Chapter 6

Generalized Linear Models

In Chapters 2 and 4 we studied how to estimate simple probability densities over a single random variable—that is, densities of the form $P(Y)$. In this chapter we move on to the problem of estimating *conditional* densities—that is, densities of the form $P(Y|X)$. Logically speaking, it would be possible to deal with this problem simply by assuming that Y may have an arbitrarily different distribution for each possible value of X, and to use techniques we've covered already to estimate a different density $P(Y|X = x_i)$ for each possible value x_i of X. However, this approach would miss the fact that X may have a *systematic* effect on Y; missing this fact when it is true would make it much more difficult to estimate the conditional distribution. Here, we cover a popular family of conditional probability distributions known as GENERALIZED LINEAR MODELS. These models can be used for a wide range of data types and have attractive computational properties.

6.1 The form of the generalized linear model

Suppose we are interested in modeling the distribution of Y conditional on a number of random variables $X_1, \ldots, X_n$. Generalized linear models are a framework for modeling this type of conditional distribution $P(Y|X_1, \ldots, X_n)$ subject to four key assumptions:

1. The influences of the $\{X_i\}$ variables on Y can be summarized into an intermediate form, the LINEAR PREDICTOR η;
2. η is a linear combination of the $\{X_i\}$;
3. There is a smooth, invertible function l mapping η to the expected value μ of Y;
4. The distribution $P(Y = y; \mu)$ of Y around μ is a member of a certain class of noise functions and is not otherwise sensitive to the X_i variables.[1]

Assumptions 1 through 3 can be expressed by the following two equations:

[1]The class of allowable noise functions is described in Section **??**.

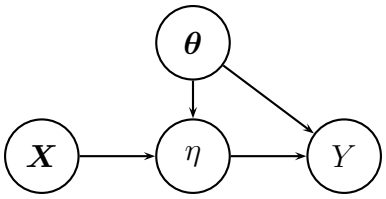


Figure 6.1: A graphical depiction of the generalized linear model. The influence of the conditioning variables $\boldsymbol{X}$ on the response Y is completely mediated by the linear predictor η.

$$\eta = \alpha + \beta_1 X_1 + \cdots + \beta_n X_n \qquad \text{(linear predictor)} \tag{6.1}$$

$$\eta = l(\mu) \qquad \text{(link function)} \tag{6.2}$$

Assumption 4 implies conditional independence of Y from the $\{X_i\}$ variables given η.

Various choices of $l(\mu)$ and $P(Y = y; \mu)$ give us different classes of models appropriate for different types of data. In all cases, we can estimate the parameters of the models using any of likelihood-based techniques discussed in Chapter 4. We cover three common classes of such models in this chapter: linear models, logit (logistic) models, and log-linear models.

6.2 Linear models and linear regression

We can obtain the classic LINEAR MODEL by chooosing the identity link function

$$\eta = l(\mu) = \mu$$

and a noise function that adds noise

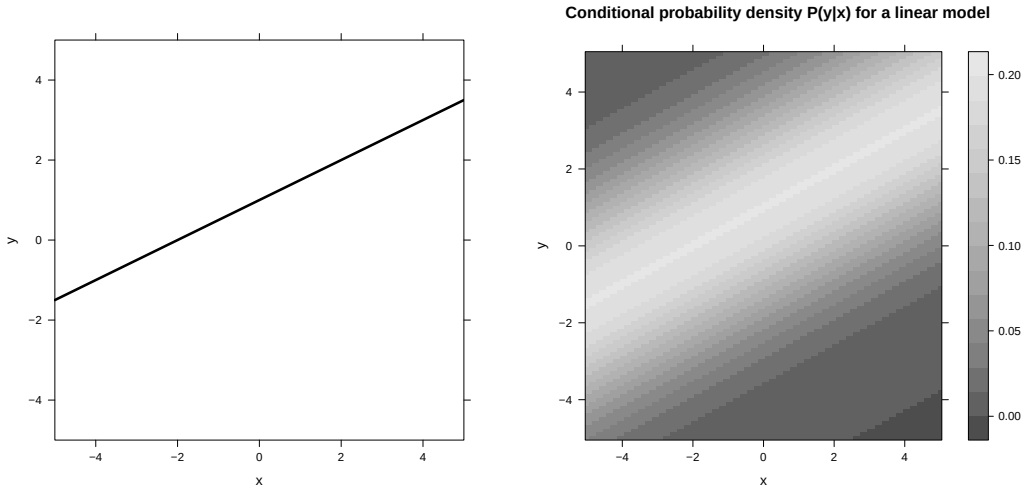
$$\epsilon \sim N(0, \sigma^2)$$

to the mean μ. Substituting these in to Equations (6.1) and 6.2, we can write Y directly as a function of $\{X_i\}$ as follows:

$$Y = \overbrace{\alpha + \beta_1 X_1 + \cdots + \beta_n X_n}^{\text{Predicted Mean}} + \overbrace{\epsilon}^{\text{Noise}\sim N(0,\sigma^2)} \tag{6.3}$$

We can also write this whole thing in more compressed form as $Y \sim N(\alpha \sum_i \beta_i X_i, \sigma^2)$.

To gain intuitions for how this model places a conditional probability density on Y, we can visualize this probability density for a single independent variable X, as in Figure 6.2—lighter means more probable. Each vertical slice of constant $X = x$ represents a conditional



(a) Predicted outcome

(b) Probability density

Figure 6.2: A plot of the probability density on the outcome of the Y random variable given the X random variable; in this case we have $\eta = \frac{1}{2}X$ and $\sigma^2 = 4$.

distribution $P(Y|x)$. If you imagine a vertical line extending through the plot at $X = 0$, you will see that the plot along this line is lightest in color at $Y = -1 = \alpha$. This is the point at which ϵ takes its most probable value, 0. For this reason, α is also called the INTERCEPT parameter of the model, and the β_i are called the SLOPE parameters.

6.2.1 Fitting a linear model

The process of estimating the parameters α and β_i of a linear model on the basis of some data (also called FITTING the model to the data) is called LINEAR REGRESSION. There are many techniques for parameter estimation in linear regression; here we will cover the method of maximum likelihood and also Bayesian linear regression.

Maximum-likelihood linear regression

Before we talk about exactly what the maximum-likelihood estimate looks like, we'll introduce some useful terminology. Suppose that we have chosen model parameters $\hat{\alpha}$ and $\{\hat{\beta}_i\}$. This means that for each point $\langle x_j, y_j \rangle$ in our dataset $\boldsymbol{y}$, we can construct a PREDICTED VALUE for $\hat{y}_j$ as follows:

$$\hat{y}_j = \hat{\alpha} + \hat{\beta}_1 x_{j1} + \ldots \hat{\beta}_n x_{jn}$$

where x_{ji} is the value of the i-th predictor variable for the j-th data point. This predicted value is both the expectated value and the modal value of Y_j due to the Gaussian-noise assumption of linear regression. We define the RESIDUAL of the j-th data point simply as

$$y_j - \hat{y}_j$$

—that is, the amount by which our model's prediction missed the observed value.

It turns out that for linear models with a normally-distributed error term ϵ, the log-likelihood of the model parameters with respect to $\boldsymbol{y}$ is proportional to the sum of the squared residuals. This means that the maximum-likelihood estimate of the parameters is also the estimate that minimizes the the sum of the squared residuals. You will often see description of regression models being fit using LEAST-SQUARES estimation. Whenever you see this, recognize that this is equivalent to maximum-likelihood estimation under the assumption that residual error is normally-distributed.

6.2.2 Fitting a linear model: case study

The dataset `english` contains reaction times for lexical decision and naming of isolated English words, as well as written frequencies for those words. Reaction times are measured in milliseconds, and word frequencies are measured in appearances in a 17.9-million word written corpus. (All these variables are recorded in log-space) It is well-established that words of high textual frequency are generally responded to more quickly than words of low textual frequency. Let us consider a linear model in which reaction time RT depends on the log-frequency, F, of the word:

$$RT = \alpha + \beta_F F + \epsilon \tag{6.4}$$

troduction to tion 11.1

This linear model corresponds to a FORMULA in `R`, which can be specified in either of the following ways:

```
RT ~ F
RT ~ 1 + F
```

The `1` in the latter formula refers to the intercept of the model; the presence of an intercept is implicit in the first formula.

The result of the linear regression is an intercept $\alpha = 843.58$ and a slope $\beta_F = -29.76$. The `WrittenFrequency` variable is in natural log-space, so the slope can be interpreted as saying that if two words differ in frequency by a factor of $e \approx 2.718$, then on average the more frequent word will be recognized as a word of English 26.97 milliseconds faster than the less frequent word. The intercept, 843.58, is the predicted reaction time for a word whose log-frequency is 0—that is, a word occurring only once in the corpus.

6.2.3 Conceptual underpinnings of best linear fit

Let us now break down how the model goes about fitting data in a simple example.

Suppose we have only three observations of log-frequency/RT pairs:

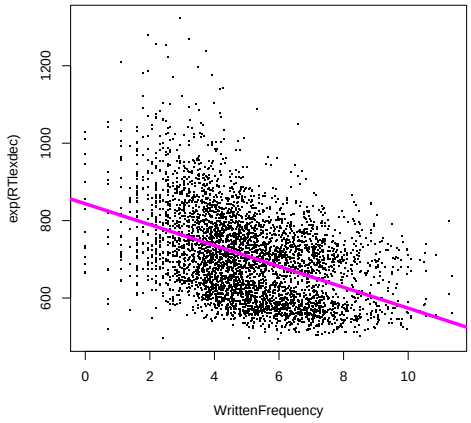


Figure 6.3: Lexical decision reaction times as a function of word frequency

$$\langle 4, 800 \rangle$$
$$\langle 6, 775 \rangle$$
$$\langle 8, 700 \rangle$$

Let use consider four possible parameter estimates for these data points. Three estimates will draw a line through two of the points and miss the third; the last estimate will draw a line that misses but is reasonably close to all the points.

First consider the solid black line, which has intercept 910 and slope -25. It predicts the following values, missing all three points:

x	$\hat{y}$	Residual $(\hat{y} - y)$
4	810	-10
6	760	15
8	710	-10

and the sum of its squared residuals is 425. Each of the other three lines has only one non-zero residual, but that residual is much larger, and in all three case, the sum of squared residuals is larger than for the solid black line. This means that the likelihood of the parameter values $\alpha = 910, \beta_F = -25$ is higher than the likelihood of the parameters corresponding to any of the other lines.

What is the MLE for α, β_F with respect to these three data points, and what are the residuals for the MLE?

Results of a linear fit (almost every statistical software package supports linear regression) indicate that the MLE is $\alpha = 908\frac{1}{3}, \beta = -25$. Thus the MLE has the same slope as the solid

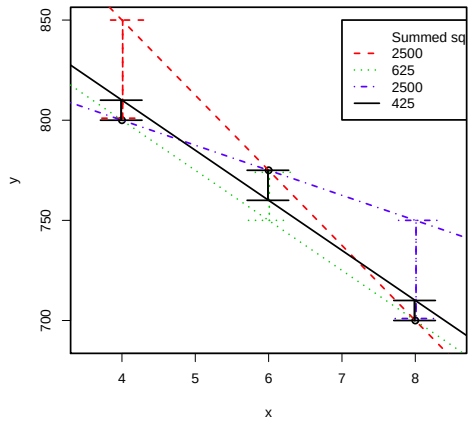


Figure 6.4: Linear regression with three points

black line in Figure 6.4, but the intercept is slightly lower. The sum of squared residuals is slightly better too.

Take-home point: for linear regression, getting everything wrong by a little bit is better than getting a few things wrong by a lot.

6.3 Handling multiple predictors

In many cases, we are interested in simultaneously investigating the linear influence of two or more predictor variables on a single response. We'll discuss two methods of doing this: RESIDUALIZING and MULTIPLE LINEAR REGRESSION.

As a case study, consider naming reaction times from the `english` dataset, and now imagine that we're interested in the influence of orthographic neighbors. (An orthographic neighbor of a word w is one that shares most of its letters with w; for example, *cat* has several orthographic neighbors including *mat* and *rat*.) The `english` dataset summarizes this information in the `Ncount` variable, which measures ORTHOGRAPHIC NEIGHBORHOOD DENSITY as (I believe) the number of maximally close orthographic neighbors that the word has. How can we investigate the role of orthographic neighborhood while simultaneously taking into account the role of word frequency?

6.3.1 Residualizing

One approach would be a two-step process: first, construct a linear regression with frequency as the predictor and RT as the response. (This is commonly called "regressing RT against frequency".) Second, construct a new linear regression with neighborhood density as the predictor the *residuals from the first regression* as the response. The transformation of a raw RT into the residual from a linear regression is called RESIDUALIZATION. Figure 6.5):

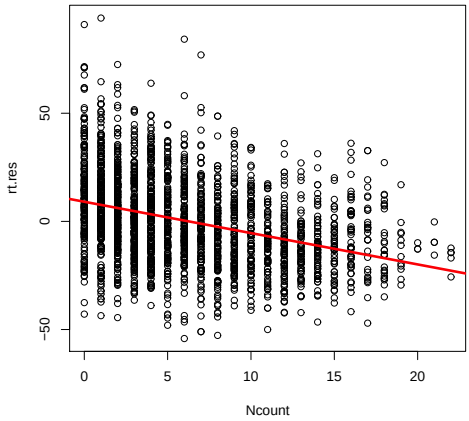


Figure 6.5: Plot of frequency-residualized word naming times and linear regression against neighborhood density

```
> english.young <- subset(english,AgeSubject=="young")
> attach(english.young)
> rt.freq.lm <- lm(exp(RTnaming) ~ WrittenFrequency)
> rt.freq.lm

Call:
lm(formula = exp(RTnaming) ~ WrittenFrequency)

Coefficients:
     (Intercept)  WrittenFrequency
         486.506            -3.307

> rt.res <- resid(rt.freq.lm)
> rt.ncount.lm <- lm(rt.res ~ Ncount)
> plot(Ncount, rt.res)
> abline(rt.ncount.lm,col=2,lwd=3)
> detach()
> rt.ncount.lm

Call:
lm(formula = rt.res ~ Ncount)

Coefficients:
(Intercept)       Ncount
      9.080       -1.449
```

Even after linear effects of frequency have been accounted for by removing them from the RT measure, neighborhood density still has some effect – words with higher neighborhood density are named more quickly.

6.3.2 Multiple linear regression

The alternative is to build a single linear model with more than one predictor. A linear model predicting naming reaction time on the basis of both frequency F and neighborhood density D would look like this:

$$RT = \alpha + \beta_F F + \beta_D D + \epsilon$$

and the corresponding `R` formula would be either of the following:

```
RT ~ F + D
RT ~ 1 + F + D
```

Plugging this in gives us the following results:

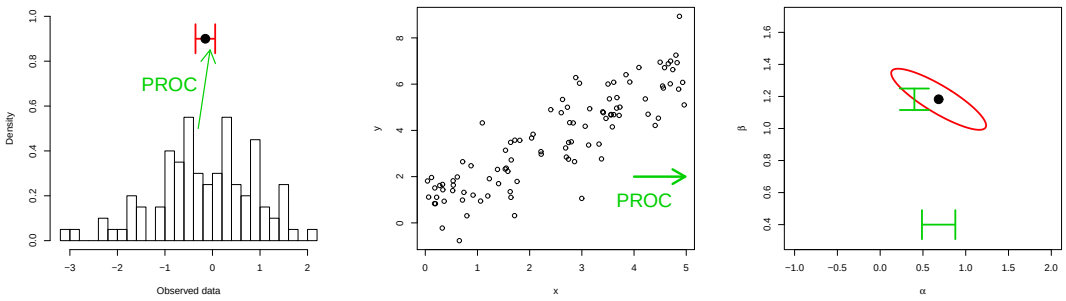
```
> rt.both.lm <- lm(exp(RTnaming) ~ WrittenFrequency + Ncount, data=english.young)
> rt.both.lm

Call:
lm(formula = exp(RTnaming) ~ WrittenFrequency + Ncount, data = english.young)

Coefficients:
     (Intercept)  WrittenFrequency            Ncount
         493.638            -2.899            -1.465
```

Note that the results are qualitatively similar but quantitatively different than for the residualization approach: larger effect sizes have been estimated for both `WrittenFrequency` and `Ncount`.

6.4 Confidence intervals and hypothesis testing for linear regression

Just as there was a close connection between hypothesis testing with the one-sample t-test and a confidence interval for the mean of a sample, there is a close connection between hypothesis testing and confidence intervals for the parameters of a linear model. We'll start by explaining the confidence interval as the fundamental idea, and see how this leads to hypothesis tests.

Figure 6.6 illustrates the procedures by which confidence intervals are constructed for a sample mean (one parameter) and for the intercept and slope of a linear regression with one



(a) Sample mean (b) Data (c) Confidence region

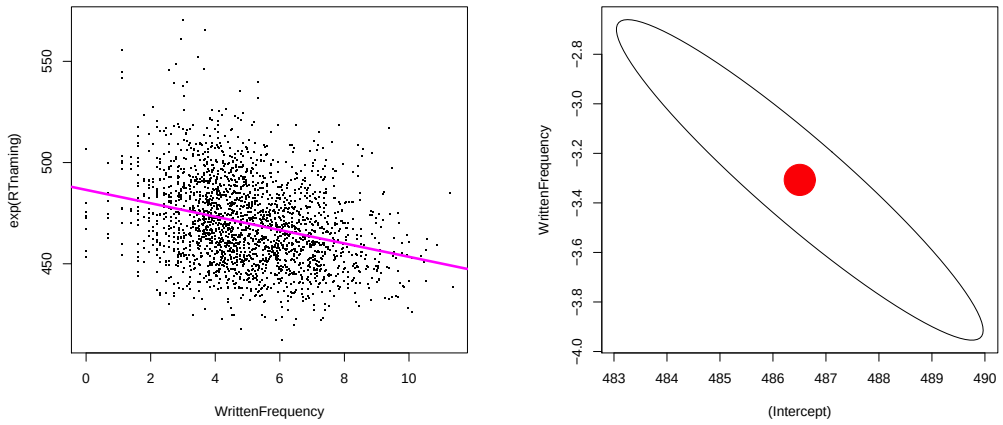
Figure 6.6: The confidence-region construction procedure for (a) sample means and (b-c) parameters of a linear model. The black dots are the maximum-likelihood estimates, around which the confidence regions are centered.

predictor. In both cases, a dataset $\boldsymbol{y}$ is obtained, and a fixed procedure is used to construct boundaries of a CONFIDENCE REGION from $\boldsymbol{y}$. In the case of the sample mean, the "region" is in one-dimensional space so it is an interval. In the case of a linear regression model, the region is in two-dimensional space, and looks like an ellipse. The size and shape of the ellipse are determined by the VARIANCE-COVARIANCE MATRIX of the linear predictors, and are determined using the fact that the joint distribution of the estimated model parameters is multivariate-normal distributed (Section 3.5). If we collapse the ellipse down to only one dimension (corresponding to one of the linear model's parameters), we have a confidence interval on that parameter; this one-dimensional confidence interval is t distributed with $N - k$ degrees of freedom (Section B.5), where N is the number of observations and k is the number of parameters in the linear model.[2]

We illustrate this in Figure 6.7 for the linear regression model of frequency against word naming latency.The model is quite certain about the parameter estimates; however, note that there is a correlation between the parameter estimates. According to the analysis, if we reran this regression many times by repeatedly drawing data from the same population and estimating parameters, whenever the resulting intercept (i.e., average predicted RT for the rarest class of word) is higher, the facilitative effect of written frequency would tend to be larger, and vice versa. This is intuitively sensible because the most important thing for the regression is where it passes through the centroid of the data; so that if the intercept drops a bit, the slope has to rise to compensate.

Perhaps a more interesting example is looking at the confidence region obtained for the parameters of two predictors. In the literature on word recognition, for example, there has been some discussion over whether word frequency or word familiarity drives variability in average word-recognition time (or whether both have independent effects). Because

[2]Formally this corresponds to marginalizing over the estimates of the other parameters that you're collapsing over.



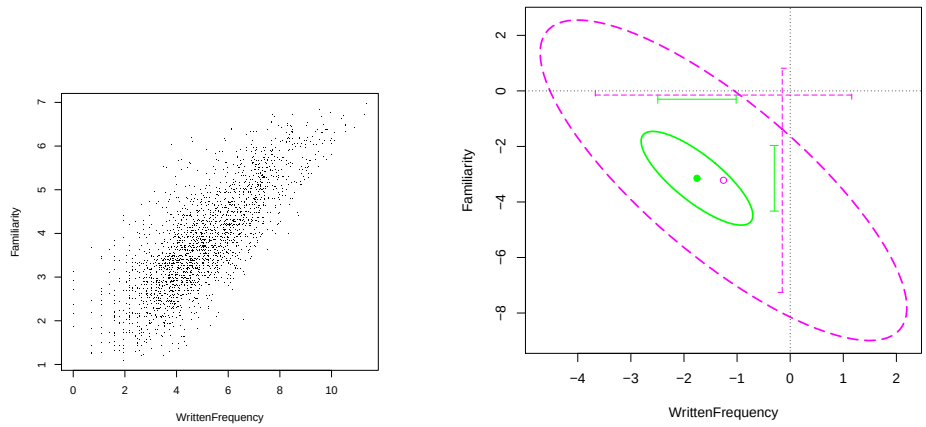
(a) Data and best linear fit

(b) Confidence ellipse

Figure 6.7: Confidence ellipse for parameters of regression of word naming latency against written frequency

subjective ratings of word familiarity are strongly correlated with word frequency, it is empirically difficult to disentangle the two. Figure 6.8a shows a scatterplot of word familiarity against word frequency ratings for 2,197 English nouns and verbs (Spieler and Balota, 1997; Balota and Spieler, 1998); the empirical correlation is 0.79. The naming study carried out by Spieler and Balota (1997) was very large, and they obtained naming times for each of these words from 31 undergraduate native English speakers. A multiple linear regression analysis with frequency and familiarity as predictors puts 95% confidence intervals for their slopes in the linear model at [-2.49,-1.02] and [-4.33,-1.96] respectively. Hence we can conclude that each of frequency and familiarity contribute independently in determining naming time (insofar as the measurements of frequency and familiarity themselves are accurately measured).

However, this was a very large study, and one might reasonably ask what conclusions one could draw from a much smaller study. The same multiple linear regression based on a random subsample of 200 of the words from Spieler and Balota's study gives us confidence intervals for the effects of word frequency and familiarity on naming time of [-3.67,1.16] and [-7.26,0.82]. With this smaller dataset, we cannot confidently conclude that either predictor is independently a determinant of naming time. Yet this negative result conceals an important conclusion that we can still draw. Figure 6.8 plots confidence *regions* for the two model parameters, as well as confidence intervals for each individual parameter, in models of the full dataset (solid green lines) and the reduced, 200-word dataset (dashed magenta lines). Although the reduced-dataset confidence region shows that we cannot be confident that either parameter is negative (i.e. that it has a facilitatory effect on naming time), we can be quite confident that it is not the case that *both* parameters are non-negative: the ellipse does not come close to encompassing the origin. That is, we can be confident that *some combination* of word frequency and familiarity has a reliable influence on naming time. We



(a) Scatterplot of word frequency & familiarity rating

(b) Confidence region for influence on word naming time

Figure 6.8: Confidence region on written word frequency and word familiarity for full dataset of Spieler and Balota (1997), and reduced subset of 200 items

return to this point in Section 6.5.2 when we cover how to compare models differing by more than one parameter through the F test.

6.5 Hypothesis testing in multiple linear regression

An extremely common use of linear models is in testing hypotheses regarding whether one or more predictor variables have a reliable influence on a continuously-distributed response. Examples of such use in the study of language might include but are not limited to:

- Does a word's frequency reliably influence how rapidly it is recognized, spoken, or read?
- Are words of different parts of speech recognized, spoken, or read at different rates above and beyond the effects of word frequency (and perhaps other properties such as word length)?
- Does the violation of a given syntactic constraint affect a native speaker's rating of the felicity of sentences with the violation (as compared to sentences without the violation)?
- Does the context in which a sound is uttered reliably influence one of its phonetic properties (such as voice-onset time for stops, or format frequency for vowels)?

All of these questions may be addressed within the Neyman-Pearson paradigm frequentist hypothesis-testing paradigm introduced in Section 5.4. Recall that the Neyman-Pearson paradigm involves specifying a null hypothesis H_0 and determining whether to reject it in

favor of a more general and complex hypothesis H_A. In many cases, we are interested in comparing whether a more complex linear regression is justified by the data over a simpler regression. Under these circumstances, we can take the simpler model M_0 as the null hypothesis, and the more complex model M_A as the alternative hypothesis. There are two statistical tests that you will generally encounter for this purpose: one based on the t statistic (which we already saw in Section 5.3) and another, the F test, which is based on what is called the F statistic. However, the former is effectively a special case of the latter, so here we'll look at how to use the F statistic for hypothesis testing with linear models; then we'll briefly cover the use of the t statistic for hypothesis testing as well.

6.5.1 Decomposition of variance

The F test takes advantage of a beautiful property of linear models to compare M_0 and M_A: the DECOMPOSITION OF VARIANCE. Recall that the VARIANCE of a sample is simply the sum of the square deviations from the mean:

$$\text{Var}(\boldsymbol{y}) = \sum_j (y_j - \bar{y})^2 \tag{6.5}$$

where $\bar{y}$ is the mean of the sample $\boldsymbol{y}$. For any model M that predicts values $\hat{y}_j$ for the data, the RESIDUAL VARIANCE or RESIDUAL SUM OF SQUARES of M is quantified in exactly the same way:

$$RSS_M(\boldsymbol{y}) = \sum_j (y_j - \hat{y}_j)^2 \tag{6.6}$$

A beautiful and extraordinarily useful property of linear models is that the sample variance can be split apart, or DECOMPOSED, into (a) the component that is explained by M, and (b) the component that remains unexplained by M. This can be written as follows (see Exercise 6.5):

$$\text{Var}(\boldsymbol{y}) = \overbrace{\sum_j (y_j - \hat{y}_j)^2}^{\text{explained by } M} + \overbrace{\sum_j (\hat{y}_j - \bar{y})^2}^{RSS_M(\boldsymbol{y}),\text{unexplained}} \tag{6.7}$$

Furthermore, if two models are nested (i.e., one is a special case of the other), then the variance can be futher subdivided among those two models. Figure 6.9 shows the partitioning of variance for two nested models.

6.5.2 Comparing linear models: the F test statistic

The F test is widely used for comparison of linear models, and forms the foundation for many analytic techniques including the analysis of variance (ANOVA).

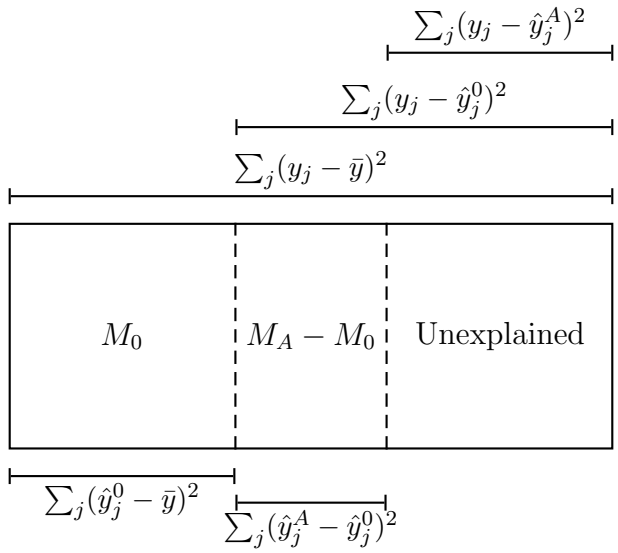


Figure 6.9: The partitioning of residual variance in linear models. Symbols in the box denote the variance explained by each model; the sums outside the box quantify the variance in each combination of sub-boxes.

Recall that our starting assumption in this section was that we had two linear models: a more general model M_A, and a special case M_0—that is, for any instance of M_0 we can achieve an instance of M_A with the same distribution over Y by setting the parameters appropriately. In this situation we say that M_0 is NESTED inside M_A. Once we have found maximum-likelihood estimates of M_0 and M_A, let us denote their predicted values for the j-th observation as $\hat{y}_j^0$ and $\hat{y}_j^A$ respectively. The sample variance unexplained by M_0 and M_A respectively is

$$\sum_j (\hat{y}_j - \hat{y}_j^0)^2 \ (M_0) \tag{6.8}$$

$$\sum_j (\hat{y}_j - \hat{y}_j^A)^2 (M_A) \tag{6.9}$$

so the additional variance explained by M_A above and beyond M_0 is

$$\sum_j (\hat{y}_j - \hat{y}_j^A)^2 - \sum_j (\hat{y}_j - \hat{y}_j^0)^2 = \sum_j (\hat{y}_j^A - \hat{y}_j^0)^2 \tag{6.10}$$

Let us suppose that M_0 has k_0 parameters, M_A has k_A parameters, and we have n observations. It turns out that the quantities in Equations (6.8), (6.9), and (6.10) are distributed proportional to χ^2 random variables with $n - k_0$, $n - k_A$, and $k_A - k_0$ degrees of freedom respectively (Section B.4). These quantities are also independent of one another; and, crucially, *if M_0 is the true model then the proportionality constants for Equations (6.9) and (6.10) are the same.*

These facts form the basis for a frequentist test of the null hypothesis $H_0 : M_0$ is correct, based on the F STATISTIC, defined below:

$$F = \frac{\sum_j (\hat{y}_j^A - \hat{y}_j^0)^2/(k_A - k_0)}{\sum_j (y_j - \hat{y}_j^A)^2/(n - k_A)} \tag{6.11}$$

Under H_0, the F statistic follows the F distribution (Section B.6)—which is parameterized by *two* degrees of freedom—with $(k_A - k_0, n - k_A)$ degrees of freedom. This follows from the fact that under H_0, we have

$$\sum_j (\hat{y}_j^A - \hat{y}_j^0)^2/(k_A - k_0) \sim C\chi^2_{k_A - k_0}; \qquad \sum_j (\hat{y}_j - \hat{y}_j^A)^2/(n - k_A) \sim C\chi^2_{n-k_A}$$

for some proportionality constant C; when the ratio of the two is taken, the two Cs cancel, leaving us with an F-distributed random variable.

Because of the decomposition of variance, the F statistic can also be written as follows:

$$F = \frac{\left[\sum_j (y_j - \hat{y}_j^0)^2 - \sum_j (y_j - \hat{y}_j^A)^2\right]/(k_A - k_0)}{\left[\sum_j (y_j - \hat{y}_j^A)^2\right]/(n - k_A)}$$

which underscores that the F statistic compares the amount of regularity in the observed data explained by M_A beyond that explained by M_0 (the numerator) with the amount of regularity unexplained by M_A (the denominator). The numerator and denominator are often called MEAN SQUARES.

Because of the decomposition of variance, the F test can be given a straightforward geometric interpretation. Take a look at the labels on the boxes in Figure 6.9 and convince yourself that the sums in the numerator and the denominator of the F statistic correspond respectively to the boxes $M_A - M_0$ and Unexplained. Thus, using the F statistic for hypothesis testing is often referred to as evaluation of the RATIO OF MEAN SQUARES.

Because of its importance for frequentist statistical inference in linear models, the F distribution has been worked out in detail and is accessible in most statistical software packages.

6.5.3 Model comparison: case study

We can bring the F test to bear in our investigation of the relative contributions of word frequency and familiarity on word naming latency; we will focus on analysis of the reduced 200-item dataset. First let us consider a test in which the null hypothesis H_0 is that only word frequency has a reliable effect, and the alternative hypothesis H_A is that both word frequency (or "Freq" for short) and familiarity ("Fam") have reliable effects. H_0 corresponds

to the model $RT \sim \mathcal{N}(\alpha + \beta_{\text{Freq}}\text{Freq}, \sigma^2)$; H_A corresponds to the model $RT \sim \mathcal{N}(\alpha + \beta_{\text{Freq}}\text{Freq} + \beta_{\text{Fam}}\text{Fam}, \sigma^2)$. After obtaining maximum-likelihood fits of both models, we can compute the residual sums of squares $\sum_j (\hat{y}_j - y_j)^2$ for each model; these turn out to be 76824.28 for M_0 and 75871.58 for M_A. M_0 has two parameters, M_A has three, and we have 200 observations; hence $k_A - k_0 = 1$ and $N - k_A = 197$. The F statistic for our hypothesis test is thus

$$\begin{aligned} F &= \frac{[76824.28 - 75871.58]/1}{[75871.58]/197} \\ &= 2.47 \end{aligned}$$

with (1,197) degrees of freedom. Consulting the cumulative distribution function for the F statistic we obtain a p-value of 0.12.

We can also apply the F test for comparisons of models differing in multiple parameters, however. For example, let M_0' be a model in which neither word frequency nor familiarity has an effect on naming time. The residual variance in this model is the entire sample variance, or 82270.49. For a comparison between M_0' and M_A we obtain an F statistic of

$$\begin{aligned} F &= \frac{[82270.49 - 75871.58]/2}{[75871.58]/197} \\ &= 8.31 \end{aligned}$$

with (2,197) degrees of freedom. The corresponding p-value is 0.00034, indicating that our data are extremely unlikely under M_0' and that M_A is far preferable. Thus, although we could not adjudicate between word frequency and familiarity with this smaller dataset, we could say confidently that *some combination of the two* has a reliable effect on word naming time.

Another widely used test for the null hypothesis that within a k-parameter model, a *single* parameter β_i is 0. This hypothesis can be tested through a t-test where the t statistic is the ratio of the parameter estimate, $\hat{\beta}_i$ to the standard error of the estimate, $\text{SE}(\hat{\beta}_i)$. For a dataset with N observations, this t-test has $N-k$ degrees of freedom. However, a F statistic with (1,m) has the same distribution as the square of a t statistic with m degrees of freedom. For this reason, the t-test for linear models can be seen as a special case of the more general F test; the latter can be applied to compare nested linear models differing in any number of parameters.

6.6 Analysis of Variance

Recall that we just covered linear models, which are conditional probability distributions of the form

$$P(Y|X) = \alpha + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \epsilon \tag{6.12}$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$. We saw how this paradigm can be put to use for modeling the predictive relationship of continuous variables, such as word frequency, familiarity, and neighborhood density, on reaction times in word recognition experiments.

In many cases, however, the predictors of interest are not continuous. For example, for the `english` dataset in `languageR` we might be interested in how naming times are influenced by the type of the initial phoneme of the word. This information is coded by the `Frication` variable of the dataset, and has the following categories:

`burst`	the word starts with a burst consonant
`frication`	the word starts with a fricative consonant
`long`	the word starts with a long vowel
`short`	the word starts with a short vowel

It is not obvious how these categories might be meaningfully arranged on the real number line. Rather, we would simply like to investigate the possibility that the mean naming time differs as a function of initial phoneme type.

The most widespread technique used to investigate this type of question is the ANALYSIS OF VARIANCE (often abbreviated ANOVA). Although many books go into painstaking detail covering different instances of ANOVA, you can gain a firm foundational understanding of the core part of the method by thinking of it as a special case of multiple linear regression.

6.6.1 Dummy variables

Let us take the example above, where `Frication` is a categorical predictor. Categorical predictors are often called FACTORS, and the values they take are often called LEVELS. (This is also the nomenclature used in `R`.) In order to allow for the possibility that each level of the factor could have arbitrarily different effects on mean naming latency, we can create DUMMY PREDICTOR VARIABLES, one per level of the factor:

Level of `Frication`	X_1	X_2	X_3	X_4
`burst`	1	0	0	0
`frication`	0	1	0	0
`long`	0	0	1	0
`short`	0	0	0	1

(Variables such as these which are 0 unless a special condition holds, in which case they are 1, are often referred to as INDICATOR VARIABLES). We then construct a standard linear model with predictors X_1 through X_4:

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \epsilon \tag{6.13}$$

When we combine the dummy predictor variables with the linear model in (9.7), we get the following equations for each level of `Frication`:

Level of `Frication`	Linear model
`burst`	$Y = \alpha + \beta_1 + \epsilon$
`frication`	$Y = \alpha + \beta_2 + \epsilon$
`long`	$Y = \alpha + \beta_3 + \epsilon$
`short`	$Y = \alpha + \beta_4 + \epsilon$

This linear model thus allows us to code a different predicted mean (and most-likely predicted value) for each level of the predictor, by choosing different values of α and β_i.

However, it should be clear from the table above that only four distinct means can be predicted in this linear model—one for each level of `Frication`. We don't need five parameters (one for α and four for the β_i) to encode four means; one of the parameters is redundant. This is problematic when fitting the model because it means that there is no unique maximum-likelihood estimate.[3] To eliminate this redundancy, we arbitrarily choose one level of the factor as the BASELINE level, and we don't introduce a dummy predictor for the baseline level. If we choose `burst` as the baseline level,[4] then we can eliminate X_4, and make X_1, X_2, X_3 dummy indicator variables for `frication`, `long`, and `short` respectively, giving us the linear model

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon \tag{6.14}$$

where predicted means for the four classes are as follows:[5]

Level of `Frication`	Predicted mean
`burst`	α
`frication`	$\alpha + \beta_1$
`long`	$\alpha + \beta_2$
`short`	$\alpha + \beta_3$

6.6.2 Analysis of variance as model comparison

Now that we have completed the discussion of using dummy variables to construct a linear model with categorical predictors (i.e., factors), we shall move on to discussing what analysis

[3] For example, if $\alpha = 0, \beta_1 = \beta_2 = \beta_3 = \beta_4 = 1$ is a maximum-likelihood estimate, then $\alpha = 1, \beta_1 = \beta_2 = \beta_3 = \beta_4 = 0$ is as well because it encodes exactly the same model.

[4] By default, `R` chooses the first level of a factor as the baseline, and the first level of a factor is whatever level comes first alphabetically unless you specified otherwise when the factor was constructed—see the `levels` argument of the function `factor()` in the documentation.

[5] This choice of coding for the dummy variables is technically known as the choice of CONTRAST MATRIX. The choice of contrast matrix described here is referred to as the TREATMENT contrast matrix, or `contr.treatment` in `R`.

of variance actually *does*. Consider that we now have two possible models of how word-initial frication affects naming time. We have the model of Equation (6.14) above, in which each class of frication predicts a different mean naming time, with noise around the mean distributed the same way for each class. We might also consider a simpler model in which frication has no effect on naming time. Such a model looks as follows:

$$Y = \alpha + \epsilon \tag{6.15}$$

Now look again at Figure 6.9 and think of the simpler model of Equation (6.15) as M_0, and the more complex model of Equation (6.14) as M_A. (Actually, the M_0 explains *no* variance in this case because it just encodes the mean.) Because ANOVA is just a comparison of linear models, we can perform a hypothesis test between M_0 and M_A by constructing an F statistic from the ratio of the amount of variance contained in the boxes $M_A - M_0$ and Unexplained. The simpler model has one parameter and the more complex model has four, so we use Equation (6.11) with $k_0 = 1, k_A = 4$ to construct the F statistic. The MLE of the single parameter for M_0 (aside from the residual noise variance) is the sample mean $\hat{\alpha} = 470$, and the sum of squared residuals in this model is 1032186. For M_A with the dummy variable coding we've used, the MLEs are $\hat{\alpha} = 471$, $\hat{\beta}_1 = 6$, $\hat{\beta}_2 = -4$, and $\hat{\beta}_3 = -16$; the sum of squared residuals is 872627. Thus the F statistic for this model comparison is

$$\begin{aligned} F(3, 2280) &= \frac{(1032186 - 872627)/3}{872627/2280} \\ &= 138.97 \end{aligned}$$

This F statistic corresponds to a p-value of 1.09×10^{-82}, yielding exceedingly clear evidence that the type of initial segment in a word affects its average naming latency.

6.6.3 Testing for interactions

The `english` dataset includes average naming latencies not only for college-age speakers but also for speakers age 60 and over. This degree of age difference turns out to have a huge effect on naming latency (Figure 6.10):

```
histogram(~ RTnaming | AgeSubject, english)
```

Clearly, college-age speakers are faster at naming words than speakers over age 60. We may be interested in including this information in our model. In Lecture 10 we already saw how to include both variables in a multiple regression model. Here we will investigate an additional possibility: that different levels of frication may have different effects on mean naming latency depending on speaker age. For example, we might think that fricatives, which our linear model above indicates are the hardest class of word onsets, might be even harder for elderly speakers than they are for the young. When these types of inter-predictor contingencies are included in a statistical model they are called INTERACTIONS.

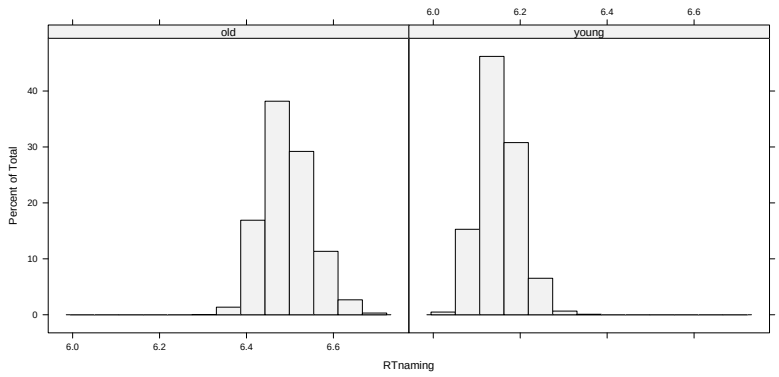


Figure 6.10: Histogram of naming latencies for young (ages $\sim$ 22.6) versus old (ages $>$ 60 speakers)

It is instructive to look explicitly at the linear model that results from introducing interactions between multiple categorical predictors. We will take `old` as the baseline value of speaker age, and leave `burst` as the baseline value of frication. This means that the "baseline" predictor set involves an old-group speaker naming a burst-initial word, and the intercept α will express the predicted mean latency for this combination. There are seven other logically possible combinations of age and frication; thus our full model will have to have seven dummy indicator variables, each with its own parameter. There are many ways to set up these dummy variables; we'll cover perhaps the most straightforward way. In addition to $X_{\{1,2,3\}}$ for the non-baseline levels of frication, we add a new variable X_4 for the non-baseline levels of speaker age (`young`). This set of dummy variables allows us to encode all eight possible groups, but it doesn't allow us to estimate separate parameters for all these groups. To do this, we need to add three more dummy variables, one for each of the non-baseline frication levels when coupled with the non-baseline age level. This gives us the following complete set of codings:

Frication	Age	X_1	X_2	X_3	X_4	X_5	X_6	X_7
burst	old	0	0	0	0	0	0	0
frication	old	1	0	0	0	0	0	0
long	old	0	1	0	0	0	0	0
short	old	0	0	1	0	0	0	0
burst	young	0	0	0	1	0	0	0
frication	young	1	0	0	1	1	0	0
long	young	0	1	0	1	0	1	0
short	young	0	0	1	1	0	0	1

We can test this full model against a strictly ADDITIVE model that allows for effects of both age and initial phoneme class, but not for interactions—that is, one with only $X_{\{1,2,3,4\}}$. It is critical to realize that the additive model is a *constrained* model: five parameters (α and β_1 through β_4) cannot be used to encode eight arbitrary condition means within the

linear framework. The best this M_0 can do—the predicted condition means in its MLE—are compared with the true condition means below:

	Predicted in M_0		Actual (and predicted in M_A)		
	Young	Older		Young	Older
Burst	662.23	470.23	Burst	661.27	471.19
Fricative	670.62	478.61	Fricative	671.76	477.48
Long vowel	653.09	461.09	Long vowel	647.25	466.93
Short vowel	647.32	455.32	Short vowel	647.72	454.92

The predicted per-category means in M_0 can be recovered from the MLE parameter estimates:

$$\hat{\alpha} = 662.23 \quad \hat{\beta}_1 = 8.39 \quad \hat{\beta}_2 = -9.14 \quad \hat{\beta}_3 = -14.91 \quad \hat{\beta}_4 = -192$$

Recovering the predicted means in M_0 from these parameter estimates is left as an exercise for the reader.

When the MLEs of M_0 and M_A are compared using the F-test, we find that our F-statistic turns out to be $F(3, 4560) = 2.69$, or $p = 0.0449$. Hence we also have some evidence that initial segment type has different effects on average naming times for younger and for older speakers—though this evidence is far less conclusive than that for differences across initial-segment type among younger speakers.

6.6.4 Repeated Measures ANOVA and Error Stratification

In the foregoing sections we have covered situations where all of the systematicity across observations can be summarized as deriving from predictors whose effects on the response are systematic and deterministic; all stochastic, idiosyncratic effects have been assumed to occur on level of the individual measurement of the response. In our analysis of average response times for recognition of English words, for example, we considered systematic effects of word frequency, familiarity, neighborhood density, and (in the case of word naming times) initial segment.

Yet it is a rare case in the study of language when there are no potential idiosyncratic effects that are incidental to the true interest of the researcher, yet affect entire *groups* of observations, rather than individual observations. As an example, Alexopolou and Keller (2007) elicited quantitative subjective ratings of sentence acceptability in a study of pronoun resumption, embedding depth, and syntactic islands. One part of one of their experiments involved investigating whether there might be an interaction between embedding and the presence of a resumptive pronoun on sentence acceptability even in cases which are not syntactic islands (Ross, 1967). That is, among the four syntactic frames below, (1-b) should be much less acceptable than (1-a), but (1-d) should not be so much less acceptable than (1-c).

(1) a. Who will we fire __? [UNEMBEDDED, −RESUMPTION]

b. Who will we evict him? [UNEMBEDDED, +RESUMPTION]
c. Who does Lucy claim we will punish __? [EMBEDDED, −RESUMPTION]
d. Who does Emily claim we will arrest him? [EMBEDDED, +RESUMPTION]

As is no doubt evident to the reader, even if we were to find that such a pattern holds for average acceptability ratings of these four sentences, a skeptic could reasonably object that the pattern might well result from the choice of words—the LEXICALIZATIONS—used to fill in the four syntactic templates. For example, *evict* is the least frequent of the four critical verbs above, and it is reasonable to imagine that sentences with less frequent words might tend to be rated as less acceptable.

Hence we want to ensure that our results *generalize* across the specific choice of lexicalizations used in this particular set of four sentences. One way of achieving this would be to prepare $k > 1$ instances of syntactic frame, choosing a separate lexicalization randomly for each of the k instances of each frame ($4k$ lexicalizations total). We might reasonably assume that the effects of choice of lexicalization on acceptability are normally distributed. Following our previous examples, we could use the following dummy-variable encodings:

	X_1	X_2	X_3
[UNEMBEDDED, −RESUMPTION]	0	0	0
[UNEMBEDDED, +RESUMPTION]	1	0	0
[EMBEDDED, −RESUMPTION]	0	1	0
[EMBEDDED, +RESUMPTION]	1	1	1

If ϵ_L is the stochastic effect of the choice of lexicalization and ϵ_E is the normally-distributed error associated with measuring the acceptability of a lexicalized frame, we get the following linear model:

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon_L + \epsilon_E$$

Typically, we can think of speaker-level stochastic effects and measurement-level stochastic effects as independent of one another; hence, because the sum of two independent normal random variables is itself normally distributed (Section 3.5.1), we can just combine these two stochastic components of this equation:

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$$

so we have a completely standard linear model. We could conduct hypothesis tests for this model in the same way as we have done previously in this chapter. For example, we could test the significance of an interaction between embedding and resumption—formally a comparison between a null-hypothesis model M_0 in which $\beta_3 = 0$ and an alternative model M_A with unconstrained β_3—by partitioning variance as in Table 6.1 and conducting an F

test comparing the variance explained by adding β_3 to the model with the residual variance left unexplained by M_A.

By choosing a different set of lexicalizations for each syntactic frame, however, we have introduced additional noise into our measurements that will only increase the difficulty of drawing reliable inferences regarding the effects of embedding, resumption, and their potential interaction. It turns out that we can in general do much better, by using the *same* lexicalizations for each syntactic frame. This is in fact what Alexopolou and Keller did, contrasting a total of nine sentence cohorts of the following types:

(2)
a. (i) Who will we fire __? [UNEMBEDDED, −RESUMPTION]
   (ii) Who will we fire him? [UNEMBEDDED, +RESUMPTION]
   (iii) Who does Mary claim we will fire __? [EMBEDDED, −RESUMPTION]
   (iv) Who does Mary claim we will fire him? [EMBEDDED, +RESUMPTION]

b. (i) Who will we evict __? [UNEMBEDDED, −RESUMPTION]
   (ii) Who will we evict him? [UNEMBEDDED, +RESUMPTION]
   (iii) Who does Elizabeth claim we will evict __? [EMBEDDED, −RESUMPTION]
   (iv) Who does Elizabeth claim we will evict him? [EMBEDDED, +RESUMPTION]

c. ...

Each cohort corresponds to a single lexicalization; in experimental studies such as these the more generic term is ITEM is often used instead of lexicalization. This experimental design is often called WITHIN-ITEMS because the manipulation of ultimate interest—the choice of syntactic frame, or the CONDITION—is conducted for each individual item. Analysis of within-items designs using ANOVA is one type of what is called a REPEATED-MEASURES ANOVA, so named because multiple measurements are made for each of the items. The set of observations obtained for a single item thus constitute a CLUSTER that we hypothesize may have idiosyncratic properties that systematically affect the response variable, and which need to be taken into account when we draw statistical inferences regarding the generative process which gave rise to our data. For this reason, repeated-measures ANOVA is an analytic technique for what are known as HIERARCHICAL MODELS. Hierarchical models are themselves an extremely rich topic, and we take them up in Chapter 8 in full detail. There is also, however, a body of analytic techniques which uses the partitioning of variance and F tests to analyze certain classes of hierarchical models using repeated-measures ANOVA. Because these techniques are extremely widespread in many literatures in the study of language and because these techniques do not require the full toolset for dealing with hierarchical models in general, we cover the repeated-measures ANOVA here. The reader is strongly encouraged, however, to compare the repeated-measure ANOVA with the analytic techniques introduced in Chapter 8, which ultimately offer greater overall flexibility and depth of analysis.

Simple random-intercepts repeated-measures ANOVA

Exactly how to conduct repeated-measures ANOVA depends on the precise nature of the idiosyncratic cluster-level properties assumed. In our current example, the simplest scenario

would be if each item (lexicalization) contributed the same fixed amount to average perceived acceptability regardless of the condition (syntactic frame) in which the lexicalization appeared. If we call the contribution of item i to acceptability a_i, then our model becomes

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + a_i + \epsilon$$

We may consider the a_i themselves to be stochastic: most canonically, they may be normally distributed around 0 with some unknown variance. Happily, the stochasticity in this model does not affect how we go about assessing the systematic effects—β_1 through β_3—of ultimate interest to us. We can partition the variance exactly as before.

6.6.5 Condition-specific random effects and error stratification

More generally, however, we might consider the possibility that idiosyncratic cluster-level properties themselves *interact* with the manipulations we intend to carry out. In our case of embedding and resumption, for example, it could be the case that some of the verbs we choose might be particularly unnatural embedded in a complement clause, particularly natural with an overt resumptive-pronoun object, and/or particularly sensitive to specific combinations of embedding and resumptivity. Such a more general model would thus say that

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + a_i + b_{i1} X_1 + b_{i2} X_2 + b_{i3} X_3 + \epsilon$$

where $\langle a_i, b_{1i}, b_{2i}, b_{3i} \rangle$ are jointly multivariate-normal with mean zero and some unknown covariance matrix Σ. [6]

With this richer structure of idiosyncratic cluster-level properties, it turns out that we *cannot* partition the variance as straightforwardly as depicted in Figure **??** and draw reliable inferences in hypothesis tests about β_1, β_2, and β_3. It is instructive to step through the precise reason for this. Suppose that we were to test for the presence of an interaction between resumption and embedding—that is, to test the null hypothesis $M_0 : \beta_3 = 0$ against the alternative, more general M_A. Even if M_0 is correct, in general the fit of M_A will account for more variance than M_0 simply because M_A is a more expressive model. As in all cases, the amount of variance that M_A fails to explain will depend on the amount of noise at the level of specific observations (the variance of ϵ). But if M_0 is true, the variance explained by M_A beyond M_0 will depend not only the amount of observation-level noise but also on the

[6] Technically, the F-tests covered in this chapter for repeated-measures ANOVA is fully appropriate only when the covariance matrix Σ is such that all differences between pairs of cluster-specific properties have equal variance: technically, for all $x, y \in \{a, b_1, \ldots, b_n\}$, $\sigma_x^2 + \sigma_y^2 - 2\sigma_{xy}$ is constant. This condition is known as SPHERICITY. Violation of sphericity can lead to anti-conservativity of F-tests; remedies include corrections for this anti-conservativity [**insert references**] as well as adopting hierarchical-model analyses of the type introduced in Chapter 8.

amount and nature of cluster-level noise—that is, the variance of b_{i3} and its correlation with a_i, b_{i1}, and b_{i2}. Exercise 6.7 asks you to demonstrate this effect through simulations.

Fortunately, there *does* turn out to be a way to test hypotheses in the face of such a rich structure of (normally-distributed) cluster-level properties: the STRATIFICATION OF VARIANCE. [**TODO: summary of how to determine what comparisons to make**]

As a first example, let us simply examine the simple effect of adding a level of embedding to object-extracted cases without resumptive pronouns: Example (1-c) versus (1-c). In these cases, according to our dummy variable scheme we have $X_1 = X_3 = 0$, giving us the simplified linear equation:

$$Y = \alpha + \beta_2 X_2 + a_i b_{i2} X_2 + \epsilon \tag{6.16}$$

Figure 6.11 demonstrates the stratification of variance. Although everything except the box labeled "Residual Error" is part of the complete model of Equation (6.16), our F-test for the presence of a significant effect of embedding will pit the variance explained by embedding against the variance explained by idiosyncratic subject sensitivities to embedding condition.

Here is code that demonstrates the execution of the repeated-measures ANOVA:

```
> set.seed(2)
> library(mvtnorm)
> n <- 20
> m <- 20
> beta <- c(0.6,0.2) ## beta[1] corresponds to the intercept; beta[2] corresponds to t
> Sigma.b <- matrix(c(0.3,0,0,0.3),2,2) ## in this case, condition-specific speaker se
> sigma.e <- 0.3
> df.1 <- expand.grid(embedding=factor(c("Unembedded","Embedded")),lexicalization=fact
> df <- df.1
> for(i in 1:(n-1))
+   df <- rbind(df,df.1)
> B <- rmvnorm(m,mean=c(0,0),sigma=Sigma.b)
> df$y <- with(df,beta[embedding] + B[cbind(lexicalization,(as.numeric(embedding)))] +
> m <- aov(y ~ embedding + Error(lexicalization/embedding),df)
```

Alexopolou & Keller 2007 data

```
> library(lme4)
```

6.6.6 Case study: two-way analysis of variance for self-paced reading

Here we cover a slightly more complex case study: a TWO-WAY (so named because we examine possible effects of two predictors and their potential interaction) ANOVA of word-by-word READING TIMES (RTs) in a moving-window self-paced reading experiment conducted

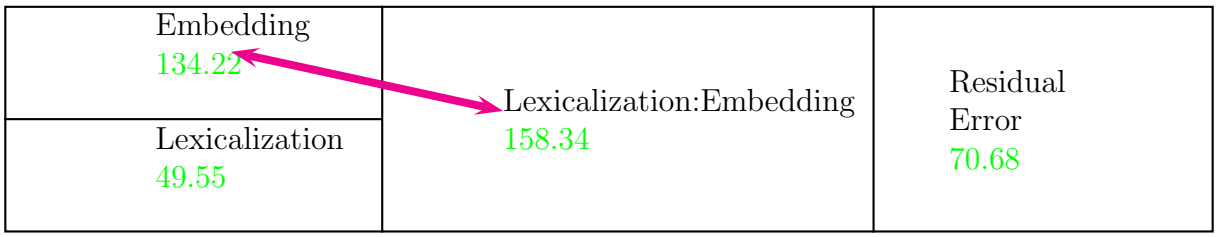


Figure 6.11: Stratification of variance in a simple repeated-measures ANOVA.

by Rohde et al. (2011).[7] In addition to the pure mathematical treatment of the ANOVA, we also cover some preliminary aspects of data analysis. The question under investigation was whether certain kinds of verbs (*implicit causality* (IC) *verbs*) such as "detest", which intuitively demand some sort of explanation, can affect readers' online syntactic attachment preferences.

(3) a. John **detests** the children of the musician who **is** generally arrogant and rude (IC,LOW)
b. John **detests** the children of the musician who **are** generally arrogant and rude (IC,HIGH)
c. John **babysits** the children of the musician who **is** generally arrogant and rude (NONIC,LOW)
d. John **babysits** the children of the musician who **are** generally arrogant and rude (NONIC,HIGH)

We hypothesized that the use of an IC verb should facilitate reading of high-attached RCs, which are generally found in English to be harder to read than low-attached RCs

[7] Moving-window self-paced reading involves presenting sentences one word or group of words at a time, masking previously presented material as new material is revealed, e.g.:

```
-----------.
The -------.
--- cat ---.
------- sat.
```

Participants control the pace at which they read through the material by pressing a button to reveal each new chunk of input; the time between consecutive button presses constitutes the reading time on the pertinent chunk of input.

(Cuetos and Mitchell, 1988). The reasoning here is that the IC verbs demand an explanation, and one way of encoding that explanation linguistically is through a relative clause. In these cases, the most plausible type of explanation will involve a clause in which the object of the IC verb plays a role, so an RC modifying the IC verb's object should become more expected. This stronger expectation may facilitate processing when such an RC is seen (Levy, 2008).

The stimuli for the experiment consist of 20 quadruplets of sentences of the sort above. Such a quadruplet is called an EXPERIMENTAL ITEM in the language of experimental psychology. The four different variants of each item are called the CONDITIONS. Since a participant who sees one of the sentences in a given item is liable to be strongly influenced in her reading of another sentence in the item, the convention is only to show each item once to a given participant. To achieve balance, each participant will be shown five items in each condition.

	Item					
Participant	1	2	3	4	5	...
1	IC,HIGH	NONIC,HIGH	IC,LOW	NONIC,LOW	IC,HIGH	...
2	NONIC,LOW	IC,HIGH	NONIC,HIGH	IC,LOW	NONIC,LOW	...
3	IC,LOW	NONIC,LOW	IC,HIGH	NONIC,HIGH	IC,LOW	...
4	NONIC,HIGH	IC,LOW	NONIC,LOW	IC,HIGH	NONIC,HIGH	...
5	IC,HIGH	NONIC,HIGH	IC,LOW	NONIC,LOW	IC,HIGH	...
⋮	⋮	⋮	⋮	⋮	⋮	⋱

The experimental data will be analyzed for effects of verb type and attachment level, and more crucially for an *interaction* between these two effects. For this reason, we plan to conduct a two-way ANOVA.

In self-paced reading, the observable effect of difficulty at a given word often shows up a word or two downstream, particularly when the word itself is quite short as in this case (short words are often read very quickly, perhaps because the preliminary cue of word length suggests that linguistic analysis of the input will be easy, inducing the reader to initiate the motor activity that will move him/her on to the next word before the difficulty of the linguistic analysis is noticed). Here we focus on the first word after the disambiguator—*generally* in III—often called the first SPILLOVER REGION.

Figure 6.12 provides scatterplots and kernel density estimates (Section 2.11.2) of RT distributions observed in each condition at this point in the sentence. The kernel density estimates make it exceedingly clear that these RTs are far from normally distributed: they are severely right-skewed. ANOVA—in particular repeated-measures ANOVA as we have here—is robust to this type of departure from normality: the non-normality will not lead to anti-conservative inferences in frequentist hypothesis tests. However, the presence of a non-negligible proportion of extremely high values means that the variance of the error is very high, which leads to a poor signal-to-noise ratio; this is a common problem when analyzing data derived from distributions heavier-tailed than the normal distribution. One common means of remedying this issue is adopting some standardized criterion for identifying some observations as OUTLIERS and excluding them from analysis. The practices and reasoning behind outlier removal will vary by data type. In self-paced reading, for example, one rationale for outlier removal is that processes unrelated to sentence comprehension can affect

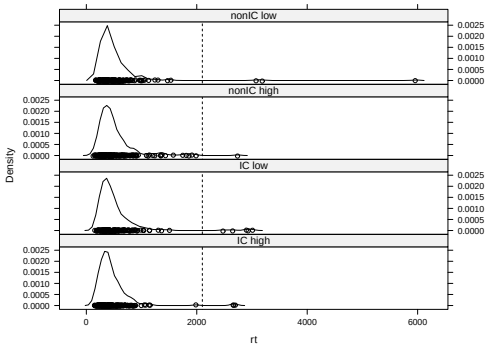


Figure 6.12: Density plots for reading times at the first spillover region for the experiment of Rohde et al. (2011)

recorded reaction times (e.g., the participant sneezes and takes a few seconds to recover); these processes will presumably be independent of the experimental manipulation itself, so if data that were probably generated by these processes can be identified and removed without biasing the outcome of data analysis, it can improve signal-to-noise ratio.

Here we'll adopt a relatively simple approach to outlier removal: binning all our observations, we determine an upper threshold of $\bar{\boldsymbol{y}} + 4\sqrt{S^2}$ where $\bar{\boldsymbol{y}}$ is the sample mean and S^2 is the unbiased estimate of the sample variance (Section 4.3.3). That threshold is plotted in Figure 6.12 as a dotted line; and any observations above that threshold are simply discarded. Note that 12 of the 933 total observations are discarded this way, or 1.3% of the total; consultation of the normal cumulative density function reveals that only 0.0032% would be expected if the data were truly normally distributed.

The comparisons to make

In this experiment, two factors characterize each stimulus: a particular individual reads a particular item that appears with particular *verbtype* (implicit-causality—IC—or non-implicit-causality) and *attachment* level of the relative clause (high or low) manipulations. `verb` and `attachment` have two levels each, so if we had m participants and n items we would in principle need at least $2 \times 2 \times m \times n$ observations to consider a full linear model with interactions of all possible types. However, because each subject saw each item only once, we only have $m \times n$ observations. Therefore it is not possible to construct the full model.

For many years dating back to Clark (1973), the standard ANOVA analysis in this situation has been to construct two separate analyses: one in which the , and one for items. In the analysis over subjects, we take as our individual data points the *mean* value of all the observations in each cell of Subject × Verb × Attachment—that is, we AGGREGATE, or average, across items. Correspondingly, in the analysis over items, we aggregate across subjects. We can use the function `aggregate()` to perform this averaging:

`aggregate()`

`with()`

```
sp.1.subj <- with(spillover.1.to.analyze,aggregate(list(rt=rt),
```

		Subject					
Verb	Attachment	1	2	3	4	5	...
IC	High	280.7	396.1	561.2	339.8	546.1	...
	Low	256.3	457.8	547.3	408.9	594.1	...
nonIC	High	340.9	507.8	786.7	369.8	453.0	...
	Low	823.7	311.4	590.4	838.3	298.9	...

Table 6.1: Repeated-measures (within-subjects) view of item-aggregated data for subjects ANOVA

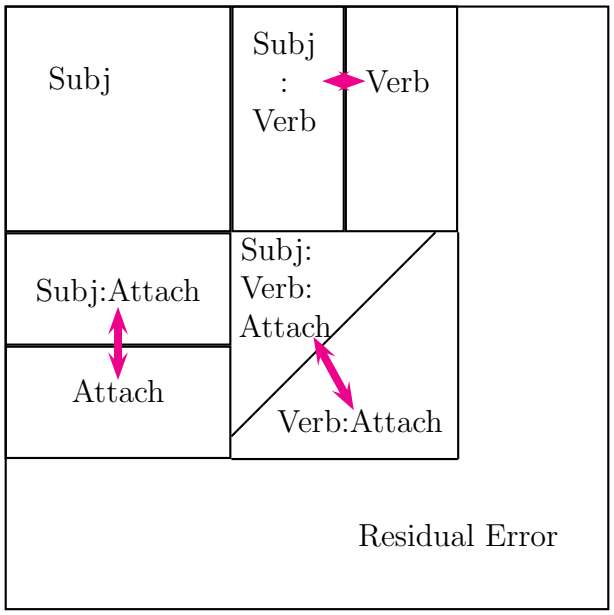


Figure 6.13: The picture for this 2×2 ANOVA, where Verb and Attachment are the fixed effects of interest, and subjects are a random factor

```
  list(subj=subj,verb=verb,attachment=attachment),mean))
sp.1.item <- with(spillover.1.to.analyze,aggregate(list(rt=rt),
  list(item=item,verb=verb,attachment=attachment),mean))
```

The view of the resulting data for the analysis over subjects can be seen in Table 6.1. This setup is called a WITHIN-SUBJECTS or REPEATED-MEASURES design because each subject participates in each condition—or, in another manner of speaking, we take multiple measurements for each subject. Designs in which, for some predictor factor, each subject participates in only one condition are called BETWEEN-SUBJECTS designs.

The way we partition the variance for this type of analysis can be seen in Figure 6.13. Because we have averaged things out so we only have one observation per Subject/Verb/Attachment combination, there will be no variation in the Residual Error box. Each test for an effect of a predictor sets of interest (`verb`, `attachment`, and `verb:attachment`) is performed by comparing the variance explained by the predictor set P with the variance associated with

arbitrary random interactions between the subject and P. This is equivalent to performing a model comparison between the following two linear models, where i range over the subjects and j over the conditions in P:

$$rt_{ij} = \alpha + B_i \text{Subj}_\text{i} + \epsilon_{ij} \qquad \text{(null hypothesis)} \tag{6.17}$$

$$rt_{ij} = \alpha + B_i \text{Subj}_\text{i} + \beta_j P_j + \epsilon_{ij} \qquad \text{(alternative hypothesis)} \tag{6.18}$$

$$\tag{6.19}$$

There is an added wrinkle here, which is that the B_i are not technically free parameters but rather are themselves assumed to be random and normally distributed. However, this difference does not really affect the picture here. (In a couple of weeks, when we get to mixed-effects models, this difference will become more prominent and we'll learn how to handle it in a cleaner and more unified way.)

Fortunately, `aov()` is smart enough to know to perform all these model comparisons in the appropriate way, by use of the `Error()` specification in your model formula. This is done as follows, for subjects:

```
> summary(aov(rt ~ verb * attachment
    + Error(subj/(verb *attachment)), sp.1.subj))

Error: subj
          Df  Sum Sq Mean Sq F value Pr(>F)
Residuals 54 4063007   75241

Error: subj:verb
          Df Sum Sq Mean Sq F value  Pr(>F)
verb       1  48720   48720  7.0754 0.01027 *
Residuals 54 371834    6886
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Error: subj:attachment
           Df Sum Sq Mean Sq F value Pr(>F)
attachment  1    327     327  0.0406  0.841
Residuals  54 434232    8041

Error: subj:verb:attachment
                Df Sum Sq Mean Sq F value  Pr(>F)
verb:attachment  1  93759   93759  6.8528 0.01146 *
Residuals       54 738819   13682
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
```

and for items:

```
> summary(aov(rt ~ verb * attachment
    + Error(item/(verb *attachment)), sp.1.item))

Error: item
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 19 203631   10717

Error: item:verb
          Df Sum Sq Mean Sq F value Pr(>F)
verb       1  21181   21181  3.5482  0.075 .
Residuals 19 113419    5969
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Error: item:attachment
           Df Sum Sq Mean Sq F value Pr(>F)
attachment  1    721     721   0.093 0.7637
Residuals  19 147299    7753

Error: item:verb:attachment
                Df Sum Sq Mean Sq F value  Pr(>F)
verb:attachment  1  38211   38211  5.4335 0.03092 *
Residuals       19 133615    7032
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
```

Fortunately, the by-subjects and by-items analysis yield largely similar results: they both point towards (a) a significant main effect of verb type; and (b) more interestingly, a significant interaction between verb type and attachment level. To interpret these, we need to look at the means of each condition. It is conventional in psychological experimentation to show the condition means from the aggregated data for the by-subjects analysis:

```
> with(sp.1.subj,tapply(rt,list(verb),mean))
      IC    nonIC
452.2940 482.0567
> with(sp.1.subj,tapply(rt,list(verb,attachment),mean))
          high      low
IC    430.4316 474.1565
nonIC 501.4824 462.6309
```

The first spillover region was read more quickly in the implicit-causality verb condition than in the non-IC verb condition. The interaction was a CROSSOVER INTERACTION: in the high

attachment conditions, the first spillover region was read more quickly for IC verbs than for non-IC verbs; but for the low attachment conditions, reading was faster for non-IC verbs than for IC verbs.

We interpreted this result to indicate that IC verbs do indeed facilitate processing of high-attaching RCs, to the extent that this becomes the preferred attachment level.

6.7 Other generalized linear models

Recall that we've looked at linear models, which specify a conditional probability density $P(Y|X)$ of the form

$$Y = \alpha + \beta_1 X_1 + \cdots + \beta_n X_n + \epsilon \tag{6.20}$$

Linear models thus assume that the only stochastic part of the data is the normally-distributed noise ϵ around the predicted mean. Yet many—probably most—types of data do not meet this assumption at all. These include:

- Continuous data in which noise is not normally distributed;
- Categorical data, where the outcome is one of a number of discrete classes;
- Count data, in which the outcome is restricted to non-negative integers.

By choosing different link and noise functions, you can help ensure that your statistical model is as faithful a reflection of possible of the major patterns in the data you are interested in representing. In the remainder of this chapter, we look at two other major classes of GLM: LOGIT and LOG-LINEAR models.

6.7.1 Logit models

Suppose we want a GLM that models binomially distributed data from n trials. We will use a slightly different formulation of the binomial distribution from what that of Chapter 2: instead of viewing the response as the number of successful trials r, we view the response as the *proportion* of successful trials $\frac{r}{n}$; call this Y. The mean proportion for binomial distribution is simply the success parameter π; hence, π is also the predicted mean μ of our GLM. This gives us enough information to specify precisely the resulting model (from now on we replace μ with π for simplicity):

$$P(Y = y; \pi) = \binom{n}{yn} \pi^{ny} (1-\pi)^{n(1-y)} \qquad \text{(or equivalently, replace } \mu \text{ with } \pi\text{)} \tag{6.21}$$

which is just the binomial distribution from back in Equation 3.8.

This is the second part of designing a GLM: choosing the distribution over Y, given the mean μ (Equation 6.1). Having done this means that we have placed ourselves in the BINOMIAL GLM FAMILY. The other part of specifying our GLM is choosing a relationship between the linear predictor η and the mean μ. Unlike the case with the classical linear model, the identity link function is not a possibility, because η can potentially be any real number, whereas the mean proportion μ of successes can only vary between 0 and 1. There are many link functions that can be chosen to make this mapping valid, but here we will use the most popular link function, the LOGIT transform:[8]

$$\log \frac{\pi}{1-\pi} = \eta \tag{6.22}$$

or equivalently the INVERSE LOGIT transform:

$$\pi = \frac{e^{\eta}}{1+e^{\eta}} \tag{6.23}$$

Figure 6.14 shows the relationship between η and π induced by the logit transform

When we insert the full form of the linear predictor from Equation (6.1) back in, we arrive at the final formula for logit models:

$$\pi = \frac{e^{\alpha+\beta_1 X_1+\cdots+\beta_n X_n}}{1+e^{\alpha+\beta_1 X_1+\cdots+\beta_n X_n}} \tag{6.24}$$

Fitting a logit model is also called LOGISTIC REGRESSION.

6.7.2 Fitting a simple logistic regression model

The most common criterion by which a logistic regression model for a dataset is fitted is exactly the way that we chose the parameter estimates for a linear regression model: the method of maximum likelihood. That is, we choose the parameter estimates that give our dataset the highest likelihood.

We will give a simple example using the `dative` dataset. The response variable here is whether the recipient was realized as an NP (i.e., the double-object construction) or as a PP (i.e., the prepositional object construction). This corresponds to the `RealizationOfRecipient` variable in the dataset. There are several options in `R` for fitting basic logistic regression models, including `glm()` in the `stats` package and `lrm()` in the `Design` package. In this case we will use `lrm()`. We will start with a simple study of the effect of recipient pronominality on the dative alternation. Before fitting a model, we examine a contingency table of the outcomes of the two factors:

[8] Two other popular link functions for binomial GLMs are the PROBIT link and the COMPLEMENTARY LOG-LOG link. See Venables and Ripley (2002, Chapter 7) for more details.

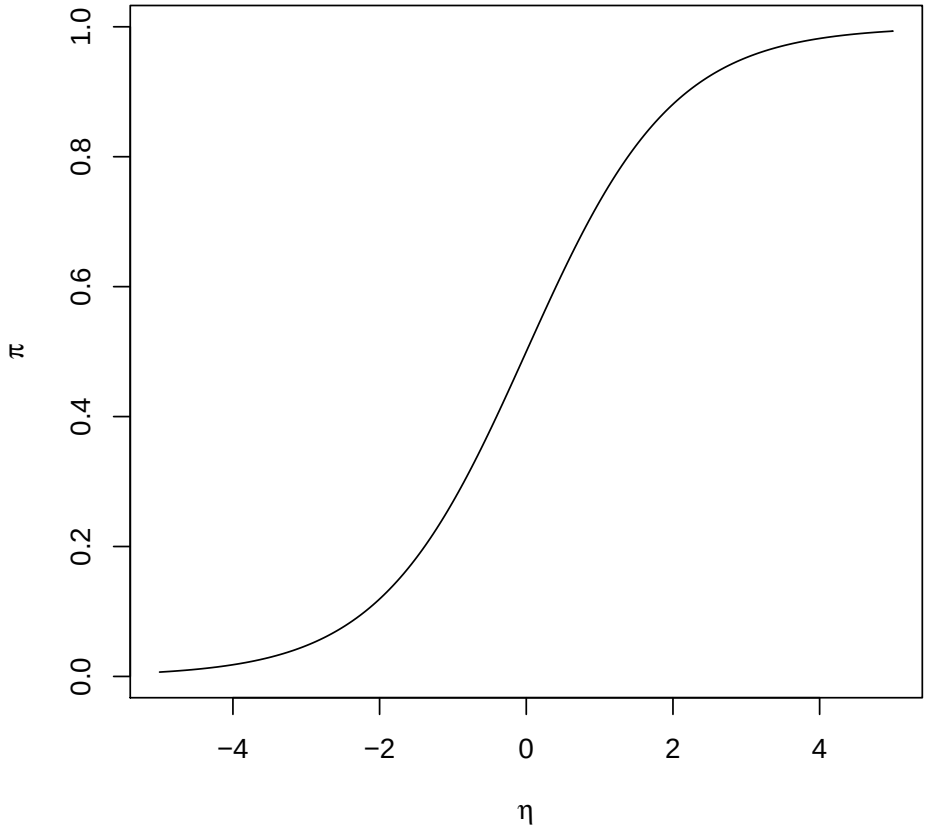


Figure 6.14: The logit transform

```
> library(languageR)
> xtabs(~ PronomOfRec + RealizationOfRecipient,dative)

               RealizationOfRecipient
PronomOfRec       NP   PP
  nonpronominal  600  629
  pronominal    1814  220
```

So sentences with nonpronominal recipients are realized roughly equally often with DO and PO constructions; but sentences with pronominal recipients are recognized nearly 90% of the time with the DO construction. We expect our model to be able to encode these findings.

It is now time to construct the model. To be totally explicit, we will choose ourselves which realization of the recipient counts as a "success" and which counts as a "failure" (although `lrm()` will silently make its own decision if given a factor as a response). In addition,

our predictor variable is a factor, so we need to use dummy-variable encoding; we will satisfice with the R default of taking the alphabetically first factor level, nonpronominal, as the baseline level.

```
> library(rms)
> response <- ifelse(dative$RealizationOfRecipient=="PP",
+                    1,0) # code PO realization as success, DO as failure
> lrm(response ~ PronomOfRec, dative)
```

The thing to pay attention to for now is the estimated coefficients for the intercept and the dummy indicator variable for a pronominal recipient. We can use these coefficients to determine the values of the linear predictor η and the predicted mean success rate p using Equations (6.1) and (6.24):

$$\eta_{--} = 0.0472 + (-2.1569) \times 0 \quad = 0.0472 \quad \text{(non-pronominal receipient)} \tag{6.25}$$

$$\eta_{+} = 0.0472 + (-2.1569) \times 1 \quad = -2.1097 \quad \text{(pronominal recipient)} \tag{6.26}$$

$$p_{\text{nonpron}} = \frac{e^{0.0472}}{1 + e^{0.0472}} \quad = 0.512 \tag{6.27}$$

$$p_{\text{pron}} = \frac{e^{-2.1097}}{1 + e^{-2.1097}} \quad = 0.108 \tag{6.28}$$

When we check these predicted probabilities of PO realization for nonpronominal and pronominal recipients, we see that they are equal to the proportions seen in the corresponding rows of the cross-tabulation we calculated above: $\frac{629}{629+600} = 0.518$ and $\frac{220}{220+1814} = 0.108$. This is exactly the expected behavior, because (a) we have two parameters in our model, α and β_1, which is enough to encode an arbitrary predicted mean for each of the cells in our current representation of the dataset; and (b) as we have seen before (Section 4.3.1), the maximum-likelihood estimate for a binomial distribution is the relative-frequency estimate—that is, the observed proportion of successes.

6.7.3 Multiple logistic regression

Just as we were able to perform multiple linear regression for a linear model with multiple predictors, we can perform multiple logistic regression. Suppose that we want to take into account pronominality of both recipient and theme. First we conduct a complete cross-tabulation and get proportions of PO realization for each combination of pronominality status:

```
> tab <- xtabs(~ RealizationOfRecipient + PronomOfRec + PronomOfTheme, dative)
> tab

, , PronomOfTheme = nonpronominal
```

```
                        PronomOfRec
RealizationOfRecipient nonpronominal pronominal
                    NP           583       1676
                    PP           512         71

, , PronomOfTheme = pronominal

                        PronomOfRec
RealizationOfRecipient nonpronominal pronominal
                    NP            17        138
                    PP           117        149
```

```
> apply(tab,c(2,3),function(x) x[2] / sum(x))

               PronomOfTheme
PronomOfRec     nonpronominal pronominal
  nonpronominal     0.4675799  0.8731343
  pronominal        0.0406411  0.5191638
```

Pronominality of the theme consistently increases the probability of PO realization; pronominality of the recipient consistently increases the probability of DO realization.

We can construct a logit model with independent effects of theme and recipient pronominality as follows:

```
> library(rms)
> dative.lrm <- lrm(response ~ PronomOfRec + PronomOfTheme, dative)
> dative.lrm
```

And once again, we can calculate the predicted mean success rates for each of the four combinations of predictor variables:

Recipient	Theme	η	$\hat{p}$
nonpron	nonpron	-0.1644	0.459
pron	nonpron	-3.0314	0.046
nonpron	pron	2.8125	0.943
pron	pron	-0.0545	0.486

In this case, note the predicted proportions of success are not the same as the observed proportions in each of the four cells. This is sensible – we cannot fit four arbitrary means with only three parameters. If we added in an interactive term, we would be able to fit four arbitrary means, and the resulting predicted proportions would be the observed proportions for the four different cells.

6.7.4 Transforming predictor variables

TODO

Predictor	Coefficient	Factor Weight	Multiplicative effect on odds
Intercept	-0.1644	0.4590	0.8484
Pronominal Recipient	-2.8670	0.0538	0.0569
Pronominal Theme	2.9769	0.9515	19.627

Table 6.2: Logistic regression coefficients and corresponding factor weights for each predictor variable in the `dative` dataset.

6.7.5 Multiplicativity of the odds

Let us consider the case of a dative construction in which both the recipient and theme are encoded with pronouns. In this situation, both the dummy indicator variables (indicating that the theme and recipient are pronouns) have a value of 1, and thus the linear predictor consists of the sum of three terms. From Equation (6.22), we can take the exponent of both sides and write

$$\frac{p}{1-p} = e^{\alpha+\beta_1+\beta_2} \tag{6.29}$$

$$= e^{\alpha} e^{\beta_1} e^{\beta_2} \tag{6.30}$$

The ratio $\frac{p}{1-p}$ is the ODDS OF SUCCESS, and in logit models the effect of any predictor variable on the response variable is multiplicative in the odds of success. If a predictor has coefficent β in a logit model, then a unit of that predictor has a multiplicative effect of e^{β} on the odds of success.

Unlike the raw coefficient β, the quantity e^{β} is not linearly symmetric—it falls in the range $(0, \infty)$. However, we can also perform the full REVERSE LOGIT TRANSFORM of Equation (6.23), mapping β to $\frac{e^{\beta}}{1+e^{\beta}}$ which ranges between zero and 1, and is linearly symmetric around 0.5. The use of logistic regression with the reverse logit transform has been used in quantitative sociolinguistics since Cedergren and Sankoff (1974) (see also Sankoff and Labov, 1979), and is still in widespread use in that field. In quantitative sociolinguistics, the use of logistic regression is often called VARBRUL (variable rule) analysis, and the parameter estimates are reported in the reverse logit transform, typically being called FACTOR WEIGHTS.

Tables 6.2 and 6.3 show the relationship between the components of the linear predictor, the components of the multiplicative odds, and the resulting predictions for each possible combination of our predictor variables.

6.8 Confidence intervals and model comparison in logit models

We'll close our introduction to logistic regression with discussion of confidence intervals and model comparison.

Recip.	Theme	Linear Predictor	Multiplicative odds	P(PO)
–pron	–pron	-0.16	0.8484	0.46
+pron	–pron	$-0.16 - 2.87 = -3.03$	$0.85 \times 0.06 = 0.049$	0.046
–pron	+pron	$-0.16 + 2.98 = 2.81$	$0.85 \times 19.6 = 16.7$	0.94
+pron	+pron	$-0.16 - 2.87 + 2.98 = -0.05$	$0.85 \times 0.06 \times 19.63 = 0.947$	0.49

Table 6.3: Linear predictor, multiplicative odds, and predicted values for each combination of recipient and theme pronominality in the `dative` dataset. In each case, the linear predictor is the log of the multiplicative odds.

6.8.1 Frequentist Confidence intervals for logit models

When there are a relatively large number of observations in comparison with the number of parameters estimated, the standardized deviation of the MLE for a logit model parameter θ is approximately normally distributed:

$$\frac{\hat{\theta} - \theta}{\text{StdErr}(\hat{\theta})} \sim \mathcal{N}(0,1) \quad \text{(approximately)} \tag{6.31}$$

This is called the Wald statistic[9]. This is very similar to the case where we used the t statistic for confidence intervals in classic linear regression (Section 6.4; remember that once the t distribution has a fair number of degrees of freedom, it basically looks like a standard normal distribution). If we look again at the output of the logit model we fitted in the previous section, we see the standard error, which allows us to construct confidence intervals on our model parameters.

```
                          Coef    S.E.    Wald Z P
Intercept                 -0.1644 0.05999  -2.74 0.0061
PronomOfRec=pronominal    -2.8670 0.12278 -23.35 0.0000
PronomOfTheme=pronominal   2.9769 0.15069  19.75 0.0000
```

Following the exact same logic as in Section 6.4, we find that the 95% confidence interval for each parameter β_i is bounded below by $\hat{\beta}_i - 1.96SE(\hat{\beta}_i)$, and bounded below by $\hat{\beta}_i + 1.96SE(\hat{\beta}_i)$. This gives us the following bounds:

```
a  -0.1673002  0.2762782
b1 -3.1076138 -2.6263766
b2  2.6815861  3.2722645
```

The Wald statistic can also be used for a frequentist test on the null hypothesis that an individual model parameter is 0. This is the source of the p-values given for the model parameters above.

[9] It is also sometimes called the Wald Z statistic, because of the convention that standard normal variables are often denoted with a Z, and the Wald statistic is distributed approximately as a standard normal.

6.8.2 Bayesian confidence intervals for logit models

In order to construct a Bayesian confidence interval for a logit model, we need to choose prior distributions on the weights α and $\{\beta_i\}$ that go into the linear predictor (Equation (6.1)), and then use sampling-based techniques (Section 4.5). As a simple example, let us take the multiple logistic regression of Section 6.7.3. The model has three parameters; we will express agnosticism about likely parameter values by using a diffuse prior. Specifically, we choose a normally-distributed prior with large variance for each parameter:

$$\begin{aligned}\alpha &\sim \mathcal{N}(0, 10000)\\ \beta_1 &\sim \mathcal{N}(0, 10000)\\ \beta_2 &\sim \mathcal{N}(0, 10000)\end{aligned}$$

With sampling we can recover 95% HPD confidence intervals (Section 5.1) for the parameters:

```
a  -0.1951817  0.2278135
b1 -3.1047508 -2.6440788
b2  2.7211833  3.2962744
```

There is large agreement between the frequentist and Bayesian confidence intervals in this case. A different choice of prior would change the HPD confidence intervals, but we have a lot of data relative to the complexity of the model we're trying to estimate, so the data dominates the prior in our case.

6.8.3 Model comparison

Just as in the analysis of variance, we are often interested in conducting tests of the hypothesis that introducing *several* model parameters simultaneously leads to a better overall model. In this case, we cannot simply use a single Wald statistic for hypothesis testing. Instead, the most common approach is to use the LIKELIHOOD-RATIO TEST, first introduced in Section 5.4.4. To review, the quantity

$$G^2 = 2\left[\log \text{Lik}_{M_1}(\boldsymbol{y}) - \log \text{Lik}_{M_0}(\boldsymbol{y})\right] \tag{6.32}$$

is approximately distributed as a χ^2_k random variable, where k is the difference in the number of free parameters between M_1 and M_0.

As an example of using the likelihood ratio test, we will hypothesize a model in which pronominality of theme and recipient both still have additive effects but that these effects may vary depending on the modality (spoken versus written) of the dataset. We fit this model and our modality-independent model using `glm()`, and use `anova()` to calculate the likelihood ratio:

```
> m.0 <- glm(response ~ PronomOfRec + PronomOfTheme,dative,family="binomial")
> m.A <- glm(response ~ PronomOfRec*Modality + PronomOfTheme*Modality,dative,family="b
> anova(m.0,m.A)
```

We can look up the p-value of this deviance result in the χ^2_3 distribution:

```
> 1-pchisq(9.07,3)

[1] 0.02837453
```

Thus there is some evidence that we should reject a model that doesn't include modality-specific effects of recipient and theme pronominality.

6.8.4 Dealing with symmetric outcomes

In the study of language, there are some types of categorical outcomes that are symmetrical in a way that can make it difficult to see how to properly assign values to explanatory variables. Consider, for example, the study of word order in the coordination of like categories. Suppose we are interested in the joint effect of word frequency and word length on ordering preferences in word pairs conjoined by *and* (called, appropriately enough, BINOMIALS), and our observation is the phrase *evasive and shifty*. The word *evasive* is longer (has more syllables) than *shifty*, but it is less frequent as well. How do we characterize these independent variables, and do we call the outcome a "success" or a "failure"?

Fortunately, we can address this problem by noticing that the central issue is really not whether *evasive and shifty* is a success or failure; the central issue is, rather, the pattern of how the explanatory variables are aligned with observed orderings. We now cover an example of how to deal with this problem taken from Benor and Levy (2006), a corpus study of English binomials. We will restrict ourselves to word pairs occurring exactly once in Benor and Levy's dataset, and look at the effects of perceptual markedness, weight (in terms of number of syllables), and word frequency. The covariates in the model are thus comparative properties—for example, whether one of the words denotes a property that is more perceptually salient, or which of the words is more frequent (also *chanted*). We can code each property P_i as a quantitative variable X_i by arbitrarily choosing an alignment direction for the property, and giving the binomial a positive value for the X_i if P_i is aligned with the binomial, a negative value of equal magnitude if P_i is aligned against the binomial, and zero if P_i is inactive. The logit response variable now serves as a dummy variable—it is always a "success". For perceptual markedness, word length, and word frequency we choose the following alignments:

- Perceptual markedness is positive if the first word in the binomial is more perceptually salient than the last word;
- Word length (in number of syllables) is positive if the last word has more syllables than the first word;

- Word frequency is positive if the first word is more frequent than the last word.

These aligned properties can be thought of as SOFT (or GRADIENT) CONSTRAINTS in the sense of Optimality Theory and similar frameworks, with statistical model fitting as a principled means of investigating whether the constraints tend not to be violated, and how strong such a tendency may be. A few such observations in the dataset thus coded are:

Word.Pair	Percept	Freq	Syl	Response
chanted and chortled	1	1	0	1
real and vibrant	0	-1	-1	1
evasive and shifty	0	1	-1	1

Note that *chanted and chortled* has a perceptual markedness value of 1, since chortling is a quieter action; *vibrant and real* has a response of 0 since it is observed in the opposite ordering; and the `Syl` covariate value for *evasive and shifty* is -1 because *evasive* has more syllables than *shifty*.

It would be nonsensical to use an intercept when fitting a model to this dataset: setting the intercept arbitrarily high, and the other model parameters to zero, would be the best fit. If, however, we remove the intercept from the model, the model expresses the tendency of each covariate to align with the binomial ordering:

```
> dat <- read.table("../data/binomials_data/single_count_binomials.txt",header=T,fill=
> summary(glm(Response ~ Percept + Syl + Freq - 1, dat,family="binomial"))$coef

         Estimate Std. Error  z value    Pr(>|z|)
Percept 1.1771339  0.5158658 2.281861 0.022497563
Syl     0.4926385  0.1554392 3.169332 0.001527896
Freq    0.3660976  0.1238079 2.956981 0.003106676
```

All three constraints have positive coefficients, indicating significant alignment with binomial ordering: the constraints do indeed tend not to be violated. It's worth noting that even though perceptual markedness is estimated to be the strongest of the three constraints (largest coefficient), its standard error is also the largest: this is because the constraint is active (non-zero) least often in the dataset.

6.9 Log-linear and multinomial logit models

A class of GLM very closely related to logit models is LOG-LINEAR MODELS. Log-linear models choose the log as the link function:

$$l(\mu) = \log \mu = \eta \qquad\qquad \mu = e^{\eta} \tag{6.33}$$

and the Poisson distribution, which ranges over non-negative integers, as the noise function:

$$P(Y = y; \mu) = e^{-\mu}\frac{\mu^y}{y!} \quad (y = 0, 1, \dots) \tag{6.34}$$

When used to model count data, this type of GLM is often called a POISSON MODEL or POISSON REGRESSION.

In linguistics, the log-linear model is most often used to model probability distributions over multi-class outcomes. Suppose that for there are M classes of possible outcomes, each with its own linear predictor η_i and random variable Y_i. If we conditionalize on the total count of all classes being 1, then the only available count outcomes for each class are 0 and 1, with probabilities:

$$P(Y_i = 1; \mu_i = e^{\eta_i}) = e^{\mu_i}e_i^{\eta} \qquad P(Y_i = 0; \mu_i = e^{\eta_i}) = e^{\mu_i} \tag{6.35}$$

and the joint probability of the single observation falling into class i is

$$\begin{aligned} P(Y_i = 1, \{Y_{j\neq i}\} = 0) &= \frac{e^{\mu_i}e_i^{\eta}\prod_{j\neq i}e^{\mu_j}}{\sum_{i'}e^{\mu_{i'}}e_{i'}^{\eta}\prod_{j\neq i'}e^{\mu_j}} \\ &= \frac{e_i^{\eta}\prod_j e^{\mu_j}}{\sum_{i'}e_{i'}^{\eta}\prod_j e^{\mu_j}} \\ &= \frac{e_i^{\eta}\prod_j e^{\mu_j}}{\prod_j e^{\mu_j}\sum_{i'}e_{i'}^{\eta}} \\ P(Y_i = 1, \{Y_{j\neq i}\} = 0) &= \frac{e_i^{\eta}}{\sum_{i'}e_{i'}^{\eta}} \end{aligned} \tag{6.36}$$

When we are thinking of a log-linear model as defining the probability distribution over which class each observation falls into, it is often useful to define the class-specific success probabilities $\pi_i \stackrel{\text{def}}{=} P(Y_i = 1, \{Y_{j\neq i}\} = 0)$. This allows us to think of a log-linear model as using a MULTINOMIAL noise distribution (Section 3.4.1).

Expressive subsumption of (multinomial) logit models by log-linear models*

Basic logit models are used to specify probability distributions over outcomes in two classes (the "failure" class 0 and the "success" class 1). Log-linear models can be used to specify probability distributions over outcomes in any number of classes. For a two-class log-linear model, the success probability for class 1 is (Equation (6.24)):

$$\pi_1 = \frac{e^{\alpha_1+\beta_{1,1}X_1+\cdots+\beta_{1,n}X_n}}{e^{\alpha_0+\beta_{0,1}X_1+\cdots+\beta_{0,n}X_n} + e^{\alpha_1+\beta_{1,1}X_1+\cdots+\beta_{1,n}X_n}} \tag{6.37}$$

$$\tag{6.38}$$

If we divide both the numerator and denominator by $e^{\alpha_0+\beta_{0,1}X_1+\cdots+\beta_{0,n}X_n}$, we get

$$\pi_1 = \frac{e^{(\alpha_1-\alpha_0)+(\beta_{1,1}-\beta_{0,1})X_1+\cdots+(\beta_{1,n}-\beta_{0,n})X_n}}{1+e^{(\alpha_1-\alpha_0)+(\beta_{1,1}-\beta_{0,1})X_1+\cdots+(\beta_{1,n}-\beta_{0,n})X_n}} \tag{6.39}$$

This is significant because the model now has exactly the same form as the logit model (Equation (6.24), except that we have parameters of the form $(\alpha_1-\alpha_0)$ and $(\beta_{1,i}-\beta_{0,i})$ rather than α and β_i respectively. This means that log-linear models EXPRESSIVELY SUBSUME logit models: any logit model can also be expressed by some log-linear model. Because of this, when maximum-likelihood estimation is used to fit a logit model and a log-linear model with the same set of variables, the resulting models will determine the same probability distribution over class proportions. There are only three differences:

1. The log-linear model can also predict the total number of observations.
2. The logit model has fewer parameters.
3. When techniques other than MLE (e.g., Bayesian inference marginalizing over model parameters) are used, the models will generally yield different predictive distributions.

6.10 Log-linear models of phonotactics

We introduce the framework of LOG-LINEAR or MAXIMUM-ENTROPY models by turning to the linguistic problem of phonotactics. A speaker's PHONOTACTIC KNOWLEDGE is their knowledge of what logically possible sound sequences constitute legitimate potential lexical items in her language. In the probabilistic setting, phonotactic knowledge can be expressed as a probability distribution over possible sound sequences. A good probabilistic model of phonotactics assigns low probability to sequences that are not possible lexical items in the language, and higher probability to sequences that are possible lexical items. A categorical characterization of sound sequences as being either impossible or possible in the language could be identified with respective assignment of zero or non-zero probability in the model. The classic example of such a distinction is that whereas native English speakers judge the non-word *blick* [blɪk] to be a possible word of English, they judge the non-word *bnick* [bnɪk] not to be a possible word of English. [citations here] However, probabilistic phonotactic models have the further advantage of being able to make *gradient* distinctions between forms that that are "more" or "less" appropriate as possible lexical items.

Construct a probabilistic phonotactic model entails putting a probability distribution over the possible sound sequences of the language. There are many approaches that could be taken to this problem; here we examine two different approaches in the context of modeling one of the best-studied problems in phonotactics: constrants on of English *word onsets*—the consonant sequences with which words begin. For simplicity, we restrict discussion here to onsets consisting of exactly two segments drawn from a subset of the inventory of English consonants, namely [f], [v], [s], [z], [sh], [p], [b], [t], [d], [l], and [r]. Table 6.4 presents a list

of the two-segment word onsets which can be constructed from these segments and which are found in the Carnegie Mellon Pronouncing Dictionary of English (Weide, 1998). Of the 121 logically possible onsets, only 30 are found. They are highly disparate in frequency, and most of the rarest (including everything on the right-hand side of the table except for [sf] as in *sphere*) are found only in loan words. In the study of phonotactics; there is some question as to exactly what counts as an "attested" sequence in the lexicon; for present purposes, I will refer to the twelve most frequent onsets plus [sf] as *unambiguously attested*.

We begin the problem of estimating a probability distribution over English two-segment onsets using simple tools from Chapters 2 and 4: multinomial models and relative frequency estimation. Let us explicitly represent the sequence structure of an English onset x_1x_2 as a $\#_L x_1 x_2 \#_R$, where $\#_L$ represents the left edge of the onset and $\#_R$ represents the right edge of the onset. Every two-segment onset can be thought of as a linearly ordered joint event comprised of the left edge, the first segment, the second segment, and the right edge. We can use the chain rule to represent this joint event as a product of conditional probabilities:

$$P(\#_L x_1 x_2 \#_R) = P(\#_L)P(x_1|\#_L)P(x_2|\#_L x_1)P(\#_R|\#_L x_1 x_2) \tag{6.40}$$

The left edge is obligatory, so that $P(\#_L) = 1$; and since we are restricting our attention to two-segment onsets, the right edge is also obligatory when it occurs, so that $P(\#_R|\#_L x_1 x_2) = 1$. We can thus rewrite Equation 6.40 as

$$P(\#_L x_1 x_2 \#_R) = P(x_1|\#_L)P(x_2|\#_L x_1) \tag{6.41}$$

We consider three possible methods for estimating this probability distribution from our data:

1. Treat each complete onset $\#_L x_1 x_2 \#_R$ as a single outcome in a multinomial model, with 121 possible outcomes; the problem then becomes estimating the parameters of this single multinomial from our data. As described in Chapter XXX, the maximum likelihood estimate for multinomials is also the relative frequency estimate, so the probability assigned to an onset in this model is directly proportional to the onset's frequency of occurrence.

   With this model it is also useful to note that for any segment x, if the event y immediately preceding it is not the left edge $\#_L$, then y itself is preceded by $\#_L$. This means that $P(x_2|\#_L x_1) = P(x_2|x_1)$. This allows us to rewrite Equation **??**:

   $$P(\#_L x_1 x_2 \#_R) = P(x_1|\#_L)P(x_2|x_1) \tag{6.42}$$

   Hence this model can also be thought of as a *bigram* model in which the probability of an event is, given the immediately preceding event, conditionally independent on everything earlier in the sequence. Note here that if we have N possible segments, we must fit $N + 1$ multinomial distributions.

2. We can introduce the strong independence assumption that the probability of a segment is entirely independent of its context: $P(x_i|x_{1\ldots i-1}) = P(x_i)$. This is a *unigram* model, giving

$$P(\#_L x_1 x_2 \#_R) = P(x_1)P(x_2) \tag{6.43}$$

   Here we need to fit only one multinomial distribution.

3. We can introduce the somewhat weaker independence assumption that the probability of a segment depends on whether it is the first or second segment in the onset, but not on what other segments occur in the onset.

$$P(\#_L x_1 x_2 \#_R) = P(x_1|\#_L)P(x_2|\#_L\_) \tag{6.44}$$

   where _ indicates the presence of *any* segment. We might call this a *positional unigram* model to emphasize the position-dependence. This model requires that we fit two multinomial distributions.

Columns 3–5 of Table 6.4 show estimated probabilities for attested onsets in these three models. Major differences among the models are immediately apparent. Among unambiguously attested onsets, [st] is much more probable in the bigram model than in either unigram model; [tr] and [sf] are much more probable in the unigram model than in the other two models; and [sp] is much less probable in the positional unigram model (see also Exercise 6.12).

A substantive claim about the nature of phonotactic knowledge put forth by researchers including Hayes and Wilson (2007) as well as XXX is that probabilistic models which do a good job accounting for the distribution of segment sequences in the lexicon should also be able to accurately predict native-speaker judgments of the acceptability of "nonce" words (sequences that are not actually words) such as *blick* and *bnick* as potential words of the language. Challenges for this approach become apparent when one examines existing datasets of native-speaker nonce-word judgments. For example, Scholes (1966) conducted a study of English onsets in nonce-word positions and uncovered regularities which seem challenging for the multinomial models we considered above. Among other results, Scholes found the following differences between onsets in the frequency with which nonce words containing them were judged acceptable:

(4) [br] > [vr] > [sr], [ml] > [sf] > [zl], [fs] > [zv]

The fact that the unattested onset [ml] leads to greater acceptability than the unambiguously attested onset [sf] clearly indicates that English phonotactic knowledge involves *some* sorts of generalization beyond the raw contents of the lexicon; hence the bigram model of Table 6.4 is

Segment	Freq	P_{unigram}	P_{unipos}	P_{bigram}	Segment	Freq	P_{unigram}	P_{unipos}	P_{bigram}
st	1784	0.0817	0.0498	0.1755	vl	15	0.0011	0.0006	0.0015
br	1500	0.1122	0.112	0.1476	vr	14	0.003	0.0016	0.0014
pr	1494	0.1405	0.1044	0.147	sf	12	0.0395	0.0003	0.0012
tr	1093	0.1555	0.0599	0.1075	sr	10	0.1553	0.154	0.001
fr	819	0.0751	0.0745	0.0806	zl	9	0.0003	0.0003	0.0009
sp	674	0.0738	0.0188	0.0663	zb	4	0.0003	0.0	0.0004
bl	593	0.0428	0.0427	0.0583	sht	4	0.0067	0.0041	0.0004
fl	572	0.0286	0.0284	0.0563	dv	3	0.0002	0.0001	0.0003
pl	458	0.0535	0.0398	0.0451	zv	2	0.0	0.0	0.0002
dr	441	0.0239	0.0239	0.0434	tv	2	0.0016	0.0003	0.0002
sl	379	0.0592	0.0587	0.0373	dz	2	0.0001	0.0	0.0002
shr	155	0.0128	0.0128	0.0152	tl	1	0.0593	0.0228	0.0001
shl	79	0.0049	0.0049	0.0078	shv	1	0.0001	0.0001	0.0001
ts	23	0.0817	0.0003	0.0023	sb	1	0.059	0.0001	0.0001
sv	19	0.0016	0.0008	0.0019	fs	1	0.0395	0.0003	0.0001

Table 6.4: The attested two-segment onsets of English, based on the segments [f], [v], [s], [z], [sh], [p], [b], [t], [d], [l], and [r], sorted by onset frequency. Probabilities are relative frequency estimates, rounded to 4 decimal places.

unacceptable. At the same time, however, [br] is clearly preferred to [sr], indicating that both unigram models are too simplistic. One might consider a mixture model which interpolates between bigram and unigram models. The difficulty with this approach, however, is that no simple mixture is obvious that would achieve the preferences necessary. The preference of [sr] over [sf] would seem to indicate that unigrams should receive considerable weighting; but the preference of [vr] over [sr] would be undermined by heavy unigram weighting.

To motivate our next development, let us consider specifically the mystery of the relative acceptability of [vr] and [sr] among onsets that are not unambiguously attested. A key piece of information we have not yet considered is the phonological substructure of the segments in question. There are many ways of representing phonological substructure, but one straightforward approach for consonants is a representation that decomposes each segment into three PHONOLOGICAL FEATURES: its PLACE of articulation, MANNER of articulation, and VOICING [refs]. The value of each of these features for each consonant used in our current example can be found in Table 6.5. The set of segments picked out by some conjunction of phonological features or their exclusion is often called a NATURAL CLASS. For example, among the consonants currently under consideration, the phonological feature [+labial] picks out the natural class {[p],[b]}; the feature [-stop] picks out {[s],[z],[f],[v],[sh],[r],[l]}; the phonological feature conjunction [+labiodental,-voiced] picks out the natural class {[f]}; and so forth.

6.10.1 Log-linear models

With multinomial models, it is not obvious how one might take advantage of the featural decomposition of segments in constructing a probability distribution over the discrete

Place	Labial [p],[b]	Labiodental [f],[v]	Alveolar [s],[z],[t],[d],[r],[l]	Alveopalatal [sh]	Velar [k],[g]
Manner	Stop [p],[b],[t],[d],[k],[g]	Fricative [s],[z],[f],[v],[sh]	Liquid [r],[l]		
Voicing	Voiced [b],[d],[g],[v],[z],[r],[l]	Unvoiced [p],[t],[k],[f],[s],[sh]			

Table 6.5: Simple phonological decomposition of the consonants used in Table 6.4

set of possible phoneme sequences. We now turn to a modeling framework that allows such decompositions to be taken into account in modeling such discrete random variables: the framework of LOG-LINEAR models. In this framework, which is intimately related to the logistic-regression models covered previously (see Section XXX), the goal is once again modeling conditional probability distributions of the form $P(Y|X)$, where Y ranges over a countable set of response *classes* $\{y_i\}$. Unlike the cases covered previously in this chapter, however, the log-linear framework is relatively agnostic to the representation of X itself. What is crucial, however, is the presence of a finite set of FEATURE FUNCTIONS $f_j(X, Y)$, each of which maps every possible paired instance of X and Y to a real number. Taken in aggregate, the feature functions map each possible response class y_i to a FEATURE VECTOR $\langle f_1(x, y_i), f_2(x, y_i), \ldots, f_n(x, y_i)\rangle$. Finally, each feature function f_j has a corresponding parameter λ_j. Given a collection of feature functions, corresponding parameter values, and a value x for the conditioning random variable X, the conditional probability of each class y_i is defined to be:

$$P(Y = y_i|X = x) = \frac{1}{Z} \exp\left[\sum_j \lambda_j f_j(x, y_i)\right] \tag{6.45}$$

where Z is a normalizing term ensuring that the probability distribution is proper.

In order to translate our phonotactic learning problem into the log-linear framework, we must identify what serves as the conditioning variable X, the response Y, and what the feature functions f_i are. Since we are putting a probability distribution over logically possible English onsets, the response must be which onset found in a (possible) lexical item. The feature functions should correspond to the phonological features identified earlier.[10] Finally, since we are only trying to fit a single probability distribution over possible English onsets that is not dependent on any other information, whatever we take the conditioning variable X to be, our feature functions will not depend on it; so we can simplify our problem somewhat so that it involves fitting the distribution $P(Y)$ using feature functions $f_j(Y)$ with

[10]Note that the term *feature* is being used in two different here: on the one hand, as part of a decomposition of individual phonological segments, on the other hand as a function that will apply to entire onsets and which is associated with a parameter in the log-linear model. Although phonological features could be used directly as features in the log-linear model, the space of possible log-linear model features is much richer than this.

parameters λ_j (also called FEATURE WEIGHTS), with the functional form of the probability distribution as follows:

$$P(Y = y_i) = \frac{1}{Z} \exp\left[\sum_j \lambda_j f_j(y_i)\right] \tag{6.46}$$

Simple log-linear models of English onsets

What remains is for us to choose the feature functions for our phonotactic model. This choice of feature functions determines what generalizations can be directly encoded in our model. As a first, highly oversimplified model, we will construct exactly one feature function for each natural class specifiable by a single phonological feature of *manner* or *voicing*. This feature function will return the number of segments in that natural class contained in the onset. That is,

$$f_j(y_i) = \begin{cases} 2 & \text{if both segments in onset } i \text{ belong to the } j\text{-th natural class;} \\ 1 & \text{if only one segment in onset } i \text{ belongs to the } j\text{-th natural class;} \\ 0 & \text{if neither segment in onset } i \text{ belongs to the } j\text{-th natural class.} \end{cases} \tag{6.47}$$

There are four manner/voicing phonological features for our segment inventory; each can be negated, giving eight natural classes.[11] Each onset is thus mapped to an eight-dimensional feature vector. In the onset [sr], for example we would have the following counts:

Natural class	Matching segments in [sr]	Natural class	Matching segments in [sr]
[+stop]	0	[-stop]	2
[+fric]	1	[-fric]	1
[+liquid]	1	[-liquid]	1
[+voiced]	1	[-voiced]	1

so that the feature vector for [sr] in this model would be $\langle 0, 2, 1, 1, 1, 1, 1, 1\rangle$.

What remains is for us to fit the parameter values $\lambda_1, \ldots, \lambda_8$ corresponding to each of these features. For a simple model like this, in which there are relatively few parameters (eight) for many outcome classes (121) and many observations (10,164), maximum likelihood estimation is generally quite reliable. We find the following maximum-likelihood estimates four our eight feature weights:

[+stop]	-0.0712	[-stop]	0.0928
[+fric]	-0.5472	[-fric]	0.5012
[+liquid]	0.5837	[-liquid]	-0.6679
[+voiced]	-0.4713	[-voiced]	0.7404

[11] We omit the phonological feature of unvoicedness because, since voicing here is a binary distinction, [+unvoiced] would be equivalent to [-voiced].

Onset	Freq	P_{M_1}	P_{M_2}	P_{M_3A}	P_{M_3B}	Onset	Freq	P_{M_1}	P_{M_2}	P_{M_3A}	P_{M_3B}
st	1784	0.0097	0.1122	0.1753	0.1587	vl	15	0.0035	0.0007	0.0013	0.0018
br	1500	0.0086	0.1487	0.1473	0.1415	vr	14	0.0035	0.0018	0.0014	0.0034
pr	1494	0.0287	0.1379	0.1468	0.1442	sf	12	0.004	0.0003	0.0009	0.001
tr	1093	0.0287	0.0791	0.1075	0.1033	sr	10	0.0119	0.0915	0.0014	0.0155
fr	819	0.0119	0.0443	0.0802	0.0726	zl	9	0.0035	0.0004	0.0009	0.0017
sp	674	0.0097	0.0423	0.066	0.056	zb	4	0.0009	0.0001	0.0001	0.0001
bl	593	0.0086	0.0567	0.0582	0.0541	sht	4	0.0097	0.0093	0.0003	0.0039
fl	572	0.0119	0.0169	0.0561	0.0454	dv	3	0.0009	0.0001	0.0001	0.0001
pl	458	0.0287	0.0526	0.045	0.046	zv	2	0.0004	0	0.0001	0
dr	441	0.0086	0.0317	0.0432	0.0391	tv	2	0.0029	0.0001	0.0001	0.0002
sl	379	0.0119	0.0349	0.0374	0.0359	dz	2	0.0009	0	0	0.0001
shr	155	0.0119	0.0076	0.0153	0.0143	tl	1	0.0287	0.0301	0.0006	0.0106
shl	79	0.0119	0.0029	0.0077	0.0067	shv	1	0.0012	0.0001	0	0
ts	23	0.0097	0.0005	0.0017	0.0002	sb	1	0.0029	0.0002	0.0002	0.0016
sv	19	0.0012	0.0011	0.0017	0.0002	fs	1	0.004	0.0003	0.0001	0.0005

Table 6.6: Probabilities estimated from four log-linear models for attested English onsets consisting of pairs from the segment inventory [f], [v], [s], [z], [sh], [p], [b], [t], [d], [l].

Similar to the case in logistic regression, positive feature weights indicates preference for onsets with large values for the feature in question, and negative feature weights indicate dispreference for such onsets. The model has learned that stops are slightly dispreferred to non-stops; fricatives and liquids are strongly preferred to non-fricatives and non-liquids; and unvoiced consonants are strongly preferred to voiced consonants. A sharper view, however, of the generalizations made by the model can be seen in Table 6.6, which shows the probabilities placed by this model on attested onsets. Although there are some things that seem to be correct about this model's generalizations—for example, none of the unambiguously attested onsets are given probability below 0.004—the model makes far too few distinctions, leading to problems such as the assignment of high probability to [tl], and the assignment of identical probabilities to [ts] and [st]. This failure should have been expected, however, given that our feature functions failed to encode any positional information, or to distinguish at all between certain segments, such as [t] and [p].

We address some of these concerns by moving on to a more complex model, which allows the following generalizations as feature functions:

- Preferences for particular segments to occur in position 1;
- Preferences for particular segments to occur in position 2;
- Preferences for particular bigrams of natural classes specifiable by a single phonological feature of either manner or voicing.

The first two types of features give the log-linear model the same generalizational power as the positional unigram model we covered earlier. The third type of feature, however,

goes beyond the positional unigram model, and allows the model to make use of abstract phonological features in generalizing over possible lexical items. Formally speaking, we have one feature function for each segment in position 1, one feature function for each segment in position 2, and one feature function for each possible pairing of single-feature natural classes. This gives us twenty-two possible single-segment feature functions and 8*8=64 possible bigram feature functions, for a total of 86. We will let each serve as an INDICATOR FUNCTION mapping those onsets it correctly describes to the value 1, and all other onsets to the value 0:

$$f_j(y_i) = \begin{cases} 1 & \text{if the } j\text{-th feature describes } y_i; \\ 0 & \text{otherwise.} \end{cases} \tag{6.48}$$

As a concrete example of how these feature functions would be applied, let us again consider the onset [sr]. It satisfies the following descriptions:

- [s] is in position 1 (we represent this feature as `s.`, with `.` indicating that anything can appear in position 2)
- [r] is in position 2 (we represent this feature as `.r`)
- [-liquid][+voice]
- All pairwise combinations of [-liquid],[-stop],[+fric],[-voice] in position 1 with [+liquid],[-stop],[-fric],[+voice] in position (16 combinations in total)

Thus the feature vector for [sr] would have eighteen entries of 1 and 68 entries of 0. Using the method of maximum likelihood to estimate values for the 86 parameters of the model, we find that the features with strongest absolute preferences and dispreferences are as follows:

[-voice][-voice]	5.62962436676025390625
.v	3.64033722877502441406
[-liquid][-stop]	2.91994524002075195312
[-voice][-stop]	2.36018443107604980469
s.	1.81566941738128662109
[-liquid][+liquid]	1.72637474536895751953
.t	1.68454444408416748047
[-stop][-liquid]	1.56158518791198730469
...	
.b	-1.03666257858276367188
z.	-1.07121777534484863281
[-liquid][-liquid]	-1.20901763439178466797
[-stop][+fric]	-1.24043428897857666016
.f	-1.30032265186309814453
[-stop][-stop]	-1.85031402111053466797
.d	-1.97170710563659667969
.sh	-3.09503102302551269531

Brief inspection indicates that most of the model's strongest preferences involve generalization on natural class cooccurences: preference for onsets to involve pairs of unvoiced segments, dispreference for pairs matching in manner of articulation, and so forth. In addition, some strong segment-specific positional dispreferences are also found, such as the preference for initial [s] and dispreference for initial [sh]. Caution is required, however, in interpreting individual feature weights too simplistically—for example, it is clear from the lexicon of English that [sh] is dispreferred even more strongly in the second position than in the first position, yet second-position [sh] feature does not appear in the list of most strongly dispreferred features. The reason for this is that several other features—including the four with the largest negative weights—strongly penalize second-position [sh] already. As with linear and logistic regression models, the proper interpretation of a feature weight is what effect a change in the associated feature value would have, *if all other feature values were kept constant.*

The other way of inspecting the generalizations made by the model is by looking at the predictive distribution on the response variable itself, as seen in Table 6.6. This model has a number of clear advantages over our simplest model: it is relatively successful at giving unambiguously attested onsets higher probability than attested onsets, but at the same time gives [sr] higher probability than many other onsets, including some that are unambiguously attested. However, it also has some weaknesses: for example, the probability for [sf] has dropped below many onsets that are not unambiguously attested, such as [vl].

Overparameterization and regularization

At some level, we might want to allow our model to have specific sensitivity to the frequency of *every* possible onset, so that each instance of a given onset x_1x_2 contributes directly and idiosyncratically to the probability of other words with that onset; but at the same time, we clearly want our model to generalize to onsets that do not occur in the English lexicon as well. Within the maximum-likelihood log-linear framework we have developed thus far, these two requirements are in conflict with one another, for the following reason. In order to allow the model sensitivity to the frequency of specific onsets, we would want to introduce one feature function for each possible onsets, giving us 121 feature functions and thus 121 parameters to estimate. However, this parameterization allows the encoding of *any* probability distribution over the 121 possible response classes. As we saw in Chapter 4, the maximum-likelihood estimate for a multinomial distribution is just the relative-frequency estimate. Hence a maximum-likelihood log-linear model with onset-specific feature functions would simply memorize the relative frequencies of the onsets. Since adding more natural class-based features to the model can only *increase* its expressivity, no ML-estimated model with these 121 features will generalize beyond relative frequencies.

It *is* possible, however, to learn both onset-specific knoweldge and natural-class-level generalizations simultaneously within the log-linear framework, however, by moving away from maximum-likelihood point estimation and instead adopting a Bayesian framework. Recall that in the Bayesian approach, the posterior probability of the model parameters λ is proportional to the likelihood of the data under λ times the prior probability of λ, which in our

case works out to:

$$P(\lambda|Y) = P(Y|\lambda)P(\lambda) \tag{6.49}$$

$$= \left[\prod_i \frac{1}{Z} \exp\left[\sum_j \lambda_j f_j(y_i)\right]\right] P(\lambda) \tag{6.50}$$

with Z a normalizing factor dependent on λ. The next step is to choose a prior distribution over the model parameters, $P(\lambda)$. In principle, any prior could be used; in practice, a popular choice is a multivariate *Gaussian* prior distribution (Section 3.5) with center μ and covariance matrix Σ, so that the prior probability of an N-dimensional parameter vector λ is

$$P(\lambda) = \frac{1}{\sqrt{(2\pi|\boldsymbol{\Sigma}|)^N}} \exp\left[\frac{(\lambda-\mu)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\lambda-\mu)}{2}\right] \tag{6.51}$$

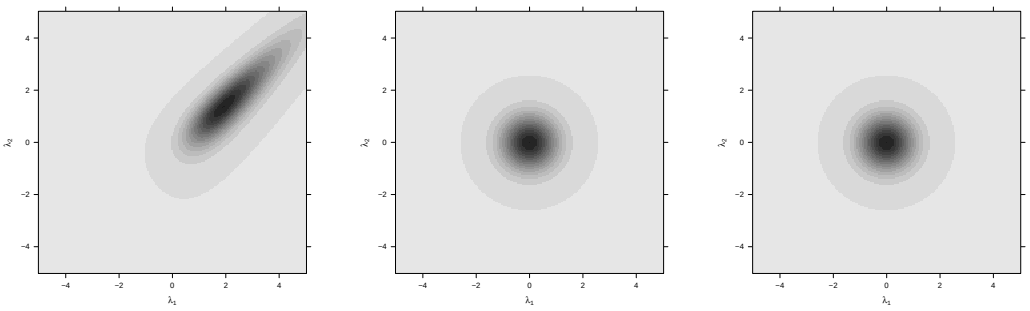
This choice of prior is popular for three reasons: (i) it has an intuitive interpretation as encoding a bias toward the parameter vector μ that is weak in the vicinity of μ but grows rapidly stronger with increasing distance from μ; (ii) for log-linear models, the posterior distribution over λ remains convex with a Gaussian prior; and (iii) Gaussian priors have been found to work well in allowing fine-grained learning while avoiding overfitting with log-linear models. The simplest choice of prior is one in which with mean $\mu = 0$ and a diagonal covariance matrix whose nonzero entries are all the same value: $\Sigma = \begin{pmatrix} \sigma^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma^2 \end{pmatrix}$. Multivariate Gaussian distributions like this are often called SPHERICAL, because surfaces of equal probability are (hyper-)spheres. With a spherical Gaussian prior, the posterior distribution can be written as follows:

$$P(\lambda|Y) \propto P(Y|\lambda)P(\lambda) \tag{6.52}$$

$$= \left[\prod_i \frac{1}{Z} \exp\left[\sum_j \lambda_j f_j(y_i)\right]\right] \exp\left[\sum_j \frac{-\lambda_j^2}{2\sigma^2}\right] \tag{6.53}$$

If we shift to log space we get

$$\log P(\lambda|Y) \propto \overbrace{\sum_i \left[\log \frac{1}{Z} + \sum_j \lambda_j f_j(y_i)\right]}^{\text{Log-likelihood}} - \overbrace{\left[\sum_j \frac{\lambda_j^2}{2\sigma^2}\right]}^{\text{Negative log-prior probability}} \tag{6.54}$$



(a) Likelihood (b) Prior (c) Posterior

Figure 6.15: A multivariate Gaussian prior ($\mu = 0, \sigma = 1$) for a simple log-linear model with three possible response classes and two indicator features functions: f_1 associated with class 1 and f_2 associated with class 2. First panel is model likelihood for class 1–3 frequencies of 7, 4, and 1 respectively; second panel is the prior distribution; third panel is the posterior distribution.

Note that the log-posterior probability falls off quadratically with the sum of the feature weights. For this reason, a Gaussian prior is sometimes called a QUADRATIC prior.

The effect of a Gaussian prior of this form can be seen in Figure 6.15: the prior penalizes deviations from its mode of 0 (a symmetric model in which all outcomes are equally likely), so that the posterior mode falls in between the MLE and the prior mode.

Let us now turn back to our study of English onsets, ready to apply our Bayesian log-linear model. We are now in a position to deploy a richer set of feature functions: on top of the positional single-segment and paired natural-class features we included in the previous model, we add paired-segment and positional single-natural-class features. This gives us an inventory of 223 total feature functions; the feature-vector representation for the onset [sr], for example, would now have the paired-segment feature `sr`, as well as the positional single-natural-class features [-stop]., [+fric]., [-liquid]., [-voiced]., .[-stop], .[-fric], .[+liquid], and .[+voiced].

As is always the case with Bayesian inference, we have a number of choices as to handle the problems of parameter estimation and prediction. Unlike the case with multinomial models, however, there are no readily available analytic techniques for dealing with Bayesian log-linear models, and sampling techniques can be quite computationally intensive. A popular approach is to use maximum a-posteriori (MAP) estimation to find the set of feature weights with (near-)maximum posterior probability, and to approximate Bayesian prediction by using these MAP parameter estimates. In our problem, using a symmetric Gaussian prior centered around $\mathbf{0}$ with standard deviation $\sigma = 1$, the features with largest and smallest weights in the MAP estimate are as follows:

st	3.24827671051025390625
sp	2.78049993515014648438
fl	2.51510095596313476562
ts	2.09755825996398925781
s.	1.87449419498443603516
[-voice][-voice]	1.80559206008911132812
fr	1.80392193794250488281
.v	1.72066390514373779297
...	
[-stop][+fric] -0.67749273777008056641	
[+voice][-voice] -0.70867472887039184570	
.d	-1.00191879272460937500
shp	-1.00633215904235839844
ss	-1.12540340423583984375
fp	-1.57261526584625244141
.sh	-1.64139485359191894531
zr	-1.65136361122131347656
ft	-1.85411751270294189453
dl	-2.24138593673706054688
tl	-3.16438293457031250000
sr	-4.12058639526367187500

Comparison with the previous model indicates important overall similarities, but it is clear that the new features are also being used by the model, perhaps most notably in encoding idiosyncratic preferences for [st] and dispreferences for [sr] and [sh]. The predictive distribution of this model, M_{3A}, can be found in Table 6.6. As expected, there is more probability mass on unambiguously attested onsets in this model than in either previous model, since this model is able to directly encode idiosyncratic preferences for specific onsets. Additionally, much of the apparent weakness of M_2 has been partly remedied—for example, the probability of [sr] has dropped below all the other unambiguously attested sequences except for [sf] while the lower probability of [vr] has stayed about the same.

Strength of the prior and generalization in log-linear models

What is the effect of increasing the strength of the prior distribution, as encoded by decreasing the standard deviation σ of the spherical multivariate Gaussian? There are two key effects we'll cover here consideration. The first effect is an overall tendency for the posterior to look more like the prior, a straightforward and intuitive consequence of the fact that in Bayesian inference, prior and likelihood stand on equal ground in determining posterior beliefs. There is a second, more subtle effect that merits attention, however, and which becomes clear from careful inspection of Equation 6.54. Consider the contribution of an individual feature weight λ_j to the posterior probability of the complete parameter vector λ. The choice of λ_j contributes directly to the log-likelihood once for *every* observation for which the corresponding

feature is implicated, but contributes to the prior log-probability only once regardless of how many observations in which the corresponding feature is implicated. This fact leads has an important consequence: a stronger prior penalizes feature weights more heavily the *sparser* the corresponding feature—that is, the less often that feature is unambiguously implicated in the data.

We can illustrate this consequence by re-fitting our previous log-linear phonotactic model using a much stronger prior: a spherical Gaussian distribution with standard deviation $\sigma = 0.01$. The resulting probability distribution over attested onsets is shown in Table 6.6 as model M_{3B}. Compared with M_{3A} (which had $\sigma = 1$), there is an overall shift of probability mass away from unambiguously attested onsets; this is the first effect described above. However, the remaining onsets do *not* all undergo similar increases in probability: the onsets [sr] and [tl], for example, undergo very large increases, whereas onsets such as [vl] and [zb] stay about the same. The reason for this is as follows. The more general features—natural-class and segment unigrams and natural-class bigrams—favor [sr] and [tl]: in our data, [s] and [t] are common as the first segment of two-segment onsets, [r] and [l] are common as the second segment of two-segment onsets, and [-voiced][+liquid] is a common natural-class bigram. The burden of fitting the low empirical frequency of [sr] and [tl] falls on the most specific features—segment bigrams—but large weights for specific features are disfavored by the strong prior, so that the resulting predictive probabilities of these onsets rises. In contrast, [vl] and [zb] are not favored by the more general features, so that their predictive probability does not rise appreciably with this moderate increase in prior strength.

A word of caution

Finally, a word of caution is necessary in the practical use of MAP estimation techniques with overparameterized log-linear models: even using Bayesian techniques so that the MAP estimate is well-defined, the posterior distribution can be *very* flat in the vicinity of its optimum, which can make it difficult to be sure how close the obtained solution may be to the true optimum. In these cases, one would do well to impose stringent convergence criteria on whatever optimization algorithm is used to search for the MAP estimate.

Log-linear distributions are maximum-entropy distributions

mention the term maxent, and point out that log-linear models satisfy the maxent property

6.10.2 Translating between logit models and log-linear models

Although we have demonstrated in Section 6.9 that log-linear models expressively subsume logit models, translating between the two can be require some care. We go through a brief example here.

SAY MORE

Gabe's needs doing example.

```
> dat <- read.table("../data/needs_doing_data/needs.txt",header=T)
> dat$Response <- ifelse(dat$Response=="ing",1,0)
> dat$Anim1 <- factor(ifelse(dat$Anim=="abst","abst","conc"))
> model.logit <- glm(Response ~ Anim1 + sqrt(Dep.Length), data=dat, family=binomial)
> # data processing to get data in format for log-linear/Poisson model
> dat.for.loglin <- with(dat,as.data.frame(as.table(tapply(Response, list(Anim1=Anim1,
> names(dat.for.loglin)[4] <- "x"
> dat.for.loglin$DL <- dat.for.loglin$Dep.Length
> dat.for.loglin$Dep.Length <- as.numeric(as.character(dat.for.loglin$DL))
> dat.for.loglin$Response <- as.numeric(as.character(dat.for.loglin$Response))
> dat.for.loglin$x <- sapply(dat.for.loglin$x, function(x) ifelse(is.na(x), 0, x))
> model.loglin <- glm(x ~ Anim1*DL + Response + Response:(Anim1 + sqrt(Dep.Length)),da
> summary(model.loglin)$coef[c(32,62,63),]

                            Estimate Std. Error   z value     Pr(>|z|)
Response                  -0.2950173 0.11070848 -2.664812 7.703144e-03
Anim1conc:Response         1.3333414 0.14315638  9.313880 1.232457e-20
Response:sqrt(Dep.Length) -0.6048434 0.06369311 -9.496215 2.176582e-21

> summary(model.logit)$coef

                   Estimate Std. Error   z value     Pr(>|z|)
(Intercept)      -0.2950173 0.11070848 -2.664812 7.703141e-03
Anim1conc         1.3333414 0.14315636  9.313881 1.232446e-20
sqrt(Dep.Length) -0.6048434 0.06369308 -9.496218 2.176519e-21
```

What we see here is that the "effect of the response variable category" in the log-linear model corresponds to the intercept in the logit model; and the interactions of response with animacy and dependency length in the log-linear model correspond to the animacy and dependency length effects in the logit model. Of course, the logit model is far more efficient to fit; it involved only three parameters, whereas the log-linear model required sixty-three.

WHAT ABOUT MODELS WHERE WE HAVE NO BASELINE CLASS BUT ALSO DON'T NEED ALL THOSE EXTRA PARAMETERS TO MODEL THE COUNTS?

...

6.11 Guide to different kinds of log-linear models

Because we have covered several types of log-linear models in this chapter, it is useful to take a moment to carefully consider the relationship among them. A diagram making these relationships explicit is given in Figure 6.16. This section briefly describes these relationships. For brevity, we have used dot-product notation instead of summation notation: model parameters and feature-function outcomes are both denoted with vectors $\boldsymbol{\lambda}$ and $\boldsymbol{f}(x, y_i)$, so

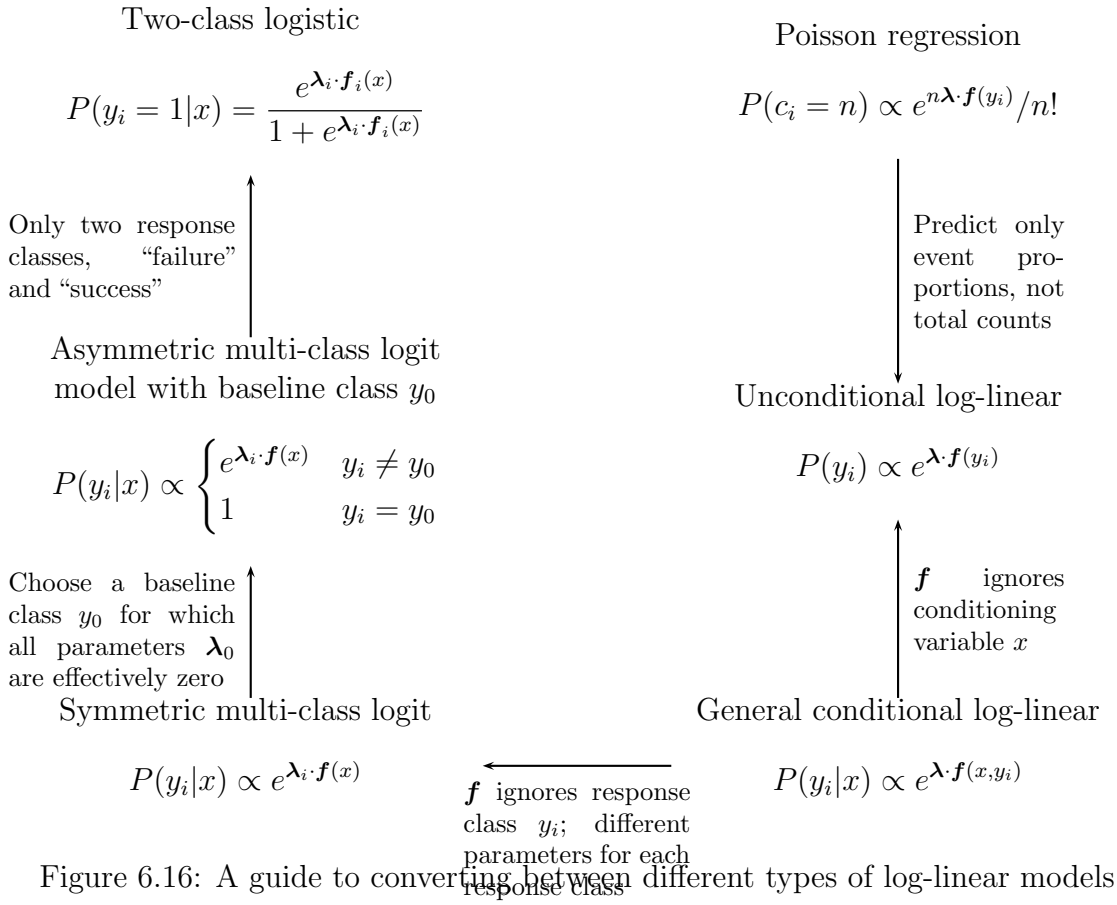


Figure 6.16: A guide to converting between different types of log-linear models

that the weighted sums $\sum_j \lambda_j f_j(x, y_i)$ we have seen previously can be succinctly expressed as dot products $\boldsymbol{\lambda} \cdot \boldsymbol{f}(x, y_i)$.

In the bottom-right corner of Figure 6.16 is the general conditional log-linear model we covered in Section XXX. In this model, there is a collection of feature functions f_j each of which maps an input x paired with a response class y_i to a real number. Each feature function f_j has an associated parameter weight λ_j. In the general conditional log-linear model, no further constraints are placed on the nature of these feature functions.

It is a common modeling decision, however, to assume that there should effectively be a single set of feature functions shared identically by all possible response classes. As an example, consider the problem of relativizer choice for non-subject extracted relative clauses with animate head nouns, such as in *the actress __ you mentioned*. In modeling this problem with conditional distributions P(Relativizer|Context), one might consider three possible response classes: *that*, *who(m)*, and relativizer omission. To examine the effect of frequency of the head noun (here, *actress*), we might want to come up with a single numerical encoding (say, log of Brown-corpus frequency) which is associated with a different feature function for each response class. Thus we would have three feature functions $f_{1,2,3}$, each of which is defined as follows:

$$f_j(x, y_i) = \begin{cases} \text{Log head-noun frequency} & j = i \\ 0 & \text{otherwise} \end{cases}$$

An equivalent approach, however, would be to say that there is only one feature function f_1 which always returns log head-noun frequency, but has a different parameter λ_{1i} for each response class. Taking this approach moves us to the lower-left corner of Figure 6.16: symmetric multi-class logistic regression. This category of model is more restrictive than the general conditional log-linear model: the latter can express probability distributions unavailable to the former, by using feature functions that are active for more than one response class, or by using feature functions active for one response class which have no matching feature function for another response class.

Some readers may have noticed that the symmetric multi-class logit model has more parameters than it needs. Let us identify one of the N response classes y_0 as the BASELINE CLASS. Then for an input x, we can the probability of any outcome y_i is as follows:

$$P(y_i|x) = \frac{e^{\boldsymbol{\lambda}_i \boldsymbol{f}(x)}}{e^{\boldsymbol{\lambda}_0 \boldsymbol{f}(x)} + e^{\boldsymbol{\lambda}_1 \boldsymbol{f}(x)} + \cdots + e^{\boldsymbol{\lambda}_N \boldsymbol{f}(x)}} \tag{6.55}$$

Let us now divide both the top and bottom of this fraction by $e^{\boldsymbol{\lambda}_0 \boldsymbol{f}(x)}$:

$$P(y_i|x) = \frac{e^{\boldsymbol{\lambda}_i \boldsymbol{f}(x)} \frac{1}{e^{\boldsymbol{\lambda}_0 \boldsymbol{f}(x)}}}{\left[e^{\boldsymbol{\lambda}_0 \boldsymbol{f}(x)} + e^{\boldsymbol{\lambda}_1 \boldsymbol{f}(x)} + \cdots + e^{\boldsymbol{\lambda}_N \boldsymbol{f}(x)}\right] \frac{1}{e^{\boldsymbol{\lambda}_0 \boldsymbol{f}(x)}}} \tag{6.56}$$

$$= \frac{e^{[\boldsymbol{\lambda}_i - \boldsymbol{\lambda}_0] \boldsymbol{f}(x)}}{e^{[\boldsymbol{\lambda}_0 - \boldsymbol{\lambda}_0] \boldsymbol{f}(x)} + e^{[\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_0] \boldsymbol{f}(x)} + \cdots + e^{[\boldsymbol{\lambda}_N - \boldsymbol{\lambda}_0] \boldsymbol{f}(x)}} \tag{6.57}$$

But $\boldsymbol{\lambda}_i - \boldsymbol{\lambda}_0 = \mathbf{0}$, so $e^{[\boldsymbol{\lambda}_0 - \boldsymbol{\lambda}_0] \boldsymbol{f}(x)} = 1$. If we now define $\boldsymbol{\lambda}'_i \equiv \boldsymbol{\lambda}_i - \boldsymbol{\lambda}_0$, we have:

$$P(y_i|x) = \frac{e^{\boldsymbol{\lambda}'_i \boldsymbol{f}(x)}}{1 + e^{\boldsymbol{\lambda}'_1 \boldsymbol{f}(x)} + \cdots + e^{\boldsymbol{\lambda}'_N \boldsymbol{f}(x)}} \tag{6.58}$$

This is a new expression of the same model, but with fewer parameters. Expressing things in this way leads us to the middle-left model in Figure 6.16. This is an *asymmetric* multiclass logit model in that we had to distinguish one class as the "baseline", but it is just as expressive as the symmetric multiclass logit model: any probability distribution that can be represented with one can be represented with the other. Therefore, any predictive inferences made using maximum-likelihood estimation techniques will be the same for the two approaches. Other techniques—such as Bayesian MAP parameter estimation or Bayesian prediction while "integrating out"—may lead to different results, however, due to the sensitivity of the prior to the structure of model parameterization.

Cases of this model where there are only two possible outcome classes are traditional *two-class* logit models (top left corner of Figure 6.16), which we covered in detail in Section

XXX. This is the type of log-linear model that the majority of readers are likely to have encountered first.

Returning to the general conditional log-linear case in the bottom-right corner of Figure 6.16, another option is to omit any sensitivity of feature functions to the input x. This is equivalent to throwing out the conditioning-variable part of the model altogether, and is sensible in cases such as our modeling of phonotactic knowledge (Section XXX), where simply wanted to infer a single probability distribution over English two-segment onsets. This decision takes us to the middle-right cell in Figure 6.16, UNCONDITIONAL log-linear models.

Finally, the unconditional log-linear model that we have here is closely related to another type of generalized linear model: POISSON REGRESSION. The key difference between unconditional log-linear models as we have described them here and Poisson regression is as follows: whereas our models have placed multinomial distributions over a set of possible response classes, the goal of Poisson regression is to put a probability distribution over *counts of observed events* in each possible response class. The two models are intimately related: if we take a fitted Poisson-regression model and use it to compute the joint probability distribution over counts in response class subject to the constraint that the total count of all response classes is 1, we get the same probability distribution that would be obtained using an unconditional log-linear model with the same parameters (Exercise 6.18). Although Poisson regression is popular in statistical modeling in general, we have not covered it here; it does turn up in some work on language modeling the frequencies of event counts in large corpora (e.g., Baayen, 2001).

6.12 Feed-forward neural networks

XXX

6.13 Further reading

There are many places to go for reading more about generalized linear models and logistic regression in particular. The classic comprehensive reference on generalized linear models is McCullagh and Nelder (1989). For GLMs on categorical data, Agresti (2002) and the more introductory Agresti (2007) are highly recommended. For more information specific to the use of GLMs and logistic regression in `R`, Venables and Ripley (2002, Section 7), Harrell (2001, Chapters 10–12), and Maindonald and Braun (2007, Section 8.2) are all good places to look.

Scheffé (1959) and Bock (1975) are comprehensive references for traditional ANOVA (including repeated-measures).

6.14 Notes and references

There are many good implementations of log-linear/maximum-entropy models publicly available; one that is simple to use from the command line, flexible, and fast is MegaM (Daumé and Marcu, 2006).

- Mention L_1 prior in addition to L_2 prior.

6.15 Exercises

Exercise 6.1: Linear regression

1. The `elp` dataset contains naming-time and lexical-decision time data by college-age native speakers for 2197 English words from a datset collected by Balota and Spieler (1998), along with a number of properties of each word. (This dataset is a slightly cleaned-up version of the `english` dataset provided by the `languageR` package; Baayen, 2008.) Use linear regression to assess the relationship between reaction time NEIGHBORHOOD DENSITY (defined as the number of words of English differing from the target word by only a single-letter edit). Is higher neighborhood density associated with faster or slower reaction times? Introduce written word (log-)frequency as a control variable. Does the direction of the neighborhood-density effect change? Is it a reliable effect (that is, what is its level of statistical significance)? Finally, is there an interaction between neighborhood density and word frequency in their effects on reaction time?

   Carry out this analysis for both word-naming and lexical-decision recognition times. In both cases, write a careful interpretation of your findings, describing not only what you found but what it might imply regarding how word recognition works. Construct visualizations of the main effects, and also of any interactions you find. If you find any qualitative differences in the way that the two predictors (and their interaction) affect reaction times, describe them carefully, and speculate why these differences might exist.

2. The dataset `nonwordsLexdec` presents average reaction times for 39 *non-word* letter sequences of English in a primed lexical decision experiment by Bicknell et al. (2010). The prime preceding the non-word always *was* a word, so trials were of the form *dish–kess*, *otter–peme*, and so forth. The dataset also contains neighborhood densities for each of the non-words, and word log-frequencies for the primes. Use linear regression to assess the relationship between neighborhood density and lexical-decision reaction time, controlling for prime log-frequency. Is the relationship between neighborhood density and reaction time the same as for the `english` dataset? Is the relationship reliable? Why do you see the results you see?

Exercise 6.2: Linear regression

The `durationsGe` dataset has as dependent variable the length of the Dutch prefix *ge-* in seconds. Use linear regression to investigate which of the following predictors have significant effects on prefix length:

- Word frequency
- Speaker sex
- Speech rate

Make sure to account for the possibility of interactions between the predictors. In addition, for word frequency and speech rate, use data visualization and `loess()` to get an intuition for whether to transform the predictors before putting them in the regression. (**Hint:** to get rid of rows in a data frame with NA's in them, the function `is.na()` is useful.)

Exercise 6.3: Analysis of variance

We talked about the idea of using a log-transformation on response variables such as reaction times to make them look more normal and hence be more faithful to the assumptions of linear models. Now suppose you are conducting a two-way ANOVA and are interested in the possibility of an interaction between the two factors. Your data are reaction times and look more normal when log-transformed. What are the potential consequences of log-transforming your response variable for investigating whether there is an interaction between your two predictors of interest? **Hint:** try constructing a set of four condition means for a two-by-two that reflect an additive pattern, and then look at the pattern when you take the log of each cell.

Exercise 6.4: Linear regression

Compare the residualization and multiple linear regression approaches. Imagine an underlying model of reading time of words in sentences in which the negative logs of raw word frequency (F_{log}) and contextual predictability (P_{log}) both play a role in determining the average reading time (RT, measured in milliseconds) of a given word. Take as the model of average reading time

$$\begin{aligned} RT &= 300 - 50F_{log} - 10P_{log} + \epsilon \\ \epsilon &\sim \mathcal{N}(0, 40) \end{aligned}$$

and suppose that F_{log} and P_{log} are generated from a multivariate normal distribution centered at $(-4, -4)$ with variance-covariance matrix $\left(\begin{smallmatrix} 0.7 & 0.5 \\ 0.5 & 1.2 \end{smallmatrix}\right)$. In this case where predictability and frequency are positively correlated, is your intuition that residualization or multiple linear regression will have greater statistical power in detecting the effect of predictability? (That is, on average which approach will yield a higher proportion of successful detections of a significance effect of predictability?) Test your intuitions by comparing residualization versus multiple linear regression approaches for detecting the effect of P_{log}. Generate 1000 sample datasets, each of size 200. Which approach has more statistical power in detecting the effect

of predictability? **Hint:** You can automatically extract a p-value from the t-statistic for a regression model parameter by looking at the fourth component of the `summary()` of an `lm` object (the result of `summary()` is a list), which is an array. For example:

```
> lexdec.lm <- lm(RT ~ Frequency, lexdec)
> summary(lexdec.lm)[[4]]
               Estimate  Std. Error    t value     Pr(>|t|)
(Intercept)  6.58877844 0.022295932 295.514824 0.000000e+00
Frequency   -0.04287181 0.004532505  -9.458744 1.026564e-20
> summary(lexdec.lm)[[4]][2,4]
[1] 1.026564e-20
```

Exercise 6.5: Decomposition of variance

Prove Equation 6.7.

Exercise 6.6: F tests and t statistics

With a randomly-selected 200-sample subset of the Spieler and Balota (1997) dataset, replicate the model comparisons reported in Section 6.5.2 and XXX

Exercise 6.7: Repeated measures and stratification of error

In English, the best-studied phonetic property distinguishing unvoiced stops ([p],[t],[k]) from voiced stops ([b],[d],[g]) is VOICE ONSET TIME (VOT): the time (typically measured in milliseconds) between (a) the acoustic burst corresponding to release of the stoppage of airflow in the vocal tract and (b) the onset of vibration of the vocal folds in the following sound (Liberman et al., 1958; Lisker and Abramson, 1967). Among other manipulations and measurements, Cho and Keating (2009) measured VOT for the first [t] in the invented name "Tebabet" (intended pronunciation [tɛbəbɛt]) in utterance-initial versus utterance-medial position, when the name was stressed:

(5) a. **Tebabet** fed them [Utterance-initial]
b. One deaf **Tebabet** [Utterance-medial]

Multiple native English speakers participated in this study, and Cho and Keating recorded several utterances of each sentence for each speaker. Hence this experiment involves a *repeated-measures* design. If we assume that different speakers may have individual idiosyncracies for the utterance-initial versus utterance-medial contrast, then we get the linear model

$$Y = \alpha + \beta X + a_i + b_i X + \epsilon$$

where X is the contrast between utterance-initial and utterance-medial position; a_i and b_i are the idiosyncracies of speaker i, distributed multivariate-normal around 0 with covariance matrix Σ; and ϵ is utterance-specific noise, also normally distributed around 0 with variance σ_2.

1. Demonstrate that applying a traditional (not repeated-measures) ANOVA according to Figure 6.9 for a repeated-measures study, in which we test a null-hypothesis model $M_0 : \beta = 0$ against an alternative-hypothesis model M_A with unconstrained β by comparing the variance explained by M_A over M_0 with the residual variance unexplained by M_A, will in general lead to anti-conservative inference. That is: assume $\beta = 0$; choose values of α, Σ, and σ^2, the number of speakers $m > 1$ and utterances per speaker $n > 1$; randomly generate N datasets using this model; analyze each dataset using a non-repeated-measures procedure; and report the proportion of models in which the null hypothesis would be rejected by the criterion $p < 0.05$.

2. Now demonstrate that the stratification-of-error procedure introduced in Section 6.6.5 avoids anti-conservative inference, through repeated generation of simulated data as in the first part of this problem.

Exercise 6.8: Outlier removal

Does outlier removal of the type introduced in Section 6.6.6 lead to anti-conservative inferences regarding differences between experimental conditions? Use simulations and/or mathematical analysis to support your claim. What if the criteria for outlier removal are determined separately for each experimental condition, instead of uniformly across conditions as done in Section 6.6.6?

Exercise 6.9: Analysis of variance.

Perform by-subjects and by-items repeated-measures ANOVA analyses of the second spillover region (`RC_VERB+2`) the Rohde et al. (2011) self-paced reading dataset. Try the results both with and without applying outlier removal; a typical outlier-removal criterion would be to discard observations more than either three or four standard deviations above the mean, with "mean" and "standard deviation" determined using only observations from the specific region being analyzed. How, if at all, does outlier removal change the results? Why do you think this is the case?

Exercise 6.10: The t-test versus Bayesian model comparison

Consider data that is generated from two normal distributions, with means $\mu_1 = 1$ and $\mu_2 = 2.5$, and with common noise $\sigma_\epsilon = 5$. Let's look at the power of the frequentist t-test versus a Bayesian model comparison in choosing between hypotheses H_0 in which the two distributions have the same mean, versus H_1 in which the two distributions have different means. Assume that our observations Y consist of 250 points from each distribution. For the Bayesian model comparison, use the specifications

$$\begin{aligned} \mu_1 &\sim \mathcal{N}(0, \sigma_\mu) \\ \mu_2 - \mu_1 &\sim \mathcal{N}(0, \sigma_\mu) \\ \sigma_\epsilon &\sim \mathcal{U}(1, 100) \end{aligned}$$

and the prior distribution $P(H_0) = P(H_1) = 1/2$. Using JAGS or similar sampling-based Bayesian inference software, plot the proportion of trials in which the posterior proability of H_0 is less then 0.05: $P(H_0|Y) < 0.05$, as a function of σ_μ. Explain the pattern you see in intuitive terms.

Exercise 6.11: Logistic regression

In analyzing the `dative` dataset [Section 6.7] we found in a logit model with linear predictor

```
RealizationOfRecipient ~ PronomOfRec + PronomOfTheme
```

that pronominality of recipient and theme had similar-sized but opposite effects (in logit space) on the probability of use of the prepositional-object construction. We tentatively interpreted this result as consistent with the idea that there is a general "pronouns like to be shifted left" constraint that operates with equal strength on recipients and themes.

1. The model above (call it M_1) has three free parameters. Define a new predictor variable that (a) is a function of the two variables `PronomOfRec` and `PronomOfTheme`; and (b) allows us to simplify the model above into a new model with only two free parameters.

2. Fit the model (call it M_2) to the `dative` dataset. How do the resulting parameter estimates compare to those of M_1?

3. Your new model M_2 should be nested inside M_1 (that is, it should be a special case of M_1). Explain this nesting—specifically, explain what special conditions imposed on M_1 result in equivalence to M_2. This nesting makes it possible to conduct a likelihood-ratio test between M_1 and M_2. Do this and report the p-value for the test. Does M_2 oversimplify the data compared with M_1?

Exercise 6.12

Use your knowledge of the English lexicon to explain why, in Table 6.4, [ts] and [sr] are so much more probable in the unigram model than in the other models, [st] is so much more probable in the bigram model than in the other models, and [tr] is so much less probable in the positional unigram model than in the other models.

Exercise 6.13

In Section **??**, the log-linear model of English onsets didn't include any conditioning information X. What conditioning information X might you include in a richer model of English onset phonotactics?

Exercise 6.14

Of the phonotactic models introduced in Section 6.10, which is the best predictive model with respect to the distribution of English onsets (as opposed to prediction of native speaker

non-word acceptability judgments)? Assess the cross-validated log-likelihood achieved by each model using ten-fold cross-validation.

Exercise 6.15

Consider the following frequency counts for the part of speech of the first word in each sentence of the parsed Brown corpus:

(Pro)noun	9192
Verb	904
Coordinator	1199
Number	237
(Pre-)Determiner	3427
Adverb	1846
Preposition or Complementizer	2418
wh-word	658
Adjective	433

Using a log-linear model with exactly one indicator feature function for each part of speech, demonstrate for yourself that the maximum-likelihood predictive distribution is simply the relative frequency estimate. Then introduce a Gaussian prior to your model. Plot the KL divergence from the MAP-estimated predictive distribution to the maximum-likelihood distribution as a function of the standard deviation of the prior.

Exercise 6.16

How would you obtain confidence regions for parameter estimates in a Bayesian log-linear model? After reading Appendix **??**, define and implement a Metropolis algorithm for sampling from the posterior distribution of a log-linear model with a Gaussian prior on the parameter estimates. Use this algorithm to generate confidence intervals for the feature weights in the models of Section 6.10.1. Which feature weights does the model have the most certainty about, and how should these features be interpreted? [HINT: you will save a lot of time if you use standard gradient-descent software to find the MAP-estimated feature weights and use these weights to initialize your sampling algorithm.]

Exercise 6.17

The file `word_suffixes` contains frequency counts from CELEX (Baayen et al., 1995) for all suffixes of English word lemmas constructible from 17 English phonemes which are of length 2 or less.

- Define a small set of feature functions (no more than in the neighborhood of 20) on the basis of generalizations you see in the data and write a script to automatically extract the outputs of these feature functions for each form in the frequency database.

- Fit a maximum-likelihood log-linear model from this output. Inspect the feature weights and the predictive distribution. What conclusions can you draw from the results?

- Introduce a Gaussian prior and fit a new log-linear model using MAP estimation. How do the feature weighs and the predictive distribution change?
- Based on any limitations you may see from the results of your first model, add new feature functions, refit the model, and discuss the changes you see between the simpler and more complex models.

Exercise 6.18

Use Bayes' Rule to show that when a fitted Poisson distribution with parameters $\boldsymbol{\lambda}$ is used to compute the probability distribution over counts in each response class subject to the constraint that the total count over all response classes is equal to 1, the resulting distribution is equivalent to that obtained by an unconditional log-linear model with the same parameters (see Figure 6.16).

Exercise 6.19: Symmetric versus baseline-class log-linear models and priors on the weights

Consider a simple model of the dative alternation, where the response variable Y is whether the recipient precedes or follows the theme, and the only predictor variable X is whether the recipient is pronominal. If we treat this as a symmetric, two-class problem we define the classes y_1 as recipient-first and y_2 as theme-first;

```
> ### part 3: no prior penalty on intercept, but penalty on all else
> library(rms)
> dat <- data.frame(x=rep(c("pro","pro","notPro","notPro"),c(8,2,2,8)),y=rep(c("NP","P
> m <- lrm(y~x,dat,penalty=1)
> predict(m,dat,type="fitted")

        1         2         3         4         5         6         7         8
0.2896893 0.2896893 0.2896893 0.2896893 0.2896893 0.2896893 0.2896893 0.2896893
        9        10        11        12        13        14        15        16
0.2896893 0.2896893 0.7103107 0.7103107 0.7103107 0.7103107 0.7103107 0.7103107
       17        18        19        20
0.7103107 0.7103107 0.7103107 0.7103107
```

Exercise 6.20: Interactions in a linear model

In the English Lexicon Project, data were collected from both younger participants (22.6± 5 y.o.) and older participants (73.4 ± 3 y.o.). For the word naming data you have availalbe from this project, analyze the effect of subject age (as a categorical variable: young vs. old) and its possible interaction with age of acquisition. Do younger participants name words faster or slower overall than older participants do? What is the effect of a word's age of acquisition on its naming latency? Is this effect any different for younger participants than for older participants? If so, how? If you see a significant difference, speculate on why the difference you see might exist.