lack of fit of these models. We focus here on the model $[AD][GD]$ that allows an association between gender and department (i.e., men and women apply at different rates to departments).

The `mosaic()` method for "glm" objects in vcdExtra provides a `residuals_type` argument, allowing `residuals_type="rstandard"` for standardized residuals. The `formula` argument here pertains to the order of the variables in the mosaic, not a model formula.

```
> library(vcdExtra)
> mosaic(berk.glm1, shade=TRUE, formula=~Admit+Dept+Gender,
+        residuals_type="rstandard", labeling=labeling_residuals,
+        main="Model: [AdmitDept][GenderDept]")
```

The mosaic display, shown in Figure 9.2, indicates that this model fits well (residuals are small) except in Department A. This suggests a model which allows an association between Admission and Gender in Department A only,

{eq:berk2}

$$\log m_{ijk} = \mu + \lambda_i^A + \lambda_j^D + \lambda_k^G + \lambda_{ij}^{AD} + \lambda_{jk}^{DG} + I(j=1)\lambda_{ik}^{AG} \ , \tag{9.13}$$

where the indicator function $I(j = 1)$ equals 1 for Department A ($j = 1$) and is zero otherwise. This

[3] Unfortunately, in the historical development of R, the `anova()` methods for linear and generalized linear models provide only *sequential* ("Type I") tests that are computationally easy, but useful only under special circumstances. The car package provides an analogous `Anova()` method that gives more generally useful *partial* ("Type II") tests for the additional contribution of each term beyond the others, taking marginal relations into account.

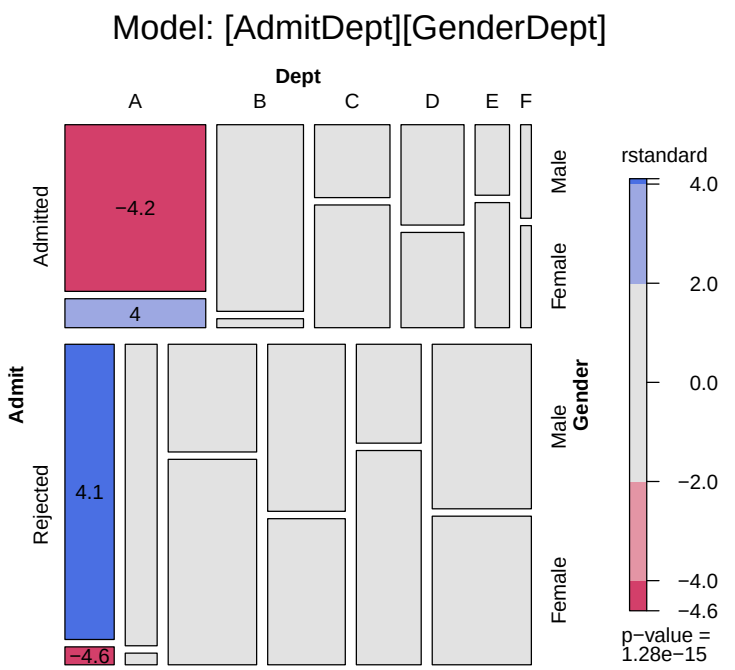

Figure 9.2: Mosaic display for the model [AD][GD], showing standardized residuals for the cell contributions to G^2

g:berk-glm1-mosaic}

model asserts that Admission and Gender are conditionally independent, given Department, except in Department A. It has one more parameter than the conditional independence model, $[AD][GD]$, and forces perfect fit in the four cells for Department A.

Model Eqn. (9.13) may be fit with `glm()` by constructing a variable equal to the interaction of `gender` and `admit` with a dummy variable having the value 1 for Department A and 0 for other departments.

```
> berkeley <- within(berkeley,
+                    dept1AG <- (Dept=='A')*(Gender=='Female')*(Admit=='Admitted'))
> head(berkeley)

      Admit Gender Dept Freq dept1AG
1 Admitted   Male    A  512       0
2 Rejected   Male    A  313       0
3 Admitted Female    A   89       1
4 Rejected Female    A   19       0
5 Admitted   Male    B  353       0
6 Rejected   Male    B  207       0
```

Fitting this model with the extra term `dept1AG` gives `berk.glm3`

```
> berk.glm3 <- glm(Freq ~ Dept * (Gender+Admit) + dept1AG,
+                  data=berkeley, family="poisson")
```

This model does indeed fit well, and represents a substantial improvement over model `berk.glm1`:

```
> vcdExtra::LRstats(berk.glm3)

Likelihood summary table:
          AIC BIC LR Chisq Df Pr(>Chisq)
berk.glm3 200 222     2.68  5       0.75

> anova(berk.glm1, berk.glm3, test="Chisq")

Analysis of Deviance Table

Model 1: Freq ~ Dept * (Gender + Admit)
Model 2: Freq ~ Dept * (Gender + Admit) + dept1AG
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         6      21.74
2         5       2.68  1     19.1  1.3e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The parameter estimate for the `dept1AG` term, $\widehat{\lambda}_{ik}^{AG} = 1.052$ may be interpreted as the log odds ratio of admission for females as compared to males in Dept. A. The odds ratio is $\exp(1.052) = 2.86$, the same as the value calculated from the raw data (see Section 4.4.2).

```
> coef(berk.glm3)[["dept1AG"]]

[1] 1.0521

> exp(coef(berk.glm3)[["dept1AG"]])

[1] 2.8636
```

Finally, Figure 9.3 shows the mosaic for this revised model. The absence of shading indicates a well-fitting model.

```
> mosaic(berk.glm3, shade=TRUE, formula=~Admit+Dept+Gender,
+        residuals_type="rstandard", labeling=labeling_residuals,
+        main="Model: [DeptGender][DeptAdmit] + DeptA*[GA]")
```

△

9.4 Equivalent logit models

{sec:loglin-logit}

Because loglinear models are formulated as models for the log (expected) frequency, they make no distinction between response and explanatory variables. In effect, they treat all variables as responses and describe their associations.

Logit (logistic regression) models, on the other hand, describe how the log odds for one variable depends on other, explanatory variables. There is a close connection between the two: When there is a response variable, each logit model for that response is equivalent to a loglinear model.

This relationship often provides a simpler way to formulate and test the model, and to plot and interpret the fitted results. Even when there is no response variable, the logit representation for one variable helps to interpret a loglinear model in terms of odds ratios. The price paid for this simplicity is that associations among the explanatory variables are not expressed in the model.

Consider, for example, the model of homogeneous association, $[AB][AC][BC]$, Eqn. (9.5) for a three-way table, and let variable C be a binary response. Under this model, the logit for variable C is

$$\begin{aligned} L_{ij} = \log\left(\frac{\pi_{ij|1}}{\pi_{ij|2}}\right) &= \log\left(\frac{m_{ij1}}{m_{ij2}}\right) \\ &= \log(m_{ij1}) - \log(m_{ij2}) \ . \end{aligned}$$

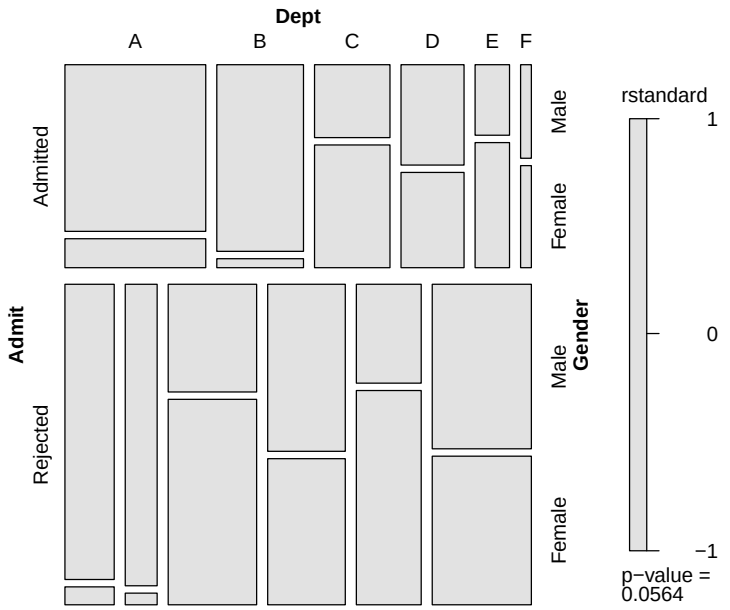

Figure 9.3: Mosaic display for the model `berk.glm3`, allowing an association of gender and admission in Department A. This model now fits the data well.

{g:berk-glm3-mosaic}

Substituting from Eqn. (9.5), all terms which do not involve variable C cancel, and we are left with

$$\begin{aligned} L_{ij} = \log(m_{ij1}/m_{ij2}) &= (\lambda_1^C - \lambda_2^C) + (\lambda_{i1}^{AC} - \lambda_{i2}^{AC}) + (\lambda_{j1}^{BC} - \lambda_{j2}^{BC}) \\ &= 2\lambda_1^C + 2\lambda_{i1}^{AC} + 2\lambda_{j1}^{BC} \ , \end{aligned} \tag{9.14}$$

{eq:logitab1}

because all λ terms sum to zero. We are interested in how these logits depend on A and B, so we can simplify the notation by replacing the λ parameters with more familiar ones, $\alpha = 2\lambda_1^C$, $\beta_i^A = 2\lambda_{i1}^{AC}$, etc., which express this relation more directly,

$$L_{ij} = \alpha + \beta_i^A + \beta_j^B \ . \tag{9.15}$$

{eq:logitab2}

In the logit model Eqn. (9.15), the response, C, is affected by both A and B, which have additive effects on the log odds of response category C_1 compared to C_2. The terms β_i^A and β_j^B correspond directly to $[AC]$ and $[BC]$ in the loglinear model Eqn. (9.5). The association among the explanatory variables, $[AB]$ is assumed in the logit model, but this model provides no explicit representation of that association. The logit model Eqn. (9.14) is equivalent to the loglinear model $[AB][AC][BC]$ in goodness-of-fit and fitted values, and parameters in the two models correspond directly.

Table 9.1 shows the equivalent relationships between all loglinear and logit models for a three-way table when variable C is a binary response. Each model necessarily includes the $[AB]$ association involving the predictor variables. The most basic model, $[AB][C]$, is the intercept-only model, asserting constant odds for variable C. The saturated loglinear model $[ABC]$, allows an interaction in the effects of A and B on C, meaning that the AC association or odds ratio varies with B.

More generally, when there is a binary response variable, say R, and one or more explanatory variables, $A, B, C, \ldots$, any logit model for R has an equivalent loglinear form. Every term in the

Table 9.1: Equivalent loglinear and logit models for a three-way table, with C as a binary response variable. {tab:loglin-logit}

Loglinear model	Logit model	Logit formula
$[AB][C]$	α	`C ~ 1`
$[AB][AC]$	$\alpha + \beta_i^A$	`C ~ A`
$[AB][BC]$	$\alpha + \beta_j^B$	`C ~ B`
$[AB][AC][BC]$	$\alpha + \beta_i^A + \beta_j^B$	`C ~ A + B`
$[ABC]$	$\alpha + \beta_i^A + \beta_j^B + \beta_{ij}^{AB}$	`C ~ A * B`

logit model, such as β_{ik}^{AC}, corresponds to an association of those factors with R, that is, $[ACR]$ in the equivalent loglinear model.

The equivalent loglinear model must also include all associations among the explanatory factors, the term $[ABC\ldots]$. Conversely, any loglinear model which includes all associations among the explanatory variables has an equivalent logit form. When the response factor has more than two categories, models for generalized logits (Section 8.3) also have an equivalent loglinear form. {ex:berkeley7}

EXAMPLE 9.3: Berkeley admissions

The homogeneous association model, $[AD][AG][DG]$ did not fit the `UCBAdmissions` data very well, and we saw that the term $[AG]$ was unnecessary. Nevertheless, it is instructive to consider the equivalent logit model. We illustrate the features of the logit model which lead to the same conclusions and simplified interpretation from graphical displays.

Because Admission is a binary response variable, model Eqn. (9.6) is equivalent to the logit model,

$$L_{ij} = \log\left(\frac{m_{\text{Admit}(ij)}}{m_{\text{Reject}(ij)}}\right) = \alpha + \beta_i^{\text{Dept}} + \beta_j^{\text{Gender}} \quad . \tag{9.16}$$ {eq:berk3}

That is, the logit model Eqn. (9.16) asserts that department and gender have additive effects on the log odds of admission. A significance test for the term β_j^{Gender} here is equivalent to the test of the $[AG]$ term for gender bias in the loglinear model. The observed log odds of admission here can be calculated as:

```
> (obs <- log(UCBAdmissions[1,,] / UCBAdmissions[2,,]))

        Dept
Gender        A      B       C      D      E      F
  Male   0.4921 0.5337 -0.5355 -0.704 -0.957 -2.770
  Female 1.5442 0.7538 -0.6604 -0.622 -1.157 -2.581
```

With the data in the form of the frequency data frame `berkeley` we used in Example 9.2, the logit model Eqn. (9.16) can be fit using `glm()` as shown below. In the model formula, the binary response is `Admit=="Admitted"`. The `weights` argument gives the frequency, `Freq` in each table cell.[4]

```
> berk.logit2 <- glm(Admit=="Admitted" ~ Dept + Gender,
+                    data=berkeley, weights=Freq, family="binomial")
> summary(berk.logit2)


Call:
glm(formula = Admit == "Admitted" ~ Dept + Gender, family = "binomial",
    data = berkeley, weights = Freq)
```

[4] Using weights gives the same fitted values, but not the same LR tests for model fit.

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-25.342  -13.058   -0.163   16.017   21.320

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)    0.5821     0.0690    8.44   <2e-16 ***
DeptB         -0.0434     0.1098   -0.40     0.69
DeptC         -1.2626     0.1066  -11.84   <2e-16 ***
DeptD         -1.2946     0.1058  -12.23   <2e-16 ***
DeptE         -1.7393     0.1261  -13.79   <2e-16 ***
DeptF         -3.3065     0.1700  -19.45   <2e-16 ***
GenderFemale   0.0999     0.0808    1.24     0.22
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 6044.3  on 23  degrees of freedom
Residual deviance: 5187.5  on 17  degrees of freedom
AIC: 5201

Number of Fisher Scoring iterations: 6
```

As in logistic regression models, parameter estimates may be interpreted as increments in the log odds, or $\exp(\beta)$ may be interpreted as the multiple of the odds associated with the explanatory categories. Because `glm()` uses a baseline category parameterization (by default) the coefficients of the first category of `Dept` and `Gender` are set to zero. You can see from the `summary()` output that the coefficients for the departments decline steadily from A–F.[5] The coefficient $\beta_F^{\text{Gender}} = 0.0999$ for females indicates that, overall, women were $\exp(0.0999) = 1.105$ times as likely as male applicants to be admitted to graduate school at U.C. Berkeley, a 10% advantage.

Similarly, the logit model equivalent of the loglinear model Eqn. (9.13) `berk.glm3` containing the extra 1 df term for an effect of gender in Department A is

$$L_{ij} = \alpha + \beta_i^{\text{Dept}} + I(j = 1)\beta^{\text{Gender}} \ . \tag{9.17}$$ {eq:berk4}

This model can be fit as follows:

```
> berkeley <- within(berkeley,
+                    dept1AG <- (Dept=='A')*(Gender=='Female'))
> berk.logit3 <- glm(Admit=="Admitted" ~ Dept + Gender + dept1AG,
+                    data=berkeley, weights=Freq, family="binomial")
```

In contrast to the tests for individual coefficients, the `Anova()` method in the car package gives likelihood-ratio tests of the terms in a model. As mentioned earlier, this provides *partial* ("Type II") tests for the additional contribution of each term beyond all others.

```
> library(car)
> Anova(berk.logit2)

Analysis of Deviance Table (Type II tests)

Response: Admit == "Admitted"
       LR Chisq Df Pr(>Chisq)
Dept      763.4  5     <2e-16 ***
Gender      1.5  1      0.216
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

[5] In fact, the departments were labeled A–F in decreasing order of rate of admission.

```
> Anova(berk.logit3)

Analysis of Deviance Table (Type II tests)

Response: Admit == "Admitted"
        LR Chisq Df Pr(>Chisq)
Dept       646.7  5    < 2e-16 ***
Gender       0.1  1      0.724
dept1AG     17.6  1   2.66e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Plotting logit models

Logit models are easier to interpret than the corresponding loglinear models because there are fewer parameters, and because these parameters pertain to the odds of a response category rather than to cell frequency. Nevertheless, interpretation is often easier still from a graph than from the parameter values.

The simple interpretation of these logit models can be seen by plotting the logits for a given model. To do that, it is necessary to construct a data frame containing the observed (`obs`) and fitted (`fit`) for the combinations of gender and department.

```
> pred2 <- cbind(berkeley[,1:3], fit=predict(berk.logit2))
> pred2 <- cbind(subset(pred2, Admit=="Admitted"), obs=as.vector(obs))
> head(pred2)

      Admit Gender Dept      fit      obs
1  Admitted   Male    A  0.58205  0.49212
3  Admitted Female    A  0.68192  1.54420
5  Admitted   Male    B  0.53865  0.53375
7  Admitted Female    B  0.63852  0.75377
9  Admitted   Male    C -0.68055 -0.53552
11 Admitted Female    C -0.58068 -0.66044
```

In this form, these results can be plotted as a line plot of the fitted logits vs. department, with separate curves for males and females, and adding points to show the observed values. Here, we use ggplot2 as shown below, with the `aes()` arguments `group=Gender, color=Gender`. This produces the left panel in Figure 9.4. The same steps for the model `berk.logit3` gives the right panel in this figure. The observed logits, of course, are the same in both plots.

```
> library(ggplot2)
> ggplot(pred2, aes(x=Dept, y=fit, group=Gender, color=Gender)) +
+   geom_line(size=1.2) +
+   geom_point(aes(x=Dept, y=obs, group=Gender, color=Gender), size=4) +
+   ylab("Log odds (Admitted)") + theme_bw() +
+   theme(legend.position=c(.8, .9),
+         legend.title=element_text(size=14),
+         legend.text=element_text(size=14))
```

The effects seen in our earlier analyses (Examples 5.14, 5.15 and 9.2) may all be observed in these plots. In the left panel of Figure 9.4, corresponding to the loglinear model $[AD][AG][DG]$, the effect of gender, β_j^{Gender}, in the equivalent logit model is shown by the constant separation between the two curves. From the plot we see that this effect is very small (and nonsignificant). In the right panel, corresponding to the logit model Eqn. (9.17), there is no effect of gender on admission, except in department A, where the extra parameter allows perfect fit.

△

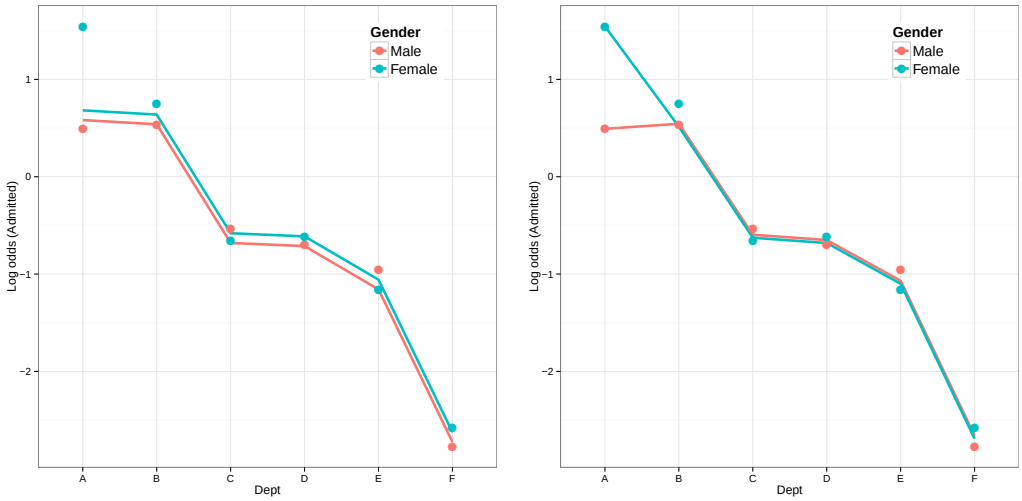

Figure 9.4: Observed (points) and fitted (lines) log odds of admissions in the logit models for the `UCBAdmissions` data. Left: the logit model Eqn. (9.16) corresponding to the loglinear model [AD] [AG] [DG]. Right: the logit model Eqn. (9.17), allowing only a 1 df term for Department A.

{fig:berk-logit}

9.5 Zero frequencies

{sec:loglin-zeros}

Cells with frequencies of zero create problems for loglinear and logit models. For loglinear models, most of the derivations of expected frequencies by maximum likelihood and other quantities that depend on these (e.g., G^2 tests) assume that all $n_{ijk\cdots} > 0$. In analogous logit models, the observed log odds (e.g., for a three-way table), $\log(n_{ij1}/n_{ij2})$, will be undefined if either frequency is zero.

Zero frequencies may occur in contingency tables for two different reasons:

- ***structural zeros*** (also called *fixed zeros*) will occur when it is impossible to observe values for some combinations of the variables. For these cases we should have $\widehat{m}_i = 0$ wherever $n_i = 0$. For example, suppose we have three different methods of contacting people at risk for some obscure genetically inherited disease: newspaper advertisement, telephone campaign, and radio appeal. If each person contacted in any way is classified dichotomously by the three methods of contact, there can never be a non-zero frequency in the 'No-No-No' cell.[6] Similarly, in a tabulation of seniors by gender and health concerns, there can never be males citing menopause or females citing prostate cancer. Square tables, such as wins and losses for sporting teams often have structural zeros in the main diagonal.

- ***sampling zeros*** (also called *random zeros*) occur when the total size of the sample is not large enough in relation to the probabilities in each of the cells to assure that someone will be observed in every cell. Here, it is permissible to have $\widehat{m}_i > 0$ when $n_i = 0$. This problem increases with the number of table variables. For example, in a European survey of religious affiliation, gender and occupation, we may not happen to observe any female Muslim vineyard-workers in France, although such individuals surely exist in the population. Even when zero frequencies do not occur, tables with many cells relative to the total frequency tend to produce small expected frequencies in at least some cells, which tends to make the G^2 statistics for model fit and likelihood-ratio statistics for individual terms unreliable.

[6] Yet, if we fit an unsaturated model, expected frequencies may be estimated for all cells, and provide a means to estimate the total number at risk in the population. See Lindsey (1995, Section 5.4).

Following Birch (1963), Haberman (1974) and many others (e.g., Bishop *et al.*, 1975) identified conditions under which the maximum likelihood estimate for a given loglinear model does not exist, meaning that the algorithms used in `loglin()` and `glm()` do not converge to a solution. The problem depends on the number and locations of the zero cells, but not on the size of the frequencies in the remaining cells. Fienberg and Rinaldo (2007) give a historical overview of the problem and current approaches and Agresti (2013, §10.6) gives a compact summary.

In R, the mechanism to handle structural zeros in the IPF approach of `loglin()` and `loglm()` is to supply the argument `start`, giving a table conforming to the data, containing values of 0 in the locations of the zero cells, and non-zero elsewhere.[7] In the `glm()` approach, the argument `subset=Freq > 0` can be used to remove the cells with zero frequencies from the data, or else, zero frequencies can be set to `NA`. This usually provides the correct degrees of freedom, however some estimated coefficients may be infinite.

For a complete table, the residual degrees of freedom are determined as

$$df = \#\text{ of cells} - \#\text{ of fitted parameters}$$

For tables with structural zeros, an analogous general formula is

{eq:dfzeros}

$$df = (\#\text{ cells} - \#\text{ of parameters}) - (\#\text{ zero cells} - \#\text{ of NA parameters}) \tag{9.18}$$

where NA parameters refers to parameters that cannot be estimated due to zero marginal totals in the model formula.

In contrast, sampling zeros are often handled by some modification of the data frequencies to ensure all non-zero cells. Some suggestions are:

- Add a small positive quantity (0.5 is often recommended) to *every* cell in the contingency table (Goodman, 1970), as is often done in calculating empirical log odds (Example 10.9); this simple approach over-smooths the data for unsaturated models, and should be deprecated, although widely used in practice.
- Replace sampling zeros by some small number, typically 10^{-10} or smaller (Agresti, 1990).
- Add a small quantity, like 0.1, to *all* zero cells, sampling or structural (Evers and Namboordiri, 1977).

In complex, sparse tables, a sensitivity analysis, comparing different approaches can help determine if the substantive conclusions vary with the approach to zero cells.

{ex:health}

EXAMPLE 9.4: Health concerns of teenagers

Fienberg (1980, Table 8-3) presented a classic example of structural zeros in the analysis of the $4 \times 2 \times 2$ table shown in Table 9.2. The data come from a survey of health concerns among teenagers, originally from Brunswick (1971). Among the health concerns, the two zero entries for menstrual problems among males are clearly structural zeros and there therefore one structural zero in the concern by gender marginal table. As usual, we abbreviate the table variables concern, age, gender by their initial letters, C, A, G below.

The *Health* data is created as a frequency data frame as follows.

```
> Health <- expand.grid(concerns = c("sex", "menstrual",
+                                    "healthy", "nothing"),
+                       age      = c("12-15", "16-17"),
+                       gender   = c("M", "F"))
> Health$Freq <- c(4, 0, 42, 57, 2, 0, 7, 20,
+                  9, 4, 19, 71, 7, 8, 10, 21)
```

[7] If structural zeros are present, the calculation of degrees of freedom may not be correct. `loglm()` deducts one degree of freedom for each structural zero, but cannot make allowance for patterns of zeros based on the fitted margins that lead to gains in degrees of freedom due to smaller dimension in the parameter space. `loglin()` makes no such correction.

{tab:health}

Table 9.2: Results from a survey of teenagers, regarding their health concerns. Two cells with structural zeros are highlighted. *Source:* Fienberg (1980, Table 8-3)

Health	Gender:	Male		Female	
Concerns	Age:	12-15	16-17	12-15	16-17
sex, reproduction		4	2	9	7
menstrual problems		**0**	**0**	4	8
how healthy I am		42	7	19	10
nothing		57	20	71	21

In this form, we first use `glm()` to fit two small models, neither of which involves the $\{CG\}$ margin. Model `health.glm0` is the model of mutual independence, $[C][A][G]$. Model `health.glm1` is the model of joint independence, $[C][AG]$, allowing an association between age and gender, but neither with concern. As noted above, the argument `subset=(Freq>0)` is used to eliminate the structural zero cells.

```
> health.glm0 <-glm(Freq ~  concerns + age + gender, data=Health,
+                   subset=(Freq>0), family=poisson)
> health.glm1 <-glm(Freq ~  concerns + age * gender, data=Health,
+                   subset=(Freq>0), family=poisson)
```

Neither of these fits the data well. To conserve space, we show only the results of the G^2 tests for model fit.

```
> vcdExtra::LRstats(health.glm0, health.glm1)

Likelihood summary table:
              AIC BIC LR Chisq Df Pr(>Chisq)
health.glm0 100.7 105     27.7  8    0.00053 ***
health.glm1  99.9 104     24.9  7    0.00080 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

To see why, Figure 9.5 shows the mosaic display for model `health.glm1`, $[C][AG]$. Note that `mosaic()` takes care to make cells of zero frequency more visible by marking them with a small "o", as these have an area of zero.

```
> mosaic(health.glm1, ~concerns+age+gender, residuals_type="rstandard")
```

This suggests that there are important associations at least between concern and gender ($[CG]$) and between concern and age ($[CA]$). These are incorporated into the next model:

```
> health.glm2 <-glm(Freq ~  concerns*gender + concerns*age, data=Health,
+                   subset=(Freq>0), family=poisson)
> vcdExtra::LRstats(health.glm2)

Likelihood summary table:
             AIC  BIC LR Chisq Df Pr(>Chisq)
health.glm2 87.7 94.7     4.66  3        0.2
```

The degrees of freedom are correct here. Eqn. (9.18), with 2 zero cells and 1 NA parameter due to the zero in the $\{CG\}$ margin gives $df = (16 - 12) - (2 - 1) = 3$. The loss of one estimable parameter can be seen in the output from `summary`.

```
> summary(health.glm2)
```

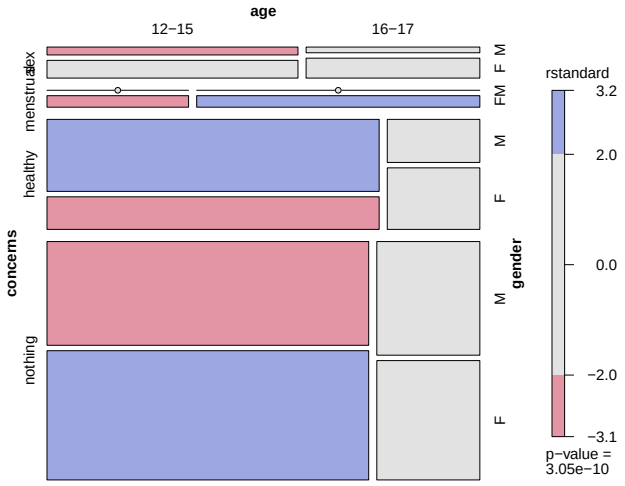

Figure 9.5: Mosaic display for the Health data, model `health.glm1`

{fig:health-mosaic}

```
Call:
glm(formula = Freq ~ concerns * gender + concerns * age, family = poisson,
    data = Health, subset = (Freq > 0))

Deviance Residuals:
     1       3       4       5       7       8       9      10      11      12
 0.236   0.585  -0.173  -0.300  -1.202   0.302  -0.149   0.000  -0.795   0.158
    13      14      15      16
 0.176   0.000   1.348  -0.282

Coefficients: (1 not defined because of singularities)
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)                 1.266      0.445    2.84   0.0045 **
concernsmenstrual          -0.860      0.586   -1.47   0.1425
concernshealthy             2.380      0.471    5.05  4.4e-07 ***
concernsnothing             2.800      0.462    6.07  1.3e-09 ***
genderF                     0.981      0.479    2.05   0.0405 *
age16-17                   -0.368      0.434   -0.85   0.3964
concernsmenstrual:genderF      NA         NA      NA       NA
concernshealthy:genderF    -1.505      0.533   -2.82   0.0047 **
concernsnothing:genderF    -0.803      0.503   -1.60   0.1105
concernsmenstrual:age16-17  1.061      0.750    1.41   0.1574
concernshealthy:age16-17   -0.910      0.513   -1.77   0.0761 .
concernsnothing:age16-17   -0.771      0.469   -1.64   0.1005
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 252.4670  on 13  degrees of freedom
Residual deviance:   4.6611  on  3  degrees of freedom
AIC: 87.66

Number of Fisher Scoring iterations: 4
```

In contrast, `loglm()` reports the degrees of freedom incorrectly for models containing zeros in any fitted margin. For use with `loglm()`, we convert it to a $4 \times 2 \times$ table.

```
> health.tab <- xtabs(Freq ~ concerns + age + gender, data = Health)
```

The same three models are fitted with `loglm()` as shown below. The locations of the positive

frequencies are marked in the array `nonzeros` and supplied as the value of the `start` argument.

```
> nonzeros <- ifelse(health.tab>0, 1, 0)
> health.loglm0 <- loglm(~ concerns + age + gender,
+                data = health.tab, start = nonzeros)
> health.loglm1 <- loglm(~ concerns + age * gender,
+                data = health.tab, start = nonzeros)
> # df is wrong
> health.loglm2 <- loglm(~ concerns*gender + concerns*age,
+                data = health.tab, start = nonzeros)
> LRstats(health.loglm0, health.loglm1, health.loglm2)

Likelihood summary table:
                AIC BIC LR Chisq Df Pr(>Chisq)
health.loglm0 104.7 111    27.74  8    0.00053 ***
health.loglm1 103.9 111    24.89  7    0.00080 ***
health.loglm2  93.7 104     4.66  2    0.09724 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The results agree with those of `glm()`, except for the degrees of freedom for the last model. △

9.6 Chapter summary

{sec:loglin-summary}

- Loglinear models provide a comprehensive scheme to describe and understand the associations among two or more categorical variables. It is helpful to think of these as discrete analogs of ANOVA models, or of regression models, where the log of cell frequency is modelled as a linear function of predictors.
- Loglinear models typically make no distinction between response and explanatory variables. When one variable *is* a response, however, any logit model for that response has an equivalent loglinear model. The logit form is usually simpler to formulate and test, and plots of the observed and fitted logits are easier to interpret.
- 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.
- Cells with zero frequencies create problems for estimation and testing hypotheses in loglinear models. Different methods are available to handle *structural zeros* and *sampling zeros*.

9.7 Lab exercises

{lab:9.1}
{sec:loglin-lab}

Exercise 9.1 Consider the data set *DaytonSurvey* (described in Example 2.6), giving results of a survey of use of alcohol (A), cigarettes (C) and marijuana (M) among high school seniors. For this exercise, ignore the variables `sex` and `race`, by working with the marginal table `Dayton.ACM`, a $2 \times 2 \times 2$ table in frequency data frame form.

```
> Dayton.ACM <- aggregate(Freq ~ cigarette + alcohol + marijuana,
+                         data=DaytonSurvey, FUN=sum)
```

(a) Use `loglm()` to fit the model of mutual independence, [A][C][M]

(b) Prepare mosaic display(s) for associations among these variables. Give a verbal description of the association between cigarette and alcohol use.
(c) Use `fourfold()` to produce fourfold plots for each pair of variables, AC, AM and CM, stratified by the remaining one. Describe these associations verbally.

{lab:9.2}

Exercise 9.2 Continue the analysis of the *DaytonSurvey* data by fitting the following models:

(a) Joint independence, [AC][M]
(b) Conditional independence, [AM][CM]
(c) Homogeneous association, [AC][AM][CM]
(d) Prepare a table giving the goodness-of-fit tests for these models, as well as the model of mutual independence, [A][C][M] , and the saturated model, [ACM] . *Hint*: `anova()` and `LRstats()` are useful here. Which model appears to give the most reasonable fit?

TODO: Add more exercises

```
> #detach(package:corrplot)
> detach(package:VGAM)
> #detach(package:logmult)
> #remove(list=objects(pattern="\\.tab|\\.df|\\.fit"))
> .locals$ch08 <- setdiff(ls(), .globals)
> #.locals$ch08
> remove(list=.locals$ch08[sapply(.locals$ch08,function(n){!is.function(get(n))})])
```

References

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

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

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principal. In B. N. Petrov and F. Czaki, eds., *Proceedings of the 2nd International Symposium on Information*. Budapest: Akademiai Kiado.

Birch, M. W. (1963). Maximum likelihood in three-way contingency tables. *Journal of the Royal Statistical Society, Series B*, 25, 220–233.

Bishop, Y. M. M., Fienberg, S. E., and Holland, P. W. (1975). *Discrete Multivariate Analysis: Theory and Practice*. Cambridge, MA: MIT Press.

Brunswick, A. F. (1971). Adolescent health, sex, and fertility. *American Journal of Public Health*, 61(4), 711–729.

Evers, M. and Namboordiri, N. K. (1977). A Monte Carlo assessment of the stability of log-linear estimates in small samples. In *Proceedings of the Social Statistics Section*. Alexandria, VA: American Statistical Association.

Fienberg, S. E. (1980). *The Analysis of Cross-Classified Categorical Data*. Cambridge, MA: MIT Press, 2nd edn.

Fienberg, S. E. and Rinaldo, A. (2007). Three centuries of categorical data analysis: Log-linear models and maximum likelihood estimation. *Journal of Statistical Planning and Inference*, 137(11), 3430–3445.

Goodman, L. A. (1970). The multivariate analysis of qualitative data: Interactions among multiple classifications. *Journal of the American Statistical Association*, 65, 226–256.

Goodman, L. A. (1971). The analysis of multidimensional contingency tables: Stepwise procedures and direct estimates for building models for multiple classifications. *Technometrics*, 13, 33–61.

Haberman, S. J. (1972). Statistical algorithms: Algorithm AS 51: Log-linear fit for contingency tables. *Applied Statistics*, 21(2), 218–225.

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

Haberman, S. J. (1974). *The Analysis of Frequency Data*. Chicago: University of Chicago Press.

Lindsey, J. K. (1995). *Modelling Frequency and Count Data*. Oxford, UK: Oxford University Press.

Raftery, A. E. (1986). Choosing models for cross-classifications. *American Sociological Review*, 51, 146–146.

Schwartz, G. (1978). Estimating the dimensions of a model. *Annals of Statistics*, 6, 461–464.