CHAPTER 11

Multilevel structures

As we illustrate in detail in subsequent chapters, multilevel models are extensions of regression in which data are structured in groups and coefficients can vary by group. In this chapter, we illustrate basic multilevel models and present several examples of data that are collected and summarized at different levels. We start with simple grouped data—persons within cities—where some information is available on persons and some information is at the city level. We then consider examples of repeated measurements, time-series cross sections, and non-nested structures. The chapter concludes with an outline of the costs and benefits of multilevel modeling compared to classical regression.

11.1 Varying-intercept and varying-slope models

With grouped data, a regression that includes indicators for groups is called a *varying-intercept model* because it can be interpreted as a model with a different intercept within each group. Figure 11.1a illustrates with a model with one continuous predictor x and indicators for $J = 5$ groups. The model can be written as a regression with 6 predictors or, equivalently, as a regression with two predictors (x and the constant term), with the intercept varying by group:

$$\text{varying-intercept model:}\;\; y_i = \alpha_{j[i]} + \beta x_i + \epsilon_i.$$

Another option, shown in Figure 11.1b, is to let the slope vary with constant intercept:

$$\text{varying-slope model:}\;\; y_i = \alpha + \beta_{j[i]} x_i + \epsilon_i.$$

Finally, Figure 11.1c shows a model in which both the intercept and the slope vary by group:

$$\text{varying-intercept, varying-slope model:}\;\; y_i = \alpha_{j[i]} + \beta_{j[i]} x_i + \epsilon_i.$$

The varying slopes are interactions between the continuous predictor x and the group indicators.

As we discuss shortly, it can be challenging to estimate all these α_j's and β_j's, especially when inputs are available at the group level. The first step of multilevel modeling is to set up a regression with varying coefficients; the second step is to set up a regression model for the coefficients themselves.

11.2 Clustered data: child support enforcement in cities

With multilevel modeling we need to go beyond the classical setup of a data vector y and a matrix of predictors X (as shown in Figure 3.6 on page 38). Each level of the model can have its own matrix of predictors.

We illustrate multilevel data structures with an observational study of the effect of city-level policies on enforcing child support payments from unmarried fathers. The treatment is at the group (city) level, but the outcome is measured on individual families.

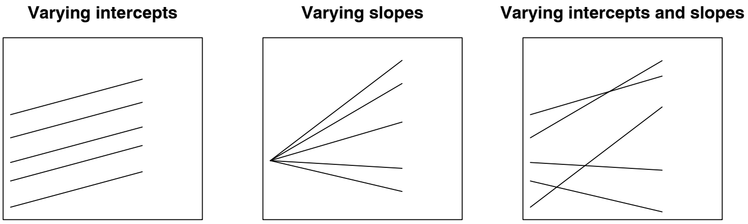

Figure 11.1 *Linear regression models with (a) varying intercepts ($y = \alpha_j + \beta x$), (b) varying slopes ($y = \alpha + \beta_j x$), and (c) both ($y = \alpha_j + \beta_j x$). The varying intercepts correspond to group indicators as regression predictors, and the varying slopes represent interactions between x and the group indicators.*

ID	dad age	mom race	informal support	city ID	city name	enforce intensity	benefit level	city indicators 1	2	⋯	20
1	19	hisp	1	1	Oakland	0.52	1.01	1	0	⋯	0
2	27	black	0	1	Oakland	0.52	1.01	1	0	⋯	0
3	26	black	1	1	Oakland	0.52	1.01	1	0	⋯	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮		⋮
248	19	white	1	3	Baltimore	0.05	1.10	0	0	⋯	0
249	26	black	1	3	Baltimore	0.05	1.10	0	0	⋯	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮		⋮
1366	21	black	1	20	Norfolk	−0.11	1.08	0	0	⋯	1
1367	28	hisp	0	20	Norfolk	−0.11	1.08	0	0	⋯	1

Figure 11.2 *Some of the data from the child support study, structured as a single matrix with one row for each person. These indicators would be used in classical regression to allow for variation among cities. In a multilevel model they are not necessary, as we code cities using their index variable ("city ID") instead. We prefer separating the data into individual-level and city-level datasets, as in Figure 11.3.*

Studying the effectiveness of child support enforcement

Cities and states in the United States have tried a variety of strategies to encourage or force fathers to give support payments for children with parents who live apart. In order to study the effectiveness of these policies for a particular subset of high-risk children, an analysis was done using a sample of 1367 noncohabiting parents from the Fragile Families study, a survey of unmarried mothers of newborns in 20 cities. The survey was conducted by sampling from hospitals which themselves were sampled from the chosen cities, but here we ignore the complexities of the data collection and consider the mothers to have been sampled at random (from their demographic category) in each city.

To estimate the effect of child support enforcement policies, the key "treatment" predictor is a measure of enforcement policies, which is available at the city level. The researchers estimated the probability that the mother received informal support, given the city-level enforcement measure and other city- and individual-level predictors.

ID	dad age	mom race	informal support	city ID
1	19	hisp	1	1
2	27	black	0	1
3	26	black	1	1
⋮	⋮	⋮	⋮	⋮
248	19	white	1	3
249	26	black	1	3
⋮	⋮	⋮	⋮	⋮
1366	21	black	1	20
1367	28	hisp	0	20

city ID	city name	enforce- ment	benefit level
1	Oakland	0.52	1.01
2	Austin	0.00	0.75
3	Baltimore	−0.05	1.10
⋮	⋮	⋮	⋮
20	Norfolk	−0.11	1.08

Figure 11.3 *Data from the child support study, structured as two matrices, one for persons and one for cities. The inputs at the different levels are now clear. Compare to Figure 11.2.*

A data matrix for each level of the model

Figure 11.2 shows the data for the analysis as it might be stored in a computer package, with information on each of the 1367 mothers surveyed. To make use of the multilevel structure of the data, however, we need to construct *two* data matrices, one for each level of the model, as Figure 11.3 illustrates. At the left is the person-level data matrix, with one row for each survey respondent, and their cities are indicated by an index variable; at the right is the city data matrix, giving the name and other information available for each city.

At a practical level, the two-matrix format of Figure 11.3 has the advantage that it contains each piece of information exactly once. In contrast, the single large matrix in Figure 11.2 has each city's data repeated several times. Computer memory is cheap so this would not seem to be a problem; however, if city-level information needs to be added or changed, the single-matrix format invites errors.

Conceptually, the two-matrix, or multilevel, data structure has the advantage of clearly showing which information is available on individuals and which on cities. It also gives more flexibility in fitting models, allowing us to move beyond the classical regression framework.

Individual- and group-level models

We briefly outline several possible ways of analyzing these data, as a motivation and lead-in to multilevel modeling.

Individual-level regression. In the most basic analysis, informal support (as reported by mothers in the survey) is the binary outcome, and there are several individual- and city-level predictors. Enforcement is considered as the treatment, and a logistic regression is used, also controlling for other inputs. This is the starting point of the observational study.

Using classical regression notation, the model is $\Pr(y_i = 1) = \text{logit}^{-1}(X_i\beta)$, where X includes the constant term, the treatment (enforcement intensity), and the other predictors (father's age and indicators for mother's race at the individual level; and benefit level at the city level). X is thus constructed from the data matrix of Figure 11.2. This individual-level regression has the problem that it ignores city-level variation beyond that explained by enforcement intensity and benefit level, which are the city-level predictors in the model.

city ID	city name	enforce- ment	benefit level	# in sample	avg. age	prop. black	proportion with informal support
1	Oakland	0.52	1.01	78	25.9	0.67	0.55
2	Austin	0.00	0.75	91	25.8	0.42	0.54
3	Baltimore	−0.05	1.10	101	27.0	0.86	0.67
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
20	Norfolk	−0.11	1.08	31	27.4	0.84	0.65

Figure 11.4 *City-level data from child support study (as in the right panel of Figure 11.3), also including sample sizes and sample averages from the individual responses.*

Group-level regression on city averages. Another approach is to perform a city-level analysis, with individual-level predictors included using their group-level averages. Figure 11.4 illustrates: here, the outcome, y_j, would be the average total support among the respondents in city j, the enforcement indicator would be the treatment, and the other variables would also be included as predictors. Such a regression—in this case, with 20 data points—has the advantage that its errors are automatically at the city level. However, by aggregating, it removes the ability of individual predictors to predict individual outcomes. For example, it is possible that older fathers give more informal support—but this would not necessarily translate into average father's age being predictive of more informal support at the city level.

Individual-level regression with city indicators, followed by group-level regression of the estimated city effects. A slightly more elaborate analysis proceeds in two steps, first fitting a logistic regression to the individual data y given individual predictors (in this example, father's age and indicators for mother's race) along with indicators for the 20 cities. This first-stage regression then has 22 predictors. (The constant term is *not* included since we wish to include indicators for all the cities; see the discussion at the end of Section 4.5.)

The next step in this two-step analysis is to perform a *linear* regression at the city level, considering the estimated coefficients of the city indicators (in the individual model that was just fit) as the "data" y_j. This city-level regression has 20 data points and uses, as predictors, the city-level data (in this case, enforcement intensity and benefit level). Each of the predictors in the model is thus included in one of the two regressions.

The two-step analysis is reasonable in this example but can run into problems when sample sizes are small in particular groups, or when there are interactions between individual- and group-level predictors. Multilevel modeling is a more general approach that can include predictors at both levels at once.

Multilevel models

The multilevel model looks something like the two-step model we have described, except that both steps are fitted at once. In this example, a simple multilevel model would have two components: a logistic regression with 1369 data points predicting the binary outcome given individual-level predictors and with an intercept that can vary by city, and a linear regression with 20 data points predicting the city intercepts from city-level predictors. In the multilevel framework, the key link between the individual and city levels is the city indicator—the "city ID" variable in Figure 11.3, which takes on values between 1 and 20.

For this example, we would have a logistic regression at the data level:

$$\Pr(y_i = 1) = \text{logit}^{-1}(X_i\beta + \alpha_{j[i]}), \text{ for } i = 1, \ldots, n, \tag{11.1}$$

where X is the matrix of individual-level predictors and $j[i]$ indexes the city where person i resides. The second part of the model—what makes it “multilevel”—is the regression of the city coefficients:

$$\alpha_j \sim \text{N}(U_j\gamma, \sigma_\alpha^2), \text{ for } j = 1, \ldots, 20, \tag{11.2}$$

where U is the matrix of city-level predictors, γ is the vector of coefficients for the city-level regression, and σ_α is the standard deviation of the unexplained group-level errors.

The model for the α’s in (11.2) allows us to include all 20 of them in model (11.1) without having to worry about collinearity. The key is the group-level variation parameter σ_α, which is estimated from the data (along with α, β, and a) in the fitting of the model. We return to this point in the next chapter.

Directions for the observational study

The “treatment” variable in this example is not randomly applied; hence it is quite possible that cities that differ in enforcement intensities could differ in other important ways in the political, economic, or cultural dimensions. Suppose the goal were to estimate the effects of potential interventions (such as increased enforcement), rather than simply performing a comparative analysis. Then it would make sense to set this up as an observational study, gather relevant pre-treatment information to capture variation among the cities, and perhaps use a matching approach to estimate effects. In addition, good pre-treatment measures on individuals should improve predictive power, thus allowing treatment effects to be estimated more accurately. The researchers studying these child support data are also looking at other outcomes, including measures of the amity between the parents as well as financial and other support.

Along with the special concerns of causal inference, the usual recommendations of regression analysis apply. For example, it might make sense to consider interactions in the model (to see if enforcement is more effective for older fathers, for example).

11.3 Repeated measurements, time-series cross sections, and other non-nested structures

Repeated measurements

Another kind of multilevel data structure involves repeated measurements on persons (or other units)—thus, measurements are clustered within persons, and predictors can be available at the measurement or person level. We illustrate with a model fitted to a longitudinal dataset of about 2000 Australian adolescents whose smoking patterns were recorded every six months (via questionnaire) for a period of three years. Interest lay in the extent to which smoking behavior can be predicted based on parental smoking and other background variables, and the extent to which boys and girls pick up the habit of smoking during their teenage years. Figure 11.5 illustrates the overall rate of smoking among survey participants.

A multilevel logistic regression was fit, in which the probability of smoking depends on sex, parental smoking, the wave of the study, and an individual parameter

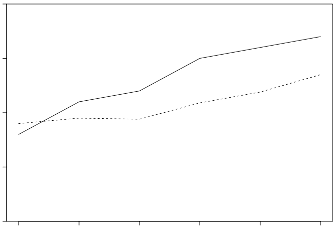

Figure 11.5 *Prevalence of regular (daily) smoking among participants responding at each wave in the study of Australian adolescents (who were on average 15 years old at wave 1).*

person		parents smoke?		wave 1		wave 2		···
ID	sex	mom	dad	age	smokes?	age	smokes?	
1	f	Y	Y	15:0	N	15:6	N	···
2	f	N	N	14:7	N	15:1	N	···
3	m	Y	N	15:1	N	15:7	Y	···
4	f	N	N	15:3	N	15:9	N	···
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

Figure 11.6 *Data from the smoking study as they might be stored in a single computer file and read into R as a matrix,* `data`*. (Ages are in years:months.) These data have a multilevel structure, with observations nested within persons.*

for the person. For person j at wave t, the modeled probability of smoking is

$$\begin{aligned}\Pr(y_{jt}=1) \;&=\; \text{logit}^{-1}(\beta_0+\beta_1\text{psmoke}_j+\beta_2\text{female}_j+ \\ &\qquad +\beta_3(1-\text{female}_j)\cdot t+\beta_4\text{female}_j\cdot t+\alpha_j), \qquad (11.3)\end{aligned}$$

where `psmoke` is the number of the person's parents who smoke and `female` is an indicator for females, so that β_3 and β_4 represent the time trends for boys and girls, respectively.[1]

Figures 11.6 and 11.7 show two ways of storing the smoking data, either of which would be acceptable for a multilevel analysis. Figure 11.6 shows a single data matrix, with one row for each person in the study. We could then pull out the smoking outcome $y = (y_{jt})$ in R, as follows:

R code

```
y <- data[,seq(6,16,2)]
female <- ifelse (data[,2]=="f", 1, 0)
mom.smoke <- ifelse (data[,3]=="Y", 1, 0)
dad.smoke <- ifelse (data[,4]=="Y", 1, 0)
psmoke <- mom.smoke + dad.smoke
```

and from there fit the model (11.3).

Figure 11.7 shows an alternative approach using two data matrices, one with a

[1] Alternatively, we could include a main effect for time and an interaction between time and sex, $\Pr(y_{jt}=1)=\text{logit}^{-1}(\beta_0+\beta_1\cdot\text{psmoke}_j+\beta_2\cdot\text{female}_j+\beta_3\cdot t+\beta_4\cdot\text{female}_j\cdot t+\alpha_j)$, so that the time trends for boys and girls are β_3 and $\beta_3+\beta_4$, respectively. This parameterization is appropriate to the extent that the comparison between the sexes is of interest; in this case we used (11.3) so that we could easily interpret β_3 and β_4 symmetrically.

age	smokes?	person ID	wave
15:0	N	1	1
14.7	N	2	1
15:1	N	3	1
15:3	N	4	1
⋮	⋮	⋮	⋮
15:6	N	1	2
15:1	N	2	2
15:7	Y	3	2
15:9	N	4	2
⋮	⋮	⋮	⋮

person		parents smoke?	
ID	sex	mom	dad
1	f	Y	Y
2	f	N	N
3	m	Y	N
4	f	N	N
⋮	⋮	⋮	⋮

Figure 11.7 *Data from the smoking study, with observational data written as a single long matrix,* `obs.data`*, with person indicators, followed by a shorter matrix,* `person.data`*, of person-level information. Compare to Figure 11.6.*

row for each observation and one with a row for each person. To model these data, one could use R code such as

R code

```
y <- obs.data[,2]
person <- obs.data[,3]
wave <- obs.data[,4]
female <- ifelse (person.data[,2]=="f", 1, 0)
mom.smoke <- ifelse (person.data[,3]=="Y", 1, 0)
dad.smoke <- ifelse (person.data[,4]=="Y", 1, 0)
psmoke <- mom.smoke + dad.smoke
```

and then parameterize the model using the index i to represent individual observations, with $j[i]$ and $t[i]$ indicating the person and wave associated with observation i:

$$\begin{aligned}\Pr(y_i=1) \;=\; & \text{logit}^{-1}(\beta_0 + \beta_1\text{psmoke}_{j[i]} + \beta_2\text{female}_{j[i]} + \\ & + \beta_3(1-\text{female}_{j[i]})\cdot t[i] + \beta_4\text{female}_{j[i]}\cdot t[i] + \alpha_{j[i]}). \qquad (11.4)\end{aligned}$$

Models (11.3) and (11.4) are equivalent, and both can be fit in Bugs (as we describe in Part 2B). Choosing between them is a matter of convenience. For data in a simple two-way structure (each adolescent is measured at six regular times), it can make sense to work with the double-indexed outcome variable, (y_{jt}). For a less rectangular data structure (for example, different adolescents measured at irregular intervals) it can be easier to string together a long data vector (y_i), with person and time recorded for each measurement, and with a separate matrix of person-level information (as in Figure 11.7).

Time-series cross-sectional data

In settings where overall time trends are important, repeated measurement data are sometimes called *time-series cross-sectional.* For example, Section 6.3 introduced a study of the proportion of death penalty verdicts that were overturned, in each of 34 states in the 23 years, 1973–1995. The data come at the state × year levels but we are also interested in studying variation among states and over time.

Time-series cross-sectional data are typically (although not necessarily) "rectangular" in structure, with observations at regular time intervals. In contrast, general repeated measurements could easily have irregular patterns (for example, in the smoking study, some children could be measured only once, others could be measured monthly and others yearly). In addition, time-series cross-sectional data commonly have overall time patterns, for example, the steady expansion of the death penalty from the 1970s through the early 1990s. In this context one must consider the state-year data as clustered within states and also within years, with the potential for predictors at all three levels. We discuss such non-nested models in Section 13.5.

Other non-nested structures

Non-nested data also arise when individuals are characterized by overlapping categories of attributes. For example, consider a study of earnings given occupation and state of residence. A survey could include, say, 1500 persons in 40 job categories in 50 states, and a regression model could predict log earnings given individual demographic predictors X, 40 indicators for job categories, and 50 state indicators. We can write the model generalizing the notation of (11.1)–(11.2):

$$y_i = X_i\beta + \alpha_{j[i]} + \gamma_{k[i]} + \epsilon_i, \text{ for } i = 1, \ldots, n, \tag{11.5}$$

where $j[i]$ and $k[i]$ represent the job category and state, respectively, for person i. The model becomes multilevel with regressions for the job and state coefficients. For example,

$$\alpha_j \sim \mathrm{N}(U_j a, \sigma_\alpha^2), \text{ for } j = 1, \ldots, 40, \tag{11.6}$$

where U is a matrix of occupation-level predictors (for example, a measure of social status and an indicator for whether it is supervisory), a is a vector of coefficients for the job model, and σ_α is the standard deviation of the model errors at the level of job category. Similarly, for the state coefficients:

$$\gamma_k \sim \mathrm{N}(V_k g, \sigma_\gamma^2) \text{ for } k = 1, \ldots, 50. \tag{11.7}$$

The model defined by regressions (11.5)–(11.7) is non-nested because neither the job categories $j[i]$ nor the states $k[i]$ are subsets of the other.

As this example illustrates, regression notation can become awkward with multilevel models because of the need for new symbols (U, V, a, g, and so forth) to denote data matrices, coefficients, and errors at each level.

11.4 Indicator variables and fixed or random effects

Classical regression: including a baseline and $J-1$ indicator variables

As discussed at the end of Section 4.5, when including an input variable with J categories into a classical regression, standard practice is to choose one of the categories as a baseline and include indicators for the other $J-1$ categories. For example, if controlling for the $J = 20$ cities in the child support study in Figure 11.2 on page 238, one could set city 1 (Oakland) as the baseline and include indicators for the other 19. The coefficient for each city then represents its comparison to Oakland.

Multilevel regression: including all J indicators

In a multilevel model it is unnecessary to do this arbitrary step of picking one of the levels as a baseline. For example, in the child support study, one would include indicators for all 20 cities as in model (11.1). In a classical regression these could not all be included because they would be collinear with the constant term, but in a multilevel model this is not a problem because they are themselves modeled by a group-level distribution (which itself can be a regression, as in (11.2)). We discuss on page 393 how the added information removes the collinearity that is present in the simple least squares estimate.

Fixed and random effects

The varying coefficients (α_j's or β_j's) in a multilevel model are sometimes called *random effects*, a term that refers to the randomness in the probability model for the group-level coefficients (as, for example, in (11.2) on page 241).

The term *fixed effects* is used in contrast to random effects—but not in a consistent way! Fixed effects are usually defined as varying coefficients that are not themselves modeled. For example, a classical regression including $J-1=19$ city indicators as regression predictors is sometimes called a "fixed-effects model" or a model with "fixed effects for cities." Confusingly, however, "fixed-effects models" sometimes refer to regressions in which coefficients do *not* vary by group (so that they are fixed, not random).[2]

A question that commonly arises is when to use fixed effects (in the sense of varying coefficients that are unmodeled) and when to use random effects. The statistical literature is full of confusing and contradictory advice. Some say that fixed effects are appropriate if group-level coefficients are of interest, and random effects are appropriate if interest lies in the underlying population. Others recommend fixed

[2] Here we outline five definitions that we have seen of fixed and random effects:

1. Fixed effects are constant across individuals, and random effects vary. For example, in a growth study, a model with random intercepts α_i and fixed slope β corresponds to parallel lines for different individuals i, or the model $y_{it} = \alpha_i + \beta t$. Kreft and De Leeuw (1998, p. 12) thus distinguish between fixed and random coefficients.
2. Effects are fixed if they are interesting in themselves or random if there is interest in the underlying population. Searle, Casella, and McCulloch (1992, section 1.4) explore this distinction in depth.
3. "When a sample exhausts the population, the corresponding variable is *fixed*; when the sample is a small (i.e., negligible) part of the population the corresponding variable is *random*" (Green and Tukey, 1960).
4. "If an effect is assumed to be a realized value of a random variable, it is called a random effect" (LaMotte, 1983).
5. Fixed effects are estimated using least squares (or, more generally, maximum likelihood) and random effects are estimated with shrinkage ("linear unbiased prediction" in the terminology of Robinson, 1991). This definition is standard in the multilevel modeling literature (see, for example, Snijders and Bosker, 1999, section 4.2) and in econometrics.

   In a multilevel model, this definition implies that fixed effects β_j are estimated conditional on a group-level variance $\sigma_\beta = \infty$ and random effects β_j are estimated conditional on σ_β estimated from data.

Of these definitions, the first clearly stands apart, but the other four definitions differ also. Under the second definition, an effect can change from fixed to random with a change in the goals of inference, even if the data and design are unchanged. The third definition differs from the others in defining a finite population (while leaving open the question of what to do with a large but not exhaustive sample), while the fourth definition makes no reference to an actual (rather than mathematical) population at all. The second definition allows fixed effects to come from a distribution, as long as that distribution is not of interest, whereas the fourth and fifth do not use any distribution for inference about fixed effects. The fifth definition has the virtue of mathematical precision but leaves unclear when a given set of effects should be considered fixed or random. In summary, it is easily possible for a factor to be "fixed" according to some definitions above and "random" for others. Because of these conflicting definitions, it is no surprise that "clear answers to the question 'fixed or random?' are not necessarily the norm" (Searle, Casella, and McCulloch, 1992, p. 15).

effects when the groups in the data represent all possible groups, and random effects when the population includes groups not in the data. These two recommendations (and others) can be unhelpful. For example, in the child support example, we are interested in these particular cities and also the country as a whole. The cities are only a sample of cities in the United States—but if we were suddenly given data from all the other cities, we would not want then to change our model.

Our advice (elaborated upon in the rest of this book) is to *always* use multilevel modeling ("random effects"). Because of the conflicting definitions and advice, we avoid the terms "fixed" and "random" entirely, and focus on the description of the model itself (for example, varying intercepts and constant slopes), with the understanding that batches of coefficients (for example, $\alpha_1, \dots, \alpha_J$) will themselves be modeled.

11.5 Costs and benefits of multilevel modeling

Quick overview of classical regression

Before we go to the effort of learning multilevel modeling, it is helpful to briefly review what can be done with classical regression:

- Prediction for continuous or discrete outcomes,
- Fitting of nonlinear relations using transformations,
- Inclusion of categorical predictors using indicator variables,
- Modeling of interactions between inputs,
- Causal inference (under appropriate conditions).

Motivations for multilevel modeling

There are various reasons why it might be worth moving to a multilevel model, whether for purposes of causal inference, the study of variation, or prediction of future outcomes:

- Accounting for individual- and group-level variation in estimating *group-level* regression coefficients. For example, in the child support study in Section 11.2, interest lies in a city-level predictor (child support enforcement), and in classical regression it is not possible to include city indicators along with city-level predictors.
- Modeling variation among *individual-level* regression coefficients. In classical regression, one can do this using indicator variables, but multilevel modeling is convenient when we want to model the variation of these coefficients across groups, make predictions for new groups, or account for group-level variation in the uncertainty for individual-level coefficients.
- Estimating regression coefficients for *particular* groups. For example, in the next chapter, we discuss the problem of estimating radon levels from measurements in several counties in Minnesota. With a multilevel model, we can get reasonable estimates even for counties with small sample sizes, which would be difficult using classical regression.

One or more of these reasons might apply in any particular study.

Complexity of multilevel models

A potential drawback to multilevel modeling is the additional complexity of coefficients varying by group. We do not mind this complexity—in fact, we embrace it in its realism—however, it does create new difficulties in understanding and summarizing the model, issues we explore in Part 3 of this book.

Additional modeling assumptions

As we discuss in the next few chapters, a multilevel model requires additional assumptions beyond those of classical regression—basically, each level of the model corresponds to its own regression with its own set of assumptions such as additivity, linearity, independence, equal variance, and normality.

We usually don't mind. First, it can be possible to check these assumptions. Perhaps more important, classical regressions can typically be identified with particular special cases of multilevel models with hierarchical variance parameters set to zero or infinity—these are the *complete pooling* and *no pooling* models discussed in Sections 12.2 and 12.3. Our ultimate justification, which can be seen through examples, is that the assumptions pay off in practice in allowing more realistic models and inferences.

When does multilevel modeling make a difference?

The usual alternative to multilevel modeling is classical regression—either ignoring group-level variation, or with varying coefficients that are estimated classically (and not themselves modeled)—or combinations of classical regressions such as the individual and group-level models described on page 239.

In various limiting cases, the classical and multilevel approaches coincide. When there is very little group-level variation, the multilevel model reduces to classical regression with no group indicators; conversely, when group-level coefficients vary greatly (compared to their standard errors of estimation), multilevel modeling reduces to classical regression with group indicators.

When the number of groups is small (less than five, say), there is typically not enough information to accurately estimate group-level variation. As a result, multilevel models in this setting typically gain little beyond classical varying-coefficient models.

These limits give us a sense of where we can gain the most from multilevel modeling—where it is worth the effort of expanding a classical regression in this way. However, there is little risk from applying a multilevel model, assuming we are willing to put in the effort to set up the model and interpret the resulting inferences.

11.6 Bibliographic note

Several introductory books on multilevel models have been written in the past decade in conjunction with specialized computer programs (see Section 1.5), including Raudenbush and Bryk (2002), Goldstein (1995), and Snijders and Bosker (1999). Kreft and De Leeuw (1998) provide an accessible introduction and a good place to start (although we do not agree with all of their recommendations). These books have a social science focus, perhaps because it is harder to justify the use of linear models in laboratory sciences where it is easier to isolate the effects of individual factors and so the functional form of responses is better understood. Giltinan and Davidian (1995) and Verbeke and Molenberghs (2000) are books on nonlinear multilevel models focusing on biostatistical applications.

Another approach to regression with multilevel data structures is to use classical estimates and then correct the standard errors to deal with the dependence in the data. We briefly discuss the connection between multilevel models and correlated-error models in Section 12.5 but do not consider these other inferential methods, which include *generalized estimating equations* (see Carlin et al., 2001, for a comparison to multilevel models) and *panel-corrected standard errors* (see Beck and Katz, 1995, 1996).

The articles in the special issue of *Political Analysis* devoted to multilevel modeling (Kedar and Shively, 2005) illustrate several different forms of analysis of multilevel data, including two-level classical regression and multilevel modeling.

Gelman (2005) discusses difficulties with the terms "fixed" and "random" effects. 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.

The child support example comes from Nepomnyaschy and Garfinkel (2005). The teenage smoking example comes from Carlin et al. (2001), who consider several different models, including a multilevel logistic regression.

11.7 Exercises

1. The file `apt.dat` in the folder `rodents` contains data on rodent infestation in a sample of New York City apartments (see codebook `rodents.doc`). The file `dist.dat` contains data on the 55 "community districts" (neighborhoods) in the city.

   (a) Write the notation for a varying-intercept multilevel logistic regression (with community districts as the groups) for the probability of rodent infestation using the individual-level predictors but no group-level predictors.

   (b) Expand the model in (a) by including the variables in `dist.dat` as group-level predictors.

2. Time-series cross-sectional data: download data with an outcome y and predictors X in each of J countries for a series of K consecutive years. The outcome should be some measure of educational achievement of children and the predictors should be a per capita income measure, a measure of income inequality, and a variable summarizing how democratic the country is. For these countries, also create country-level predictors that are indicators for the countries' geographic regions.

   (a) Set up the data as a wide matrix of countries $\times$ measurements (as in Figure 11.6).

   (b) Set up the data as two matrices as in Figure 11.7: a long matrix with JK rows with all the measurements, and a matrix with J rows, with information on each country.

   (c) Write a multilevel regression as in (11.5)–(11.7). Explain the meaning of all the variables in the model.

3. The folder `olympics` has seven judges' ratings of seven figure skaters (on two criteria: "technical merit" and "artistic impression") from the 1932 Winter Olympics.

   (a) Construct a $7 \times 7 \times 2$ array of the data (ordered by skater, judge, and judging criterion).

   (b) Reformulate the data as a 98×4 array (similar to the top table in Figure 11.7), where the first two columns are the technical merit and artistic impression scores, the third column is a skater ID, and the fourth column is a judge ID.

(c) Add another column to this matrix representing an indicator variable that equals 1 if the skater and judge are from the same country, or 0 otherwise.

4. The folder `cd4` has CD4 percentages for a set of young children with HIV who were measured several times over a period of two years. The dataset also includes the ages of the children at each measurement.

(a) Graph the outcome (the CD4 percentage, on the square root scale) for each child as a function of time.

(b) Each child's data has a time course that can be summarized by a linear fit. Estimate these lines and plot them for all the children.

(c) Set up a model for the children's slopes and intercepts as a function of the treatment and age at baseline. Estimate this model using the two-step procedure–first estimate the intercept and slope separately for each child, then fit the between-child models using the point estimates from the first step.

CHAPTER 12

Multilevel linear models: the basics

Multilevel modeling can be thought of in two equivalent ways:

- We can think of a generalization of linear regression, where intercepts, and possibly slopes, are allowed to vary by group. For example, starting with a regression model with one predictor, $y_i = \alpha + \beta x_i + \epsilon_i$, we can generalize to the varying-intercept model, $y_i = \alpha_{j[i]} + \beta x_i + \epsilon_i$, and the varying-intercept, varying-slope model, $y_i = \alpha_{j[i]} + \beta_{j[i]} x_i + \epsilon_i$ (see Figure 11.1 on page 238).
- Equivalently, we can think of multilevel modeling as a regression that includes a categorical input variable representing group membership. From this perspective, the group index is a factor with J levels, corresponding to J predictors in the regression model (or $2J$ if they are interacted with a predictor x in a varying-intercept, varying-slope model; or $3J$ if they are interacted with two predictors $X_{(1)}, X_{(2)}$; and so forth).

In either case, $J-1$ linear predictors are added to the model (or, to put it another way, the constant term in the regression is replaced by J separate intercept terms). The crucial multilevel modeling step is that these J coefficients are then themselves given a model (most simply, a common distribution for the J parameters α_j or, more generally, a regression model for the α_j's given group-level predictors). The group-level model is estimated simultaneously with the data-level regression of y.

This chapter introduces multilevel linear regression step by step. We begin in Section 12.2 by characterizing multilevel modeling as a compromise between two extremes: *complete pooling*, in which the group indicators are not included in the model, and *no pooling*, in which separate models are fit within each group. After laying out some notational difficulties in Section 12.5, we discuss in Section 12.6 the different roles of the individual- and group-level regressions. Chapter 13 continues with more complex multilevel structures.

12.1 Notation

We briefly review the notation for classical regression and then outline how it can be generalized for multilevel models. As we illustrate in the examples, however, no single notation is appropriate for all problems. We use the following notation for classical regression:

- Units $i = 1, \dots, n$. By *units*, we mean the smallest items of measurement.
- Outcome measurements $y = (y_1, \dots, y_n)$. These are the unit-level data being modeled.
- Regression predictors are represented by an $n \times k$ matrix X, so that the vector of predicted values is $\hat{y} = X\beta$, where $\hat{y}$ and β are column vectors of length n and k, respectively. We include in X the constant term (unless it is explicitly excluded from the model), so that the first column of X is all 1's. We usually label the coefficients as $\beta_0, \dots, \beta_{k-1}$, but sometimes we index from 1 to k.
- For each individual unit i, we denote its row vector of predictors as X_i. Thus, $\hat{y}_i = X_i\beta$ is the prediction for unit i.

- For each predictor κ, we label the $(\kappa{+}1)^{st}$ column of X as $X_{(\kappa)}$ (assuming that $X_{(0)}$ is a column of 1's).
- Any information contained in the unit labels i should be coded in the regression inputs. For example, if $i = 1, \dots, n$ represents the order in which persons i enrolled in a study, we should create a time variable t_i and, for example, include it in the matrix X of regression predictors. Or, more generally, consider transformations and interactions of this new input variable.

For multilevel models, we label:

- Groups $j = 1, \dots, J$. This works for a single level of grouping (for example, students within schools, or persons within states).
- We occasionally use $k = 1, \dots, K$ for a second level of grouping (for example, students within schools within districts; or, for a non-nested example, test responses that can be characterized by person or by item). In any particular example, we have to distinguish this k from the number of predictors in X. For more complicated examples we develop idiosyncratic notation as appropriate.
- Index variables $j[i]$ code group membership. For example, if $j[35] = 4$, then the 35^{th} unit in the data ($i = 35$) belongs to group 4.
- Coefficients are sometimes written as a vector β, sometimes as α, β (as in Figure 11.1 on page 238), with group-level regression coefficients typically called γ.
- We make our R and Bugs code more readable by typing α, β, γ as `a,b,g`.
- We write the varying-intercept model with one additional predictor as $y_i = \alpha_{j[i]} + \beta x_i + \epsilon_i$ or $y_i \sim \mathrm{N}(\alpha_{j[i]} + \beta x_i, \sigma_y^2)$. Similarly, the varying-intercept, varying-slope model is $y_i = \alpha_{j[i]} + \beta_{j[i]} x_i + \epsilon_i$ or $y_i \sim \mathrm{N}(\alpha_{j[i]} + \beta_{j[i]} x_i, \sigma_y^2)$.
- With multiple predictors, we write $y_i = X_i B + \epsilon_i$, or $y_i \sim \mathrm{N}(X_i B, \sigma_y^2)$. B is a matrix of coefficients that can be modeled using a general varying-intercept, varying-slope model (as discussed in the next chapter).
- Standard deviation is σ_y for data-level errors and $\sigma_\alpha, \sigma_\beta$, and so forth, for group-level errors.
- Group-level predictors are represented by a matrix U with J rows, for example, in the group-level model, $\alpha_j \sim \mathrm{N}(U_j \gamma,\, \sigma_\alpha^2)$. When there is a single group-level predictor, we label it as lowercase u.

12.2 Partial pooling with no predictors

As noted in Section 1.3, multilevel regression can be thought of as a method for compromising between the two extremes of excluding a categorical predictor from a model (*complete pooling*), or estimating separate models within each level of the categorical predictor (*no pooling*).

Complete-pooling and no-pooling estimates of county radon levels

We illustrate with the home radon example, which we introduced in Section 1.2 and shall use throughout this chapter. Consider the goal of estimating the distribution of radon levels of the houses within each of the 85 counties in Minnesota.[1] This seems

[1] Radon levels are always positive, and it is reasonable to suppose that effects will be multiplicative; hence it is appropriate to model the data on the logarithmic scale (see Section 4.4). For some purposes, though, such as estimating total cancer risk, it makes sense to estimate averages on the original, unlogged scale; we can obtain these inferences using simulation, as discussed at the end of Section 12.8.

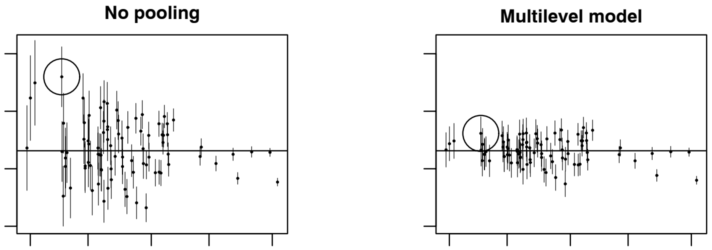

Figure 12.1 *Estimates ± standard errors for the average log radon levels in Minnesota counties plotted versus the (jittered) number of observations in the county: (a) no-pooling analysis, (b) multilevel (partial pooling) analysis, in both cases with no house-level or county-level predictors. The counties with fewer measurements have more variable estimates and larger higher standard errors. The horizontal line in each plot represents an estimate of the average radon level across all counties. The left plot illustrates a problem with the no-pooling analysis: it systematically causes us to think that certain counties are more extreme, just because they have smaller sample sizes.*

simple enough. One estimate would be the average that completely pools data across all counties. This ignores variation among counties in radon levels, however, so perhaps a better option would be simply to use the average log radon level in each county. Figure 12.1a plots these averages against the number of observations in each county.

Whereas complete pooling ignores variation between counties, the no-pooling analysis overstates it. To put it another way, the no-pooling analysis overfits the data within each county. To see this, consider Lac Qui Parle County (circled in the plot), which has the highest average radon level of all 85 counties in Minnesota. This average, however, is estimated using only two data points. Lac Qui Parle may very well be a high-radon county, but do we really believe it is *that* high? Maybe, but probably not: given the variability in the data we would not have much trust in an estimate based on only two measurements.

To put it another way, looking at all the counties together: the estimates from the no-pooling model overstate the variation among counties and tend to make the individual counties look more different than they actually are.

Partial-pooling estimates from a multilevel model

The multilevel estimates of these averages, displayed in Figure 12.1b, represent a compromise between these two extremes. The goal of estimation is the average log radon level α_j among all the houses in county j, for which all we have available are a random sample of size n_j. For this simple scenario with no predictors, the multilevel estimate for a given county j can be approximated as a weighted average of the mean of the observations in the county (the unpooled estimate, $\bar{y}_j$) and the mean over all counties (the completely pooled estimate, $\bar{y}_{\rm all}$):

$$\hat{\alpha}_j^{\rm multilevel} \approx \frac{\frac{n_j}{\sigma_y^2}\bar{y}_j + \frac{1}{\sigma_\alpha^2}\bar{y}_{\rm all}}{\frac{n_j}{\sigma_y^2} + \frac{1}{\sigma_\alpha^2}}, \qquad (12.1)$$

where n_j is the number of measured houses in county j, σ_y^2 is the within-county variance in log radon measurements, and σ_α^2 is the variance among the average log radon levels of the different counties. We could also allow the within-county variance to vary by county (in which case σ_y would be replaced by $\sigma_{y\,j}$ in the preceding formula) but for simplicity we assume it is constant.

The weighted average (12.1) reflects the relative amount of information available about the individual county, on one hand, and the average of all the counties, on the other:

- Averages from counties with smaller sample sizes carry less information, and the weighting pulls the multilevel estimates closer to the overall state average. In the limit, if $n_j = 0$, the multilevel estimate is simply the overall average, $\bar{y}_{\text{all}}$.
- Averages from counties with larger sample sizes carry more information, and the corresponding multilevel estimates are close to the county averages. In the limit as $n_j \to \infty$, the multilevel estimate is simply the county average, $\bar{y}_j$.
- In intermediate cases, the multilevel estimate lies between the two extremes.

To actually apply (12.1), we need estimates of the variation within and between counties. In practice, we estimate these variance parameters together with the α_j's, either with an approximate program such as `lmer()` (see Section 12.4) or using fully Bayesian inference, as implemented in Bugs and described in Part 2B of this book. For now, we present inferences (as in Figure 12.1) without dwelling on the details of estimation.

12.3 Partial pooling with predictors

The same principle of finding a compromise between the extremes of complete pooling and no pooling applies for more general models. This section considers partial pooling for a model with unit-level predictors. In this scenario, no pooling might refer to fitting a separate regression model within each group. However, a less extreme and more common option that we also sometimes refer to as "no pooling" is a model that includes group indicators and estimates the model classically.[2]

As we move on to more complicated models, we present estimates graphically but do not continue with formulas of the form (12.1). However, the general principle remains that multilevel models compromise between pooled and unpooled estimates, with the relative weights determined by the sample size in the group and the variation within and between groups.

Complete-pooling and no-pooling analyses for the radon data, with predictors

Continuing with the radon data, Figure 12.2 shows the logarithm of the home radon measurement versus floor of measurement[3] for houses sampled from eight of the 85 counties in Minnesota. (We fit our model to the data from all 85 counties, including a total of 919 measurements, but to save space we display the data and estimates for a selection of eight counties, chosen to capture a range of the sample sizes in the survey.)

In each graph of Figure 12.2, the dashed line shows the linear regression of log

[2] This version of "no pooling" does not pool the estimates for the intercepts—the parameters we focus on in the current discussion—but it does completely pool estimates for any slope coefficients (they are forced to have the same value across all groups) and also assumes the residual variance is the same within each group.

[3] Measurements were taken in the lowest living area of each house, with basement coded as 0 and first floor coded as 1.

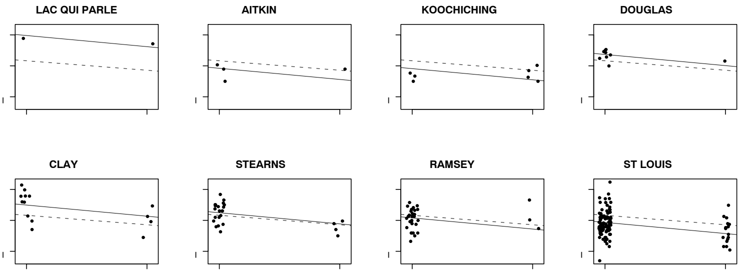

Figure 12.2 *Complete-pooling (dashed lines, $y = \alpha + \beta x$) and no-pooling (solid lines, $y = \alpha_j + \beta x$) regressions fit to radon data from the 85 counties in Minnesota, and displayed for eight of the counties. The estimated slopes β differ slightly for the two models, but here our focus is on the intercepts.*

radon, given the floor of measurement, using a model that pools all counties together (so the same line appears in all eight plots), and the solid line shows the no-pooling regressions, obtained by including county indicators in the regression (with the constant term removed to avoid collinearity; we also could have kept the constant term and included indicators for all but one of the counties). We can write the complete-pooling regression as $y_i = \alpha + \beta x_i + \epsilon_i$ and the no-pooling regression as $y_i = \alpha_{j[i]} + \beta x_i + \epsilon_i$, where $j[i]$ is the county corresponding to house i. The solid lines then plot $y = \hat{\alpha} + \hat{\beta} x$ from the complete-pooling model, and the dashed lines show $y = \hat{\alpha}_j + \hat{\beta} x$, for $j = 1, \ldots, 8$, from the no-pooling model.

Here is the complete-pooling regression for the radon data:

R output

```
lm(formula = y ~ x)
             coef.est coef.se
(Intercept)  1.33     0.03
x           -0.61     0.07
  n = 919, k = 2
  residual sd = 0.82
```

To fit the no-pooling model in R, we include the county index (a variable named `county` that takes on values between 1 and 85) as a factor in the regression—thus, predictors for the 85 different counties. We add "−1" to the regression formula to remove the constant term, so that all 85 counties are included. Otherwise, R would use county 1 as a baseline.

R output

```
lm(formula = y ~ x + factor(county) - 1)
                  coef.est coef.sd
x                 -0.72     0.07
factor(county)1    0.84     0.38
factor(county)2    0.87     0.10
. . .
factor(county)85   1.19     0.53
  n = 919, k = 86
  residual sd = 0.76
```

The estimated slopes β differ slightly for the two regressions. The no-pooling model includes county indicators, which can change the estimated coefficient for x, if the proportion of houses with basements varies among counties. This is just

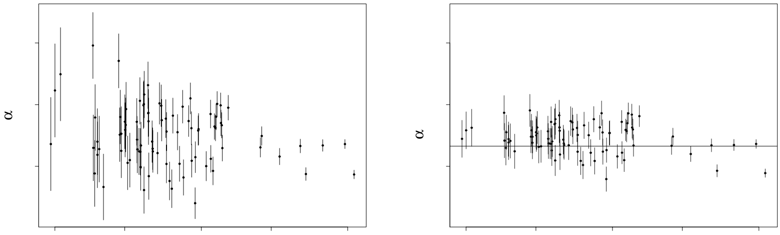

Figure 12.3 *(a) Estimates ± standard errors for the county intercepts α_j in the model $y_i = \alpha_{j[i]} + \beta x_i + \text{error}_i$, for the no-pooling analysis of the radon data, plotted versus number of observations from the county. The counties with fewer measurements have more variable estimates with higher standard errors. This graph illustrates a problem with classical regression: it systematically causes us to think that certain counties are more extreme, just because they have smaller sample sizes.*
(b) Multilevel (partial pooling) estimates ± standard errors for the county intercepts α_j for the radon data, plotted versus number of observations from the county. The horizontal line shows the complete pooling estimate. Comparing to the left plot (no pooling), which is on the same scale, we see that the multilevel estimate is typically closer to the complete-pooling estimate for counties with few observations, and closer to the no-pooling estimates for counties with many observations.
These plots differ only slightly from the no-pooling and multilevel estimates without the house-level predictor, as displayed in Figure 12.1.

a special case of the rule that adding new predictors in a regression can change the estimated coefficient of x, if these new predictors are correlated with x. In the particular example shown in Figure 12.2, the complete-pooling and no-pooling estimates of β differ only slightly; in the graphs, the difference can be seen most clearly in Stearns and Ramsey counties.

Problems with the no-pooling and complete-pooling analyses

Both the analyses shown in Figure 12.2 have problems. The complete-pooling analysis ignores any variation in average radon levels between counties. This is undesirable, particularly since the goal of our analysis was to identify counties with high-radon homes. We do not want to pool away the main subject of our study!

The no-pooling analysis has problems too, however, which we can again see in Lac Qui Parle County. Even after controlling for the floors of measurement, this county has the highest fitted line (that is, the highest estimate $\hat{\alpha}_j$), but again we do not have much trust in an estimate based on only two observations.

More generally, we would expect the counties with the least data to get more extreme estimates $\hat{\alpha}_j$ in the no-pooling analyses. Figure 12.3a illustrates with the estimates ± standard errors for the county intercepts α_j, plotted versus the sample size in each county j.

Multilevel analysis

The simplest multilevel model for the radon data with the floor predictor can be written as

$$y_i \sim \mathrm{N}(\alpha_{j[i]} + \beta x_i, \sigma_y^2), \quad \text{for } i = 1, \ldots, n, \tag{12.2}$$

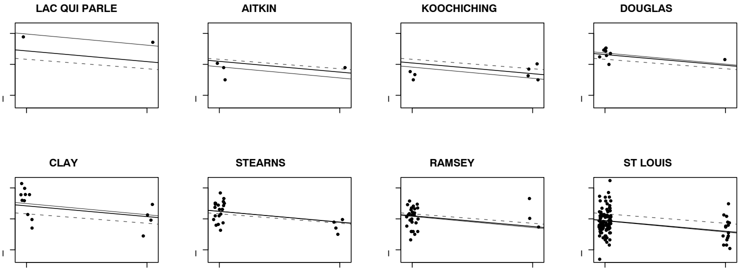

Figure 12.4 *Multilevel (partial pooling) regression lines $y = \alpha_j + \beta x$ fit to radon data from Minnesota, displayed for eight counties. Light-colored dashed and solid lines show the complete-pooling and no-pooling estimates, respectively, from Figure 12.3a.*

which looks like the no-pooling model but with one key difference. In the no-pooling model, the α_j's are set to the classical least squares estimates, which correspond to the fitted intercepts in a model run separately in each county (with the constraint that the slope coefficient equals β in all models). Model (12.2) also looks a little like the complete-pooling model except that, with complete pooling, the α_j's are given a "hard constraint"—they are all fixed at a common α.

In the multilevel model, a "soft constraint" is applied to the α_j's: they are assigned a probability distribution,

$$\alpha_j \sim \mathrm{N}(\mu_\alpha, \sigma_\alpha^2), \quad \text{for } j = 1, \ldots, J, \tag{12.3}$$

with their mean μ_α and standard deviation σ_α estimated from the data. The distribution (12.3) has the effect of pulling the estimates of α_j toward the mean level μ_α, but not all the way—thus, in each county, a *partial-pooling* compromise between the two estimates shown in Figure 12.2. In the limit of $\sigma_\alpha \to \infty$, the soft constraints do nothing, and there is no pooling; as $\sigma_\alpha \to 0$, they pull the estimates all the way to zero, yielding the complete-pooling estimate.

Figure 12.4 shows, for the radon example, the estimated line from the multilevel model (12.2), which in each county lies between the complete-pooling and no-pooling regression lines. There is strong pooling (solid line closer to complete-pooling line) in counties with small sample sizes, and only weak pooling (solid line closer to no-pooling line) in counties containing many measurements.

Going back to Figure 12.3, the right panel shows the estimates and standard errors for the county intercepts α_j from the multilevel model, plotted versus county sample size. Comparing to the left panel, we see more pooling for the counties with fewer observations. We also see a trend that counties with larger sample sizes have lower radon levels, indicating that "county sample size" is correlated with some relevant county-level predictor.

Average regression line and individual- and group-level variances

Multilevel models typically have so many parameters that it is not feasible to closely examine all their numerical estimates. Instead we plot the estimated group-level models (as in Figure 12.4) and varying parameters (as in Figure 12.3b) to look for patterns and facilitate comparisons across counties. It can be helpful, however, to look at numerical summaries for the *hyperparameters*—those model parameters without group-level subscripts.

For example, in the radon model, the hyperparameters are estimated as $\hat{\mu}_\alpha = 1.46$, $\hat{\beta} = -0.69$, $\hat{\sigma}_y = 0.76$, and $\hat{\sigma}_\alpha = 0.33$. (We show the estimates in Section 12.4.) That is, the estimated average regression line for all the counties is $y = 1.46 - 0.69x$, with error standard deviations of 0.76 at the individual level and 0.33 at the county level. For this dataset, variation within counties (after controlling for the floor of measurement) is comparable to the average difference between measurements in houses with and without basements.

One way to interpret the variation between counties, σ_α, is to consider the variance ratio, $\sigma_\alpha^2/\sigma_y^2$, which in this example is estimated at $0.33^2/0.76^2 = 0.19$, or about one-fifth. Thus, the standard deviation of average radon levels between counties is the same as the standard deviation of the average of 5 measurements within a county (that is, $0.76/\sqrt{5} = 0.33$). The relative values of individual- and group-level variances are also sometimes expressed using the *intraclass correlation*, $\sigma_\alpha^2/(\sigma_\alpha^2 + \sigma_y^2)$, which ranges from 0 if the grouping conveys no information to 1 if all members of a group are identical.

In our example, the group-level model tells us that the county intercepts, α_j, have an estimated mean of 1.46 and standard deviation of 0.33. (What is relevant to our discussion here is the standard deviation, not the mean.) The amount of information in this distribution is the same as that in 5 measurements within a county. To put it another way, for a county with a sample size less than 5, there is more information in the group-level model than in the county's data; for a county with more than 5 observations, the within-county measurements are more informative (in the sense of providing a lower-variance estimate of the county's average radon level). As a result, the multilevel regression line in a county is closer to the complete-pooling estimate when sample size is less than 5, and closer to the no-pooling estimate when sample size exceeds 5. We can see this in Figure 12.4: as sample size increases, the multilevel estimates move closer and closer to the no-pooling lines.

Partial pooling (shrinkage) of group coefficients α_j

Multilevel modeling partially pools the group-level parameters α_j toward their mean level, μ_α. There is more pooling when the group-level standard deviation σ_α is small, and more smoothing for groups with fewer observations. Generalizing (12.1), the multilevel-modeling estimate of α_j can be expressed as a weighted average of the no-pooling estimate for its group $(\bar{y}_j - \beta\bar{x}_j)$ and the mean, μ_α:

$$\text{estimate of } \alpha_j \approx \frac{\frac{n_j}{\sigma_y^2}}{\frac{n_j}{\sigma_y^2} + \frac{1}{\sigma_\alpha^2}}(\bar{y}_j - \beta\bar{x}_j) + \frac{\frac{1}{\sigma_\alpha^2}}{\frac{n_j}{\sigma_y^2} + \frac{1}{\sigma_\alpha^2}}\mu_\alpha. \tag{12.4}$$

When actually fitting multilevel models, we do not actually use this formula; rather, we fit models using `lmer()` or Bugs, which automatically perform the calculations, using formulas such as (12.4) internally. Chapter 19 provides more detail on the algorithms used to fit these models.

Classical regression as a special case

Classical regression models can be viewed as special cases of multilevel models. The limit of $\sigma_\alpha \to 0$ yields the complete-pooling model, and $\sigma_\alpha \to \infty$ reduces to the no-pooling model. Given multilevel data, we can estimate σ_α. Therefore we see no reason (except for convenience) to accept estimates that arbitrarily set this parameter to one of these two extreme values.

12.4 Quickly fitting multilevel models in R

We fit most of the multilevel models in this part of the book using the `lmer()` function, which fits linear and generalized linear models with varying coefficients.[4] Part 2B of the book considers computation in more detail, including a discussion of why it can be helpful to make the extra effort and program models using Bugs (typically using a simpler `lmer()` fit as a starting point). The `lmer()` function is currently part of the R package `Matrix`; see Appendix C for details. Here we introduce `lmer()` in the context of simple varying-intercept models.

The lmer function

Varying-intercept model with no predictors. The varying intercept model with no predictors (discussed in Section 12.2) can be fit and displayed using `lmer()` as follows:

R code

```
M0 <- lmer (y ~ 1 + (1 | county))
display (M0)
```

This model simply includes a constant term (the predictor "1") and allows it to vary by county. We next move to a more interesting model including the floor of measurement as an individual-level predictor.

Varying-intercept model with an individual-level predictor. We shall introduce multilevel fitting with model (12.2)–(12.3), the varying-intercept regression with a single predictor. We start with the call to `lmer()`:

R code

```
M1 <- lmer (y ~ x + (1 | county))
```

This expression starts with the no-pooling model, "`y ~ x`," and then adds "`(1 | county)`," which allows the intercept (the coefficient of the predictor "1," which is the column of ones—the constant term in the regression) to vary by county.

We can then display a quick summary of the fit:

R code

```
display (M1)
```

which yields

R output

```
lmer(formula = y ~ x + (1 | county))
            coef.est coef.se
(Intercept)  1.46     0.05
x           -0.69     0.07
Error terms:
 Groups   Name        Std.Dev.
 county   (Intercept) 0.33
 Residual             0.76
# of obs: 919, groups: county, 85
deviance = 2163.7
```

[4] The name `lmer` stands for "linear mixed effects in R," but the function actually works for generalized linear models as well. The term "mixed effects" refers to random effects (coefficients that vary by group) and fixed effects (coefficients that do not vary). We avoid the terms "fixed" and "random" (see page 245) and instead refer to coefficients as "modeled" (that is, grouped) or "unmodeled."

The top part of this display shows the inference about the intercept and slope for the model, averaging over the counties. The bottom part gives the estimated variation: $\hat{\sigma}_\alpha = 0.33$ and $\hat{\sigma}_y = 0.76$. We also see that the model was fit to 919 houses within 85 counties. We shall ignore the deviance for now.

Estimated regression coefficients

To see the estimated model within each county. We type

R code

```
coef (M1)
```

which yields

R output

```
$county
   (Intercept)     x
1         1.19 -0.69
2         0.93 -0.69
3         1.48 -0.69
. . .
85        1.39 -0.69
```

Thus, the estimated regression line is $y = 1.19 - 0.69x$ in county 1, $y = 0.93 + 0.69x$ in county 2, and so forth. The slopes are all identical because they were specified thus in the model. (The specification `(1|county)` tells the model to allow only the intercept to vary. As we shall discuss in the next chapter, we can allow the slope to vary by specifying `(1+x|county)` in the regression model.)

Fixed and random effects. Alternatively, we can separately look at the estimated model averaging over the counties—the "fixed effects"—and the county-level errors—the "random effects." Typing

R code

```
fixef (M1)
```

yields

R output

```
(Intercept)           x
       1.46       -0.69
```

The estimated regression line in an average county is thus $y = 1.46 - 0.69x$. We can then look at the county-level errors:

R code

```
ranef (M1)
```

which yields

R output

```
   (Intercept)
1        -0.27
2        -0.53
3         0.02
. . .
85       -0.08
```

These tell us how much the intercept is shifted up or down in particular counties. Thus, for example, in county 1, the estimated intercept is 0.27 lower than average, so that the regression line is $(1.46 - 0.27) - 0.69x = 1.19 - 0.69x$, which is what we saw earlier from the call to `coef()`. For some applications, it is best to see the estimated model within each group; for others, it is helpful to see the estimated average model and group-level errors.

Uncertainties in the estimated coefficients

We wrote little functions `se.fixef()` and `se.ranef()` for quickly pulling out these standard errors from the model fitted by `lmer()`. In this example,

R code

```
se.fixef (M1)
```

yields

R output

```
(Intercept)           x
      0.05         0.07
```

and

R code

```
se.ranef (M1)
```

yields,

R output

```
$county
   (Intercept)
1         0.25
2         0.10
3         0.26
. . .
85        0.28
```

As discussed in Section 12.3, the standard errors differ according to the sample size within each county; for example, counties 1, 2, and 85 have 4, 52, and 2 houses, respectively, in the sample. For the within-county regressions, standard errors are only given for the intercepts, since this model has a common slope for all counties.

Summarizing and displaying the fitted model

We can access the components of the estimates and standard errors using list notation in R. For example, to get a 95% confidence interval for the slope (which, in this model, does not vary by county):

R code

```
fixef(M1)["x"] + c(-2,2)*se.fixef(M1)["x"]
```

or, equivalently, since the slope is the second coefficient in the regression,

R code

```
fixef(M1)[2] + c(-2,2)*se.fixef(M1)[2]
```

The term "fixed effects" is used for the regression coefficients that do not vary by group (such as the coefficient for x in this example) or for group-level coefficients or group averages (such as the average intercept, μ_α in (12.3)).

Identifying the batches of coefficients. In pulling out elements of the coefficients from `coef()` or `ranef()`, we must first identify the grouping (`county`, in this case). The need for this labeling will become clear in the next chapter in the context of non-nested models, where there are different levels of grouping and thus different structures of varying coefficients.

For example, here is a 95% confidence interval for the intercept in county 26:

R code

```
coef(M1)$county[26,1] + c(-2,2)*se.ranef(M1)$county[26]
```

and here is a 95% confidence interval for the error in the intercept in that county (that is, the deviation from the average):

R code

```
as.matrix(ranef(M1)$county)[26] + c(-2,2)*se.ranef(M1)$county[26]
```

For a more elaborate example, we make Figure 12.4 using the following commands:

R code

```
a.hat.M1 <- coef(M1)$county[,1]    # 1st column is the intercept
b.hat.M1 <- coef(M1)$county[,2]    # 2nd element is the slope
x.jitter <- x + runif(n,-.05,.05)    # jittered data for plotting
par (mfrow=c(2,4))                   # make a 2x4 grid of plots
for (j in display8){
  plot (x.jitter[county==j], y[county==j], xlim=c(-.05,1.05),
    ylim=y.range, xlab="floor", ylab="log radon level", main=uniq.name[j])
## [uniq.name is a vector of county names that was created earlier]
  curve (coef(lm.pooled)[1] + coef(lm.pooled)[2]*x, lty=2, col="gray10",
    add=TRUE)
  curve (coef(lm.unpooled)[j+1] + coef(lm.unpooled)[1]*x, col="gray10",
    add=TRUE)
  curve (a.hat.M1[j] + b.hat.M1[j]*x, lwd=1, col="black", add=TRUE)
}
```

Here, lm.pooled and lm.unpooled are the classical regressions that we have already fit.

More complicated models

The lmer() function can also handle many of the multilevel regressions discussed in this part of the book, including group-level predictors, varying intercepts and slopes, nested and non-nested structures, and multilevel generalized linear models. Approximate routines such as lmer() tend to work well when the sample size and number of groups is moderate to large, as in the radon models. When the number of groups is small, or the model becomes more complicated, it can be useful to switch to Bayesian inference, using the Bugs program, to better account for uncertainty in model fitting. We return to this point in Section 16.1.

12.5 Five ways to write the same model

We begin our treatment of multilevel models with the simplest structures—*nested* models, in which we have observations $i = 1, \dots, n$ clustered in groups $j = 1, \dots, J$, and we wish to model variation among groups. Often, predictors are available at the individual and group levels. We shall use as a running example the home radon analysis described above, using as predictors the house-level x_i and a measure of the logarithm of soil uranium as a county-level predictor, u_j. For some versions of the model, we include these both as individual-level predictors and label them as X_{i1} and X_{i2}.

There are several different ways of writing a multilevel model. Rather than introducing a restrictive uniform notation, we describe these different formulations and explain how they are connected. It is useful to be able to express a model in different ways, partly so that we can recognize the similarities between models that only appear to be different, and partly for computational reasons.

Allowing regression coefficients to vary across groups

Perhaps the simplest way to express a multilevel model generally is by starting with the classical regression model fit to all the data, $y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \epsilon_i$, and then generalizing to allow the coefficients β to vary across groups; thus,

$$y_i = \beta_{0\,j[i]} + \beta_{1\,j[i]} X_{i1} + \beta_{2\,j[i]} X_{i2} + \cdots + \epsilon_i .$$

The "multilevel" part of the model involves assigning a multivariate distribution to the vector of β's within each group, as we discuss in Section 13.1.

For now we will focus on *varying-intercept models*, in which the only coefficient that varies across groups is the constant term β_0 (which, to minimize subscripting, we label α). For the radon data that include the floor and a county-level uranium predictor, the model then becomes

$$y_i = \alpha_{j[i]} + \beta_1 X_{i1} + \beta_2 X_{i2} + \epsilon_i$$

where X_{i1} is the i^{th} element of the vector $X_{(1)}$ representing the first-floor indicators and X_{i2} is the i^{th} element of the vector $X_{(2)}$ representing the uranium measurement in the county containing house i. We can also write this in matrix notation as

$$y_i = \alpha_{j[i]} + X_i\beta + \epsilon_i$$

with the understanding that X includes the first-floor indicator and the county uranium measurement but not the constant term. This is the way that models are built using `lmer()`, including all predictors at the individual level, as we discuss in Section 12.6.

The second level of the model is simply

$$\alpha_j \sim \mathrm{N}(\mu_\alpha, \sigma_\alpha^2). \tag{12.5}$$

Group-level errors. The model (12.5) can also be written as

$$\alpha_j = \mu_\alpha + \eta_j, \quad \text{with } \eta_j \sim \mathrm{N}(0, \sigma_\alpha^2). \tag{12.6}$$

The group-level errors η_j can be helpful in understanding the model; however, we often use the more compact notation (12.5) to reduce the profusion of notation. (We have also toyed with notation such as $\alpha_j = \mu^\alpha + \epsilon_j^\alpha$ in which ϵ is consistently used for regression errors—but the superscripts seem too confusing. As illustrated in Part 2B of this book, we sometimes use such notation when programming models in Bugs.)

Combining separate local regressions

An alternative way to write the multilevel model is as a linking of local regressions in each group. Within each group j, a regression is performed on the local predictors (in this case, simply the first-floor indicator, x_i), with a constant term α that is indexed by group:

$$\text{within county } j\text{:} \;\; y_i \sim \mathrm{N}(\alpha_j + \beta x_i, \sigma_y^2), \quad \text{for } i = 1, \ldots, n_j. \tag{12.7}$$

The county uranium measurement has not yet entered the model since we are imagining separate regressions fit to each county—there would be no way to estimate the coefficient for a county-level predictor from any of these within-county regressions.

Instead, the county-level uranium level, u_j, is included as a predictor in the second level of the model:

$$\alpha_j \sim \mathrm{N}(\gamma_0 + \gamma_1 u_j, \sigma_\alpha^2). \tag{12.8}$$

We can also write the distribution in (12.8) as $\mathrm{N}(U_j\gamma, \sigma_\alpha^2)$, where U has two columns: a constant term, $U_{(0)}$, and the county-level uranium measurement, $U_{(1)}$. The errors in this model (with mean 0 and standard deviation σ_α) represent variation *among counties* that is not explained by the local and county-level predictors.

The multilevel model combines the J local regression models (12.7) in two ways: first, the local regression coefficients β are the same in all J models (an assumption we will relax in Section 13.1). Second, the different intercepts α_j are connected through the group-level model (12.8), with consequences to the coefficient estimates that we discuss in Section 12.6.

Group-level errors. We can write (12.8) as

$$\alpha_j = \gamma_0 + \gamma_1 u_j + \eta_j, \quad \text{with } \eta_j \sim \mathrm{N}(0, \sigma_\alpha^2), \tag{12.9}$$

explicitly showing the errors in the county-level regression.

Modeling the coefficients of a large regression model

The identical model can be written as a single regression, in which the local and group-level predictors are combined into a single matrix X:

$$y_i \sim \mathrm{N}(X_i\beta, \sigma_y^2), \tag{12.10}$$

where, for our example, X includes vectors corresponding to:

- A constant term, $X_{(0)}$;
- The floor where the measurement was taken, $X_{(1)}$;
- The county-level uranium measure, $X_{(2)}$;
- J (not $J-1$) county indicators, $X_{(3)}, \ldots, X_{(J+2)}$.

At the upper level of the model, the J county indicators (which in this case are $\beta_3, \ldots, \beta_{J+2}$) follow a normal distribution:

$$\beta_j \sim \mathrm{N}(0, \sigma_\alpha^2), \quad \text{for } j = 3, \ldots, J+2. \tag{12.11}$$

In this case, we have centered the β_j distribution at 0 rather than at an estimated μ_β because any such μ_β would be statistically indistinguishable from the constant term in the regression. We return to this point shortly.

The parameters in the model (12.10)–(12.11) can be identified exactly with those in the separate local regressions above:

- The local predictor x in model (12.7) is the same as $X_{(1)}$ (the floor) here.
- The local errors ϵ_i are the same in the two models.
- The matrix of group-level predictors U in (12.8) is just $X_{(0)}$ here (the constant term) joined with $X_{(2)}$ (the uranium measure).
- The group-level errors $\eta_1, \ldots, \eta_J$ in (12.9) are identical to $\beta_3, \ldots, \beta_{J+2}$ here.
- The standard-deviation parameters σ_y and σ_α keep the same meanings in the two models.

Moving the constant term around. The multilevel model can be written in yet another equivalent way by moving the constant term:

$$\begin{aligned} y_i &= \mathrm{N}(X_i\beta, \sigma_y^2), \quad \text{for } i = 1, \ldots, n \\ \beta_j &\sim \mathrm{N}(\mu_\alpha, \sigma_\alpha^2), \quad \text{for } j = 3, \ldots, J+2. \end{aligned} \tag{12.12}$$

In this version, we have removed the constant term from X (so that it now has only $J+2$ columns) and replaced it by the equivalent term μ_α in the group-level model. The coefficients $\beta_3, \ldots, \beta_{J+2}$ for the group indicators are now centered around μ_α rather than 0, and are equivalent to $\alpha_1, \ldots, \alpha_J$ as defined earlier, for example, in model (12.9).

Regression with multiple error terms

Another option is to re-express model (12.10), treating the group-indicator coefficients as error terms rather than regression coefficients, in what is often called a "mixed effects" model popular in the social sciences:

$$\begin{aligned} y_i &\sim \mathrm{N}(X_i\beta + \eta_{j[i]}, \sigma_y^2), \quad \text{for } i = 1, \dots, n \\ \eta_j &\sim \mathrm{N}(0, \sigma_\alpha^2), \end{aligned} \tag{12.13}$$

where $j[i]$ represents the county that contains house i, and X now contains only three columns:

- A constant term, $X_{(0)}$;
- The floor, $X_{(1)}$;
- The county-level uranium measure, $X_{(2)}$.

This is the same as model (12.10)–(12.11), simply renaming some of the β_j's as η_j's. All our tools for multilevel modeling will automatically work for models with multiple error terms.

Large regression with correlated errors

Finally, we can express a multilevel model as a classical regression with correlated errors:

$$y_i = X_i\beta + \epsilon_i^{\mathrm{all}}, \quad \epsilon^{\mathrm{all}} \sim \mathrm{N}(0, \Sigma), \tag{12.14}$$

where X is now the matrix with three predictors (the constant term, first-floor indicator, and county-level uranium measure) as in (12.13), but now the errors $\epsilon_i^{\mathrm{all}}$ have an $n \times n$ covariance matrix Σ. The error $\epsilon_i^{\mathrm{all}}$ in (12.14) is equivalent to the sum of the two errors, $\eta_{j[i]} + \epsilon_i$, in (12.13). The term $\eta_{j[i]}$, which is the same for all units i in group j, induces correlation in ϵ^{all}.

In multilevel models, Σ is parameterized in some way, and these parameters are estimated from the data. For the nested multilevel model we have been considering here, the variances and covariances of the n elements of ϵ^{all} can be derived in terms of the parameters σ_y and σ_α:

$$\begin{aligned} \text{For any unit } i\text{:} \qquad & \Sigma_{ii} = \mathrm{var}(\epsilon_i^{\mathrm{all}}) = \sigma_y^2 + \sigma_\alpha^2 \\ \text{For any units } i,k \text{ within the same group } j\text{:} \qquad & \Sigma_{ik} = \mathrm{cov}(\epsilon_i^{\mathrm{all}}, \epsilon_k^{\mathrm{all}}) = \sigma_\alpha^2 \\ \text{For any units } i,k \text{ in different groups:} \qquad & \Sigma_{ik} = \mathrm{cov}(\epsilon_i^{\mathrm{all}}, \epsilon_k^{\mathrm{all}}) = 0. \end{aligned}$$

It can also be helpful to express Σ in terms of standard errors and correlations:

$$\begin{aligned} \mathrm{sd}(\epsilon_i) &= \sqrt{\Sigma_{ii}} = \sqrt{\sigma_y^2 + \sigma_\alpha^2} \\ \mathrm{corr}(\epsilon_i, \epsilon_k) &= \frac{\Sigma_{ik}}{\sqrt{\Sigma_{ii}\Sigma_{kk}}} = \begin{cases} \frac{\sigma_\alpha^2}{\sigma_y^2+\sigma_\alpha^2} & \text{if } j[i] = j[k] \\ 0 & \text{if } j[i] \neq j[k]. \end{cases} \end{aligned}$$

We generally prefer modeling the multilevel effects explicitly rather than burying them as correlations, but once again it is useful to see how the same model can be written in different ways.

12.6 Group-level predictors

Adding a group-level predictor to improve inference for group coefficients α_j

We continue with the radon example from Sections 12.2–12.3 to illustrate how a multilevel model handles predictors at the group as well as the individual levels.

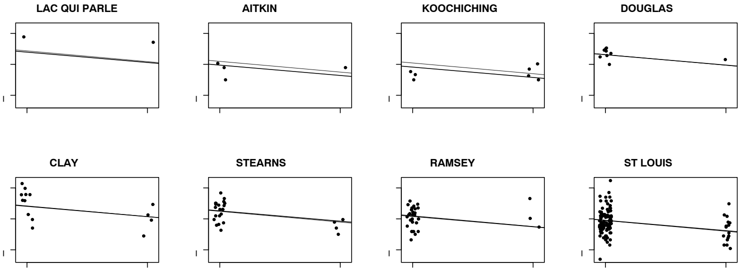

Figure 12.5 *Multilevel (partial pooling) regression lines $y = \alpha_j + \beta x$ fit to radon data, displayed for eight counties, including uranium as a county-level predictor. Light-colored lines show the multilevel estimates, without uranium as a predictor, from Figure 12.4.*

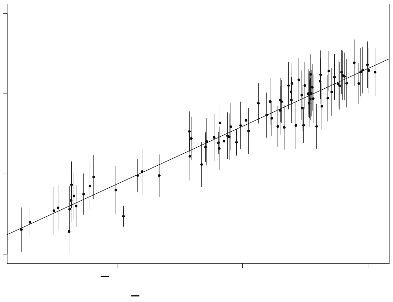

Figure 12.6 *Estimated county coefficients α_j ($\pm$1 standard error) plotted versus county-level uranium measurement u_j, along with the estimated multilevel regression line $\alpha_j = \gamma_0 + \gamma_1 u_j$. The county coefficients roughly follow the line but not exactly; the deviation of the coefficients from the line is captured in σ_α, the standard deviation of the errors in the county-level regression.*

We use the formulation

$$\begin{aligned} y_i &\sim \mathrm{N}(\alpha_{j[i]} + \beta x_i,\, \sigma_y^2), \text{ for } i = 1, \ldots, n \\ \alpha_j &\sim \mathrm{N}(\gamma_0 + \gamma_1 u_j,\, \sigma_\alpha^2), \text{ for } j = 1, \ldots, J, \end{aligned} \qquad (12.15)$$

where x_i is the house-level first-floor indicator and u_j is the county-level uranium measure.

R code

```
u.full <- u[county]
M2 <- lmer (y ~ x + u.full + (1 | county))
display (M2)
```

This model includes floor, uranium, and intercepts that vary by county. The `lmer()` function only accepts predictors at the individual level, so we have converted u_j to $u_i^{\text{full}} = u_{j[i]}$ (with the variable `county` playing the role of the indexing $j[i]$), to pull out the uranium level of the county where house i is located.

The display of the `lmer()` fit shows coefficients and standard errors, along with estimated residual variation at the county and individual ("residual") level:

R output

```
lmer(formula = y ~ x + u.full + (1 | county))
             coef.est coef.se
(Intercept)   1.47     0.04
x            -0.67     0.07
u.full        0.72     0.09
Error terms:
 Groups   Name        Std.Dev.
 county   (Intercept) 0.16
 Residual             0.76
# of obs: 919, groups: county, 85
deviance = 2122.9
```

As in our earlier example on page 261, we use `coef()` to pull out the estimated coefficients,

R code

```
coef (M2)
```

yielding

R output

```
$county
   (Intercept)     x u.full
1         1.45 -0.67   0.72
2         1.48 -0.67   0.72
. . .
85        1.42 -0.67   0.72
```

Only the intercept varies, so the coefficients for `x` and `u.full` are the same for all 85 counties. (Actually, `u.full` is constant within counties so it cannot have a varying coefficient here.) On page 280 we shall see a similar display for a model in which the coefficient for x varies by county.

As before, we can also examine the estimated model averaging over the counties:

R code

```
fixef (M2)
```

yielding

R output

```
(Intercept)           x      u.full
       1.47       -0.67        0.72
```

and the county-level errors:

R code

```
ranef (M2)
```

yielding

R output

```
   (Intercept)
1        -0.02
2         0.01
. . .
85       -0.04
```

The results of `fixef()` and `ranef()` add up to the coefficients in `coef()`: for county 1, $1.47 - 0.02 = 1.45$, for county 2, $1.47 + 0.01 = 1.48$, ..., and for county 85, $1.47 - 0.04 = 1.42$ (up to rounding error).

Interpreting the coefficients within counties

We can add the unmodeled coefficients (the "fixed effects") to the county-level errors to get an intercept and slope for each county. We start with the model that averages over all counties, $y_i = 1.47 - 0.67x_i + 0.72u_{j[i]}$ (as obtained from `display(M2)` or `fixef(M2)`.

Now consider a particular county, for example county 85. We can determine its fitted regression line in two ways from the `lmer()` output, in each case using the log uranium level in county 85, $u_{85} = 0.36$.

First, using the the last line of the display of `coef(M2)`, the fitted model for county 85 is $y_i = 1.42 - 0.67x_i + 0.72u_{85} = (1.42 + 0.72 \cdot 0.36) - 0.67x_i = 1.68 - 0.67x_i$, that is, 1.68 for a house with a basement and 1.01 for a house with no basement. Exponentiating gives estimated geometric mean predictions of 5.4 pCi/L and 2.7 pCi/L for houses in county 85 with and without basements.

Alternatively, we can construct the fitted line for county 85 by starting with the results from `fixef(M2)`—that is, $y_i = 1.47 - 0.67x_i + 0.72u_{j[i]}$, setting $u_{j[i]} = u_{85} = 0.36$—and adding the group-level error from `ranef(M2)`, which for county 85 is -0.04. The resulting model is $y_i = 1.47 - 0.67x_i + 0.72 \cdot 0.36 - 0.04 = 1.68 - 0.67x_i$, the same as in the other calculation (up to rounding error in the last digit of the intercept).

Figure 12.5 shows the fitted line for each of a selection of counties, and Figure 12.6 shows the county-level regression, plotting the estimated coefficients α_j versus the county-level predictor u_j. These two figures represent the two levels of the multilevel model.

The group-level predictor has increased the precision of our estimates of the county intercepts α_j: the ±1 standard-error bounds are narrower in Figure 12.6 than in Figure 12.3b, which showed α_j's estimated without the uranium predictor (note the different scales on the y-axes of the two plots and the different county variables plotted on the x-axes).

The estimated individual- and county-level standard deviations in this model are $\hat{\sigma}_y = 0.76$ and $\hat{\sigma}_\alpha = 0.16$. In comparison, these residual standard deviations were 0.76 and 0.33 without the uranium predictor. This predictor has left the within-county variation unchanged—which makes sense, since it is a county-level predictor which has no hope of explaining variation within any county—but has drastically reduced the unexplained variation between counties. In fact, the variance ratio is now only $\sigma_\alpha^2/\sigma_y^2 = 0.16^2/0.76^2 = 0.044$, so that the county-level model is as good as $1/0.044 = 23$ observations within any county. The multilevel estimates under this new model will be close to the complete-pooling estimates (with county-level uranium included as a predictor) for many of the smaller counties in the dataset because a county would have to have more than 23 observations to be pulled closer to the no-pooling estimate than the complete-pooling estimate.

Interpreting the coefficient of the group-level predictor

The line in Figure 12.6 shows the prediction of average log radon in a county (for homes with basements—that is, $x_i = 0$—since these are the intercepts α_j), as a function of the log uranium level in the county. This estimated group-level regression line has an estimated slope of about 0.7. Coefficients between 0 and 1 are typical in a log-log regression: in this case, each increase of 1% in uranium level corresponds to a 0.7% predicted increase in radon.

It makes sense that counties higher in uranium have higher radon levels, and it also makes sense that the slope is less than 1. Radon is affected by factors other than soil uranium, and the "uranium" variable in the dataset is itself an imprecise measure of actual soil uranium in the county, and so we would expect a 1% increase in the uranium variable to match to something less than a 1% increase in radon. Compared to classical regression, the estimation of this coefficient is trickier (since the α_j's—the "data" for the county-level regression—are not themselves observed) but the principles of interpretation do not change.

A multilevel model can include county indicators along with a county-level predictor

Users of multilevel models are often confused by the idea of including county indicators along with a county-level predictor. Is this possible? With 85 counties in the dataset, how can a regression fit 85 coefficients for counties, plus a coefficient for county-level uranium? This would seem to induce perfect collinearity into the regression or, to put it more bluntly, to attempt to learn more than the data can tell us. Is it really possible to estimate 86 coefficients from 85 data points?

The short answer is that we really have more than 85 data points. There are hundreds of houses with which to estimate the 85 county-level intercepts, and 85 counties with which to estimate the coefficient of county-level uranium. In a classical regression, however, the 85 county indicators and the county-level predictor would indeed be collinear. This problem is avoided in a multilevel model because of the partial pooling of the α_j's toward the group-level linear model. This is illustrated in Figure 12.6, which shows the estimates of all these 86 parameters—the 85 separate points and the slope of the line. In this model that includes a group-level predictor, the estimated intercepts are pulled toward this group-level regression line (rather than toward a constant, as in Figure 12.3b). The county-level uranium predictor u_j thus helps us estimate the county intercepts α_j but without overwhelming the information in individual counties.

Partial pooling of group coefficients α_j in the presence of group-level predictors

Equation (12.4) on page 258 gives the formula for partial pooling in the simple model with no group-level predictors. Once we add a group-level regression, $\alpha_j \sim \mathrm{N}(U_j\gamma, \sigma_\alpha^2)$, the parameters α_j are shrunk toward their regression estimates $\hat{\alpha}_j = U_j\gamma$. Equivalently, we can say that the group-level errors η_j (in the model $\alpha_j = U_j\gamma + \eta_j$) are shrunk toward 0. As always, there is more pooling when the group-level standard deviation σ_α is small, and more smoothing for groups with fewer observations. The multilevel estimate of α_j is a weighted average of the no-pooling estimate for its group $(\bar{y}_j - \overline{X}_j\beta)$ and the regression prediction $\hat{\alpha}_j$:

$$\begin{aligned}\text{estimate of } \alpha_j \quad \approx \quad & \frac{\frac{n_j}{\sigma_y^2}}{\frac{n_j}{\sigma_y^2} + \frac{1}{\sigma_\alpha^2}} \cdot (\text{estimate from group } j) + \\ & + \frac{\frac{1}{\sigma_\alpha^2}}{\frac{n_j}{\sigma_y^2} + \frac{1}{\sigma_\alpha^2}} \cdot (\text{estimate from regression}). \qquad (12.16)\end{aligned}$$

Equivalently, the group-level errors η_j are partially pooled toward zero:

$$\text{estimate of } \eta_j \approx \frac{\frac{n_j}{\sigma_y^2}}{\frac{n_j}{\sigma_y^2} + \frac{1}{\sigma_\alpha^2}}(\bar{y}_j - \overline{X}_j\beta - U_j\gamma) + \frac{\frac{1}{\sigma_\alpha^2}}{\frac{n_j}{\sigma_y^2} + \frac{1}{\sigma_\alpha^2}} \cdot 0.$$

12.7 Model building and statistical significance

From classical to multilevel regression

When confronted with a multilevel data structure, such as the radon measurements considered here or the examples in the previous chapter, we typically start by fitting some simple classical regressions and then work our way up to a full multilevel model. The four natural starting points are:

- Complete-pooling model: a single classical regression completely ignoring the group information—that is, a single model fit to all the data, perhaps including group-level predictors but with no coefficients for group indicators.
- No-pooling model: a single classical regression that includes group indicators (but no group-level predictors) but with no model for the group coefficients.
- Separate models: a separate classical regression in each group. This approach is not always possible if there are groups with small sample sizes. (For example, in Figure 12.4 on page 257, Aitkin County has three measurements in homes with basements and one in a home with no basement. If the sample from Aitkin County had happened to contain only houses with basements, then it would be impossible to estimate the slope β from this county alone.)
- Two-step analysis: starting with either the no-pooling or separate models, then fitting a classical group-level regression using, as "data," the estimated coefficients for each group.

Each of these simpler models can be informative in its own right, and they also set us up for understanding the partial pooling in a multilevel model, as in Figure 12.4.

For large datasets, fitting a model separately in each group can be computationally efficient as well. One might imagine an iterative procedure that starts by fitting separate models, continues with the two-step analysis, and then returns to fitting separate models, but using the resulting group-level regression to guide the estimates of the varying coefficients. Such a procedure, if formalized appropriately, is in fact the usual algorithm used to fit multilevel models, as we discuss in Chapter 17.

When is multilevel modeling most effective?

Multilevel model is most important when it is close to complete pooling, at least for some of the groups (as for Lac Qui Parle County in Figure 12.4 on page 257). In this setting we can allow estimates to vary by group while still estimating them precisely. As can be seen from formula (12.16), estimates are more pooled when the group-level standard deviation σ_α is small, that is, when the groups are similar to each other. In contrast, when σ_α is large, so that groups vary greatly, multilevel modeling is not much better than simple no-pooling estimation.

At this point, it might seem that we are contradicting ourselves. Earlier we motivated multilevel modeling as a compromise between no pooling and complete pooling, but now we are saying that multilevel modeling is effective when it is close to complete pooling, and ineffective when it is close to no pooling. If this is so, why not just always use the complete-pooling estimate?

We answer this question in two ways. First, when the multilevel estimate is close to complete pooling, it still allows variation between groups, which can be important, in fact can be one of the goals of the study. Second, as in the radon example, the multilevel estimate can be close to complete pooling for groups with small sample size and close to no pooling for groups with large sample size, automatically performing well for both sorts of group.

Using group-level predictors to make partial pooling more effective

In addition to being themselves of interest, group-level predictors play a special role in multilevel modeling by reducing the unexplained group-level variation and thus reducing the group-level standard deviation σ_α. This in turn increases the amount of pooling done by the multilevel estimate (see formula (12.16)), giving more precise estimates of the α_j's, especially for groups for which the sample size n_j is small. Following the template of classical regression, multilevel modeling typically proceeds by adding predictors at the individual and group levels and reducing the unexplained variance at each level. (However, as discussed in Section 21.7, adding a group-level predictor can actually increase the unexplained variance in some situations.)

Statistical significance

It is *not* appropriate to use statistical significance as a criterion for including particular group indicators in a multilevel model. For example, consider the simple varying-intercept radon model with no group-level predictor, in which the average intercept μ_α is estimated at 1.46, and the within-group intercepts α_j are estimated at $1.46 - 0.27 \pm 0.25$ for county 1, $1.46 - 0.53 \pm 0.10$ for county 2, $1.46 + 0.02 \pm 0.28$ for county 3, and so forth (see page 261).

County 1 is thus approximately 1 standard error away from the average intercept of 1.46, county 2 is more than 4 standard errors away, . . . and county 85 is less than 1 standard error away. Of these three counties, only county 2 would be considered "statistically significantly" different from the average.

However, we should include all 85 counties in the model, and nothing is lost by doing so. The purpose of the multilevel model is not to see whether the radon levels in county 1 are statistically significantly different from those in county 2, or from the Minnesota average. Rather, we seek the best possible estimate in each county, with appropriate accounting for uncertainty. Rather than make some significance threshold, we allow all the intercepts to vary and recognize that we may not have much precision in many of the individual groups. We illustrate this point in another example in Section 21.8.

The same principle holds for the models discussed in the following chapters, which include varying slopes, non-nested levels, discrete data, and other complexities. Once we have included a source of variation, we do not use statistical significance to pick and choose indicators to include or exclude from the model.

In practice, our biggest constraints—the main reasons we do not use extremely elaborate models in which all coefficients can vary with respect to all grouping factors—are fitting and understanding complex models. The `lmer()` function works well when it works, but it can break down for models with many grouping factors. Bugs is more general (see Part 2B of this book) but can be slow with large datasets or complex models. In the meantime we need to start simple and build up gradually, a process during which we can also build understanding of the models being fit.

12.8 Predictions for new observations and new groups

Predictions for multilevel models can be more complicated than for classical regression because we can apply the model to existing groups or new groups. After a brief review of classical regression prediction, we explain in the context of the radon model.

Review of prediction for classical regression

In classical regression, prediction is simple: specify the predictor matrix $\tilde{X}$ for a set of new observations[5] and then compute the linear predictor $\tilde{X}\beta$, then simulate the predictive data:

- For linear regression, simulate independent normal errors $\tilde{\epsilon}_i$ with mean 0 and standard deviation σ, and compute $\tilde{y} = \tilde{X}\beta + \tilde{\epsilon}$; see Section 7.2.
- For logistic regression, simulate the predictive binary data: $\Pr(\tilde{y}_i) = \text{logit}^{-1}(\tilde{X}_i\beta)$ for each new data point i; see Section 7.4.
- With binomial logistic regression, specify the number of tries $\tilde{n}_i$ for each new unit i, and simulate $\tilde{y}_i$ from the binomial distribution with parameters $\tilde{n}_i$ and $\text{logit}^{-1}(\tilde{X}_i\beta)$; see Section 7.4.
- With Poisson regression, specify the exposures $\tilde{u}_i$ for the new units, and simulate $\tilde{y}_i \sim \text{Poisson}(\tilde{u}_i e^{\tilde{X}_i\beta})$ for each new i; see Section 7.4.

As discussed in Section 7.2, the estimation for a regression in R gives a set of n_{sims} simulation draws. Each of these is used to simulate the predictive data vector $\tilde{y}$, yielding a set of n_{sims} simulated predictions. For example, in the election forecasting example of Figure 7.5 on page 146:

R code

```
model.1 <- lm (vote.88 ~ vote.86 + party.88 + inc.88)
display (model.1)
n.sims <- 1000
sim.1 <- sim (model.1, n.sims)
beta.sim <- sim.1$beta
sigma.sim <- sim.1$sigma
n.tilde <- length (vote.88)
X.tilde <- cbind (rep(1,n.tilde), vote.88, party.90, inc.90)
y.tilde <- array (NA, c(n.sims, n.tilde))
for (s in 1:n.sims) {
  y.tilde[s,] <- rnorm (n.tilde, X.tilde%*%beta.sim[s,], sigma.sim[s])
}
```

This matrix of simulations can be used to get point predictions (for example, `median(y.tilde[,3])` gives the median estimate for $\tilde{y}_3$) or predictive intervals (for example, `quantile(y.tilde[,3],c(.025,.975))`) for individual data points or for more elaborate derived quantities, such as the predicted number of seats won by the Democrats in 1990 (see the end of Section 7.3). For many applications, the `predict()` function in R is a good way to quickly get point predictions and intervals (see page 48); here we emphasize the more elaborate simulation approach which allows inferences for arbitrary quantities.

[5] Predictions are more complicated for time-series models: even when parameters are fit by classical regression, predictions must be made sequentially. See Sections 8.4 and 24.2 for examples.

Prediction for a new observation in an existing group

We can make two sorts of predictions for the radon example: predicting the radon level for a new house within one of the counties in the dataset, and for a new house in a new county. We shall work with model (12.15) on page 266, with floor as an individual-level predictor and uranium as a group-level predictor

For example, suppose we wish to predict $\tilde{y}$, the log radon level for a house with no basement (thus, with radon measured on the first floor, so that $\tilde{x} = 1$) in Hennepin County ($j = 26$ of our Minnesota dataset). Conditional on the model parameters, the predicted value has a mean of $\alpha_{26} + \beta$ and a standard deviation of σ_y. That is,

$$\tilde{y}|\theta \sim \mathrm{N}(\alpha_{26} + \beta\tilde{x}, \sigma_y^2),$$

where we are using θ to represent the entire vector of model parameters.

Given estimates of α, β, and σ_y, we can create a predictive simulation for $\tilde{y}$ using R code such as

R code

```
x.tilde <- 1
sigma.y.hat <- sigma.hat(M2)$sigma$data
coef.hat <- as.matrix(coef(M2)$county)[26,]
y.tilde <- rnorm (1, coef.hat %*% c(1, x.tilde, u[26]), sigma.y.hat)
```

More generally, we can create a vector of `n.sims` simulations to represent the predictive uncertainty in $\tilde{y}$:

R code

```
n.sims <- 1000
coef.hat <- as.matrix(coef(M2)$county)[26,]
y.tilde <- rnorm (1000, coef.hat %*% c(1, x.tilde, u[26]), sigma.y.hat)
```

Still more generally, we can add in the inferential uncertainty in the estimated parameters, α, β, and σ. For our purposes here, however, we shall ignore inferential uncertainty and just treat the parameters $\alpha, \beta, \sigma_y, \sigma_\alpha$ as if they were estimated perfectly from the data.[6] In that case, the computation gives us 1000 simulation draws of $\tilde{y}$, which we can summarize in various ways. For example,

R code

```
quantile (y.tilde, c(.25,.5,.75))
```

gives us a predictive median of 0.76 and a 50% predictive interval of $[0.26, 1.27]$. Exponentiating gives us a prediction on the original (unlogged) scale of $\exp(0.76) = 2.1$, with a 50% interval of $[1.3, 3.6]$.

For some applications we want the average, rather than the median, of the predictive distribution. For example, the expected risk from radon exposure is proportional to the predictive average or mean, which we can compute directly from the simulations:

R code

```
unlogged <- exp(y.tilde)
mean (unlogged)
```

In this example, the predictive mean is 2.9, which is a bit higher than the median of 2.1. This makes sense: on the unlogged scale, this predictive distribution is skewed to the right.

[6] One reason we picked Hennepin County ($j = 26$) for this example is that, with a sample size of 105, its average radon level is accurately estimated from the available data.

Prediction for a new observation in a new group

Now suppose we want to predict the radon level for a house, once again with no basement, but this time in a county not included in our analysis. We then must generate a new county-level error term, $\tilde{\alpha}$, which we sample from its $\mathrm{N}(\gamma_0+\gamma_1\tilde{u}_j, \sigma_\alpha^2)$ distribution. We shall assume the new county has a uranium level equal to the average of the uranium levels in the observed counties:

R code

```
u.tilde <- mean (u)
```

grab the estimated $\gamma_0, \gamma_1, \sigma_\alpha$ from the fitted model:

R code

```
g.0.hat <- fixef(M2)["(Intercept)"]
g.1.hat <- fixef(M2)["u.full"]
sigma.a.hat <- sigma.hat(M2)$sigma$county
```

and simulate possible intercepts for the new county:

R code

```
a.tilde <- rnorm (n.sims, g.0.hat + g.1.hat*u.tilde, sigma.a.hat)
```

We can then simulate possible values of the radon level for the new house in this county:

R code

```
y.tilde <- rnorm (n.sims, a.tilde + b.hat*x.tilde, sigma.y.hat)
```

Each simulation draw of $\tilde{y}$ uses a different simulation of $\tilde{\alpha}$, thus propagating the uncertainty about the new county into the uncertainty about the new house in this county.

Comparison of within-group and between-group predictions. The resulting prediction will be more uncertain than for a house in a known county, since we have no information about $\tilde{\alpha}$. Indeed, the predictive 50% interval of this new $\tilde{y}$ is $[0.28, 1.34]$, which is slightly wider than the predictive interval of $[0.26, 1.27]$ for the new house in county 26. The interval is only slightly wider because the within-county variation in this particular example is much higher than the between-county variation.

More specifically, from the fitted model on page 266, the within-county (residual) standard deviation σ_y is estimated at 0.76, and the between-county standard deviation σ_α is estimated at 0.16. The log radon level for a new house in an already-measured county can then be measured to an accuracy of about ±0.76. The log radon level for a new house in a new county can be predicted to an accuracy of about $\pm\sqrt{0.76^2+0.16^2} = \pm 0.78$. The ratio 0.78/0.76 is 1.03, so we would expect the predictive interval for a new house in a new county to be about 3% wider than for a new house in an already-measured county. The change in interval width is small here because the unexplained between-county variance is so small in this dataset.

For another example, the 50% interval for the log radon level of a house with no basement in county 2 is $[0.28, 1.30]$, which is centered in a different place but also is narrower than the predictive interval for a new county.

Nonlinear predictions

Section 7.3 illustrated the use of simulation for nonlinear predictions from classical regression. We can perform similar calculations in multilevel models. For example, suppose we are interested in the average radon level among all the houses in Hennepin County ($j = 26$). We can perform this inference using poststratification, first estimating the average radon level of the houses with and without basements in the county, then weighting these by the proportion of houses in the county that have basements. We can look up this proportion from other data sources on homes, or we can estimate it from the available sample data.

For our purposes here, we shall assume that 90% of all the houses in Hennepin County have basements. The average radon level of all the houses in the county is then 0.1 times the average for the houses in Hennepin County without basements, plus 0.9 times the average for those with basements. To simulate in R:

R code

```
y.tilde.basement <- rnorm (n.sims, a.hat[26], sigma.y.hat)
y.tilde.nobasement <- rnorm (n.sims, a.hat[26] + b.hat, sigma.y.hat)
```

We then compute the estimated mean for 1000 houses of each type in the county (first exponentiating since our model was on the log scale):

R code

```
mean.radon.basement <- mean (exp (y.tilde.basement))
mean.radon.nobasement <- mean (exp (y.tilde.nobasement))
```

and finally poststratify given the proportion of houses of each type in the county:

R code

```
mean.radon <- .9*mean.radon.basement + .1*mean.radon.basement
```

In Section 16.6 we return to the topic of predictions, using simulations from Bugs to capture the uncertainty in parameter estimates and then propagating inferential uncertainty into the predictions, rather than simply using point estimates `a.hat`, `b.hat`, and so forth.

12.9 How many groups and how many observations per group are needed to fit a multilevel model?

Advice is sometimes given that multilevel models can only be used if the number of groups is higher than some threshold, or if there is some minimum number of observations per groups. Such advice is misguided. Multilevel modeling includes classical regression as a limiting case (complete pooling when group-level variances are zero, no pooling when group-level variances are large). When sample sizes are small, the key concern with multilevel modeling is the estimation of variance parameters, but it should still work at least as well as classical regression.

How many groups?

When J, the number of groups, is small, it is difficult to estimate the between-group variation and, as a result, multilevel modeling often adds little in such situations, beyond classical no-pooling models. The difficulty of estimating variance parameters is a technical issue to which we return in Section 19.6; to simplify, when σ_α cannot be estimated well, it tends to be overestimated, and so the partially pooled estimates are close to no pooling (this is what happens when σ_α has a high value in (12.16) on page 269).

At the same time, multilevel modeling should not do any worse than no-pooling regression and sometimes can be easier to interpret, for example because one can include indicators for all J groups rather than have to select one group as a baseline category.

One or two groups

With only one or two groups, however, multilevel modeling reduces to classical regression (unless "prior information" is explicitly included in the model; see Section 18.3). Here we usually express the model in classical form (for example, including a single predictor for `female`, rather than a multilevel model for the two levels of the `sex` factor).

Even with only one or two groups in the data, however, multilevel models can be useful for making predictions about new groups. See also Sections 21.2–22.5 for further connections between classical and multilevel models, and Section 22.6 for hierarchical models for improving estimates of variance parameters in settings with many grouping factors but few levels per factor.

How many observations per group?

Even two observations per group is enough to fit a multilevel model. It is even acceptable to have one observation in many of the groups. When groups have few observations, their α_j's won't be estimated precisely, but they can still provide partial information that allows estimation of the coefficients and variance parameters of the individual- and group-level regressions.

Larger datasets and more complex models

As more data arise, it makes sense to add parameters to a model. For example, consider a simple medical study, then separate estimates for men and women, other demographic breakdowns, different regions of the country, states, smaller geographic areas, interactions between demographic and geographic categories, and so forth. As more data become available it makes sense to estimate more. These complexities are latent everywhere, but in small datasets it is not possible to learn so much, and it is not necessarily worth the effort to fit a complex model when the resulting uncertainties will be so large.

12.10 Bibliographic note

Multilevel models have been used for decades in agriculture (Henderson, 1950, 1984, Henderson et al., 1959, Robinson, 1991) and educational statistics (Novick et al., 1972, 1973, Bock, 1989), where it is natural to model animals in groups and students in classrooms. More recently, multilevel models have become popular in many social sciences and have been reviewed in books by Longford (1993), Goldstein (1995), Kreft and De Leeuw (1998), Snijders and Bosker (1999), Verbeke and Molenberghs (2000), Leyland and Goldstein (2001), Hox (2002), and Raudenbush and Bryk (2002). We do not attempt to trace here the many applications of multilevel models in various scientific fields.

It might also be useful to read up on Bayesian inference to understand the theoretical background behind multilevel models.[7] Box and Tiao (1973) is a classic reference that focuses on linear models. It predates modern computational methods but might be useful for understanding the fundamentals. Gelman et al. (2003) and Carlin and Louis (2000) cover applied Bayesian inference including the basics of multilevel modeling, with detailed discussions of computational algorithms. Berger

[7] As we discuss in Section 18.3, multilevel inferences can be formulated non-Bayesianly; however, understanding the Bayesian derivations should help with the other approaches too. All multilevel models are Bayesian in the sense of assigning probability distributions to the varying regression coefficients. The distinction between Bayesian and non-Bayesian multilevel models arises only for the question of modeling the other parameters—the nonvarying coefficients and the variance parameters—and this is typically a less important issue, especially when the number of groups is large.

(1985) and Bernardo and Smith (1994) cover Bayesian inference from two different theoretical perspectives.

The R function `lmer()` is described by Bates (2005a, b) and was developed from the linear and nonlinear mixed effects software described in Pinheiro and Bates (2000).

Multilevel modeling used to be controversial in statistics; see, for example, the discussions of the papers by Lindley and Smith (1972) and Rubin (1980) for some sense of the controversy.

The Minnesota radon data were analyzed by Price, Nero, and Gelman (1996); see also Price and Gelman (2004) for more on home radon modeling.

Statistical researchers have studied partial pooling in many ways; see James and Stein (1960), Efron and Morris (1979), DuMouchel and Harris (1983), Morris (1983), and Stigler (1983). Louis (1984), Shen and Louis (1998), Louis and Shen (1999), and Gelman and Price (1999) discuss some difficulties in the interpretation of partially pooled estimates. Zaslavsky (1993) discusses adjustments for undercount in the U.S. Census from a partial-pooling perspective. Normand, Glickman, and Gatsonis (1997) discuss the use of multilevel models for evaluating health-care providers.

12.11 Exercises

1. Using data of your own that are appropriate for a multilevel model, write the model in the five ways discussed in Section 12.5.
2. Continuing with the analysis of the CD4 data from Exercise 11.4:
   (a) Write a model predicting CD4 percentage as a function of time with varying intercepts across children. Fit using `lmer()` and interpret the coefficient for time.
   (b) Extend the model in (a) to include child-level predictors (that is, group-level predictors) for treatment and age at baseline. Fit using `lmer()` and interpret the coefficients on time, treatment, and age at baseline.
   (c) Investigate the change in partial pooling from (a) to (b) both graphically and numerically.
   (d) Compare results in (b) to those obtained in part (c).
3. Predictions for new observations and new groups:
   (a) Use the model fit from Exercise 12.2(b) to generate simulation of predicted CD4 percentages for each child in the dataset at a hypothetical next time point.
   (b) Use the same model fit to generate simulations of CD4 percentages at each of the time periods for a new child who was 4 years old at baseline.
4. Posterior predictive checking: continuing the previous exercise, use the fitted model from Exercise 12.2(b) to simulate a new dataset of CD4 percentages (with the same sample size and ages of the original dataset) for the final time point of the study, and record the average CD4 percentage in this sample. Repeat this process 1000 times and compare the simulated distribution to the observed CD4 percentage at the final time point for the actual data.
5. Using the radon data, include county sample size as a group-level predictor and write the varying-intercept model. Fit this model using `lmer()`.
6. Return to the beauty and teaching evaluations introduced in Exercise 3.5 and 4.8.

(a) Write a varying-intercept model for these data with no group-level predictors. Fit this model using `lmer()` and interpret the results.

(b) Write a varying-intercept model that you would like to fit including three group-level predictors. Fit this model using `lmer()` and interpret the results.

(c) How does the variation in average ratings across instructors compare to the variation in ratings across evaluators for the same instructor?

7. This exercise will use the data you found for Exercise 4.7. This time, rather than repeating the same analysis across each year, or country (or whatever group the data varies across), fit a multilevel model using `lmer()` instead. Compare the results to those obtained in your earlier analysis.

8. Simulate data (outcome, individual-level predictor, group indicator, and group-level predictor) that would be appropriate for a multilevel model. See how partial pooling changes as you vary the sample size in each group and the number of groups.

9. Number of observations and number of groups:

(a) Take a simple random sample of one-fifth of the radon data. (You can create this subset using the `sample()` function in R.) Fit the varying-intercept model with floor as an individual-level predictor and log uranium as a county-level predictor, and compare your inferences to what was obtained by fitting the model to the entire dataset. (Compare inferences for the individual- and group-level standard deviations, the slopes for floor and log uranium, the average intercept, and the county-level intercepts.)

(b) Repeat step (a) a few times, with a different random sample each time, and summarize how the estimates vary.

(c) Repeat step (a), but this time taking a cluster sample: a random sample of one-fifth of the counties, but then all the houses within each sampled county.

CHAPTER 13

Multilevel linear models: varying slopes, non-nested models, and other complexities

This chapter considers some generalizations of the basic multilevel regression. Models in which slopes and intercepts can vary by group (for example, $y_i = \alpha_{j[i]} + \beta_{j[i]}x_i + \cdots$, where α and β both vary by group j; see Figure 11.1c on page 238) can also be interpreted as interactions of the group index with individual-level predictors.

Another direction is non-nested models, in which a given dataset can be structured into groups in more than one way. For example, persons in a national survey can be divided by demographics or by states. Responses in a psychological experiment might be classified by person (experimental subject), experimental condition, and time.

The chapter concludes with some examples of models with nonexchangeable multivariate structures. We continue with generalized linear models in Chapters 14–15 and discuss how to fit all these models in Chapters 16–19.

13.1 Varying intercepts and slopes

The next step in multilevel modeling is to allow more than one regression coefficient to vary by group. We shall illustrate with the radon model from the previous chapter, which is relatively simple because it only has a single individual-level predictor, x (the indicator for whether the measurement was taken on the first floor).

We begin with a varying-intercept, varying-slope model including x but without the county-level uranium predictor; thus,

$$\begin{aligned} y_i &\sim \mathrm{N}(\alpha_{j[i]} + \beta_{j[i]}x_i,\, \sigma_y^2), \text{ for } i = 1, \ldots, n \\ \begin{pmatrix} \alpha_j \\ \beta_j \end{pmatrix} &\sim \mathrm{N}\left(\begin{pmatrix} \mu_\alpha \\ \mu_\beta \end{pmatrix}, \begin{pmatrix} \sigma_\alpha^2 & \rho\sigma_\alpha\sigma_\beta \\ \rho\sigma_\alpha\sigma_\beta & \sigma_\beta^2 \end{pmatrix} \right), \text{ for } j = 1, \ldots, J, \end{aligned} \tag{13.1}$$

with variation in the α_j's and the β_j's and also a between-group correlation parameter ρ. In R:

R code

```
M3 <- lmer (y ~ x + (1 + x | county))
display (M3)
```

which yields

R output

```
lmer(formula = y ~ x + (1 + x | county))
            coef.est coef.se
(Intercept)  1.46     0.05
x           -0.68     0.09
Error terms:
 Groups   Name        Std.Dev. Corr
 county   (Intercept) 0.35
          x           0.34     -0.34
```

```
  Residual            0.75
 # of obs: 919, groups: county, 85
 deviance = 2161.1
```

In this model, the unexplained within-county variation has an estimated standard deviation of $\hat{\sigma}_y = 0.75$; the estimated standard deviation of the county intercepts is $\hat{\sigma}_\alpha = 0.35$; the estimated standard deviation of the county slopes is $\hat{\sigma}_\beta = 0.34$; and the estimated correlation between intercepts and slopes is -0.34.

We then can type

R code

```
coef (M3)
```

to yield

R output

```
$county
   (Intercept)     x
1         1.14 -0.54
2         0.93 -0.77
3         1.47 -0.67
. . .
85        1.38 -0.65
```

Or we can separately look at the estimated population mean coefficients μ_α, μ_β and then the estimated errors for each county. First, we type

R code

```
fixef (M3)
```

to see the estimated average coefficients ("fixed effects"):

R output

```
(Intercept)          x
       1.46      -0.68
```

Then, we type

R code

```
ranef (M3)
```

to see the estimated group-level errors ("random effects"):

R output

```
   (Intercept)     x
1        -0.32  0.14
2        -0.53 -0.09
3         0.01  0.01
. . .
85       -0.08  0.03
```

We can regain the estimated intercept and slope α_j, β_j for each county by simply adding the errors to μ_α and μ_β; thus, the estimated regression line for county 1 is $(1.46 - 0.32) + (-0.68 + 0.14)x = 1.14 - 0.54x$, and so forth.

The group-level model for the parameters (α_j, β_j) allows for partial pooling in the estimated intercepts and slopes. Figure 13.1 shows the results—the estimated lines $y = \alpha_j + \beta_j x$—for the radon data in eight different counties.

Including group-level predictors

We can expand the model of (α, β) in (13.1) by including a group-level predictor (in this case, soil uranium):

$$\begin{pmatrix} \alpha_j \\ \beta_j \end{pmatrix} \sim \mathrm{N}\left(\begin{pmatrix} \gamma_0^\alpha + \gamma_1^\alpha u_j \\ \gamma_0^\beta + \gamma_1^\beta u_j \end{pmatrix}, \begin{pmatrix} \sigma_\alpha^2 & \rho\sigma_\alpha\sigma_\beta \\ \rho\sigma_\alpha\sigma_\beta & \sigma_\beta^2 \end{pmatrix} \right), \text{ for } j = 1, \ldots, J. \qquad (13.2)$$

The resulting estimates for the α_j's and β_j's are changed slightly from what is displayed in Figure 13.1, but more interesting are the second-level models themselves, whose estimates are shown in Figure 13.2. Here is the result of fitting the model in R:

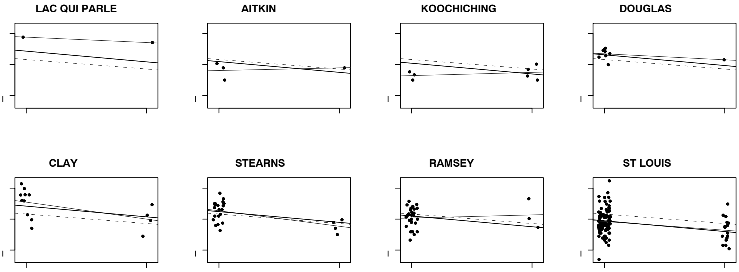

Figure 13.1 *Multilevel (partial pooling) regression lines $y = \alpha_j + \beta_j x$, displayed for eight counties j. In this model, both the intercept and the slope vary by county. The light solid and dashed lines show the no-pooling and complete pooling regression lines. Compare to Figure 12.4, in which only the intercept varies.*

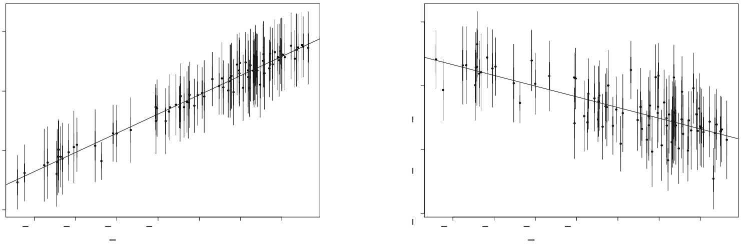

Figure 13.2 *(a) Estimates ± standard errors for the county intercepts α_j, plotted versus county-level uranium measurement u_j, along with the estimated multilevel regression line, $\alpha = \gamma_0^\alpha + \gamma_1^\alpha u$. (b) Estimates ± standard errors for the county slopes β_j, plotted versus county-level uranium measurement u_j, along with the estimated multilevel regression line, $\beta = \gamma_0^\beta + \gamma_1^\beta u$. Estimates and standard errors are the posterior medians and standard deviations, respectively. For each graph, the county coefficients roughly follow the line but not exactly; the discrepancies of the coefficients from the line are summarized by the county-level standard-deviation parameters $\sigma_\alpha, \sigma_\beta$.*

R output

```
lmer(formula = y ~ x + u.full + x:u.full + (1 + x | county))
            coef.est coef.se
(Intercept)  1.47     0.04
x           -0.67     0.08
u.full       0.81     0.09
x:u.full    -0.42     0.23
Error terms:
 Groups   Name        Std.Dev. Corr
 county   (Intercept) 0.12
          x           0.31     0.41
 Residual             0.75
# of obs: 919, groups: county, 85
deviance = 2114.3
```

The parameters $\gamma_0^\alpha, \gamma_0^\beta, \gamma_1^\alpha, \gamma_1^\beta$ in model (13.2) are the coefficients for the intercept, x, u.full, and x:u.full, respectively, in the regression. In particular, the interaction corresponds to allowing uranium to be a predictor in the regression for the slopes.

The estimated coefficients in each group (from coef(M4)) are:

R output

```
$county
   (Intercept)     x u.full x:u.full
1         1.46 -0.65   0.81    -0.42
2         1.50 -0.89   0.81    -0.42
. . .
85        1.44 -0.70   0.81    -0.42
```

Or we can display the average coefficients (using fixef(M4)):

R output

```
(Intercept)          x      u.full    x:u.full
       1.47      -0.67        0.81       -0.42
```

and the group-level errors for the intercepts and slopes (using ranef(M4)):

R output

```
   (Intercept)     x
1        -0.01  0.02
2         0.03 -0.21
. . .
85       -0.02 -0.03
```

The coefficients for the intercept and x vary, as specified in the model. This can be compared to the model on page 267 in which only the intercept varies.

Going from lmer output to intercepts and slopes

As before, we can combine the average coefficients with the group-level errors to compute the intercepts α_j and slopes β_j of model (13.2). For example, the fitted regression model in county 85 is $y_i = 1.47 - 0.67x_i + 0.81u_{85} - 0.42x_iu_{85} - 0.02 - 0.03x_i$. The log uranium level in county 85, u_{85}, is 0.36, and so the fitted regression line in county 85 is $y_i = 1.73 - 0.85x_i$. More generally, we can compute a vector of county intercepts α and slopes β:

R code

```
a.hat.M4 <- coef(M4)[,1] + coef(M4)[,3]*u
b.hat.M4 <- coef(M4)[,2] + coef(M4)[,4]*u
```

Here it is actually useful to have the variable u defined at the county level (as compared to u.full = u[county] which was used in the lmer() call). We next consider these linear transformations algebraically.

Varying slopes as interactions

Section 12.5 gave multiple ways of writing the basic multilevel model. These same ideas apply to models with varying slopes, which can be considered as interactions between group indicators and an individual-level predictor. For example, consider the model with an individual-level predictor x_i and a group-level predictor u_j,

$$\begin{aligned} y_i &= \alpha_{j[i]} + \beta_{j[i]}x_i + \epsilon_i \\ \alpha_j &= \gamma_0^\alpha + \gamma_1^\alpha u_j + \eta_j^\alpha \\ \beta_j &= \gamma_0^\beta + \gamma_1^\beta u_j + \eta_j^\beta. \end{aligned}$$

We can re-express this as a single model by substituting the formulas for α_j and β_j into the equation for y_i:

$$y_i = \left[\gamma_0^\alpha + \gamma_1^\alpha u_{j[i]} + \eta_{j[i]}^\alpha\right] + \left[\gamma_0^\beta + \gamma_1^\beta u_{j[i]} + \eta_{j[i]}^\beta\right] x_i + \epsilon_i. \qquad (13.3)$$

This expression looks messy but it is really just a regression including various interactions. If we define a new individual-level predictor $v_i = u_{j[i]}$ (in the radon example, this is the uranium level in the county where your house is located), we can re-express (13.3) term by term as

$$y_i = a + bv_i + c_{j[i]} + dx_i + ev_ix_i + f_{j[i]}x_i + \epsilon_i.$$

This can be thought of in several ways:

- A varying-intercept, varying-slope model with four individual-level predictors (the constant term, v_i, x_i, and the interaction v_ix_i) and varying intercepts and slopes that are centered at zero.
- A regression model with $4 + 2J$ predictors: the constant term, v_i, x_i, v_ix_i, indicators for the J groups, and interactions between x and the J group indicators.
- A regression model with four predictors and three error terms.
- Or, to go back to the original formulation, a varying-intercept, varying-slope model with one group-level predictor.

Which of these expressions is most useful depends on the context. In the radon analysis, where the goal is to predict radon levels in individual counties, the varying-intercept, varying-slope formulation, as pictured in Figure 13.2, seems most appropriate. But in a problem where interest lies in the regression coefficients for x_i, u_j, and their interaction, it can be more helpful to focus on these predictors and consider the unexplained variation in intercepts and slopes merely as error terms.

13.2 Varying slopes without varying intercepts

Figure 11.1 on page 238 displays a varying-intercept model, a varying-slope model, and a varying-intercept, varying-slope model. Almost always, when a slope is allowed to vary, it makes sense for the intercept to vary also. That is, the graph in the center of Figure 11.1b usually does not make sense. For example, if the coefficient of floor varies with county, then it makes sense to allow the intercept of the regression to vary also. It would be an implausible scenario in which the counties were all identical in radon levels for houses without basements, but differed in their coefficients for x.

A situation in which a constant-intercept, varying-slope model is appropriate

Occasionally it is reasonable to allow the slope but not the intercept to vary by group. For example, consider a study in which J separate experiments are performed on samples from a common population, with each experiment randomly assigning a control condition to half its subjects and a treatment to the other half. Further suppose that the "control" conditions are the same for each experiment but the "treatments" vary. In that case, it would make sense to fix the intercept and allow the slope to vary—thus, a basic model of:

$$\begin{aligned} y_i &\sim \mathrm{N}(\alpha + \theta_{j[i]}T_i,\, \sigma_y^2) \\ \theta_j &\sim \mathrm{N}(\mu_\theta, \sigma_\theta^2), \end{aligned} \tag{13.4}$$

where $T_i = 1$ for treated units and 0 for controls. Individual-level predictors could be added to the regression for y, and any interactions with treatment could also have varying slopes; for example,

$$\begin{array}{rcl} y_i & \sim & \mathrm{N}\left(\alpha + \beta x_i + \theta_{1,j[i]} T_i + \beta_{2,j[i]} x_i T_i,\ \sigma_y^2\right) \\ \begin{pmatrix} \theta_{1,j} \\ \theta_{2,j} \end{pmatrix} & \sim & \mathrm{N}\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}\right), \text{ for } j = 1, \ldots, J, \end{array} \qquad (13.5)$$

The multilevel model could be further extended with group-level predictors characterizing the treatments.

Fitting in R

To fit such a model in `lmer()`, we must explicitly remove the intercept from the group of coefficients that vary by group; for example, here is model (13.4) including the treatment indicator T as a predictor:

R code

```
lmer (y ~ T + (T - 1 | group))
```

The varying slope allows a different treatment effect for each group.

And here is model (13.5) with an individual-level predictor `x`:

R code

```
lmer (y ~ x + T + (T + x:T - 1 | group))
```

Here, the treatment effect and its interaction with x vary by group.

13.3 Modeling multiple varying coefficients using the scaled inverse-Wishart distribution

When more than two coefficients vary (for example, $y_i \sim \mathrm{N}(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2}, \sigma^2)$, with β_0, β_1, and β_2 varying by group), it is helpful to move to matrix notation in modeling the coefficients and their group-level regression model and covariance matrix.

Simple model with two varying coefficients and no group-level predictors

Starting with the model that begins this chapter, we can rewrite the basic varying-intercept, varying-slope model (13.1) in matrix notation as

$$\begin{array}{rcl} y_i & \sim & \mathrm{N}(X_i B_{j[i]}, \sigma_y^2), \text{ for } i = 1, \ldots, n \\ B_j & \sim & \mathrm{N}(M_B, \Sigma_B), \text{ for } j = 1, \ldots, J, \end{array} \qquad (13.6)$$

where

- X is the $n \times 2$ matrix of predictors: the first column of X is a column of 1's (that is, the constant term in the regression), and the second column is the predictor x. X_i is then the vector of length 2 representing the i^{th} row of X, and $X_i B_{j[i]}$ is simply $\alpha_{j[i]} + \beta_{j[i]} x_i$ from the top line of (13.1).
- $B = (\alpha, \beta)$ is the $J \times 2$ matrix of individual-level regression coefficients. For any group j, B_j is a vector of length 2 corresponding to the j^{th} row of B (although for convenience we consider B_j as a column vector in the product $X_i B_{j[i]}$ in model (13.6)). The two elements of B_j correspond to the intercept and slope, respectively, for the regression model in group j. $B_{j[i]}$ in the first line of (13.6) is the $j[i]^{th}$ row of B, that is, the vector representing the intercept and slope for the group that includes unit i.
- $M_B = (\mu_\alpha, \mu_\beta)$ is a vector of length 2, representing the mean of the distribution of the intercepts and the mean of the distribution of the slopes.

- Σ_B is the 2×2 covariance matrix representing the variation of the intercepts and slopes in the population of groups, as in the second line of (13.1).

We are following our general notation in which uppercase letters represent matrices: thus, the vectors α and β are combined into the matrix B.

In the fitted radon model on page 279, the parameters of the group-level model are estimated at $\widehat{M_B} = (1.46, -0.68)$ and $\widehat{\Sigma}_B = \begin{pmatrix} \hat{\sigma}_a^2 & \hat{\rho}\hat{\sigma}_a\hat{\sigma}_b \\ \hat{\rho}\hat{\sigma}_a\hat{\sigma}_b & \hat{\sigma}_b^2 \end{pmatrix}$, where $\hat{\sigma_a} = 0.35$, $\hat{\sigma}_b = 0.34$, and $\hat{\rho} = -0.34$. The estimated coefficient matrix $\widehat{B}$ is given by the 85×2 array at the end of the display of `coef(M3)` on page 280.

More than two varying coefficients

The same expression as above holds, except that the 2's are replaced by K's, where K is the number of individual-level predictors (including the intercept) that vary by group. As we discuss shortly in the context of the inverse-Wishart model, estimation becomes more difficult when $K > 2$ because of constraints among the correlation parameters of the covariance matrix Σ_B.

Including group-level predictors

More generally, we can have J groups, K individual-level predictors, and L predictors in the group-level regression (including the constant term as a predictor in both cases). For example, $K = L = 2$ in the radon model that has floor as an individual predictor and uranium as a county-level predictor.

We can extend model (13.6) to include group-level predictors:

$$\begin{array}{rcl} y_i & \sim & \mathrm{N}(X_i B_{j[i]},\, \sigma_y^2), \text{ for } i = 1, \ldots, n \\ B_j & \sim & \mathrm{N}(U_j G, \Sigma_B), \text{ for } j = 1, \ldots, J, \end{array} \qquad (13.7)$$

where B is the $J \times K$ matrix of individual-level coefficients, U is the $J \times L$ matrix of group-level predictors (including the constant term), and G is the $L \times K$ matrix of coefficients for the group-level regression. U_j is the j^{th} row of U, the vector of predictors for group j, and so $U_j G$ is a vector of length K.

Model (13.1) is a special case with $K = L = 2$, and the coefficients in G are then $\gamma_0^\alpha, \gamma_0^\beta, \gamma_1^\alpha, \gamma_1^\beta$. For the fitted radon model on page 279, the γ's are the four unmodeled coefficients (for the intercept, `x`, `u.full`, and `x:u.full`, respectively), and the two columns of the estimated coefficient matrix $\widehat{B}$ are estimated by `a.hat` and `b.hat`, as defined by the R code on page 282.

Including individual-level predictors whose coefficients do not vary by group

The model can be further expanded by adding unmodeled individual-level coefficients, so that the top line of (13.7) becomes

$$y_i \sim \mathrm{N}(X_i^0 \beta^0 + X_i B_{j[i]},\, \sigma_y^2), \text{ for } i = 1, \ldots, n, \qquad (13.8)$$

where X^0 is a matrix of these additional predictors and β^0 is the vector of their regression coefficients (which, by assumption, are common to all the groups).

Model (13.8) is sometimes called a *mixed-effects* regression, where the β^0's and the B's are the *fixed* and *random* effects, respectively. As noted on pages 2 and 245, we avoid these terms because of their ambiguity in the statistical literature. For example, sometimes unvarying coefficients such as the β^0's in model (13.8) are called "fixed," but sometimes the term "fixed effects" refers to intercepts that vary by groups but are not given a multilevel model (this is what we call the "no-pooling model," as pictured, for example, by the solid lines in Figure 12.2 on page 255).

Equivalently, model (13.8) can be written by folding X^0 and X into a common predictor matrix X, folding β^0 and B into a common coefficient matrix B, and using model (13.1), with the appropriate elements in Σ_B set to zero, implying no variation among groups for certain coefficients.

Modeling the group-level covariance matrix using the scaled inverse-Wishart distribution

When the number K of varying coefficients per group is more than two, modeling the correlation parameters ρ is a challenge. In addition to each of the correlations being restricted to fall between -1 and 1, the correlations are jointly constrained in a complicated way—technically, the covariance matrix Σ_β must be positive definite. (An example of the constraint is: if $\rho_{12} = 0.9$ and $\rho_{13} = 0.9$, then ρ_{23} must be at least 0.62.)

Modeling and estimation are more complicated in this jointly constrained space. We first introduce the inverse-Wishart model, then generalize to the scaled inverse-Wishart, which is what we recommend for modeling the covariance matrix of the distribution of varying coefficients.

Inverse-Wishart model. One model that has been proposed for the covariance matrix Σ_β is the *inverse-Wishart* distribution, which has the advantage of being computationally convenient (especially when using Bugs, as we illustrate in Section 17.1) but the disadvantage of being difficult to interpret.

In the model $\Sigma_B \sim \text{Inv-Wishart}_{K+1}(I)$, the two parameters of the inverse-Wishart distribution are the *degrees of freedom* (here set to $K+1$, where K is the dimension of B, that is, the number of coefficients in the model that vary by group) and the *scale* (here set to the $K \times K$ identity matrix).

To understand this model, we consider its implications for the standard deviation and correlations. Recall that if there are K varying coefficients, then Σ_B is a $K \times K$ matrix, with diagonal elements $\Sigma_{kk} = \sigma_k^2$ and off-diagonal-elements $\Sigma_{kl} = \rho_{kl}\sigma_k\sigma_l$ (generalizing models (13.1) and (13.2) to $K > 2$).

Setting the degrees-of-freedom parameter to $K+1$ has the effect of setting a uniform distribution on the individual correlation parameters (that is, they are assumed equally likely to take on any value between -1 and 1).

Scaled inverse-Wishart model. When the degrees of freedom parameter of the inverse-Wishart distribution is set to K+1, the resulting model is reasonable for the correlations but is quite constraining on the scale parameters σ_k. This is a problem because we would like to estimate σ_k from the data. Changing the degrees of freedom allows the σ_k's to be estimated more freely, but at the cost of constraining the correlation parameters.

We get around this problem by expanding the inverse-Wishart model with a new vector of scale parameters ξ_k:

$$\Sigma_B = \text{Diag}(\xi) Q \text{Diag}(\xi),$$

with the *unscaled covariance matrix* Q being given the inverse-Wishart model:

$$Q \sim \text{Inv-Wishart}_{K+1}(I).$$

The variances then correspond to the diagonal elements of the unscaled covariance

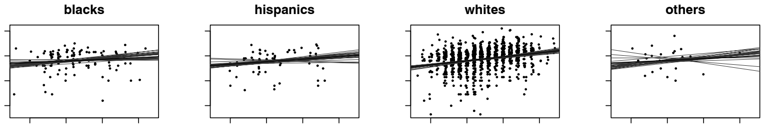

Figure 13.3 *Multilevel regression lines $y = \alpha_j + \beta_j x$ for log earnings on height (among those with positive earnings), in four ethnic categories j. The gray lines indicate uncertainty in the fitted regressions.*

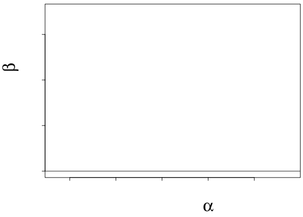

Figure 13.4 *Scatterplot of estimated intercepts and slopes (for whites, hispanics, blacks, and others), (α_j, β_j), for the earnings-height regressions shown in Figure 13.3. The extreme negative correlation arises because the center of the range of height is far from zero. Compare to the coefficients in the rescaled model, as displayed in Figure 13.7.*

matrix Q, multiplied by the appropriate scaling factors ξ:

$$\sigma_k^2 = \Sigma_{kk} = \xi_k^2 Q_{kk}, \text{ for } k = 1, \dots, K,$$

and the covariances are

$$\Sigma_{kl} = \xi_k \xi_l Q_{kl}, \text{ for } k, l = 1, \dots, K,$$

We prefer to express in terms of the standard deviations,

$$\sigma_k = |\xi_k| \sqrt{Q_{kk}},$$

and correlations

$$\rho_{kl} = \Sigma_{kl} / (\sigma_k \sigma_l).$$

The parameters in ξ and Q cannot be interpreted separately: they are a convenient way to set up the model, but it is the standard deviations σ_k and the correlations ρ_{kl} that are of interest (and which are relevant for producing partially pooled estimates for the coefficients in B).

As with the unscaled Wishart, the model implies a uniform distribution on the correlation parameters. As we discuss next, it can make sense to transform the data to remove any large correlations that could be expected simply from the structure of the data.

13.4 Understanding correlations between group-level intercepts and slopes

Recall that varying slopes can be interpreted as interactions between an individual-level predictor and group indicators. As with classical regression models with interactions, the intercepts can often be more clearly interpreted if the continuous

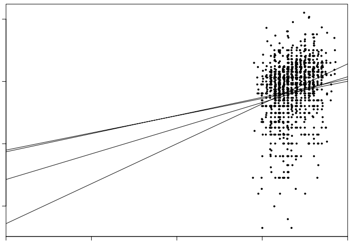

Figure 13.5 *Sketch illustrating the difficulty of simultaneously estimating α and β. The lines show the regressions for the four ethnic groups as displayed in Figure 13.3: the center of the range of x values is far from zero, and so small changes in the slope induce large changes in the intercept.*

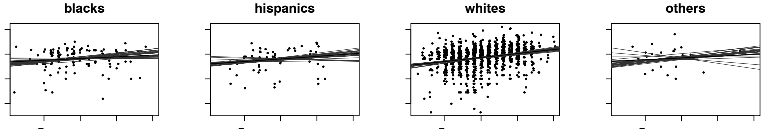

Figure 13.6 *Multilevel regression lines $y = \alpha_j + \beta_j z$, for log earnings given mean-adjusted height ($z_i = x_i - \bar{x}$), in four ethnic groups j. The gray lines indicate uncertainty in the fitted regressions.*

predictor is appropriately centered. We illustrate with the height and earnings example from Chapter 4.

We begin by fitting a multilevel model of log earnings given height, allowing the coefficients to vary by ethnicity. The data and fitted model are displayed in Figure 13.3. (Little is gained by fitting a multilevel model here—with only four groups, a classical no-pooling model would work nearly as well, as discussed in Section 12.9—but this is a convenient example to illustrate a general point.)

Figure 13.4 displays the estimates of (α_j, β_j) for the four ethnic groups, and they have a strong negative correlation: the groups with high values of α have relatively low values of β, and vice versa. This correlation occurs because the center of the x-values of the data is far from zero. The regression lines have to go roughly through the center of the data, and then changes in the slope induce opposite changes in the intercept, as illustrated in Figure 13.5.

There is nothing wrong with a high correlation between the α's and β's, but it makes the estimated intercepts more difficult to interpret. As with interaction models in classical regression, it can be helpful to subtract the average value of the continuous x before including it in the regression; thus, $y_i \sim \mathrm{N}(\alpha_{j[i]} + \beta_{j[i]} z_i, \sigma_y^2)$, where $z_i = x_i - \bar{x}$. Figures 13.6 and 13.7 show the results for the earnings regression: the correlation has pretty much disappeared. Centering the predictor x will not necessarily remove correlations between intercepts and slopes—but any correlation that remains can then be more easily interpreted. In addition, centering can speed convergence of the Gibbs sampling algorithm used by Bugs and other software.

We fit this model, and the subsequent models in this chapter, in Bugs (see Chap-

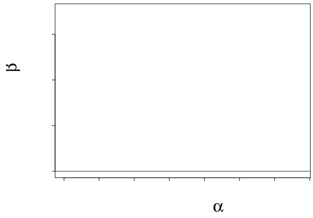

Figure 13.7 *Scatterplot of estimated intercepts and slopes, (α_j, β_j), for the regression of earnings on mean-adjusted height z, for the four groups j displayed in Figure 13.6. The coefficients are no longer strongly correlated (compare to Figure 13.4).*

ter 17 for examples of code) because, as discussed in Section 12.4, the current version of `lmer()` does not work so well when the number of groups is small—and, conversely, with these small datasets, Bugs is not too slow.

13.5 Non-nested models

So far we have considered the simplest hierarchical structure of individuals i in groups j. We now discuss models for more complicated grouping structures such as introduced in Section 11.3.

Example: a psychological experiment with two potentially interacting factors

Figure 13.8 displays data from a psychological experiment of pilots on flight simulators, with $n = 40$ data points corresponding to $J = 5$ treatment conditions and $K = 8$ different airports. The responses can be fit to a *non-nested* multilevel model of the form

$$\begin{aligned} y_i &\sim \mathrm{N}(\mu + \gamma_{j[i]} + \delta_{k[i]}, \sigma_y^2), \text{ for } i = 1, \dots, n \\ \gamma_j &\sim \mathrm{N}(0, \sigma_\gamma^2), \text{ for } j = 1, \dots, J \\ \delta_k &\sim \mathrm{N}(0, \sigma_\delta^2), \text{ for } k = 1, \dots, K. \end{aligned} \tag{13.9}$$

The parameters γ_j and δ_k represent treatment effects and airport effects. Their distributions are centered at zero (rather than given mean levels μ_γ, μ_δ) because the regression model for y already has an intercept, μ, and any nonzero mean for the γ and δ distributions could be folded into μ. As we shall see in Section 19.4, it can sometimes be effective for computational purposes to add extra mean-level parameters into the model, but the coefficients in this expanded model must be interpreted with care.

We can perform a quick fit as follows:

```
lmer (y ~ 1 + (1 | group.id) + (1 | scenario.id))
```

R code

where `group.id` and `scenario.id` are the index variables for the five treatment conditions and eight airports, respectively.

When fit to the data in Figure 13.8, the estimated residual standard deviations at the individual, treatment, and airport levels are $\hat{\sigma}_y = 0.23$, $\hat{\sigma}_\gamma = 0.04$, and $\hat{\sigma}_\delta = 0.32$. Thus, the variation among airports is huge—even larger than that among individual measurements—but the treatments vary almost not at all. This general pattern can be seen in Figure 13.8.

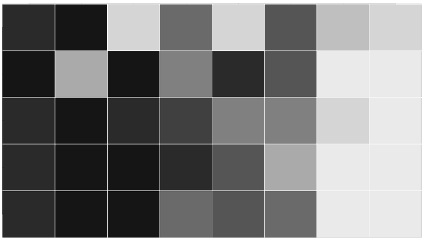

Figure 13.8 *Success rates of pilots training on a flight simulator with five different treatments and eight different airports. Shadings in the 40 cells i represent different success rates y_i, with black and white corresponding to 0 and 100%, respectively. For convenience in reading the display, the treatments and airports have each been sorted in increasing order of average success. These 40 data points have two groupings—treatments and airports—which are not nested.*

Data in matrix form

airport	treatment conditions				
1	0.38	0.25	0.50	0.14	0.43
2	0.00	0.00	0.67	0.00	0.00
3	0.38	0.50	0.33	0.71	0.29
4	0.00	0.12	0.00	0.00	0.86
5	0.33	0.50	0.14	0.29	0.86
6	1.00	1.00	1.00	1.00	0.86
7	0.12	0.12	0.00	0.14	0.14
8	1.00	0.86	1.00	1.00	0.75

Data in vector form

y	j	k
0.38	1	1
0.00	1	2
0.38	1	3
0.00	1	4
0.33	1	5
1.00	1	6
0.12	1	7
1.00	1	8
0.25	2	1
...	...	...

Figure 13.9 *Data from Figure 13.8 displayed as an array (y_{jk}) and in our preferred notation as a vector (y_i) with group indicators $j[i]$ and $k[i]$.*

Model (13.9) can also be written more cleanly as $y_{jk} \sim \mathrm{N}(\mu + \gamma_j + \delta_k, \sigma_y^2)$, but we actually prefer the more awkward notation using $j[i]$ and $k[i]$ because it emphasizes the multilevel structure of the model and is not restricted to balanced designs. When modeling a data array of the form (y_{jk}), we usually convert it into a vector with index variables for the rows and columns, as illustrated in Figure 13.9 for the flight simulator data.

Example: regression of earnings on ethnicity categories, age categories, and height

All the ideas of the earlier part of this chapter, introduced in the context of a simple structure of individuals within groups, apply to non-nested models as well. For example, Figure 13.10 displays the estimated regression of log earnings, y_i, on height, z_i (mean-adjusted, for reasons discussed in the context of Figures 13.3–13.6), applied to the $J = 4$ ethnic groups and $K = 3$ age categories. In essence, there is a separate regression model for each age group and ethnicity combination. The multilevel model can be written, somewhat awkwardly, as a data-level model,

$$y_i \sim \mathrm{N}(\alpha_{j[i],k[i]} + \beta_{j[i],k[i]} z_i,\ \sigma_y^2), \text{ for } i = 1, \dots, n,$$

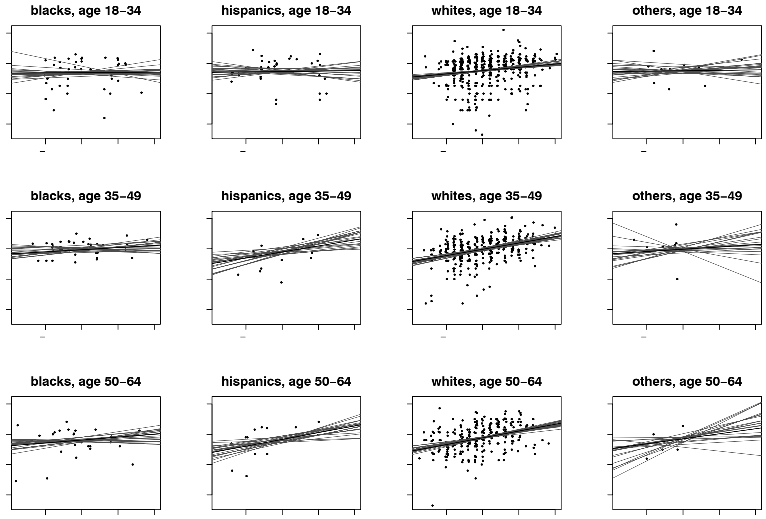

Figure 13.10 *Multilevel regression lines* $y = \beta^0_{j,k} + \beta^1_{j,k} z$, *for log earnings* y *given mean-adjusted height* z, *for four ethnic groups* j *and three age categories* k. *The gray lines indicate uncertainty in the fitted regressions.*

a decomposition of the intercepts and slopes into terms for ethnicity, age, and ethnicity × age,

$$\begin{pmatrix} \alpha_{j,k} \\ \beta_{j,k} \end{pmatrix} = \begin{pmatrix} \mu_0 \\ \mu_1 \end{pmatrix} + \begin{pmatrix} \gamma^{\rm eth}_{0j} \\ \gamma^{\rm eth}_{1j} \end{pmatrix} + \begin{pmatrix} \gamma^{\rm age}_{0k} \\ \gamma^{\rm age}_{1k} \end{pmatrix} + \begin{pmatrix} \gamma^{\rm eth\times age}_{0jk} \\ \gamma^{\rm eth\times age}_{1jk} \end{pmatrix},$$

and models for variation,

$$\begin{aligned}
\begin{pmatrix} \gamma^{\rm eth}_{0j} \\ \gamma^{\rm eth}_{1j} \end{pmatrix} &\sim \mathrm{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \Sigma^{\rm eth}\right), \text{ for } j = 1, \dots, J \\
\begin{pmatrix} \gamma^{\rm age}_{0k} \\ \gamma^{\rm age}_{1k} \end{pmatrix} &\sim \mathrm{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \Sigma^{\rm age}\right), \text{ for } k = 1, \dots, K \\
\begin{pmatrix} \gamma^{\rm eth\times age}_{0jk} \\ \gamma^{\rm eth\times age}_{1jk} \end{pmatrix} &\sim \mathrm{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \Sigma^{\rm eth\times age}\right), \text{ for } j = 1, \dots, J;\ k = 1, \dots, K.
\end{aligned}$$

Because we have included means μ_0, μ_1 in the decomposition above, we can center each batch of coefficients at 0.

Interpretation of data-level variance. The data-level errors have estimated residual standard deviation $\hat{\sigma}_y = 0.87$. That is, given ethnicity, age group, and height, log earnings can be predicted to within approximately ±0.87, and so earnings themselves can be predicted to within a multiplicative factor of $e^{0.87} = 2.4$. So earnings cannot be predicted well at all by these factors, which is also apparent from the scatter in Figure 13.10.

Interpretation of group-level variances. The group-level errors can be separated into intercept and slope coefficients. The intercepts have estimated residual stan-

B: 257	E: 230	A: 279	C: 287	D: 202
D: 245	A: 283	E: 245	B: 280	C: 260
E: 182	B: 252	C: 280	D: 246	A: 250
A: 203	C: 204	D: 227	E: 193	B: 259
C: 231	D: 271	B: 266	A: 334	E: 338

Figure 13.11 *Data from a 5×5 latin square experiment studying the effects of five ordered treatments on the yields of millet crops, from Snedecor and Cochran (1989). Each cell shows the randomly assigned treatment and the observed yield for the plot.*

dard deviations of $(\widehat{\Sigma}^{\rm eth}_{00})^{1/2} = 0.08$ at the ethnicity level, $(\widehat{\Sigma}^{\rm age}_{00})^{1/2} = 0.25$ at the age level, and $(\widehat{\Sigma}^{\rm eth\times age}_{00})^{1/2} = 0.11$ at the ethnicity × age level. Because we have rescaled height to have a mean of zero (see Figure 13.10), we can interpret these standard deviations as the relative importance of each factor (ethnicity, age group, and their interaction) on log earnings at the average height in the population.

This model fits earnings on the log scale and so these standard deviations can be interpreted accordingly. For example, the residual standard deviation of 0.08 for the ethnicity coefficients implies that the predictive effects of ethnic groups in the model are on the order of ± 0.08, which correspond to multiplicative factors from about $e^{-0.08} = 0.92$ to $e^{0.08} = 1.08$.

The slopes have estimated residual standard deviations of $(\widehat{\Sigma}^{\rm eth}_{11})^{1/2} = 0.03$ at the ethnicity level, $(\widehat{\Sigma}^{\rm age}_{11})^{1/2} = 0.02$ at the age level, and $(\widehat{\Sigma}^{\rm eth\times age}_{11})^{1/2} = 0.02$ at the ethnicity × age level. These slopes are per inch of height, so, for example, the predictive effects of ethnic groups in the model are in the range of $\pm 3\%$ in income per inch of height. One can also look at the estimated correlation between intercepts and slopes for each factor.

Example: a latin square design with grouping factors and group-level predictors

Non-nested models can also include group-level predictors. We illustrate with data from a 5×5 latin square experiment, a design in which 25 units arranged in a square grid are assigned five different treatments, with each treatment being assigned to one unit in each row and each column. Figure 13.11 shows the treatment assignments and data from a small agricultural experiment. There are three non-nested levels of grouping—rows, columns, and treatments—and each has a natural group-level predictor corresponding to a linear trend. (The five treatments are ordered.)

The corresponding multilevel model can be written as

$$\begin{aligned}
y_i &\sim \mathrm{N}(\mu + \beta^{\rm row}_{j[i]} + \beta^{\rm column}_{k[i]} + \beta^{\rm treat}_{l[i]},\, \sigma^2_y), \text{ for } i = 1, \ldots, 25\\
\beta^{\rm row}_j &\sim \mathrm{N}(\gamma^{\rm row}\cdot(j-3), \sigma^2_{\beta\,\rm row}), \text{ for } j = 1, \ldots, 5\\
\beta^{\rm column}_k &\sim \mathrm{N}(\gamma^{\rm column}\cdot(k-3), \sigma^2_{\beta\,\rm column}), \text{ for } k = 1, \ldots, 5\\
\beta^{\rm treat}_l &\sim \mathrm{N}(\gamma^{\rm treat}\cdot(l-3), \sigma^2_{\beta\,\rm treat}), \text{ for } l = 1, \ldots, 5.
\end{aligned} \qquad (13.10)$$

Thus j, k, and l serve simultaneously as values of the row, column, and treatment predictors.

By subtracting 3, we have centered the row, column, and treatment predictors at zero; the parameter μ has a clear interpretation as the grand mean of the data, with the different β's supplying deviations for rows, columns, and treatments. As with group-level models in general, the linear trends at each level potentially allow more precise estimates of the group effects, to the extent that these trends are supported by the data. An advantage of multilevel modeling here is that it doesn't force a

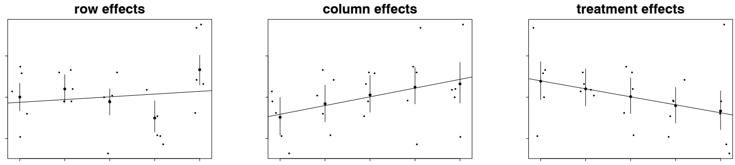

Figure 13.12 *Estimates ±1 standard error for the row, column, and treatment effects for the latin square data in Figure 13.11. The five levels of each factor are ordered, and the lines display the estimated group-level regressions,* $y=\mu+\gamma^{\text{row}}\cdot(x-3)$, $y=\mu+\gamma^{\text{column}}\cdot(x-3)$, *and* $y=\mu+\gamma^{\text{treat}}\cdot(x-3)$.

choice between a linear fit and separate estimates for each level of a predictor. (This is an issue we discussed more generally in Chapter 11 in the context of including group indicators as well as group-level predictors.)

Figure 13.12 shows the estimated row, column, and treatment effects on graphs, along with the estimated linear trends. The grand mean μ has been added back to each of these observations so that the plots are on the scale of the original data. This sort of data structure is commonly studied using the analysis of variance, whose connections with multilevel models we discuss fully in Chapter 22, including a discussion of this latin square example in Section 22.5.

13.6 Selecting, transforming, and combining regression inputs

As with classical regression (see Section 4.5), choices must be made in multilevel models about which input variables to include, and how best to transform and combine them. We discuss here how some of these decisions can be expressed as particular choices of parameters in a multilevel model. The topic of formalizing modeling choices is currently an active area of research—key concerns include using information in potential input variables without being overwhelmed by the complexity of the relating model, and including model choice in uncertainty estimates. As discussed in Section 9.5, the assumption of ignorability in observational studies is more plausible when controlling for more pre-treatment inputs, which gives us a motivation to include more regression predictors.

Classical models for regression coefficients

Multilevel modeling includes classical least squares regression as a special case. In a multilevel model, each coefficient is part of a model with some mean and standard deviation. (These mean values can themselves be determined by group-level predictors in a group-level model.) In classical regression, every predictor is either in or out of the model, and each of these options corresponds to a special case of the multilevel model.

- If a predictor is "in," this corresponds to a coefficient model with standard deviation of ∞: no group-level information is used to estimate this parameter, so it is estimated directly using least squares. It turns out that in this case the group-level mean is irrelevant (see formula (12.16) on page 269 for the case $\sigma_\alpha = \infty$); for convenience we often set it to 0.
- If a predictor is "out," this corresponds to a group-level model with group-level

mean 0 and standard deviation 0: the coefficient estimate is then fixed at zero (see (12.16) for the case $\sigma_\alpha = 0$) with no uncertainty.

Multilevel modeling as an alternative to selecting regression predictors

Multilevel models can be used to combine inputs into more effective regression predictors, generalizing some of the transformation ideas discussed in Section 4.6. When many potential regression inputs are available, the fundamental approach is to include as many of these inputs as possible, but not necessarily as independent least squares predictors.

For example, Witte et al. (1994) describe a logistic regression in a case-control study of 362 persons, predicting cancer incidence given information on consumption of 87 different foods (and also controlling for five background variables which we do not discuss further here). Each of the foods can potentially increase or decrease the probability of cancer, but it would be hard to trust the result of a regression with 87 predictors fit to only 362 data points, and classical tools for selecting regression predictors do not seem so helpful here. In our general notation, the challenge is to estimate the logistic regression of cancer status y on the 362×87 matrix X of food consumption (and the 362×6 matrix X^0 containing the constant term and the 5 background variables).

More information is available, however, because each of the 87 foods can be characterized by its level of each of 35 nutrients, information that can be expressed as an 87×36 matrix of predictors Z indicating how much of each nutrient is in each food. Witte et al. fit the following multilevel model:

$$\begin{aligned} \Pr(y_i = 1) &= \operatorname{logit}^{-1}(X_i^0\beta^0 + X_i B_{j[i]}), \text{ for } i = 1, \dots, 362 \\ B_j &\sim \mathrm{N}(Z_j\gamma, \sigma_\beta^2), \text{ for } j = 1, \dots, 87. \end{aligned} \tag{13.11}$$

The food-nutrient information in Z allows the multilevel model to estimate separate predictive effects for foods, after controlling for systematic patterns associated with nutrients. In the extreme case that $\sigma_\beta = 0$, all the variation associated with the foods is explained by the nutrients. At the other extreme, $\sigma_\beta = \infty$ would imply that the nutrient information is not helping at all.

Model (13.11) is helpful in reducing the number of food predictors from 87 to 35. At this point, Witte et al. used substantive understanding of diet and cancer to understand the result. Ultimately, we would like to have a model that structures the 35 predictors even more, perhaps by categorizing them into batches or combining them in some way. The next example sketches how this might be done; it is currently an active research topic to generally structure large numbers of regression predictors.

Linear transformation and combination of inputs in a multilevel model

For another example, we consider the problem of forecasting presidential elections by state (see Section 1.2). A forecasting model based on 11 recent national elections has more than 500 "data points"—state-level elections—and can then potentially include many state-level predictors measuring factors such as economic performance, incumbency, and popularity. However, at the national level there are really only 11 observations and so one must be parsimonious with national-level predictors. In practice, this means performing some preliminary data analysis to pick a single economic predictor, a single popularity predictor, and maybe one or two other predictors based on incumbency and political ideology.

Setting up a model to allow partial pooling of a set of regression predictors

A more general approach to including national predictors is possible using multilevel modeling. For example, suppose we wish to include five measures of the national economy (for example, change in GDP per capita, change in unemployment, and so forth). The usual approach (which we have followed in the past in this problem) is to choose one of these as the economic predictor, x, thus writing the model as

$$y_i = \alpha + \beta x_i + \cdots , \tag{13.12}$$

where the dots indicate all the rest of the model, including other state-level and national predictors, as well as error terms at the state, regional, and national levels. Here we focus on the economic inputs, for simplicity setting aside the rest of the model.

Instead of choosing just one of the five economic inputs, it would perhaps be better first to standardize each of them (see Section 4.2), orient them so they are in the same direction, label these standardized variables as $X_{(j)}$, for $j = 1, \dots, 5$, and then average them into a single predictor, defined for each data point as

$$x_i^{\text{avg}} = \frac{1}{5}\sum_{j=1}^{5}\sum X_{ij}, \text{ for } i = 1, \dots. n. \tag{13.13}$$

This new x^{avg} can be included in place of x as the regression predictor in (13.12), or, equivalently,

$$\begin{aligned} y_i &= \alpha + \beta x_i^{\text{avg}} + \cdots \\ &= \alpha + \frac{1}{5}\beta X_{i1} + \cdots + \frac{1}{5}\beta X_{i5} + \cdots . \end{aligned}$$

The resulting model will represent an improvement to the extent that the average of the five standardized economy measures is a better predictor than the single measure chosen before.

However, model (13.13) is limited in that it restricts the coefficients of the five separate x^j's to be equal. More generally, we can replace (13.13) by a weighted average:

$$x_i^{\text{w.avg}} = \frac{1}{5}\sum_{j=1}^{5}\gamma_j X_{ij}, \text{ for } i = 1, \dots, n, \tag{13.14}$$

so that the data model becomes

$$\begin{aligned} y_i &= \alpha + \beta x_i^{\text{w.avg}} + \cdots \\ &= \alpha + \frac{1}{5}\gamma_1\beta X_{i1} + \cdots + \frac{1}{5}\gamma_5\beta X_{i5} + \cdots . \end{aligned} \tag{13.15}$$

We would like to estimate the relative coefficients γ_j from the data, but we cannot simply use classical regression, since this would then be equivalent to estimating a separate coefficient for each of the five predictors, and we have already established that not enough data are available to do a good job of this.

Instead, one can set up a model for the γ_j's:

$$\gamma_j \sim \text{N}(1, \sigma_\gamma^2), \text{ for } j = 1, \dots, 5, \tag{13.16}$$

so that, in the model (13.15), the common coefficient β can be estimated classically, but the relative coefficients γ_j are part of a multilevel model. The hyperparameter σ_γ can be interpreted as follows:

- If $\sigma_\gamma = 0$, the model reduces to the simple averaging (13.14): *complete pooling* of the γ_j's to the common value of 1, so that the combined predictor $x^{\text{w.avg}}$ is simply x^{avg}, the average of the five individual $X_{(j)}$'s.

- If $\sigma_\gamma = \infty$, there is *no pooling*, with the individual coefficients $\frac{1}{5}\gamma_j\beta$ estimated separately using least squares.
- When σ_γ is positive but finite, the γ_j's are *partially pooled*, so that the five predictors x_j have coefficients that are near each other but not identical.

Depending on the amount of data available, σ_γ can be estimated as part of the model or set to a value such as 0.3 that constrains the γ_j's to be fairly close to 1 and thus constrains the coefficients of the individual x^j's toward each other in the data model (13.15).

Connection to factor analysis

A model can include multiplicative parameters for both modeling and computational purposes. For example, we could predict the election outcome in year t in state s within region $r[s]$ as

$$y_{st} = \beta^{(0)} X_{st}^{(0)} + \alpha_1 \sum_{j=1}^{5} \beta_j^{(1)} X_{jt}^{(1)} + \alpha_2 \gamma_t + \alpha_3 \delta_{r[s],t} + \epsilon_{st},$$

where $X^{(0)}$ is the matrix of state × year-level predictors, $X^{(1)}$ is the matrix of year-level predictors, and γ, δ, and ϵ are national, regional, and statewide error terms. In this model, the auxiliary parameters α_2 and α_3 exist for purely computational reasons, and they can be estimated, with the understanding that we are interested only in the products $\alpha_2\gamma_t$ and $\alpha_3\delta_{r,t}$. More interestingly, α_1 serves both a computational and modeling role—the $\beta_j^{(1)}$ parameters have a common $\mathrm{N}(\frac{1}{5}, \sigma_m^2)$ model, and α_1 has the interpretation as the overall coefficient for the economic predictors.

More generally, we can imagine K batches of predictors, with the data-level regression model using a weighted average from each batch:

$$y = X^{(0)}\beta^{(0)} + \beta_1 x^{\text{w.avg},\,1} + \cdots + \beta_k x^{\text{w.avg},\,K} + \cdots,$$

where each predictor $x_k^{\text{w.avg}}$ is a combination of J_k individual predictors x^{jk}:

$$\text{for each } k\text{:}\;\; x_i^{\text{w.avg},\,k} = \frac{1}{J_k} \sum_{j=1}^{J_k} \gamma_{jk} x_i^{jk}, \;\text{ for } i = 1, \ldots, n.$$

This is equivalent to a regression model on the complete set of available predictors, $x^{11}, \ldots, x^{J_1 1}; x^{12}, \ldots, x^{J_2 2}; \ldots, x^{1K}, \ldots, x^{J_K K}$, where the predictor x^{jk} gets the coefficient $\frac{1}{J_k}\gamma_{jk}\beta_k$. Each batch of relative weights γ is then modeled hierarchically:

$$\text{for each } k\text{:}\;\; \gamma_{jk} \sim \mathrm{N}(1, \sigma_{\gamma\,k}^2), \;\text{ for } j = 1, \ldots, J_k,$$

with the hyperparameters $\sigma_{\gamma\,k}$ estimated from the data or set to low values such as 0.3.

In this model, each combined predictor $x^{\text{w.avg},\,k}$ represents a "factor" formed by a linear combination of the J_k individual predictors, β_k represents the importance of that factor, and the γ_{jk}'s give the relative importance of the different components.

As noted at the beginning of this section, these models are currently the subject of active research, and we suggest that they can serve as a motivation to specially tailored models for individual problems rather than as off-the-shelf solutions to generic multilevel problems with many predictors.

13.7 More complex multilevel models

The models we have considered so far can be generalized in a variety of ways. Chapters 14 and 15 discuss multilevel logistic and generalized linear models. Other extensions within multilevel linear and generalized linear models include the following:

- Variances can vary, as parametric functions of input variables, and in a multilevel way by allowing different variances for groups. For example, the model $y_i \sim \mathrm{N}(X_i\beta, \sigma_i^2)$, with $\sigma_i = \exp(X_i\gamma)$, allows the variance to depend on the predictors in a way that can be estimated from the data, and similarly, in a multilevel context, a model such as $\sigma_i = \exp(a_{j[i]} + bx_i)$ allows variances to vary by group. (It is natural to model the parameters σ on the log scale because they are restricted to be positive.)
- Models with several factors can have many potential interactions, which themselves can be modeled in a structured way, for example with larger variances for coefficients of interactions whose main effects are large. This is a model-based, multilevel version of general advice for classical regression modeling.
- Regression models can be set up for multivariate outcomes, so that vectors of coefficients become matrices, with a data-level covariance matrix. These models become correspondingly more complex when multilevel factors are added.
- Time series can be modeled in many ways going beyond simple autoregressions, and these parameters can vary by group with time-series cross-sectional data. This can be seen as a special case of non-nested groupings (for example, country × year), with calendar time being a group-level predictor.
- One way to go beyond linearity is with nonparametric regression, with the simplest version being $y_i = g(X_i, \theta) + \epsilon_i$, and the function g being allowed to have some general form (for example, cubic splines, which are piecewise-continuous third-degree polynomials). Versions of such models can also be estimated using locally weighted regression, and again can be expanded to multilevel structures as appropriate.
- More complicated models are appropriate to data with spatial or network structure. These can be thought of as generalizations of multilevel models in which groups (for example, social networks) are not necessarily disjoint, and in which group membership can be continuous (some connections are stronger than others) rather than simply "in" or "out."

We do not discuss any of these models further here, but we wanted to bring them up to be clear that the particular models presented in this book are just the starting point to our general modeling approach.

13.8 Bibliographic note

The textbooks by Kreft and De Leeuw (1998), Raudenbush and Bryk (2002), and others discuss multilevel models with varying intercepts and slopes. For an early example, see Dempster, Rubin, and Tsutakawa (1981). Non-nested models are discussed by Rasbash and Browne (2003). The flight simulator example comes from Gawron et al. (2003), and the latin square example comes from Snedecor and Cochran (1989).

Models for covariance matrices have been presented by Barnard, McCulloch, and Meng (1996), Pinheiro and Bates (1996), Daniels and Kass (1999, 2001), Daniels and Pourahmadi (2002). Boscardin and Gelman (1996) discuss parametric models for unequal variances in multilevel linear regression. The scaled inverse-Wishart model we recommend comes from O'Malley and Zaslavsky (2005).

The models for combining regression predictors discussed in Section 13.6 appear in Witte et al. (1994), Greenland (2000), Gelman (2004b), and Gustafson and Greenland (2005). See also Hodges et al. (2005) and West (2003) on methods of including many predictors and interactions in a regression. Other work on selecting and combining regression predictors in multilevel models includes Madigan and Raftery (1994), Hoeting et al. (1999), Chipman, George, and McCulloch (2001), and Dunson (2006). The election forecasting example is discussed in Gelman and King (1993) and Gelman et al. (2003, section 15.2); see Fair (1978), Rosenstone (1983), Campbell (1992), and Wlezien and Erikson (2004, 2005) for influential work in this area.

Some references for hierarchical spatial and space-time models include Besag, York, and Mollie (1991), Waller et al. (1997), Besag and Higdon (1999), Wikle et al. (2001), and Bannerjee, Gelfand, and Carlin (2003). Jackson, Best, and Richardson (2006) discuss hierarchical models combining aggregate and survey data in public health. Datta et al. (1999) compare hierarchical time series models; see also Fay and Herriot (1979). Girosi and King (2005) present a multilevel model for estimating trends within demographic subgroups.

For information on nonparametric methods such as lowess, splines, wavelets, hazard regression, generalized additive models, and regression trees, see Hastie, Tibshirani, and Friedman (2002), and, for examples in R, see Venables and Ripley (2002). Crainiceanu, Ruppert, and Wand (2005) fit spline models using Bugs. MacLehose et al. (2006) combine ideas of nonparametric and multilevel models.

13.9 Exercises

1. Fit a multilevel model to predict course evaluations from beauty and other predictors in the `beauty` dataset (see Exercises 3.5, 4.8, and 12.6) allowing the intercept and coefficient for beauty to vary by course category:
   (a) Write the model in statistical notation.
   (b) Fit the model using `lmer()` and discuss the results: the coefficient estimates and the estimated standard deviation and correlation parameters. Identify each of the estimated parameters with the notation in your model from (a).
   (c) Display the estimated model graphically in plots that also include the data.

2. Models for adjusting individual ratings: a committee of 10 persons is evaluating 100 job applications. Each person on the committee reads 30 applications (structured so that each application is read by three people) and gives each a numerical rating between 1 and 10.
   (a) It would be natural to rate the applications based on their combined scores; however, there is a worry that different raters use different standards, and we would like to correct for this. Set up a model for the ratings (with parameters for the applicants and the raters).
   (b) It is possible that some persons on the committee show more variation than others in their ratings. Expand your model to allow for this.

3. Non-nested model: continuing the Olympic ratings example from Exercise 11.3:
   (a) Write the notation for a non-nested multilevel model (varying across skaters and judges) for the technical merit ratings and fit using `lmer()`.

(b) Fit the model in (a) using the artistic impression ratings.

(c) Display your results for both outcomes graphically.

(d) Use posterior predictive checks to investigate model fit in (a) and (b).

4. Models with unequal variances: the folder `age.guessing` contains a dataset from Gelman and Nolan (2002) from a classroom demonstration in which 10 groups of students guess the ages of 10 different persons based on photographs. The dataset also includes the true ages of the people in the photographs.

   Set up a non-nested model to these data, including a coefficient for each of the persons in the photos (indicating their apparent age), a coefficient for each of the 10 groups (indicating potential systematic patterns of groups guessing high or low), and a separate error variance for each group (so that some groups are more consistent than others).

5. Return to the CD4 data introduced from Exercise 11.4.

   (a) Extend the model in Exercise 12.2 to allow for varying slopes for the time predictor.

   (b) Next fit a model that does not allow for varying slopes but does allow for different coefficients for each time point (rather than fitting the linear trend).

   (c) Compare the results of these models both numerically and graphically.

6. Using the time-series cross-sectional dataset you worked with in Exercise 11.2, fit the model you formulated in part (c) of that exercise.