CHAPTER 21

Understanding and summarizing the fitted models

Now that we can fit multilevel models, we should consider how to understand and summarize the parameters (and important transformations of these parameters) thus estimated.

Inferences from classical regression are typically summarized by a table of coefficient estimates and standard errors, sometimes with additional information on residuals and statistical significance (see, for example, the R output on page 39). With multilevel models, however, the sheer number of parameters adds a challenge to interpretation. The coefficient list in a multilevel model can be arbitrarily long (for example, the radon analysis has 85 county-level coefficients for the varying-intercept model, or 170 coefficients if the slope is allowed to vary also), and it is unrealistic to expect even the person who fit the model to be able to interpret each number separately. We prefer graphical displays such as the generic `plot` of a Bugs object or plots of fitted multilevel models such as displayed in the examples in Part 2A of this book.

Our general plan is to follow the same structures when plotting as when modeling. Thus, we plot data with data-level regressions (as in Figure 12.5 on page 266), and estimated group coefficients with group-level regressions (as in Figure 12.6). More complicated plots can be appropriate for non-nested models (for example, Figure 13.10 on page 291 and Figure 13.12 on page 293). More conventional plots of parameter estimates and standard errors (such as Figure 14.1 on page 306) can be helpful in multilevel models too. It is also sometimes feasible to display between-group and within-group information on the same graph (see Figure 14.11 on page 313). Finally, specific models can inspire new ideas for graphs, as in Figure 15.4 on page 330.

21.1 Uncertainty and variability

Uncertainty reflects lack of complete knowledge about a parameter; *variability* refers to underlying differences among individuals or groups. As sample size goes to infinity, uncertainty goes to zero (more precisely, in a well-designed study, uncertainty about key parameters goes to zero), but variability will always be present.

Distinguishing between uncertainty and variability in a multilevel model

For example, consider the varying-intercept radon model, $y_i \sim \text{N}(\alpha_{j[i]} + \beta x_i, \sigma_y^2)$. Here is a subset of the Bugs inferences for the parameters (more details appear on page 351):

R output

```
           mean   sd
a[1]        1.2  0.3
a[2]        0.9  0.1
. . .
```

```
a[85]       1.4  0.3
b          -0.7  0.1
mu.a        1.5  0.1
sigma.y     0.8  0.05
sigma.a     0.3  0.05
```

The numbers in the last column of this table show the *uncertainty* about each parameter. For example, we estimate the intercept for county 1 to be 1.2 with an uncertainty of 0.3. The parameters σ_y and σ_α represent *variability* among houses within a county and variability among counties. Thus, the unexplained variation of radon levels of houses within a county (after controlling for floor of measurement), is estimated at 0.8, and the unexplained variation among county-average radon levels is estimated at 0.3.

As we get more data within *existing* counties, uncertainty about individual α_j's would be expected to decline. If data from *more* counties were added, uncertainty about the group-level parameters μ_α and σ_α would be expected to decline. But under either (or both) of these scenarios, there is no reason to expect the estimates of σ_y and σ_α to change.

A varying-intercept, varying-slope example

For another example, consider the model from page 449 of variation in time trends of CD4 counts in a sample of children. The model is $y_i \sim \text{N}(\alpha_{j[i]} + \beta_{j[i]} t_i, \sigma_y^2)$. The `lmer` display includes both uncertainty and variation, and they can easily be confused:

R output

```
lmer(formula = y ~ time + (1 + time | person))
             coef.est coef.se
(Intercept)  4.8      0.2
time        -0.5      0.1
Error terms:
 Groups   Name        Std.Dev. Corr
 person   (Intercept) 1.3
          time        0.7      0.1
 Residual             0.7
# of obs: 369, groups: person, 83
```

There is no confusion about the coefficient estimates: 4.8 and -0.5 are the estimates of the average α_j, β_j in the population, so the "average person," in some sense, has a time trend of $4.8 - 0.5t$. But care must be taken in interpreting standard errors and standard deviations. We will focus here on the time trend (the slope in the model) because it is of more substantive interest. The *standard error* of the time coefficient is 0.1, which tells us the precision of the "-0.5" estimate: we are essentially certain that the population-average slope is negative, that is, that CD4 counts are on average declining in this population. The estimated *standard deviation* of the slopes is 0.7, which implies that the individual slopes vary greatly, with some being very negative and some being moderately positive. In this example, both the uncertainty in the mean and the variation in the population are important.

Variability can be interesting in itself

In some cases, the magnitude of variation can be substantively relevant. For example, Kane et al. (2006) estimate the "value added" by teachers in elementary and middle schools in New York City. They fit regression models of students' test

scores, given previous test scores, background information on students, classes, and schools, and varying intercepts for teachers.

The models were actually fit with classical regression using an indicator variable for each teacher, and then the estimates were post-processed to distinguish inferential variability and consistent teacher effects, with the latter identified based on correlations between the estimated teacher effects in successive years and in different classes taught by the same teacher within a year. For our purposes we can imagine that a multilevel varying-intercept model was fit. We have no particular interest in thousands of individual teacher effects, but we do care about the group-level coefficients (in this case, the "groups" are the teachers, and the predictors describe teacher certification and experience) and the residual standard deviation of teacher effects.

The analysis found the teacher characteristics to be weak predictors of value added—in particular, teachers with formal certifications and better college grades did essentially the same (as measured by the test scores of the students they taught), on average, compared to seemingly less-well-qualified teachers. In contrast, the unexplained teacher-level variation was large, with students of the best teachers scoring about 0.2 to 0.4 better than students of the worst teachers (after controlling for previous test scores and other covariates). This unexplained variation in teacher effects (on a scale in which 1 unit represents the standard deviation of scores of all students within a grade) is moderately large, and it led the researchers to conclude that "policies that enable districts to attract and retain high quality teachers (or screen-out less effective teachers) have potentially large benefits for student achievement."

21.2 Superpopulation and finite-population variances

Each factor in a multilevel model corresponds to a set of linear parameters or coefficients. Label the coefficients in a particular factor as $\alpha_1, \ldots, \alpha_J$, where J is the number of groups or categories of this factor. The variation among the α_j's can be summarized in two ways:

- The *superpopulation* standard deviation σ_α, which represents the variation among the modeled probability distribution from which the α_j's were drawn, is relevant for determining the uncertainty about the value of a new group not in the original set of J.
- The *finite-population* standard deviation s_α of the particular J values α_j describes variation within the existing data.

Example of a non-nested model

We illustrate with the simple model (13.9) for the flight simulator example introduced in Section 13.5. The model has three (non-nested) levels:

$$\begin{aligned} y_i &\sim \text{N}(\gamma_{j[i]} + \delta_{k[i]}, \sigma_y^2), \text{ for } i = 1, \ldots, n \\ \gamma_j &\sim \text{N}(\mu_\gamma, \sigma_\gamma^2), \text{ for } j = 1, \ldots, J \\ \delta_k &\sim \text{N}(\mu_\delta, \sigma_\delta^2), \text{ for } k = 1, \ldots, K. \end{aligned}$$

with diffuse normal prior distributions specified for μ_γ and μ_δ. When we specify the model this way, μ_γ and μ_δ are not separately identified; however we can compute summaries of interest using the ideas of redundant additive parameterization as described in Section 19.4.

The superpopulation standard deviations at each level are simply $\sigma_\gamma, \sigma_\delta, \sigma_y$. The finite-population standard deviations are

$$\begin{aligned} s_\gamma &= \sqrt{\frac{1}{J-1}\sum_{j=1}^{J}(\gamma_j - \bar{\gamma})^2} \\ s_\delta &= \sqrt{\frac{1}{K-1}\sum_{k=1}^{K}(\delta_k - \bar{\delta})^2} \\ s_y &= \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\epsilon_i - \bar{\epsilon})^2}. \end{aligned} \tag{21.1}$$

In defining s_y, we work with the data-level errors

$$\epsilon_i = y_i - \hat{y}_i = y_i - (\mu + \gamma_{j[i]} + \delta_{k[i]}).$$

Computation

In Bugs, the superpopulation standard deviations $\sigma_y, \sigma_\gamma, \sigma_\delta$ are already defined in the model (see the model at the beginning of Section 17.3). To define the finite-population standard deviations, we add the lines

Bugs code

```
for (i in 1:n){
  e.y[i] <- y[i] - y.hat[i]
}
s.y <- sd(e.y[])

s.g <- sd(g[])
s.d <- sd(d[])
```

When group-level predictors are present, they must be subtracted before computing the standard deviations as defined above. Suppose, for example, the airport coefficients γ_j had a group-level model with two predictors, u_1 and u_2:

Bugs code

```
for (i in 1:J){
  g[j] ~ dnorm (g.hat[j], tau.g)
  g.hat[j] <- a.0 + a.1*u.1[j] + a.2*u[2]
}
tau.g <- pow(sigma.g, -2)
```

Then `sigma.gamma` is the superpopulation standard deviation, and the corresponding finite-population standard deviation is defined by

Bugs code

```
for (j in 1:J){
  e.g[j] <- g[j] - g.hat[j]
}
s.g <- sd(e.g[])
```

When using the convenient $\gamma_j \sim \text{N}(\hat{\gamma}_j, \sigma_\gamma^2)$ notation, we can simply subtract $\hat{\gamma}_j$ to define the group-level error; we need not worry about how $\hat{\gamma}$ is constructed. (For example, the model for γ could itself include group-level parameters.)

The finite-population standard deviation is estimated more precisely than the superpopulation standard deviations

The superpopulation and finite-population standard deviations are *not* two different statistical "estimators" of a common quantity; rather, they are two different

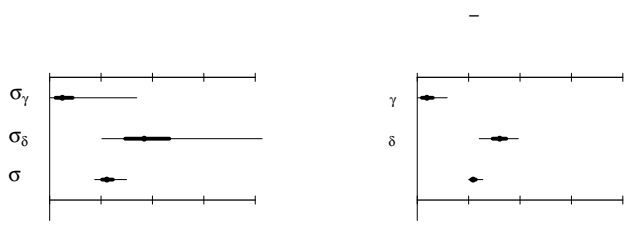

Figure 21.1 *Median estimates, 50% intervals, and 95% intervals for (a) superpopulation and (b) finite-population standard deviations of the treatment-level, airport-level, and data-level errors in the flight simulator example. The two sorts of standard-deviation parameters have essentially the same estimates, but the finite-population quantities are estimated much more precisely.*

quantities that can both be estimated from the multilevel model. We can get a point estimate and uncertainty intervals for both. In general, the point estimates of σ and s will be similar to each other, but s will have less uncertainty than σ. That is, the variation is more precisely estimated for the finite population than the superpopulation. This makes sense because we have more information about the units we observe than the full population from which they are sampled.

More generally, consider a hypothetical group-level model,

$$\alpha_j = U_j\gamma + \eta_j, \quad \eta_j \sim \text{N}(0, \sigma_\alpha^2), \quad \text{for } j = 1, \ldots, J.$$

The superpopulation standard deviation is simply σ_α, and the finite-population standard deviation is $s_\alpha = \text{sd}_{j=1}^J \eta_j = \sqrt{\frac{1}{J-1}\sum_{j=1}^J (\eta_j - \bar{\eta})^2}$.

Example with only two groups. Both measures of variation are important. To see how they differ, consider the extreme case in which $J = 2$ and there are no group-level predictors (so that, in classical estimation, there would be a constraint such as $\alpha_1 + \alpha_2 = 0$) and a large amount of data are available in both groups. In that case, almost nothing can be said about the superpopulation standard deviation, but the finite-population standard deviation can be still be estimated well if the sample size within groups is large. The two parameters α_1 and α_2 will be estimated accurately (in either a classical or a multilevel model) and so will $s_\alpha^2 = \frac{1}{2-1}((\alpha_1 - \bar{\alpha})^2 + (\alpha_2 - \bar{\alpha})^2) = (\alpha_1 - \alpha_2)^2/2$. The superpopulation variance σ_α^2, on the other hand, is only being estimated by a measurement from a distribution that is proportional to a χ^2 with 1 degree of freedom. We know much about the two parameters α_1, α_2 but can say little about others from their batch.

Fixed and random effects

As we discussed in Section 11.4, we believe that much of the statistical literature on fixed and random effects can be fruitfully reexpressed in terms of finite-population and superpopulation inferences. In some contexts (for example, collecting data on the 50 states of the United States), the finite population seems more meaningful; whereas in others (for example, subject-level effects in a psychological experiment), interest clearly lies in the superpopulation.

Chapter 22 applies finite-population standard deviations to the analysis of variance.

Finite-population standard deviations for regression coefficients that are not part of a multilevel model

A key idea in analysis of variance, as we discuss in Chapter 22, is to summarize each factor in a regression model by its finite-population standard deviation. As we have seen above, this is typically straightforward for batches of varying coefficients. For an unmodeled coefficients we can define finite-population standard deviations as the standard deviation of the coefficients in the population of predicted values.

We illustrate with some examples:

- *Binary predictor.* Suppose sex is included in a model as an indicator x that equals 1 for men and 0 for women. Label the proportion of men in the sample is λ. If the coefficient for x is β, then the finite-population standard deviation is simply the standard deviation of a random variable that equals β with probability λ and 0 with probability $1-\lambda$; this comes to $\sqrt{\lambda(1-\lambda)}|\beta|$.
- *Unmodeled categorical predictor.* Suppose age is included using three indicator variables corresponding to the categories 18–29, 30–44, and 45–64 (with the final category, 65+, being the baseline), and label the proportion of the sample in each category as $\lambda_1, \lambda_2, \lambda_3$, and $\lambda_4 = 1-\lambda_1-\lambda_2-\lambda_3$. If the coefficients for the three indicators are $\beta_1, \beta_2, \beta_3$, and (implicitly) $\beta_4 = 0$, then the finite-sample standard deviation is the standard deviation of a random variable that takes on these four values with these four probabilities. This can be easily computed in R from the vectors λ and β:

R code
```
b.mean <- sum (b*lambda)
b.sd <- sqrt (sum (lambda*(b-b.mean)^2))
```

Alternatively, we can perform the calculation directly using indexing:

R code
```
b.sd <- sd (b[x])
```

assuming `x` is the index variable for age that takes on the values 1, 2, 3, 4.

- *Continuous linear predictor.* Suppose age is simply included as a linear predictor x with coefficient β. The corresponding finite-population standard deviation is simply the standard deviation of x in the sample, multiplied by the absolute value of β.
- *Continuous nonlinear predictor.* Suppose age is included as a continuous predictor x with linear and quadratic coefficients, β_1 and β_2. The finite-population standard deviation is then the standard deviation of $\beta_1 + \beta_2 x$ in the data, and can simply be calculated in R as `sd(b[1]+b[2]*x)`, using the data vector `x`.

These examples do not cover all cases. For example, it is not clear how to define the standard deviation of variables that are also included as interactions.

21.3 Contrasts and comparisons of multilevel coefficients

A key difference between classical and multilevel models is the treatment of categorical or discrete inputs. As discussed in Section 11.4, in a classical regression it is possible to include group indicators or group-level predictors, but it is difficult to do both. (It is possible to perform two-level regression, first including group indicators and then regressing the estimated group indicators on group-level predictors—but this approach runs into difficulties when within-group sample sizes are small.)

Contrasts: including an input both numerically and categorically

Fortunately, including group indicators and also group-level predictors is straightforward and often useful in multilevel modeling. The only challenge comes in the interpretation of the group-level coefficients. We illustrate with an example from education.

Advanced Placement (AP) tests are taken by students in high school and are used by colleges to place students in introductory courses. As part of a study of the relevance of AP tests to the college curriculum, we performed a regression predicting the grades in introductory calculus classes for students who had taken the AP Calculus exam. The outcome of the regression was course grade,[1] and the inputs were AP score (which took on the discrete values 1, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, and 6.5; we label these as $u_j, j = 1, \ldots, 10$)[2] and Scholastic Aptitude Test (SAT) score (which we treated as a continuous predictor).

There are two natural ways to code AP score as regression predictors: as a single continuous predictor taking on numerical values u_j from 1 through 6.5, or as 10 indicators corresponding to its 10 different levels. Either of these approaches could make sense: on one hand, the AP grades are ordered, and it is reasonable, at least as a starting point, to consider the jump from a score of 1 to 2, or 2 to 3, and so forth, as corresponding to roughly equal jumps in course grade. On the other hand, nonlinear patterns are possible: for example, perhaps scores of 4 and 5 are both high enough to assure a good grade in the calculus class, so that little benefit would be seen beyond a score of 4.

Classical regression. In a classical regression, we can include AP score as a continuous predictor (and also, for example, include the square of AP score to model nonlinearity)—or we could include indicators for four of the five AP scores, for example, treating a score of 1 as the baseline category. But it would not be possible to do both, because the linear trend is collinear with the set of indicators.

That is, we can model the coefficients for the AP scores, $\alpha_j, j = 1, \ldots, 5$, as $\alpha_j = \gamma_0 + \gamma_1 u_j$ (the linear model), or $\alpha_j = \gamma_0 + \gamma_1 u_j + \gamma_2 u_j^2$ (the quadratic model), or simply as 10 different α_j's—but there is no way in a classical model to include the linear or quadratic trend *and* the separate coefficients for each level.

Multilevel regression. In a multilevel model we can include AP both as a continuous predictor and as indicators: $\alpha_j \sim \mathrm{N}(\gamma_0 + \gamma_1 u_j, \sigma_\alpha^2)$ or, equivalently,

$$\alpha_j = \gamma_0 + \gamma_1 u_j + \eta_j, \tag{21.2}$$

with group-level errors $\eta_j \sim \mathrm{N}(0, \sigma_\alpha^2)$. This model allows the α_j's to have arbitrary levels, but with the group-level model (21.2) pulling the estimates toward linearity to the extent supported by the data.

The two classical models (linearity at one extreme or 10 separate coefficients at the other extreme) are special cases of the multilevel model, corresponding to $\sigma_\alpha = 0$ or ∞, respectively.

For another example of this sort of multilevel model, see Figure 13.12 on page 293. We further discuss contrasts in the context of the analysis of variance in Section 22.5.

[1] Grades of A+, A, A-, B+, ... were coded as 4.3, 4.0, 3.7, 3.3, ... For simplicity we treated the grades as continuous measurements.

[2] The AP Calculus test had two different forms, each on its own 1–5 scale. We performed a preliminary analysis to equate the two tests and put them on a common scale by adding 1.5 to scores on the more difficult test form.

The finite-population group-level coefficient. A challenge remains in interpreting the group-level slope, γ_1 in model (21.2). The difficulty is in the partial nonidentifiability between γ_1 and the η_j's—more precisely, between γ_1 and whatever linear trend there is in the η_j's. This is only a partial nonidentifiability, because the η_j's are pooled toward zero by the multilevel model—but with only 10 groups, it creates practical problems.

In this particular example, the estimated slope among the 10 groups was clearly positive—α_j increased with j, roughly linearly with the score u_j. But the inference for the group-level slope γ_1 had a wide standard error—the point estimate of γ_1 was positive but its 95% confidence interval actually included zero and negative values. The problem arose because there was a possibility, with only $J = 10$ groups, that the finite-sample regression of the η_j's was strongly positive or negative.

Thinking about the problem, we realized that what we really wanted was the *finite-population* slope, that is, the linear regression of the α_j's on the u_j's for these 10 groups. This finite-population quantity will be estimated more precisely than the superpopulation slope γ_1.

Having fit the model in Bugs, one can quickly compute a 95% interval, for example, for the finite-population slope in R:

R code

```
attach.bugs (AP.fit)
finite.slope <- rep (NA, n.sims)
for (s in 1:n.sims){
  finite.pop <- lm (a[s,] ~ u)
  finite.slope[s] <- coef(finite.pop)["u"]
}
quantile (finite.slope, c(.025,.975))
```

This can be compared, for example, to the 95% interval for γ_1, which will be wider. Performing this regression on the multilevel coefficients is equivalent to constraining the group-level errors η_j in (21.2) to have a mean of 0 and a slope of 0. The step can thus be thought of as a generalization of the additive redundant parameterization of Section 19.4.

Inferences for multilevel parameters defined relative to their finite-population average

Sometimes each individual coefficient is not of intrinsic interest but we are interested in comparisons among the coefficients. Moreover, defining predictors relative to each other yields much more stable estimates when sample sizes are small. We here discuss both the finite-population and superpopulation contrasts that can be estimated using multilevel models.

Consider the flight simulator data in Section 13.5, where the airport coefficients δ_k come from a distribution with common mean μ_δ. The left panel of Figure 21.2 shows the estimates and standard errors for these parameters. These reflect superpopulation contrasts because they measure the position of each airport relative to the population mean μ_δ.

Alternatively, we can define the varying airport intercepts relative to the mean of these coefficients in our sample, computing an adjusted coefficient $\delta_k^{\text{adj}} = \delta_k - \bar{\delta}$ for each airport k. This sort of mean shift is motivated in Section 19.4 for computational purposes—to help the Gibbs sampler converge faster—but it also gives us more precise inferences about the relative values of the airport coefficients for this sample. The right panel of Figure 21.2 shows the estimates and standard errors for the mean-centered airport coefficients. The point estimates are in the same place but

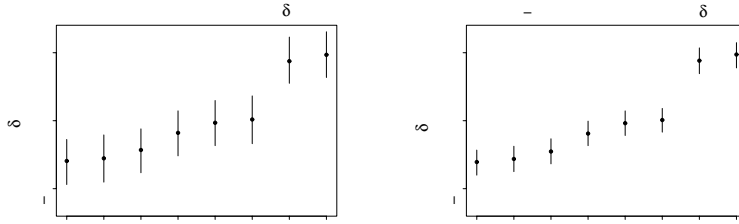

Figure 21.2 *Estimates and standard errors for the airport coefficients in the flight simulator example: (a) parameters with population mean subtracted out, $\delta_k - \mu_\delta$; (b) parameters centered around the sample mean $\delta_k^{\text{adj}} = \delta_k - \bar{\delta}$. The point estimates are the same for the two sets of parameters, but the centered versions have smaller, finite-sample standard errors. The airports have been ordered in increasing average response (see Figure 13.8 on page 290).*

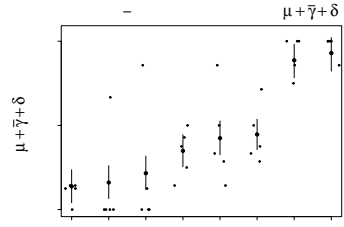

Figure 21.3 *Estimates and standard errors for the airport coefficients in the flight simulator example, after adding in the average values of the other terms in the regression model, thus moving to the scale of the data. The small dots in the graphs show the data. (It makes sense that many of the data fall outside the standard-error bounds, which represent inferential uncertainty about the airport coefficients but do not include the data-level error that is in the model.)*

the finite-population standard errors are smaller than the superpopulation standard errors used in the $\delta_k - \mu_\delta$ contrast.

Adding the average value of the other predictors back in. Another way of summarizing the differences across airports is to add the average values of the other terms in the regression, to create coefficients that are on the scale of the original data. For the flight simulator example, the other terms in the regression model are the mean level μ and the treatment effects γ_j:

$$y_i \sim \text{N}(\mu + \gamma_{j[i]} + \delta_{k[i]}, \sigma_y^2).$$

The average value of the other coefficients is μ plus the average of the γ_j's, and so the shifted-to-the-data-scale airport coefficients are $\mu + \bar{\gamma} + \delta_k$ for the airports $k = 1, \ldots, K$. Figure 21.3 shows the inference for these shifted parameters, with the data displayed also. In essence, then, these parameters represent the predicted success rates for each airport for the average treatment. (The plot reveals some potential poor fit of the linear model as fit to the data y_i, which in fact are success rates that

are bounded between 0 and 1. But here we are simply focusing on the mechanics of displaying and understanding the coefficients in a non-nested multilevel model.)

This is the method we used to display the fitted model and data in the various plots in Chapters 12–15.

21.4 Average predictive comparisons

Section 5.7 discusses how to calculate average predictive differences for logistic regressions. The challenge is that the model is on the logistic scale but we would like to interpret changes on the probability scale. Further complications arise with multilevel models, and we discuss the general issue of average predictive comparisons here.

Notation and basic definition of predictive comparisons

Assume you have fit a probability model predicting a numerical outcome y based on vectors x of inputs and θ of parameters. As in Section 3.4, we denote the separate sources of information in x that are used in the predictive model as *inputs* or *input variables*, as distinguished from the *linear predictors* that form the columns of the design matrix of a linear or generalized linear model. For example, consider a logistic regression of some individual outcome on age, sex, an age × sex interaction, and age^2. There are two inputs (age and sex) but five linear predictors (including the constant term).

In any model, we consider the scalar inputs one at a time, using the notation

u : the input of interest,

v : all the other inputs.

Thus, $x = (u, v)$. We focus on the expected predictive difference in y per unit difference in the input of interest, u, with v (the other components of x) held constant:

$$b_u(u^{(\text{lo})}, u^{(\text{hi})}, v, \theta) = \frac{\text{E}(y|u^{(\text{hi})}, v, \theta) - \text{E}(y|u^{(\text{lo})}, v, \theta)}{u^{(\text{hi})} - u^{(\text{lo})}} \tag{21.3}$$

and the average predictive difference per unit of u:

$$B_u(u^{(\text{lo})}, u^{(\text{hi})}) = \frac{1}{n}\sum_{i=1}^{n} b_u(u^{(\text{lo})}, u^{(\text{hi})}, v_i, \theta). \tag{21.4}$$

We assume that $\text{E}(y|x, \theta)$ is a known function (such as the inverse-logit) that can be computed directly. For a linear model with no interactions, b_u does not depend on $u^{(1)}, u^{(2)}$, or v, and is simply the regression coefficient associated with u. More generally, however, b_u varies as a function of these inputs, and it can be useful to summarize the predicted difference with some sort of weighted average as in (21.4). In practice, one must also average over θ (or plug in a point estimate).

Expressions (21.3) and (21.4) are similar to (5.8) and (5.9) on page 103 except in dividing by $u^{(\text{hi})} - u^{(\text{lo})}$. For some settings the total difference is relevant; other times we are interested in the difference per unit of u.[3] As we shall see, options also arise in the definition and estimation of average predictive changes with continuous inputs u. This section presents one particular set of choices and illustrates with an example.

[3] We use the notation δ for predictive differences and Δ for their averages, and the notation b, B for ratio comparisons of the form (21.3) and (21.4), by analogy to regression coefficients β.

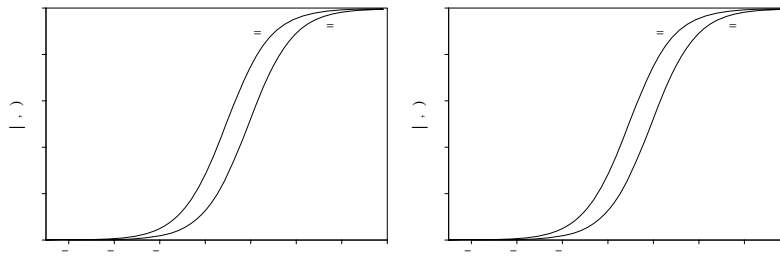

Figure 21.4 *Hypothetical examples illustrating the need for averaging over the distribution of v (rather than simply working with a central value) in computing the predictive comparison. Each graph shows a hypothesized logistic regression model for an outcome y given a binary input of interest, u, and a continuous input variable, v. The vertical lines on the x-axis indicate the values of v in the hypothetical dataset. (a) In the left plot, data v are concentrated near the ends of the predictive range. Hence the average predictive comparison is small. In contrast, $E(v)$ is near the center of the range; hence the predictive comparison at this average value is large, even though this is not appropriate for any of the data points individually. (b) Conversely, in the right plot, the average predictive comparison is reasonably large, but this would not be seen if the predictive comparison were evaluated at the average value of v.*

Problems with evaluating predictive comparisons at a central value

A rough approach that is sometimes used is to evaluate $\mathrm{E}(y|u, v)$ at a central value v_0—perhaps the mean or the median of the data—and then estimate predictive comparisons by holding v constant at this value. Evaluating differences about a central value can work well in practice, but one can run into problems when the space of inputs is very spread out (in which case no single central value can be representative) or if many of the inputs are binary or bimodal (in which case the concept of a "central value" is less meaningful). In addition, this approach is hard to automate since it requires choices about how to set up the range for each input variable. In fact, our research in this area was motivated by practical difficulties that can arise in trying to implement this central-value approach.

We illustrate some challenges in defining predictive comparisons with a simple hypothetical example of a logistic regression model of data y on a binary input of interest, u, and a continuous input variable, v. The curves in each plot of Figure 21.4 show the assumed predictive relationship. In this example, u has a constant effect on the logit scale but, on the scale of $\mathrm{E}(y)$, the predictive comparison b_u (as defined in (21.3), with $u^{(\mathrm{lo})} = 0$ and $u^{(\mathrm{hi})} = 1$) is high for v in the middle of the plotted range and low at the extremes. As a result, the average predictive comparison B_u, defined in (21.4), depends on the distribution of the other input, v.

The two plots in Figure 21.4 show two examples in which the average predictive comparison differs from the predictive comparison evaluated at a central value. In the first plot, the data are at the extremes, so the average predictive comparison is small—but the predictive comparison evaluated at $\mathrm{E}(v)$ is misleadingly large. The predictive comparison in y corresponding to a difference in u is small, because switching u from 0 to 1 typically has the effect of switching $\mathrm{E}(y|u, v)$ from, say, 0.02 to 0.05 or from 0.96 to 0.99. In contrast, the predictive comparison if evaluated at

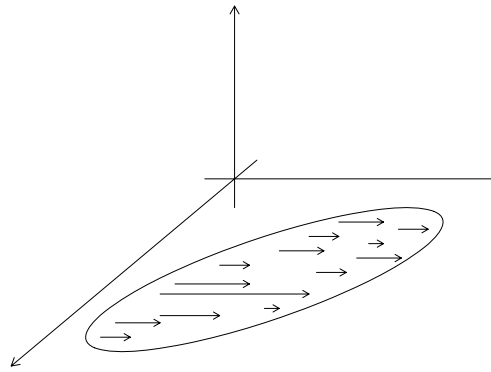

Figure 21.5 *Diagram illustrating differences in the input of interest u, with the other inputs v held constant. The ellipse in (u, v)-space represents the joint distribution $p(u, v)$, and as the arrows indicate, we wish to consider differences in u in the region of support of this distribution.*

the mean of the data is large, where switching u from 0 to 1 switches $\mathrm{E}(y|u, v)$ from 0.36 to 0.60.

The second plot in Figure 21.4 shows a similar example, but where the centrally computed predictive comparison is too low compared to the population-average predictive comparison. Here, the centrally located value of v is already near the edge of the curve, at which point a difference in u corresponds to only a small difference in $\mathrm{E}(y|u, v)$ of only 0.03. In comparison, the average predictive comparison, averaging over the data locations, has the larger value of 0.11, which appropriately reflects that many of the sample data are in the range where a difference in u can correspond to a large difference in $\mathrm{E}(y)$.

General approach to defining population predictive comparisons

The basic predictive comparison b_u defined in (21.3) depends in general on $u^{(\mathrm{lo})}$ and $u^{(\mathrm{hi})}$ (the beginning and end points of the hypothesized difference in the input of interest), v (the values of the other inputs), and θ (the parameters of the model). We define the *average predictive comparison* b_u as the mean value of b_u over some specified distribution of the inputs and parameters. We apply this idea to various sorts of inputs u, starting with numerical input variables, including continuous and binary inputs as special cases, and moving to unordered categorical variables, interactions, and constraints.

It turns out that the form of the input of interest u, not the form of data y or other predictors v, is crucial in deciding how to define average predictive comparisons. In any application, we would compute the average predictive comparison of each of the inputs to a model one at a time—that is, treating each component of x in turn as the "input of interest." This is often a goal of regression modeling: estimating the predictive comparison of each input with all other inputs held constant.

Averaging over $u^{(\mathrm{lo})}, u^{(\mathrm{hi})}$, and v in this way is equivalent to counting all pairs of transitions of $(u^{(\mathrm{lo})}, v^{(1)})$ to $(u^{(\mathrm{hi})}, v^{(2)})$ in which $v^{(1)} = v^{(2)}$—that is, differences in u with v held constant. Figure 21.5 illustrates. We average over θ using simulations

from the fitted model (as obtained, for example, in Bugs), which in a Bayesian context is the posterior distribution, or classically could be a point estimate or an uncertainty distribution defined by simulations (as discussed at the end of the previous section). The distributions of (u, v) and θ are independent because we are working in a regression context in which θ represents the parameters of the model for y conditional on u and v.

In the special case in which u is a binary input, the average predictive comparison is a simple average of the differences $(\mathrm{E}(y|u^{(\mathrm{hi})}, v, \theta) - \mathrm{E}(y|u^{(\mathrm{lo})}, v, \theta))$. More generally, the average predictive comparison has the form of a ratio of averages.

Models with interactions

These definitions automatically apply to models with interactions. The key is that u represents a single input, and $x = (u, v)$ represents the vector of inputs to the predictive model. As discussed throughout this book, the vector of inputs is not in general the same as the vector of linear predictors. For example, in the simple example at the beginning of this section, sex is included on its own and also interacted with age. When defining the predictive comparison for sex, we must alter this input wherever it occurs in the model—that is, both the "sex" predictor and the "sex × age" predictor must be changed. For another example, the constant term in a regression is *not* an input in our sense and has no corresponding predictive comparison, since it can take on only one possible value.

From a computational perspective, it is important that the model be coded in terms of its separate inputs. Thus, to compute predictive comparisons, it is not enough simply to specify the design matrix of a regression model; one must be able to evaluate $\mathrm{E}(y)$ as a function of the original inputs.

Inputs that are not always active

A model will sometimes have inputs that are involved in prediction for only some of the data. For example, consider an experiment in which some units are given the control (no treatment) and others are given the treatment, in doses ranging from 10 to 20 (on some meaningful scale). Suppose the data are fit by a generalized linear model with treatment indicator, dose, and some pre-treatment measurements as predictors.

Now consider how to define the average predictive comparison for dose. One approach is to consider treatment and dose to be a single input with value 0 for control units and the dose for treated units. This will not be appropriate, however, if we are particularly interested in the effect of dose in the range 10 to 20, conditional on treatment. We can define the predictive comparison for dose, restricting all integrals over v to the subspace in which dose is defined (in this case, the treated units).

We can formally define the average predictive comparison for a partially active input u by introducing a function $\zeta_u(v)$ that equals 1 wherever u is defined and 0 elsewhere. Then all the earlier definitions hold, as long as we insert the factor $\zeta_u(v)$ in all the integrals.

Average predictive comparisons for multilevel models

In a linear multilevel model, one might use the standard deviation of a batch of coefficients as a measure of their importance for predicting the outcome variable. But in a nonlinear model, this does not directly transfer to the scale of measurement.

For the purpose of defining average predictive comparisons, we can treat a batch of K parameters $\phi_k, k = 1, \ldots, K$, as an unordered categorical input variable with K levels $u^{(k)}$. The essence of a variance components model is that the parameters ϕ_k in a batch are considered to have been drawn from a continuous population distribution. In the following example we consider vector ϕ_k's.

Example: a multilevel logistic regression of prison sentences

We illustrate average predictive comparisons in multilevel models with a model of the geographic variation of the severity of prison sentences in the United States. A study was performed of the sentences of 8446 convicted felons in 39 of the 75 most populous counties in the United States during May 1998. The outcome variable for this study was "sentence severity," defined as $y_{ij} = 1$ if the offender i in county j received a prison sentence, or 0 for a jail or noncustodial sentence (considered to be much less severe than prison). A multilevel logistic regression model was fit with 12 individual-level variables from the State Court Processing Statistics program of the Bureau of Justice Statistics, linked to six county-level variables using the Federal Information Processing Standards code. Information collected in this program includes demographic characteristics, criminal history, details of pretrial processing, disposition, and sentencing of felony defendants.

Under the model, the outcomes are independent with probabilities

$$\Pr(y_i = 1) = \text{logit}^{-1}\left(X_i G_{j[i]} \eta + X_i \alpha_{j[i]}\right), \tag{21.5}$$

where X_i represents a vector of measurements on K individual-level variables and G_j is a $K \times M$ block-diagonal matrix of measurements on L county-level variables. In particular, interactions between individual- and county-level variables account for dependence of individual-level comparisons across counties, so that $M = KL$ if all county-level variables are used to explain these individual-level comparisons. The coefficients η in (21.5) represent main effects and interactions of the predictors and are constant across counties, and the unmodeled coefficients in the vector α have a j-subscript and represent varying coefficients across counties, or can be viewed as interactions between the predictors X and the county indicators j.

Applying average predictive comparisons to a single model

Fitting this two-level multilevel model is straightforward; however, the presence of individual-county interactions and varying county coefficients complicates the interpretation of the parameters η and α_j. In contrast, average predictive comparisons provide a clear indication of the overall contribution of each variable to the probability of receiving a prison sentence (rather than a jail or noncustodial sentence).

Figure 21.6 displays the average predictive comparison for each variable in the model, together with a average predictive comparison for the county indicators. Horizontal bars indicate $\pm$ 1 standard error for each average predictive comparison, calculated as described earlier in this section. Due to computational limitations, we based all calculations on a randomly drawn subset of $L = 100$ posterior samples, with $n = 4500$ data points for the binary inputs, $n = 450$ for the continuous inputs,

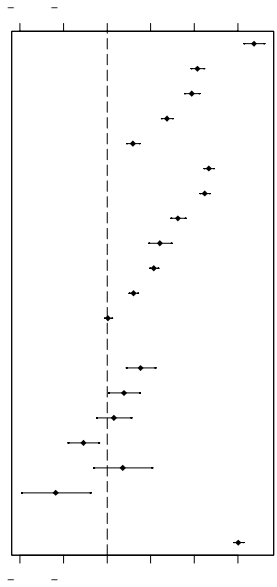

Figure 21.6 *Estimated average predictive comparisons B_u for the probability of a prison sentence (rather than a jail or noncustodial sentence), for each input variable u in the prison example. Horizontal lines show ±1 standard-error bounds. The first set of inputs, with initial letters I, are at the level of the individual offender; the second set, with initial letters C, are county-level inputs; and the last corresponds to the differences associated with varying the county indicator, keeping all other inputs constant. Many of the individual predictors are associated with large differences and county itself predicts a fair amount of variability, but the county-level variables are associated with relatively small differences (recall that these average predictive comparisons correspond to differences of one standard deviation in each input).*

and $n = 4000$ for the varying county coefficients. Results varied little on repeating the calculations with different random subsets.

Individual-level variables in Figure 21.6 are denoted with an initial "I," and county-level variables are denoted with an initial "C." The five individual-level variables for "most serious conviction charge" (ICVIOL1, ICTRAF, ICVIOL2, ICPROP, and ICDRUG) are relative to a reference category of weapons, driving-related, and other public order offenses. The 12 individual-level variables and 2 of the county-level variables are binary, and the remaining 4 county-level variables are continuous. Finally, the county indicators are a batch of parameters in a multilevel model.

The individual-level predictor associated with the largest difference in the probability of receiving a prison sentence is ICVIOL1 (murder, rape, or robbery), with an estimated average predictive comparison of 0.38 (and standard error 0.03). That is, the expected difference in the probability of receiving a prison sentence between a randomly chosen individual in the population charged with murder, rape, or robbery and a similar individual charged with a reference category offense is 0.38.

Other charges are less highly associated with a prison sentence, with decreasing probability: drug trafficking (with an estimated average predictive comparison of 0.21), then assault (0.19), then property offenses (0.14), and finally drug possession offenses (0.05). Other individual-level variables can be interpreted similarly, as can the two binary county-level variables (which compare individuals in southern and non-southern counties, and individuals in counties with and without state sentencing guidelines).

Predictive comparisons for the continuous input variables correspond to changes of 1 standard deviation in each, one at a time. Standard deviations for the four variables (CCONS, CCRIME, CBLPCT, and CUNEMP) are 13%, 220 per 10,000, 12%, and 1.8%, respectively. The positive average predictive comparisons for CCONS and CCRIME suggest that, comparing otherwise-similar cases, those in counties with higher conservative populations or higher crime rates have slightly higher probabilities of receiving a prison sentence. Conversely, the negative average predictive comparison for CUNEMP suggests lessened sentence severity in high-unemployment counties, with all other inputs fixed. Taking into account all other factors, the proportion of the county's population that is African American (CBLPCT) has little bearing by itself on sentence severity. (However, this factor does play a role in reducing or increasing the contributions of various individual-level variables, as can be seen in more detailed analysis.)

The average predictive comparison for the county indicators differs from the other average predictive comparisons in this example in that it considers just the magnitude, rather than the sign, of the comparisons. This is because "county" is the only unordered categorical variable in the model. To understand its average predictive comparison, consider the probabilities of receiving a prison sentence for two individuals who are identical in all respects except that they are in different counties. So, the two individuals will share the same values for individual-level variables but have different values for county-level variables (and county indicators). The county average predictive comparison of 0.37 represents the root mean square of the difference in the probability of receiving a prison sentence between a randomly chosen individual in one county and a similar individual in another county.

Using average predictive comparisons to compare models

The multilevel logistic regression model (21.5) has varying coefficients for the within-county intercepts as well as for each individual predictor. We also fit a multilevel model with varying intercepts only, as well as a non-multilevel complete pooling model that ignores the multilevel nature of the data and excludes county indicators. Whereas the regression coefficients have different interpretations for the different models, predictive comparisons allow for direct comparison.

Figure 21.7 displays the average predictive comparison for each variable across all three models. The average predictive comparisons and standard errors are very similar across the two multilevel models, perhaps suggesting that the additional variation for each individual predictor may be redundant. The individual-level comparisons are also very similar for the non-multilevel model. However, the county-level comparisons tend to be smaller in magnitude and have smaller standard errors for the non-multilevel model; the average predictive comparison for unemployment, CUNEMP, even has the opposite sign. The non-multilevel model did not fit this dataset well, and the average predictive comparisons displayed here suggest that while individual-level comparisons may be robust to model misspecification of this nature, higher-level comparisons can clearly be adversely affected.

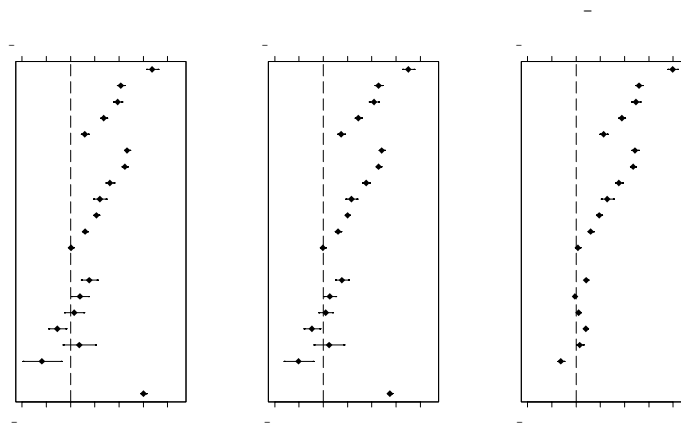

Figure 21.7 *Estimated average predictive comparisons for the probability of a prison sentence for each input variable across three models in the prison example. Horizontal lines show* ± 1 *standard-error bounds. Estimates and standard errors are very similar across the two multilevel models. However, although the individual-level comparisons are similar for the non-multilevel model, the county-level comparisons tend to be smaller in magnitude and have smaller standard errors.*

Predictive summaries in practice

Predictive comparisons, like any other automatic summary of a model, cannot be universally applicable, because the best approach in any problem must be tailored to the specifics of the application. We agree with this point but note that the overwhelming current practice in applied statistics of regression models is simply to report coefficient estimates (and standard errors), with no sense of their implications on the original scale of the data. Average predictive comparisons are not intended to be a replacement for regression coefficients; rather, they summarize a model in a way that can complement the coefficient estimates in order to make their scale more interpretable. Thus, we agree that there is no such thing as a "one size fits all" method—but that is what the current standard approach implicitly assumes.

The example illustrates the effectiveness and convenience of predictive comparisons. In this multilevel dataset with a binary outcome measure, the comparisons clarify the overall role of each individual- and group-level predictor in the presence of multiple interactions, as well as illustrate the relative size of the varying coefficients. They can also be used to understand and compare models directly, in a way that is difficult to do using logistic regression coefficients.

21.5 R^2 and explained variance

As discussed in the context of the examples in Chapters 3 and 4, it can be helpful to understand a model through R^2, the proportion of variance explained by its linear predictors. Although explained variance can be misleading (as illustrated in Figure 3.9 on page 42), it can be a useful measure of the relative importance of different sources of variation in a particular dataset. Here we discuss how to generalize R^2 to multilevel models.

Explained variation in linear models

Consider a linear regression written as $y_i = X_i\beta + \epsilon_i$, $i = 1, \ldots, n$. One way to summarize the fit of the regression is by the proportion of variance explained:

$$R^2 = 1 - \frac{\mathop{\mathrm{V}}_{i=1}^{n} \epsilon_i}{\mathop{\mathrm{V}}_{i=1}^{n} y_i}, \tag{21.6}$$

where V represents the finite-sample variance operator, $\mathop{\mathrm{V}}_{i=1}^{n} x_i = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$. In a multilevel model, the predictors "explain" the data at different levels, and R^2 can be generalized in a variety of ways.

One sort of definition of "explained variance," which we do *not* want to use, is $R^2 = 1 - \frac{\text{residual variance under the larger model}}{\text{residual variance under the null model}}$, with various choices of the null model corresponding to predictions at different levels. This definition is not so helpful to us because our goal is to understand a particular model being fit on its own terms.

Here we present a slightly different approach, computing (21.6) at each level of the model and thus coming up with several R^2 values for any particular multilevel model. This approach has the virtue of summarizing the fit at each level and requiring no additional null models to be fit. In defining this summary, our goal is not to dismiss other definitions of R^2 but rather to add another tool to the understanding of multilevel models.

We introduce notation (to which we shall return in the next chapter in the context of the analysis of variance) for a multilevel model with M error terms. (For example, $M = 2$ in the varying-intercept model of radon in houses within counties. For the varying-intercept, varying-slope model, $M = 3$, with regression models for house radon levels, county intercepts, and county slopes.) For convenience, in this section and the next, we shall refer to each error term—that is, the data and each batch of modeled parameters—as a "level" of the model.

For each level m, we write the model as

$$\theta_k^{(m)} = \hat{\theta}_k^{(m)} + \epsilon_k^{(m)}, \quad \text{for } k = 1, \ldots, K^{(m)}, \tag{21.7}$$

where the $\hat{\theta}_k^{(m)}$'s are the linear predictors (that is, the linear combination of coefficients and predictors) at that level of the model, and the errors $\epsilon_k^{(m)}$ come from a distribution with mean zero and standard deviation $\sigma^{(m)}$. At the lowest (data) level of the model, the $\theta_k^{(m)}$'s correspond to the individual data points (the y_i's in the radon model). At higher levels of the model, the $\theta_k^{(m)}$'s represent batches of comparisons or regression coefficients (county intercepts α_j and slopes β_j in the radon model). Because we work with each batch of parameters separately, we shall suppress the superscripts (m) for the rest of this section.

Proportion of variance explained at each level

For each level (21.7) of the model, we first consider the variance explained by the linear predictors $\hat{\theta}_k$. Generalizing from the classical expression (21.6), we define

$$R^2 = 1 - \frac{\mathrm{E}\left(\mathop{\mathrm{V}}_{k=1}^{K} \epsilon_k\right)}{\mathrm{E}\left(\mathop{\mathrm{V}}_{k=1}^{K} \theta_k\right)}, \tag{21.8}$$

where the expectations average over the uncertainty in the fitted model (using the posterior simulations). Any particular multilevel model will then have more than one R^2: one for the data level and one for each batch of modeled parameters. In our simulation-based approach to inference, the expectations in the numerator and denominator of (21.8) can be evaluated by averaging over posterior simulation draws, as we discuss later in this section.

R^2 will be close to 0 when the average residual error variance is approximately equal to the average variance of the θ_k's. R^2 will be close to 1 when the residual errors ϵ_k are close to zero for each posterior sample. Thus, R^2 is larger when the $\hat{\theta}_k$'s more closely approximate the θ_k's.

In classical least squares regression, (21.8) reduces to the usual definition of R^2: the numerator of the ratio becomes the residual variance, and the denominator is simply the variance of the data. Averaging over uncertainty in the regression coefficients leads to a lower value for R^2, as with the classical "adjusted R^2" measure. We shall discuss this connection further. It is possible for our measure (21.8) to be negative, much like adjusted R^2, if a model predicts so poorly that, on average, the residual error variance is larger than the variance of the data.

As a model improves (by adding better predictors and thus improving the $\hat{\theta}_k$'s), we would generally expect both R^2 and the amount of pooling (discussed more thoroughly in the next section) to increase for all levels of the model. Increasing R^2 corresponds to more of the variation being explained at that level of the regression model, and a high level of pooling implies that the model is pulling the ϵ_k's strongly toward the population mean for that level.

Adding a predictor at one level does *not* necessarily increase R^2 and the amount of pooling at other levels of the model, however. In fact, it is possible for an individual-level predictor to improve prediction at the data level but decrease R^2 at the group level (see Section 21.7). Here we merely note that a model can have different explanatory power at different levels.

Connections to classical definitions

Our general expression for explained variance reduces to classical R^2 for simple linear regression with the least squares estimate for the vector of coefficients. We present this correspondence here, together with the less frequently encountered explained variance for the basic multilevel model.

The classical normal linear regression model can be written as $y_i = X_i\beta + \epsilon_i, i = 1, \ldots, n$, with a $n \times p$ matrix X of predictors and errors ϵ_i that are normal with zero mean and constant variance σ^2.

If we plug in the least squares estimate, $\hat{\beta} = (X^t X)^{-1} X^t y$, then the proportion of variance explained (21.8) simply reduces to the classical definition

$$R^2 = 1 - \frac{\mathrm{E}\left(\mathop{\mathrm{V}}_{i=1}^{n} \epsilon_i\right)}{\mathrm{E}\left(\mathop{\mathrm{V}}_{i=1}^{n} y_i\right)} = 1 - \frac{y^t (I - H)\, y}{y^t I_c\, y},$$

where I is the $n \times n$ identity matrix, $H = X(X^t X)^{-1} X^t$, and I_c is the $n \times n$ matrix with $1 - 1/n$ along the diagonal and $1/n$ off the diagonal.

In a simulation-based context, to fully evaluate our expression (21.8) for R^2, one would also average over posterior uncertainty in β and σ. Under the standard noninformative prior density that is uniform on $(\beta, \log \sigma)$, the proportion of variance

explained (21.8) becomes

$$R^2 = 1 - \left(\frac{n-3}{n-p-2}\right)\frac{y^t(I-H)\,y}{y^t I_c\,y},$$

where p is the number of columns of X.

This is remarkably similar to the classical adjusted R^2. In fact, if we plug in the classical estimate, $\hat{\sigma}^2 = y^t(I-H)\,y/(n-p)$, rather than averaging over the marginal posterior distribution for σ^2, then (21.8) becomes

$$R^2 = 1 - \left(\frac{n-1}{n-p}\right)\frac{y^t(I-H)\,y}{y^t I_c\,y},$$

which is exactly classical adjusted R^2. Because $\frac{n-3}{n-p-2} > \frac{n-1}{n-p}$ for $p>1$, our Bayesian adjusted R^2 leads to a lower measure of explained variance than the classical adjusted R^2. This makes sense, since the classical adjusted R^2 could be considered too high because it does not account for uncertainty in σ.

Setting up the computations in R and Bugs

We can add the R^2 computation to a multilevel model in three steps:

1. For each level of the model, add one line in the Bugs code to define the residual error: `e.y[i]<-y[i]-y.hat[i]`, and so forth.
2. Add the errors to the list of parameters saved in the `bugs()` call from R.
3. For each level of the model, compute R^2 as defined in (21.8) in R.

We illustrate with the varying-intercept, varying-slope model for the radon data. We take the Bugs model from Sections 17.1–17.2 and add lines at each level to compute the error terms:

Bugs code

```
model {
  for (i in 1:n){
    y[i] ~ dnorm (y.hat[i], tau.y)
    y.hat[i] <- a[county[i]] + b[county[i]]*x[i]
    e.y[i] <- y[i] - y.hat[i]                          # data-level errors
  }
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif (0, 100)

  for (j in 1:J){
    a[j] <- B[j,1]
    b[j] <- B[j,2]
    B[j,1:2] ~ dmnorm (B.hat[j,], Tau.B[,])
    B.hat[j,1] <- g.a.0 + g.a.1*u[j]
    B.hat[j,2] <- g.b.0 + g.b.1*u[j]
    for (k in 1:2){                                    # group-level errors
      E.B[j,k] <- B[j,k] - B.hat[j,k]
    }
  }
. . .
```

In the call to `bugs()`, the parameters `e.y` and `E.B` must be saved[4] (along with `a`, `b`, and any other parameters of interest). We can then compute explained variance in R as follows:

[4] The label `E.B` follows our general convention of labeling matrices with capital letters.

R code
```
rsquared.y <- 1 - mean (apply (e.y, 1, var)) / var (y)
e.a <- E.B[,,1]
e.b <- E.B[,,2]
rsquared.a <- 1 - mean (apply (e.a, 1, var)) / mean (apply (a, 1, var))
rsquared.b <- 1 - mean (apply (e.b, 1, var)) / mean (apply (b, 1, var))
```

Alternatively, we could define `e.a` and `e.b` by name in the Bugs model.

The inner variances (within the above `apply()` calls) represent variation across the K items at each level (in the radon example, these items are the n data points, the J slopes, and the J intercepts). The outer means represent averages over the simulations representing the uncertainty distributions. The denominator of the expression for R_y^2 has a simpler format because y is an observed vector and does not have any simulation uncertainty.

21.6 Summarizing the amount of partial pooling

When fitting a multilevel model, it can be useful to get a sense of how much pooling is being performed for each group of parameters. The amount of pooling depends on the group-level variance and the information available within each group. Here we define a simple numerical summary of the amount of pooling of a set of group effects α_j, $j = 1, \ldots, J$. These could be varying intercepts in a simple nested model or a batch of varying parameters in a more complicated model with varying slopes and non-nested levels.

Consider the basic multilevel model with data $y_i \sim \mathrm{N}(\alpha_{j[i]}, \sigma_y^2)$, with population distribution $\alpha_j \sim \mathrm{N}(\mu_\alpha, \sigma_\alpha^2)$ and hyperparameters $\mu_\alpha, \sigma_y, \sigma_\alpha$ known. Let n_j be the number of measurements in each group j, and label $\bar{y}_j$ as the average of the y_i's within the group. For each group j, the multilevel estimate of the parameter α_j is

$$\hat{\alpha}_j^{\text{multilevel}} = \omega_j \mu_\alpha + (1 - \omega_j)\bar{y}_j, \tag{21.9}$$

where

$$\omega_j = 1 - \frac{\sigma_\alpha^2}{\sigma_\alpha^2 + \sigma_y^2/n_j} \tag{21.10}$$

is a "pooling factor" that represents the degree to which the estimates are pooled together (that is, based on μ_α) rather than estimated separately (based on the raw data $\bar{y}_j$). The extreme possibilities, $\omega = 0$ and 1, correspond to no pooling ($\hat{\alpha}_j = \bar{y}_j$) and complete pooling ($\hat{\alpha}_j = \mu_\alpha$), respectively. The (posterior) variance of the parameter α_j is

$$\mathrm{var}(\alpha_j) = (1 - \omega_j)\sigma_y^2/n_j. \tag{21.11}$$

The statistical literature sometimes labels $1-\omega$ as the "shrinkage" factor, a notation we find confusing since a shrinkage factor of zero corresponds to complete shrinkage toward the population mean. To avoid ambiguity, we use the "pooling factor" terminology instead. The form of expression (21.10) matches the form of the definition (21.6) of R^2 in Section 21.5.

The concept of pooling is used to help understand multilevel models in two distinct ways: comparing the estimates of different parameters in a group, and summarizing the pooling of the model as a whole. When comparing, it is usual to consider several parameters α_j with a common population (prior) distribution but different data variances; thus, $\bar{y}_j \sim \mathrm{N}(\alpha_j, \sigma_{y\,j}^2)$. Then ω_j can be defined as in (21.10), with $\sigma_{y\,j}$ in place of σ_y. Then each group is associated with a different pooling factor that is larger if the amount of information ($1/\sigma_{y\,j}^2$) in the data for that group is small.

Pooling factors for individual coefficients in a multilevel model

The pooling factor for an individual coefficient captures the weighted averaging between the within-group data and the group-level model:

$$\hat{\alpha}_j^{\text{multilevel}} = \omega_j \hat{\alpha}_j^{\text{complete pooling}} + (1 - \omega_j)\hat{\alpha}_j^{\text{no pooling}}. \tag{21.12}$$

The pooling factor ω_j can range from 0 (no pooling) to 1 (complete pooling). Coefficients with higher values of ω are estimated more from the group-level model and less from the within-group data.

Computing the pooling factor. It can be helpful to use ω_j to summarize the amount of pooling of a multilevel parameter; however, it is generally impractical to compute it using the implicit definition in (21.12). The difficulty is that additional computations would be required to determine the "no pooling" and "complete pooling" estimates.

Instead, we make use of the following formula from the linear model for a parameter $\alpha_j = \hat{\alpha}_j + \epsilon_j$, where ϵ_j has a $\text{N}(0, \sigma_\alpha^2)$ prior distribution:

$$\text{pooling factor } \lambda_j = \frac{(\text{standard error of } \epsilon_j)^2}{\sigma_\alpha^2}. \tag{21.13}$$

Computation in Bugs. To apply formula (21.13), we must perform inferences about the group-level errors ϵ_j, which can be defined easily enough in the Bugs model. For example, the varying intercepts for the counties j in the radon example have the model, $\alpha_j \sim \text{N}(\hat{\alpha}_j, \sigma_\alpha^2)$, where the $\hat{\alpha}_j$'s are the linear predictors at the group level. In the Bugs code, we simply add this sort of line inside the "`for (j in 1:J)`" loop:

Bugs code

```
e.a[j] <- a[j] - a.hat[j]
```

In the R code, we must then add `e.a` to the list of parameters to be saved, and then we can calculate the vector of J pooling factors. For example,

R code

```
omega <- (sd(e.a)/sigma.a)^2
omega <- pmin (omega, 1)
```

The second line above[5] is needed to keep the estimated pooling factors below 1, which occasionally happens due to simulation variability (and can also happen with non-normal models).

Figure 21.8a illustrates with a plot of the pooling factors versus sample sizes for the varying intercepts for the 85 counties in the radon model with the floor predictor and county effects but no county-level predictors. The counties with small sample sizes have pooling factors ω_j near 1 (that is, nearly complete pooling), whereas the larger counties have pooling factors near 0, close to the no-pooling estimates.

Figure 21.8b displays pooling factors for the radon model with floor of measurement as an individual-level predictor and uranium as a county-level predictor. Adding the county-level predictor increases the pooling, which is appropriate since the new model has a lower group-level error. (As discussed in Section 12.6, σ_α declines from 0.34 to 0.13 when county uranium is added to the model.) There is more pooling to this better-fitting model, yielding more precise estimates of the county effects.

[5] The R function `pmin()` performs parallel minimization, in this case computing, for each element of ω, the minimum of itself and 1.

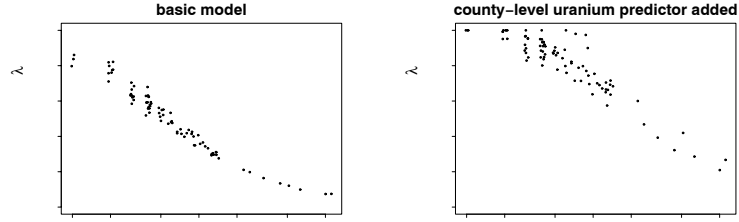

Figure 21.8 *Pooling factors for county intercepts in two versions of the radon model, (a) with no county-level predictors and (b) including the county-level uranium predictor. For each model, ω_j decreases with sample size—that is, the most pooling occurs with the small counties. There is more pooling in the second model, which makes sense—adding the county-level predictor reduces the variance of the group-level errors.*

Summary pooling factor for each batch of parameters

We can define a summary measure, λ, for the average amount of pooling at each level of a multilevel model, summarizing the extent to which the variance of the residuals ϵ_k is reduced by the pooling of the multilevel model:

$$\lambda = 1 - \frac{\mathop{\mathrm{V}}_{k=1}^{K} \mathrm{E}(\epsilon_k)}{\mathrm{E}\left(\mathop{\mathrm{V}}_{k=1}^{K} \epsilon_k\right)}. \tag{21.14}$$

The denominator in this expression is the numerator in expression (21.8)—the average variance in the ϵ_k's, that is, the unexplained component of the variance of the θ_k's. The numerator in the ratio term of (21.14) is the variance among the point estimates (the partial-pooling estimators) of the ϵ_k's. If this variance is high (close to the average variance in the ϵ_k's), then λ will be close to 0 and there is little pooling. If this variance is low, then the estimated ϵ_k's are pooled closely together, and the pooling factor λ will be close to 1.

The striking similarity of expressions (21.6) and (21.14), which define R^2 and λ, respectively, suggests that the two concepts can be understood in a common framework. We consider each to represent the fraction of variance explained, first by the linear predictor μ and then by the multilevel model for ϵ.

Setting up the computation in R and Bugs

We can compute λ at each level using the errors `e.y, e.a, e.b` defined at the end of Section 21.5 for computing R^2. Once the Bugs model has been fit, the pooling factors can be computed in R as follows:

R code
```
lambda.y <- 1 - var (apply (e.y, 2, mean)) / mean (apply (e.y, 1, var))
lambda.a <- 1 - var (apply (e.a, 2, mean)) / mean (apply (e.a, 1, var))
lambda.b <- 1 - var (apply (e.b, 2, mean)) / mean (apply (e.b, 1, var))
```

Discussion

The proportion of variance explained (21.8) and the pooling factor (21.14) can be easily calculated at each stage of a multilevel model. In general, R^2 will be

informative wherever regression predictors (including group indicators) are present, and λ will be relevant at the hierarchical stages of the model. The measures of explained variance and partial pooling conveniently summarize the fit at each level of the model and the degree to which estimates are pooled toward their population models. Together, they clarify the role of predictors at different levels of a multilevel model. They can be derived from a common framework of comparing variances at each level of the model, which also means that they do not require the fitting of additional null models.

Expressions (21.8) and (21.14) are closely related to the usual definitions of adjusted R^2 in simple linear regression and pooling in balanced one-way hierarchical models. From this perspective, they unify the data-level concept of R^2 and the group-level concept of pooling or shrinkage, and also generalize these concepts to account for uncertainty in the variance components. Further, as illustrated for the radon application, they can help us understand more complex multilevel models.

Other challenges include defining explained variance and partial pooling factors for generalized linear models, either on the scale of the data or of the latent parameters.

21.7 Adding a predictor can *increase* the residual variance!

Multilevel models can behave in ways that are unexpected from the perspective of classical statistics. We illustrate with the radon model from Chapter 12. We first fit a stripped-down multilevel model for the home radon levels, including the county-level uranium predictor but no individual-level predictors (not even the floor of measurement); thus,

$$\text{model 1:} \quad \begin{array}{rcl} y_i & \sim & \text{N}(\alpha_{j[i]}, \sigma_y^2) \\ \alpha_j & \sim & \text{N}(\gamma_0 + \gamma_1 u_j, \sigma_\alpha^2). \end{array}$$

Fitting this model to the Minnesota radon data yields estimated variance components $\sigma_y = 0.80, \sigma_\alpha = 0.12$.

We then add the house-level floor indicator x_i:

$$\text{model 2:} \quad \begin{array}{rcl} y_i & \sim & \text{N}(\alpha_{j[i]} + \beta x_i,\, \sigma_y^2) \\ \alpha_j & \sim & \text{N}(\gamma_0 + \gamma_1 u_j, \sigma_\alpha^2), \end{array}$$

yielding new estimates of $\sigma_y = 0.76, \sigma_\alpha = 0.16$. The house-level standard deviation has decreased—which makes sense since we have added a predictor at that level—but the variation at the county level has *increased*, which is a surprise. In classical regression models, the residual variance can only go down, not up, when a predictor is added.[6]

What is going on? After some thought, we realized that in model 2, the counties with more basements happened to have higher county coefficients α_j. In model 1, some of the variation in county radon levels was canceled by an opposite variation in the proportion of basements. The increased between-county variance in model 2 indicates true variation among counties that happened to be masked by the first model.

Figure 21.9 shows a hypothetical extreme version of this situation: the three counties have identical average radon levels (thus, $\sigma_\alpha = 0$ for model 1, which has no basement predictor), but only because the naturally low-radon county has many basements and the naturally high-radon county has few basements. (Such a pattern

[6] We ignore the minor increase in the variance estimate that corresponds to reducing the degrees of freedom by 1 and thus dividing by $n - k - 1$ instead of $n - k$ in the variance calculation.

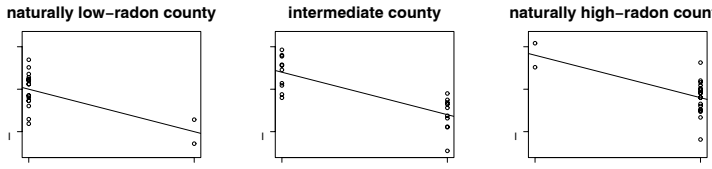

Figure 21.9 *Hypothetical data from three counties illustrating how adding an individual-level predictor can decrease the group-level variance. In each county, radon levels are higher in homes with basements. The county with low natural radon levels has more homes with basements, and the county with high natural radon levels has fewer homes with basements. As a result, the average radon level in the three counties is identical, but when floor of measurement is (appropriately) included as a predictor, the counties appear more different.*

can happen, for example, if low-radon areas have sandy soil in which basements are easy to dig, with high-radon areas having rocky soil where basements are less commonly built.) Model 2, which controls for basements, reveals the true underlying variation among the counties, and thus σ_α increases.

This pattern, caused by correlation between individual-level variables and group-level errors, does not occur in classical regression. When it occurs in multilevel regression, the model fit can be improved by including the average of x as a group-level predictor (in this example, the proportion of houses in the county that have basements). When the county-level basement proportion is added as a group-level predictor, its coefficient is estimated at -0.41 (with a standard error of 0.2), and the estimated residual standard deviations at the data and county levels are 0.76 and 0.14.

For the radon problem, the county-level basement proportion is difficult to interpret directly but rather serves as a proxy for underlying variables (for example, the type of soil that is prevalent in the county).

In other settings, especially in social science, individual averages that are used as group-level predictors are often interpreted as "contextual effects." For example, in the police stops example in Section 15.1, one might suspect that police behavior in a precinct is influenced by the ethnic composition of the local residents. However, we must be suspicious of this sort of conclusion without further information. As the radon example illustrates, it is possible to have between-level correlations without the need for a "contextual" story. As usual in these models, we try to be careful to state the regression results in a predictive rather than causal manner ("the counties with more basements tend to have lower radon levels, after controlling for the basement statuses of the individual houses").

21.8 Multiple comparisons and statistical significance

A meta-analysis of a set of randomized experiments

In the wake of concerns about the health effects of low-frequency electric and magnetic fields, an experiment was performed to measure the effect of electromagnetic fields at various frequencies on the functioning of chick brains. At each of several frequencies of electromagnetic fields (1 Hz, 15 Hz, 30 Hz, ..., 510 Hz), a randomized experiment was performed to estimate the effect of exposure, compared to a control condition of no electromagnetic field. The researchers reported, for each fre-

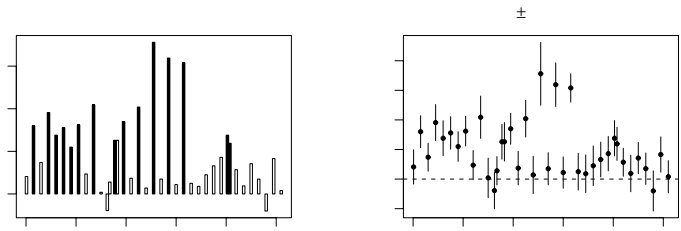

Figure 21.10 *(a) Estimated effects of electromagnetic fields on calcium efflux from chick brains, shaded to indicate different levels of statistical significance, adapted from Blackman et al. (1988). A separate experiment was performed at each frequency. (b) Same results presented as estimates ± standard errors. As discussed in the text, the first plot, with its emphasis on statistical significance, is misleading.*

quency, the estimated treatment effect (the average difference between treatment and control measurements) and the standard error (that is, $\sqrt{\sigma_T^2/n_T + \sigma_C^2/n_C}$; see Section 2.3).

In the article reporting this study, the estimates at the different frequencies were summarized by their statistical significance, as we illustrate in Figure 21.10a by using different shading for results that are more than 2.3 standard errors from zero (that is, statistically significant at the 99% level), between 2.0 and 2.3 standard errors from zero (statistically significant at the 95% level), and so forth. The researchers used this sort of display to hypothesize that one process was occurring at 255, 285, and 315 Hz (where effects were highly significant), another at 135 and 225 Hz (where effects were only moderately significant), and so forth. The estimates are all of relative calcium efflux, so that an effect of 0.1, for example, corresponds to a 10% increase compared to the control condition.

In the chick brain experiment, the researchers made the common mistake of using statistical significance as a criterion for separating the estimates of different effects. As we discuss in Section 2.5, this approach does not make sense. At the very least, it is more informative to show the estimated treatment effect and standard error at each frequency, as in Figure 21.10b.

Multilevel model

The confidence intervals at different frequencies in Figure 21.10b overlap substantially, which implies that the estimates could be usefully pooled using a multilevel model. The raw data from these experiments were not available,[7] so we analyzed the estimates, which we shall label as y_j for each subexperiment j. Because the chicken brains were randomly assigned to the treatment and control groups, we can assume the y_j's are unbiased estimates of the treatment effects θ_j; and because the sample sizes were not tiny, it is reasonable to assume that the estimation errors are approximately normally distributed; thus,

$$y_j \sim \text{N}(\theta_j, \sigma_j^2).$$

[7] When we asked the experimenter to share the data with us, he refused. This was disturbing considering this was a government-funded study of public health interest, but we did not feel like putting in the effort to wrest the data from him.

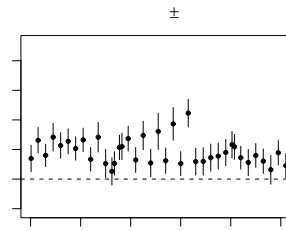

Figure 21.11 *Multilevel estimates and standard errors for the effects of magnetic fields, partially pooled from the separate estimates displayed in Figure 21.10. The standard errors of the original estimates were large, and so the multilevel estimates are pooled strongly toward the common mean estimate of 0.1.*

Our default model for the treatment effects is simply

$$\theta_j \sim \mathrm{N}(\mu_\theta, \sigma_\theta^2).$$

If we assume each σ_j is known and equal to the standard error of the estimate y_j, we can easily perform inference about the θ_j's (as well as the hyperparameters $\mu_\theta, \sigma_\theta$) using Bugs, supplying y, σ, and J as data:

Bugs code

```
model {
  for (j in 1:J){
    y[j] ~ dnorm (theta[j], tau.y[j])
    tau.y[j] <- pow(sigma.y[j], -2)
  }
  for (j in 1:J){
    theta[j] ~ dnorm (mu.theta, tau.theta)
    e.theta[j] <- theta[j] - mu.theta
  }
  tau.theta <- pow(sigma.theta, -2)
  mu.theta ~ dnorm (0, .0001)
  sigma.theta ~ dunif (0, 100)
}
```

(The line defining `e.theta` is there to allow the computation of the partial pooling factor λ, which turns out to be 0.49, implying that the estimates are pooled, on average, halfway toward the group-level model, which in this case is simply the average of all the treatment effects.)

Figure 21.11 displays the inferences for the treatment effects θ_j, as estimated from the multilevel model. The inferences shown here represent partial pooling of the separate estimates y_j toward the grand mean μ_θ.

More generally, the multilevel model can be seen as a way to estimate the effects at each frequency j, without setting "nonsignificant" results to zero. Some of the apparently dramatic features of the original data as plotted in Figure 21.10a—for example, the negative estimate at 480 Hz and the pair of statistically significant estimates at 405 Hz—do not stand out so much in the multilevel estimates, indicating that these features could be easily explained by sampling variability and do not necessarily represent real features of the underlying parameters.

Potential criticisms of the multilevel model

The above multilevel model can be criticized because it pools all the estimates toward their common mean of 0.1 with an assumed normal distribution for the true θ_j's. This would not be appropriate if, for example, the treatment had a positive effect at some frequencies and a zero effect at others. Or, as hypothesized by the authors of the original study, that the true treatment effects could fall into three groups: a set of large effects near 0.3, a set of moderate effects near 0.15, and a set of zero effects.

We could explore this possibility by fitting a mixture model for the θ_j's. But the resulting inferences would not differ much from our multilevel analysis that used a normal distribution. The reason that changing the model would not do much is that the uncertainty bounds for the individual estimates are so high. Even if, for example, we fit a model with three clusters of effects, it would not be so clear which points correspond to which cluster. The estimates at 255, 285, and 315 Hz appear to be one cluster, but in fact the point at 255 Hz has a high standard error (see Figure 21.10b) and could very well belong to a lower cluster, whereas the estimates at 45, 135, 225, and other points are consistent with being in the higher group. Similarly, several of the estimates are not statistically significant (see Figure 21.10a) but they are almost all positive, and in aggregate they do not appear to be zero. There is certainly no sharp dividing line between the "low" and the "moderate" estimates.

To put it another way: we do not "believe" the estimates in Figure 21.11 in the same way as we trust the estimated county radon levels in Chapters 12 and 13. The difference is that the lognormal distribution for county radon levels seems reasonable enough (in that radon in a house is affected by many small multiplicative factors), whereas it seems more plausible that the effect of electromagnetic fields on calcium efflux could be concentrated at a few frequencies. However, given the variability in the parameter estimates, we believe the unpooled estimates in Figure 21.10 even less, and we would recommend using the multilevel model and estimates as a starting point for further analysis of these data.

Relevance to multiple comparisons

One of the risks in statistical analysis of complex data is overinterpretation of patterns that could be explained by random variation. Multilevel modeling is sometimes viewed with skepticism as just one more kind of model that can be fit without the evidence of the data. Actually, though, multilevel modeling can reduce overinterpretation. For example, Figure 21.10 shows a dramatic pattern of three points that stand apart from all the others, but the multilevel estimates in Figure 21.11 show these to be part of a larger group of relatively large effects for frequencies up to 300 Hz or so. The partial pooling has revealed the fragility of the patterns in the raw data (or in the no-pooling estimates), which can be explained by sampling variability.

21.9 Bibliographic note

Achen (1982), Aitkin and Longford (1986), and Kiss (2003) present some methods for understanding multilevel models. Textbooks such as Ramsey and Schafer (2001) and Fox (2002) are useful places to start.

The distinction between finite-population and superpopulation variances is fundamental in multilevel models and has been considered by many researchers, including Searle, Casella, and McCulloch (1992) and Gelman (2005). The different

definitions of "fixed" and "random" effects, with references, appear in Gelman (2005). See also Kreft and De Leeuw (1998, section 1.3.3) for a discussion of the multiplicity of definitions of fixed and random effects and coefficients, and Robinson (1998) for a historical overview. See Rosenthal, Rosnow, and Rubin (2000) for more on contrasts in linear models.

The material on average predictive comparisons is from Gelman and Pardoe (2007); the prison example appears in Pardoe and Weidner (2006) along with a lively discussion. See also Graubard and Korn (1999), King, Tomz, and Wittenberg (2000), Pardoe (2001), and Pardoe and Cook (2002) for related work on numerical and graphical summaries of marginal predictive comparisons. The measures of R^2 and partial pooling for multilevel models come from Gelman and Pardoe (2006); see Wherry (1931), Pratt (1987), Bentler and Raykov (2000), Goldstein, Browne, and Rasbash (2002), Afshartous and De Leeuw (2002), Xu (2003), Gustafson and Clarke (2004), and Browne et al. (2005) for related ideas.

Kane, Rockoff, and Staiger (2006) describe the study of value added in teaching; some important earlier work in this area includes Hanushek (1971), Summers and Wolfe (1977), and Ehrenberg and Brewer (1994). The Advanced Placement study is described by Wainer, Gelman, and Wang (2000). The example of the increasing residual variance comes from Gelman and Price (1998). The chick brain studies, along with the inappropriate analysis based on statistical significance, come from Blackman et al. (1988). For more on multiple comparisons in hierarchical models, see Gelman and Tuerlinckx (2000).

21.10 Exercises

1. Uncertainty and variability: for the radon model in Section 21.1, give examples of how the parameter estimates and standard deviations might look if the sample size is increased in the following ways:

   (a) 4 times as many houses measured within each existing county.

   (b) 4 times as many counties, but the same number of houses measured in each county.

   (c) 4 times as many counties, with 4 times the number of houses measured in each county (thus, 16 times as many houses in total).

2. Superpopulation and finite-population standard deviations: fit a non-nested multilevel model to the Winter Olympics data (see Exercises 11.3 and 13.3).

   (a) Fit the model using `lmer()` to get quick estimates of the standard-deviation parameters (at the levels of data, judge, and skater).

   (b) Fit the model in Bugs and get point estimates and 50% intervals for the superpopulation standard-deviation parameters and the corresponding finite-sample standard deviations.

3. Superpopulation and finite-population standard deviations:

   (a) Fit a varying-intercept, varying-slope model to the data in the folder `radon`. Get point estimates and 50% intervals for the superpopulation and finite-population standard deviations.

   (b) Repeat step (a) but just using a sample of 10 counties. The inference for the superpopulation standard deviation should now be much more uncertain than the finite-population standard deviation.

(c) Repeat but just using a sample of 5 counties. The inferences should be even more different now.

4. Contrasts: fit in Bugs a varying-intercept model to the radon data with log uranium as a group-level predictor. You will compare inferences for the superpopulation contrast (that is, the slope for log uranium in the county-level model) and the corresponding finite-population contrast (that is, the coefficient of log uranium for the intercepts for the particular counties in the data). You will need to postprocess the simulations in R in order to get simulations for the finite-population contrast.

   (a) Compare the inferences (estimates and standard errors) for the superpopulation and finite-population contrasts.

   (b) Repeat part (a), but fitting the model just to the first three counties in the dataset.

5. Average predictive comparisons:

   (a) Take the model of well switching from Exercise 14.2 and estimate average predictive comparisons for the input variables in the model (including the index variable for villages).

   (b) Take the model of rodent infestation from Exercise 14.3 and estimate average predictive comparisons for the input variables in the model (including the index variables for buildings and for community districts).

6. Explained variance and partial pooling: go to the model you fit to the one-fifth sample of the radon data in Exercise 12.9. Fit the model in Bugs and compute the percentage of explained variance and partial pooling factor at each of the two levels of the model.