Handout on Random Effects

Richard L. Smith

April 23, 2019

The notes have been produced for the course STOR 556 and are intended to supplement the course text [2].

1 Random Effects Analysis of Variance

A classical *fixed effects analysis of variance* could be based on either of the formulas

$$y_{ij} = \mu + \tau_i + \nu_j + \epsilon_{ij},\ 1 \le i \le I,\ 1 \le j \le J, \tag{1}$$

or

$$y_{ijk} = \mu + \tau_i + \nu_j + \epsilon_{ijk},\ 1 \le i \le I,\ 1 \le j \le J\ 1 \le k \le K, \tag{2}$$

where $\mu,\ \tau_i,\ \nu_j$ represent an overall mean, a treatment effect and a block effect, and ϵ_{ij} or ϵ_{ijk} represent a random error term which is typically, though not necessarily, assumed to be independent $N(0, \sigma_\epsilon^2)$ for some common variance σ_ϵ^2. Both models (1) and (2) may be fitted using the `lm` command where the different levels of treatment i and block j are defined to be factor variables; the advantage of (2) with $K > 1$ is that it also allows us to test *interactions* or cross-products between the treatment and block variables.

The idea of "random effects" is that we may find it more appropriate to treat either the τ_i's or the ν_j's as random variables from some common distribution, for example $\tau_i \sim N[0, \sigma_\tau^2]$ or $\nu_j \sim N[0, \sigma_\nu^2]$ where σ_τ^2 or σ_ν^2 are additional variance terms. Very often, we make some of the terms fixed effects and some of them random effects, in which case it is called a *mixed effects* model.

2 Estimation

Consider the simplest possible model of this form, represented by

$$y_{ij} = \mu + \alpha_i + \epsilon_{ij},\ 1 \le i \le a,\ 1 \le j \le n, \tag{3}$$

with $\alpha_i \sim N[0, \sigma_\alpha^2],\ \epsilon_{ij} \sim N[0, \sigma_\epsilon^2]$, all mutually independent. In this case,

$$\begin{aligned} \mathrm{E}(y_{ij}) &= \mu, \\ \mathrm{Var}(y_{ij}) &= \mathrm{E}\{(\alpha_i + \epsilon_{ij})^2\} \\ &= \sigma_\alpha^2 + \sigma_\epsilon^2, \end{aligned}$$

and for given i, j, j' with $j \neq j'$,

$$\begin{aligned} \mathrm{Cov}(y_{ij}, y_{ij'}) &= \mathrm{E}\{(\alpha_i + \epsilon_{ij})(\alpha_i + \epsilon_{ij'})\} \\ &= \sigma_\alpha^2. \end{aligned}$$

So

$$\rho = \frac{\sigma_\alpha^2}{\sigma_\alpha^2 + \sigma_\epsilon^2}$$

is the correlation between y_{ij} and $y_{ij'}$. This is known as the *intraclass correlation coefficient* (ICC). The question is how we can estimate the quantities σ_α^2, σ_ϵ^2 and ρ.

For notational purposes, we use dots to denote averages over the variable with a dot in place of the index, for example $y_{i\cdot} = \frac{1}{n}\sum_{j=1}^n y_{ij}$, $y_{\cdot\cdot} = \frac{1}{an}\sum_{i=1}^a \sum_{j=1}^n y_{ij}$.

One possible way is to start with the *analysis of variance decomposition*:

$$\begin{aligned} \sum_{i=1}^a \sum_{j=1}^n (y_{ij} - y_{\cdot\cdot})^2 &= \sum_{i=1}^a \sum_{j=1}^n (y_{ij} - y_{i\cdot} + y_{i\cdot} - y_{\cdot\cdot})^2 \\ &= \sum_{i=1}^a \sum_{j=1}^n (y_{ij} - y_{i\cdot})^2 + 2\sum_{i=1}^a \sum_{j=1}^n (y_{ij} - y_{i\cdot})(y_{i\cdot} - y_{\cdot\cdot}) + \sum_{i=1}^a \sum_{j=1}^n (y_{i\cdot} - y_{\cdot\cdot})^2 \\ &= \sum_{i=1}^a \sum_{j=1}^n (y_{ij} - y_{i\cdot})^2 + n\sum_{i=1}^a (y_{i\cdot} - y_{\cdot\cdot})^2 \end{aligned}$$

where the middle term vanishes because $\sum_{j=1}^n (y_{ij} - y_{i\cdot}) = 0$ for any i. We can write this expression in the form

$$SST = SSE + SSA$$

where SST is known as the total sum of squares, SSE is the residual sum of squares, and SSA is often called the treatment sum of squares.

Let's now figure out the expectations of each of these terms. Based on the model (3) we have

$$\begin{aligned} \mathrm{E}\{SST\} &= \mathrm{E}\left\{\sum_{i=1}^a \sum_{j=1}^n (\alpha_i - \alpha_\cdot + \epsilon_{ij} - \epsilon_{\cdot\cdot})^2\right\} \\ &= \mathrm{E}\left\{\sum_{i=1}^a n(\alpha_i - \alpha_\cdot)^2 + \sum_{i=1}^a \sum_{j=1}^n (\epsilon_{ij} - \epsilon_{\cdot\cdot})^2\right\} \\ &= n(a-1)\sigma_\alpha^2 + (na-1)\sigma_\epsilon^2. \end{aligned}$$

Similarly,

$$\begin{aligned} \mathrm{E}\{SSE\} &= \mathrm{E}\left\{\sum_{i=1}^a \sum_{j=1}^n (\epsilon_{ij} - \epsilon_{i\cdot})^2\right\} \\ &= a(n-1)\sigma_\epsilon^2. \end{aligned}$$

Hence

$$\begin{aligned} \mathrm{E}\{SSA\} &= n(a-1)\sigma_\alpha^2 + (na-1)\sigma_\epsilon^2 - a(n-1)\sigma_\epsilon^2 \\ &= (a-1)(n\sigma_\epsilon^2 + \sigma_a^2). \end{aligned}$$

These formulas suggest we define

$$MSE = \frac{SSE}{a(n-1)},\ MSA = \frac{SSA}{a-1}$$

and hence define unbiased estimators

$$\begin{aligned} \hat{\sigma}_\epsilon^2 &= MSE, \\ n\hat{\sigma}_\alpha^2 + \hat{\sigma}_\epsilon^2 &= MSA, \end{aligned} \tag{4}$$

which simplifies to

$$\hat{\sigma}_\alpha^2 = \frac{MSA - MSE}{n}. \tag{5}$$

It should be emphasized that the formulas (4) and (5) were the standard way of estimating these models until about twenty years ago, when the introduction of the `nmle` R package (now updated to `lme4`) made it possible to use the maximum likelihood approach, which we are going to describe. However, there are some disadvantages to these formulas:

1. For more complicated models than (3), the algebra gets very messy;
2. For more complicated models than (3), the solution may not be unique;
3. There is no guarantee that $MSA > MSE$. If it is not, $\hat{\sigma}_\alpha^2$ could be negative!

There are various ad hoc fixes to these issues, but the problem is that each model requires its own set of estimators and there isn't a unified theory.

This approach is sometimes called the *method of moments* (MOM), since it amounts to estimating σ_α^2 and σ_ϵ^2 by equating the means (first-order moments) of MSE and MSA.

3 Likelihood-Based Methods

The alternative approach that has now become much more popular begins by rewriting the linear model in the form

$$y = X\beta + Z\gamma + \epsilon \tag{6}$$

where some of the elements are standard: y is a vector of n observations, X is a $n \times p$ matrix of covariates, β is a vector of p (fixed effect) parameters and ϵ is a vector of errors, typically assumed to be normal with mean 0 and some common variance σ^2. The new contribution is the term $Z\gamma$, where:

- Z is a matrix of covariates (in the case of a simple model like (2) or (3), this will be a matrix of zeros and ones);
- γ is a vector of random effects, typically assumed to have a normal distribution of the form $\gamma \sim N[0, D\sigma^2]$ with D a diagonal matrix, and to be independent of ϵ. For example, if γ is the same as α in (3), D will be an $a \times a$ matrix with off-diagonal entries all 0, diagonal entries all some constant d where $d = \sigma_\alpha^2/\sigma_\epsilon^2$ in the notation of our previous example (3). The problem is then to estimate d.

In the model (6), the covariance matrix of y is now

$$\begin{aligned} \mathrm{E}\left\{(y - X\beta)(y - X\beta)^T\right\} &= \mathrm{E}\left\{(Z\gamma + \epsilon)(Z\gamma + \epsilon)^T\right\} \\ &= \mathrm{E}\left\{(Z\gamma\gamma^T Z^T + Z\gamma\epsilon^T + \epsilon\gamma^T Z^T + \epsilon\epsilon^T\right\} \\ &= (ZDZ^T + I)\sigma^2 \end{aligned}$$

where the middle two terms have expectation zero because γ and ϵ are uncorrelated (one of the assumptions of the model).

Therefore, we write the new model in matrix notation as

$$y \sim N[X\beta, \sigma^2(I + ZDZ^T)] \tag{7}$$

(see equation on bottom line, page 197 of text) where X is the matrix of fixed-effect covariates, Z is the matrix of random-effect covariates, β is the vector of fixed effects and D are the diagonal entries that define the variances of the random effects. The problems are:

(a) Estimate β and σ^2 given the values of D,

(b) Estimate D.

To solve problem (a), the standard method is called *generalized least squares*, which you may have seen in STOR 455. To be specific, if we have a model of the form $y \sim N[X\beta, V\sigma^2]$ where β and σ^2 are unknown but V is a known matrix (the covariance matrix of y, up to a constant σ^2) then the standard estimates are

$$\hat{\beta} = (X^T V^{-1} X)^{-1} X^T V^{-1} y, \tag{8}$$

$$\hat{\sigma}^2 = \frac{(y - X\hat{\beta})^T V^{-1} (y - X\hat{\beta})}{n - p}. \tag{9}$$

To solve problem (b), the best-known general method is the *method of maximum likelihood* which we have already discussed extensively in the GLM part of the course. For model (7), the joint density of all the observations is

$$(2\pi\sigma^2)^{-n/2} |V|^{-1/2} \exp\left\{-\frac{(y - X\beta)^T V^{-1} (y - X\beta)}{2\sigma^2}\right\} \tag{10}$$

where $|V|$ denotes the determinant of the matrix $V = I + ZDZ^T$. so the log likelihood is

$$\ell = -\frac{n}{2}\log 2\pi - \frac{n}{2}\log\sigma^2 - \frac{1}{2}\log|V| - \frac{(y - X\beta)^T V^{-1} (y - X\beta)}{2\sigma^2}. \tag{11}$$

We maximize (11) in stages:

(a) Minimize $(y - X\beta)^T V^{-1}(y - X\beta)$ with respect to β. By calculus or a geometric argument, this leads us back to (8).

(b) Minimize $(n \log \sigma^2)/2 + (y - X\hat{\beta})^T V^{-1}(y - X\hat{\beta})/(2\sigma^2)$ with respect to σ^2. This leads us to

$$\hat{\sigma}^2 = \frac{(y - X\hat{\beta})^T V^{-1}(y - X\hat{\beta})}{n} \tag{12}$$

(c) Substitute $\hat{\beta}$ and $\hat{\sigma}^2$ back into (11) which leads us to try to minimize (ignoring constants)

$$\frac{n}{2} \log \hat{\sigma}^2 + \frac{1}{2} \log |V| \tag{13}$$

and then we choose D (which determines both V and $\hat{\sigma}^2$) to minimize (13). Although there is no possibility of a closed-form solution in this case, the function (13) is fairly easily programmed into a numerical optimization routine and this is used to determine the estimator $\hat{D}$.

There is one catch, however: estimator (12) for σ^2 is not the same as estimator (9). In fact, you probably know already that estimator (12) is biased, and the correction of dividing by $n - p$ rather than n is there to make the estimator unbiased. This is true even in the simpler case where V is the identity matrix and this is a known disadvantage of the method of maximum likelihood for the general linear model.

The solution is that, instead of using maximum likelihood, we adopt a variant known as *restricted maximum likelihood* estimation (sometimes also called the *reduced maximum likelihood* method, but either way generally abbreviated to *REML*). This is described on p. 198 of Faraway, but here is a different and more explicit (but equivalent) formulation.

Replace (11) by

$$\ell_R = -\frac{n-p}{2} \log 2\pi - \frac{n-p}{2} \log \sigma^2 - \frac{1}{2} \log |V| - \frac{1}{2} \log |X^T V^{-1} X| - \frac{(y - X\beta)^T V^{-1}(y - X\beta)}{2\sigma^2}. \tag{14}$$

We now choose successively β, σ^2 and V to maximize ℓ_R. The estimator (8) for $\hat{\beta}$ is the same, and the estimator for σ^2 now turns out to be (9), with the correct divisor $n - p$. Finally, we substitute $\hat{\beta}$ and $\hat{\sigma}^2$ into (14) and choose D to minimize

$$\frac{n-p}{2} \log \hat{\sigma}^2 + \frac{1}{2} \log |V| + \frac{1}{2} \log |X^T V^{-1} X|. \tag{15}$$

One point to note is that the estimator (9) is no longer an exactly unbiased estimator when V is estimated. However, both theoretical and numerical studies have generally confirmed that both $\hat{D}$ (the values of D that minimize (15)) and $\hat{\sigma}^2$ are more accurate under the *REML* method than direct maximum likelihood.

4 Fitting in R

The package to fit these models is called `lme4` and the main function within that package is called `lmer`. A typical syntax of the command (see text) is

```
mmod=lmer(bright~1+(1|operator),pulp)
```

Here we have a dataframe called `pulp`; the response is `bright` and there is one factor variable called `operator`. The first `1+` is just fitting a constant (intercept) as the sole fixed effect; it would be possible to introduce additional fixed effects (or regression coefficients) here by specifying the appropriate x variables. The second term `(1|operator)` (always within parentheses) tells us that the observations are to be grouped according to the variable `operator`, and the `1|` part specifies that the random effect is constant within each group. The default is REML estimation but you can also get maximum likelihood by including the option `REML=F`. As we go on, we will see more complicated formulas for specifying the structure of random effects models we want.

5 Testing

This section considers testing for the significance of a random effects term. In the notation of (3): test $H_0 : \ \sigma_\alpha^2 = 0$ versus $H_A : \ \sigma_\alpha^2 > 0$. One possible method is just to use the same F test as you would in a fixed effects ANOVA — see the example at the top of p. 199 of the course text, where we found $MSA = 0.447,\ MSE = 0.106$, and the F test (for the null hypothesis that the operator effect is 0) has a p-value of 0.023. There is actually nothing wrong with this approach for the random effects model as well (if the null hypothesis is that there's no treatment effect at all, it really doesn't matter whether it's a fixed or random effect....) but the difficulty is that the ANOVA table itself becomes increasingly difficult to calculate (or, in some cases, even to define) as the model becomes complicated and unbalanced.

As an alternative to this approach, we consider likelihood ratio tests as a general all-purpose method of comparing two nested hypotheses. The likelihood ratio test (LRT) statistic is defined by

$$W \ = \ 2\left\{\ell(\hat\beta_1, \hat\sigma_1, \hat D_1|y) - \ell(\hat\beta_0, \hat\sigma_0, \hat D_0|y)\right\}$$

where $\hat\beta_0,\ \hat\sigma_0,\ \hat D_0$ are MLEs under H_0 and $\hat\beta_1,\ \hat\sigma_1,\ \hat D_1$ are MLEs under H_1.

Caution: To calculate a LRT you must use the actual MLEs, not the REML estimators (use `REML=F` when you fit the model in R). The reason is the REML estimators for H_0 and H_1 are optimizing in different spaces and therefore not comparable.

Some notes on the use of LRT tests:

- The obvious starting point is to assume $W \sim \chi^2_\nu$ where ν is the number of additional parameters in H_1 compared with H_0 (in the simple case we've been discussing so far, $\nu = 1$). However, **further caution**, this may not be a good approximation to the true distribution of W. The reason is that although the MLE cannot give a negative value for a parameter like σ_α^2, it can give a zero value, in which case $W = 0$. This is inconsistent with a chi-squared distribution for W. To see that this is something that actually happens, see the example at

the top of page 205, where it is calculated by simulation that $\Pr\{W < 0.00001\} \approx 0.7$ for the particular dataset discussed there.

- As an alternative, we can use the **bootstrap method**, which uses simulated datasets where H_0 is assumed correct to approximate the sampling distribution of W. In the example on pp. 204–205, the sharply reduces the calculated p-value (from 0.109 by the χ^2 approximation to about 0.019) and changes the conclusion from accepting H_0 to rejecting it.
- A third method uses the `RLRsim` package (p. 205), whose point (I think) is to use a more efficient method of programming the simulation method to speed up the analysis (I suspect what they have done is to code the simulation part of the program in C++, which would be much faster than writing a loop in R).

The conclusion: LRTs are a flexible general method for testing these models, but don't rely on the χ^2 approximation to the distribution; better to use a bootstrap or a package such as `RLRsim` which computes what are in principle exact tests.

6 Estimating Random Effects

So far, the discussion has been about how to estimate the variance of the random effects and test the statistical significance of that variance, but we have not talked about estimating the effects themselves (unlike fixed effects, which show themselves as just coefficients of the regression equation). The text (Section 10.3) discusses this, but I think it's a little misleading the way they talk about Bayesian calculations — the method is Bayesian in the sense of calculaing a conditional distribution, but it doesn't rely on having prior distributions for the unknown parameters, which most statisticians would consider the central idea of Bayesian statistics.

The essence of the problem is this: the model is defined by two sets of random quantities γ and y, where y is observed but γ is not. What we are trying to do it to predict the unobserved γ as a function of the observed y. This is an exercise in conditional probabilities, but it has nothing directly to do with Bayesian statistics.

Although we haven't formally studied the "multivariate normal distribution," there have been places where we have used it, for example, equation (10) gives the probability density function for a multivariate normal random variable y with mean $X\beta$ and covariance matrix $\sigma^2 V$. We won't need that formula for what follows, but the key concept is that we can define something called the multivariate normal distribution where the mean vector and the covariance matrix are given.

Now we extend this concept to that of a *partitioned multivariate normal distribution.* Suppose the vector y is partitioned into two vectors $\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$ and we correspondingly partition the means and covariance matrices as $\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$ and $\begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix}$.

In other words, y_1 is a vector with mean μ_1 and covariance matrix V_{11}; y_2 is similar with μ_2 and V_{22}, while $V_{12} = V_{21}^T$ is the matrix of cross-covariances between y_1 and y_2. In this context, we note the following:

Very useful fact. The conditional distribution of y_1 given y_2 is also multivariate normal, with mean $\mu_1 + V_{12}V_{22}^{-1}(y_2 - \mu_2)$ and covariance matrix $V_{11} - V_{12}V_{22}^{-1}V_{21}$.

I won't try to give the proof — this is standard in books that cover the multivariate normal distribution and it can in fact be derived directly from the formula (10) (though this is tedious).

Now let's apply this to our model (6). We identify y_1 and y_2 with γ and y respectively, The means are $\mu_1 = 0$ and $\mu_2 = X\beta$ and the covariance matrices are $V_{11} = D\sigma^2$ and $V_{22} = (I + ZDZ^T)\sigma^2$. Also, the matrix of cross-covariances is

$$\begin{aligned} V_{21} \quad = \quad V_{12}^T \quad &= \quad \mathrm{E}\left\{(Y - X\beta)\gamma^T\right\} \\ &= \quad \mathrm{E}\left\{(Z\gamma + \epsilon)\gamma^T\right\} \\ &= \quad ZD\sigma^2 \end{aligned}$$

since the cross-covariances of γ and ϵ are 0.

Therefore, the conditional mean of γ given y is

$$V_{12}V_{22}^{-1}(y - X\beta) \quad = \quad DZ^T(I + ZDZ^T)^{-1}(y - X\beta)$$

and the conditional covariance matrix is

$$V_{11} - V_{12}V_{22}^{-1}V_{21} \quad = \quad \left\{D - DZ^T(I + ZDZ^T)^{-1}ZD\right\}\sigma^2.$$

The first expression is exactly that for $\hat{\alpha}$, p. 207. The second formula does not appear anywhere in the text (I don't think), but it presumably underlies the calculation of `condVar` and the confidence interval plots on p. 208.

6.1 Implementation in R

Let's suppose a random effects model has been previously fitted in the R object `mmod`. The command

```
r1=ranef(mmod)
```

creates a vector of estimated values for the random effects. If this is modified to

```
r1=ranef(mmod,condVar=T)
```

then this also includes conditional variances, though it does not seem so easy to extract the numerical values of these. The best method I have been able to figure out uses

```
str(r1)
```

which includes this information along with other stuff that you may not want.

A graphical display of the result (see p. 208) is possible through

```
library(lattice)
dotplot(r1)
```

7 Testing Hypotheses for Fixed or for Random Effects

The *Kenward-Roger test* is a test for the presence of certain fixed effects in a model that also has random effects. The format is typically something like

```
library(pbkrtest)
KRmodcomp(m1,m2)
```

where `m1` is the fitted model under the alternative hypothesis and `m2` is the fitted model under the null hypothesis. I think there is a small error in the text at the bottom of page 213: the Kenward-Roger test is designed to be used with a REML procedure for the random effect, not the MLE. (The MLE would be needed if we were using a bootstrap, as occurred in at least one of your homework examples, but K-R is not a bootstrap procedure; it's based on an asymptotic approximation to improve the properties of a likelihood ratio test).

In the above example, `m2` is the null model containing at least one random effect and possible also some fixed effects. `m1` is the alternative model which contains everything in `m2` and some additional fixed effect(s). The result is a p-value for the significance of the additional fixed effect(s).

A second kind of testing problem is for the significance of a random effect, typically in the presence of other random effects. In this case, there may or may not be fixed effects. One method for this is the "parametric bootstrap" method of sampling the likelihood ratio statistic from the null model and then comparing the observed LRT statistic with the bootstrap distribution. A command of the form `PBmodcomp(m1,m2)` can be performed within the `pbkrtest` package. Note that this often produces a large number of warning measures due to singularities in the fit (this typically happens when one of the estimated variances is zero). An alternative is the `exactRLRT` method within the `RLRsim` package. The syntax of this is a little unintuitive but see Section 9 for a worked example.

More the about Kenward-Roger test. Recall that the random effects model leads to a joint distribution of the form (6), where the covariance matrix is of the form $\sigma^2 V$, $V = I + Z^T D Z$, where the diagonal entries of the diagonal matrix D depend on the variances of the random effects. When V is known, unbiased estimates of β and σ^2 are given by (7) and (8), and though we didn't state it here, for testing linear hypotheses about the regression coefficients β, it's possible to develop an exact test based on the F distribution, just like the standard linear model where $V = I$.

However, neither the unbiasedness of $\hat{\beta}$ and $\hat{\sigma}^2$ nor the exactness of the resulting F test holds if D and hence V is estimated. The idea of the Kenward-Roger test is to develop a correction for that. Specifically, for testing a hypothesis of the form $H_0 : \ C\beta = 0$ against the alternative $H_0 : \ C\beta \neq 0$, where we assume β is p-dimensional and C is a $q \times p$ matrix for some $q < p$, the Kenward-Roger procedure calculates a p-value by applying asymptotic corrections to the standard F test. We don't need to go into details about the algebraic formulas involved.

Kenward-Roger or bootstrap? The bootstrap test can also be run for two models which have the same random effects and which differ only in their fixed effects, with one model nested inside the other. Which one is best in this case?

As noted above, the Kenward-Roger test is based on an asymptotic approximation, which means that it won't be an exact result for small sample sizes. The form of bootstrap test we

have used is known as the parametric bootstrap because it relies on generating simulations from the null model, as opposed to the nonparametric bootstrap which relies on resampling the data (however, the nonparametric bootstrap would be hard to apply in a complicated model with many covariates). The parametric bootstrap does not rely on asymptotic calculations but only on Monte Carlo simulation, and in that sense, could be a more accurate way to calculate p-values. However, there are two caveats: it could take a very large number of Monte Carlo simulations to produce a result that is more accurate than Kenward-Roger, and the parametric bootstrap is still not an exact test, in the sense that even the null model usually contains unknown parameters are these are estimated prior to drawing the samples.

It seems to me that Kenward-Roger is usually a pretty accurate procedure and therefore should be preferred in practice to the bootstrap method. However, the bootstrap method is always available as a backup, and if a situation arose where the two methods were giving substantially different results, I would tend to prefer the bootstrap method.

8 Split Plot Design: Irrigation Example (Section 10.7)

- Two crop varieties, four methods of irrigation, eight fields
- Each field can use only one method of irrigation
- However, we can and do split each field into two — one for each variety
- Variety, irrigation and the variety $\times$ irrigation interaction are fixed effects; field effect is random; we could in principle also have a field $\times$ interaction random effect
- y_{ijk} is yield for irrigation method i with variety j in field k

Most general model:

$$y_{ijk} = \mu + i_i + v_j + (iv)_{ij} + f_k + (vf)_{jk} + \epsilon_{ijk}$$

Fitted with:

```
m1=lmer(yield~irrigation+variety+irrigation:variety+(1|field)+(1|field:variety),irrigation)
```

This doesn't work — cannot separate field $\times$ variety effect from field effect. See:

```
Error: number of levels of each grouping factor must be < number of observations
```

So we drop the field $\times$ irrigation interaction.

```
m2=lmer(yield~irrigation+variety+irrigation:variety+(1|field),irrigation)
m3=lmer(yield~irrigation+variety+(1|field),irrigation)
m4=lmer(yield~variety+(1|field),irrigation)
m5=lmer(yield~irrigation+(1|field),irrigation)
m6=lmer(yield~1+(1|field),irrigation)
```

Each of these commands fits the data but successive `summary(mx)` commands cast doubt on the sgnificance of any of the fixed effect parameters.

We can formally test this using the Kenward-Roger test:

```
library(pbkrtest)
KRmodcomp(m2,m3)
KRmodcomp(m3,m4)
KRmodcomp(m3,m5)
KRmodcomp(m4,m6)
KRmodcomp(m5,m6)
```

None of these gives a statistically significant result (all the p-values are well above 0.05). Therefore, we conclude that neither variety nor irrigation is significant as a field effect.

We can test whether the random effect term in `m6` is significant by the bootstrap method:

```
library(RLRsim)
exactRLRT(m6)
```

The p-value is around 0.0002, highly signiicant.

We can also get confidence intervals for the standard deviations of the random effects (first row for field random effect, second term for residuals).

```
> confint(m6)
Computing profile confidence intervals ...
                 2.5 %    97.5 %
.sig01       1.9795357  5.926239
.sigma       0.8112183  2.219072
(Intercept) 37.6047703 42.845229
```

We can also do diagnostics:

```
par(mfrow=c(1,2),cex=1.2)
plot(fitted(m6),residuals(m6),xlab='Fitted',ylab='Residuals',pch=20)
qqnorm(residuals(m6),main='',pch=20)
```

This is similar to Figure 10.6 of the text, though not identical because it's a slightly different model.

Conclusions: None of the fixed effects due to variety or irrigation (or their interaction) is significant. However, there is a clearly significant random effect due to field.

9 Nested Effects: Eggs Example (Section 10.8)

- 48 samples of egg powder, variable of interest is `Fat`
- Six labs, labelled I, II, III, IV, V, VI
- Two technicians in each lab, labelled `one` and `two` — *but they are different people in each lab — technician is nested within lab*
- Two Samples of egg powder, labelled G and H (but they are really the same; purpose was to test lab consistency)

- Two measurements for each combination of lab, technician and sample
- $y_{ijk\ell}$ is fat level for labroatory i, technician j, Sample k, measurement ℓ

Initial data loading and graphical display:

```
data(eggs)
library(ggplot2)
ggplot(eggs,aes(y=Fat,x=Lab,color=Technician,shape=Sample))+
geom_point(position=position_jitter(width=0.1,height=0.0))+scale_color_grey()
```

Except for a bit of random jitter, this is the same as Figure 10.7 of the text.

All the effects except the overall mean should be treated as random effects because both Technician and Sample are nested within Lab. Therefore, the appropriate model is

$$y_{ijk\ell} \quad = \quad \mu + L_i + T_{ij} + S_{ijk} + \epsilon_{ijk\ell}.$$

We can try this model, and also the model omitting the S_{ijk} term:

```
m1=lmer(Fat~1+(1|Lab)+(1|Lab:Technician)+(1|Lab:Technician:Sample),data=eggs)
m2=lmer(Fat~1+(1|Lab)+(1|Lab:Technician),data=eggs)
```

We can do `summary` with both `m1` and `m2` — the point the text makes here is that the estimated standard deviations are quite similar under both models — casts doubt on significance of the extra term.

More formal test based on bootstrap:

```
m3=lmer(Fat~1+(1|Lab:Technician:Sample),data=eggs)
library(RLRsim)
exactRLRT(m3,m1,m2)
```

Note the format of the `exactRLRT` command. The models being tested are `m1` (alternative hypothesis) and `m2` (null hypothesis). However, in order to run `exactRLRT` it is necessary first to fit a model containing *only* the random effect being set to zero under the null hypothesis — this explains model `m3`. The result is a p-value of about 0.1 — not significant.

We could have also performed the above test with

```
library(pbkrtest)
PBmodcomp(m1,m2)
```

though this seems less precise in this example; see Section 7 for further discussion.

We therefore eliminate the `Sample` effect but retain a test for the `Technician` effect. This can be done via

```
m4=lmer(Fat~1+(1|Lab),data=eggs)
m5=lmer(Fat~1+(1|Lab:Technician),data=eggs)
exactRLRT(m5,m2,m4)
```

In this case the p-value is about 0.002 — clearly, the Technician effect is significant.

The text also illustrates confidence intervals —

```
> confint(m1)
Computing profile confidence intervals ...
                 2.5 %    97.5 %
.sig01      0.00000000 0.1151956
.sig02      0.00000000 0.1745313
.sig03      0.00000000 0.1794302
.sigma      0.06550185 0.1158823
(Intercept) 0.29651556 0.4784844
```

This unlines the point that we cold drop any of the three random effect terms and there is no clear-cut guidance which.

Side Comment. It's not obvious to me that you couldn't treat Sample as a fixed effect. If Samples G and H were really products from two different manufacturers, this would seem the way to go. So an alternative procedure would be to fit a mixed model with Sample as a fixed effect, and then the Kenward-Roger test for the significance of Sample:

```
m6=lmer(Fat~1+Sample+(1|Lab)+(1|Lab:Technician),data=eggs)
KRmodcomp(m6,m2)
```

The p-value is about 0.076 — not significant, but maybe not so far off.

10 Crossed Effects: Abrasion Example (Section 10.9)

This is an example of a latin square design. There are four materials A, B, C, D and our ultimate interest is in comparing them for the response "wear". However the position in the test specimen (1, 2, 3, 4) is also important and we cannot compare all possible combinations of material and position (a "complete factorial" design would require 16 runs but we have only four). So we try to balance them by using each combination of material and position exactly once, as shown by the following:

```
data(abrasion)
matrix(abrasion$material,4,4)
```

```
     [,1] [,2] [,3] [,4]
[1,] "C"  "A"  "D"  "B"
[2,] "D"  "B"  "C"  "A"
[3,] "B"  "D"  "A"  "C"
[4,] "A"  "C"  "B"  "D"
```

A fixed effects model could be fitted by:

```
m1=aov(wear~material+run+position,abrasion)
summary(m1)
```

This makes it pretty clear that the `material` effect is significant (p-value 0.00085) but the `run` and `position` effects look significant as well. However, the latter two effects are less easy to interpret given that they are not really fixed effects.

Therefore, we do the equivalent random effects model:

```
m2=lmer(wear~material+(1|run)+(1|position),abrasion)
summary(m2)
```

We can test for the fixed effect using Kenward-Roger:

```
m3=lmer(wear~1+(1|run)+(1|position),abrasion)
library(pbkrtest)
KRmodcomp(m2,m3)
```

This again produces a p-value about 0.00085.

We can also use the RLRsim package to test the random effects:

```
m4=lmer(wear~material+(1|run),abrasion)
m5=lmer(wear~material+(1|position),abrasion)
library(RLRsim)
exactRLRT(m4,m2,m5)
exactRLRT(m5,m2,m4)
```

In both cases the result is statistically significant.

The result in this example is that all three variable (material, position and run) are statistically significant, but treating the latter two as random effects seems more informative for generalization to future tests. (I could also imagine a case for treating position as a fixed effect — it depends what this variable really means — but I doubt that one would obtain different results by formally testing that.)

11 Multilevel Models: School Testing Example (Section 10.10)

This example is considerably more complicated than the preceding ones, but it contains no fundamentally new concepts: the key ideas are to fit various combinations of fixed and random effects, using both formal tests of fit and comparisons based on AIC or BIC to reduce the number of models compared.

Following the introductory code in the text we begin with

```
data(jsp)
# drop everything outside year 2
jspr=jsp[jsp$year==2,]
library(ggplot2)
ggplot(jspr,aes(x=raven,y=math))+xlab('Raven Score')+ylab('Math Score')+
geom_point(position=position_jitter(),alpha=0.3)
ggplot(jspr,aes(x=social,y=math))+xlab('Social Class')+ylab('Math Score')+
geom_boxplot()
```

Key points of the subsequent analysis:

- Initial analysis using `lm`: gender can be dropped, raven and social both significant, *but* it's not necessarily correct analysis because ignores grouping into schools and classes (not really independent observations)
- Using random effects models by school and class allows for grouped effects
- After several tests we decide to remove all gender effects
- Use centered raven scores to better interpret interactions
- Use model to calculate adjusted scores for each school — different interpretation from raw scores, adjusts for different entering raven scores
- Test for significance of random effects — class effect marginal but school effect is clear
- Compositional effect — is mean raven score per school significant? — probably not.

12 Random Effects Models for Nonnormal Responses (Chapter 13)

Recall the definition of an exponential family:

$$f(y_i \mid \theta_i, \phi) \quad = \quad \exp\left\{\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y, \phi)\right\}. \tag{16}$$

In a GLM, there is some function ψ so that $\theta_i = \psi(x_i^T\beta)$ where x_i is a set of covariates associated with observation i and β is a vector of regression coefficients. (In the case of a canonical link function, $\psi(x_i^T\beta)$ is just $x_i^T\beta$ and the text makes that assumption, but it seems to me we might want to consider non-canonical link so I've used this slightly more general notation.)

The basic idea of a random effects GLMM, similar to that of a random effects linear model, is to extend the regression function to

$$\theta_i = \psi(x_i^T\beta + z_i^T\gamma) \tag{17}$$

where γ is a vector of random effects assuming $\gamma \sim N[0, D\sigma^2]$ for some matrix D. It is not actually necessary that D be diagonal, and in some of our examples, it will not be.

Hence the likelihood function may be written

$$L(\beta, \phi, D \mid y) \quad = \quad \int \prod_{i=1}^{n} f(y_i \mid \beta, \phi, \gamma) h(\gamma \mid D) d\gamma. \tag{18}$$

[I think this is the right way round: integral and then product sign. Very often several observations y depend on a single element of γ, for example, any time we have repeated measurements on the same individual. The observations are conditionally independent given the random effects,

but the resulting joint likelihood must be integrated out with respect to the random effects. This is different from the way it is written on page 275 of the text.]

The model defined by (18) is complicated because of the integral sign: we could evaluate $\prod_{i=1}^{n} f(y_i \mid \beta, \phi, \gamma) h(\gamma \mid D)$ and then plug it into a numerical optimization routine (in effect, this is what the `glm` function does, though it uses a rather particular optimization algorithm). However, without an analytic form for the integral, there's no direct way to evaluate (18). The normal linear model with random effects does not suffer from this difficulty because the integral is evaluated analytically to give formula (10).

The text describes five ways of dealing with this issue:

- Penalized quasi-likelihood (PQL — Section 12.2)
- Integrated likelihood method (Section 12.3)
- Bayes method (Section 12.4)
- INLA method (Section 12.5)
- Generalized estimating equations (Section 12.6)

First we describe the data we're going to use to evaluate all these methods. I'm skipping over Section 13.3 of the text and going straight to the epilepsy example in Section 13.4.

12.1 Epilepsy data

The data consist of 59 epileptic patients, each observed over five time period (one initial-eight-week period called the baseline, then four two-week periods called the experiment. Each patient is given a placebo during the baseline period. During the experimental period, 28 patient are kept on the placebo and the other 31 are given a drug which is supposed to reduce epileptic seizures. The primary variables of interest are `seizures` (number of seizures in each period of observation), `id` (patient id, 1–59), `treat` (0 for the placebo, 1 for treatment), `expind` (0 during the baseline period, 1 for experiment), `timeadj` (length of the period, either 2 weeks or 8) and `age` (this doesn't seem to get used anywhere). The first few lines of code create some additional variables (mainly for labelling purposes), draws some figures (Fig. 13.5 on page 287), and computes the following table:

	Drug	
phase	placebo	treatment
baseline	3.848	3.956
experiment	4.304	3.984

Variable tabulated is rate of seizures in number per week: this suggests the rate rises in both the placebo and treatment group from the baseline to the experiment period, but possibly rises more for the placebo patients. If true, this would confirm the effectiveness of the drug, though in the rather negative sense that the seizure rate of the treatment patients didn't rise as much as that of the placebo patients dring the experimental period.

Note that one outcome of this initial analysis is to exclude patient number 49. As the text acknowledges, it's not clear whether this action is either necessary or appropriate.

An initial GLM fit shows significant negative interaction between `expind` and `treat`. The interpretation of this is that te treatment group does significantly better (lower rate of seizures) during the experimental part of the trial. However, this analysis does not account for the grouping effect that there are multiple observations per patient:

```
modglm=glm(seizures~offset(log(timeadj))+expind+treat+I(expind*treat),family=poisson,epilo)
sumary(modglm)
```

We therefore look for ways to incorporate the patient random effect.

12.2 PQL method

The PQL method extends the quasi-likelihood approach for GLMs which (recall Chapter 8) breaks up the calculation into a series of iterated steps each of which is a weighted ordinary linear regression. The PQL method essentially replaced the linear regression step with a random effects linear regression. The code is this:

```
library(MASS)
modpql=glmmPQL(seizures~offset(log(timeadj))+expind+treat+I(expind*treat),
random=~1|id,family=poisson,epilo)
summary(modpql)
```

The result shows an almost identical point estimate for the `I(expind*treat)` interaction, but the standard error is larger, reflecting the grouping effect. It is still statistically significant with a p-value of about 0.008.

12.3 Integrated likelihood method

This is handled by the function `glmer` within the package `lme4`. It is therefore a direct generalization of the earlier `lmer` function. The integral in (18) is handled by a numerical approximation — either the Laplace method (essentially based on the function value at its maximum and its second derivative) or Gauss-Hermite quadrature (used a weighted sum of up to 25 points). Gauss-Hermite is slower but more accurate and is reflected by the `nAGQ` parameter which indicated the number of points at which the likelihood function is evaluated (`nAGQ=1` reduces to the Laplace method). Here is the code used in this example:

```
library(lme4)
modgh=glmer(seizures~offset(log(timeadj))+expind+treat+I(expind*treat)+(1|id),
nAGQ=25,family=poisson,epilo)
summary(modgh)
```

Compared with PQL, we again get an almost identical value of the point estimate of the experiment $\times$ treatment interaction, but the standard error is smaller — in fact, we get an almost identical p-value as the `glm` fit (1.4×10^{-5}). However, the `treat` estimate is quite different, so the `glmer` and `glm` models are not the same.

12.4 Bayes method

I am only going to describe this briefly since there is a whole chapter on Bayesian methods which we have skipped over (Chapter 12 of [2]). The core of a Bayesian method replaces the likelihood function (18) by a formula of form

$$f(\gamma, \beta, \phi, D \mid y_1, ..., y_n) \quad = \quad \frac{\{\prod_{i=1}^n f(y_i \mid \beta, \phi, \gamma)\} \; h(\gamma \mid D) \; \pi(\beta, \phi, D)}{\int\int\int\int \{\prod_{i=1}^n f(y_i \mid \beta, \phi, \gamma)\} \; h(\gamma \mid D) \; \pi(\beta, \phi, D) d\gamma d\beta d\phi dD}. \quad (19)$$

Here, the quantity $\pi(\beta, \phi, D)$ is known as the prior density of the unknown parameters β, ϕ, D. It has no prescribed form but is usually taken to be a very diffuse density function. The function (19) is known as a posterior density function and represents the joint density of all the unknowns, β, ϕ, D and γ, conditional upon the observations $y_1, ..., y_n$. The computational difficulty with (19) arises from the quadruple integral integral in the denominator (for our model, a triple integral because there is no parameter ϕ) but this is resolved by an algorithm called *Markov chain Monte Carlo* (MCMC for short) which in effect creates simulated samples from the full joint distribution (19). Although it is possible (with some training) to write your own code for MCMC, here we use a package called STAN, named after Stanislav Ulam, a mid-twentieth-century mathematician and physicist who was a pioneer of Monte Carlo methods. The result of the analysis is a file of Monte Carlo simulations which may be used to draw posterior density plots such as Figure 13.6 of the text, or the table in the middle of page 290. The results are quite consistent with the `glmer` results.

Pointers to the implementation: it's necessary to create some basic STAN code which I have put online as the file `glmmpois.stan`. You will have to download this file into your own home directory in order to run the code. The rest is the same as the R code on pages 289–290. My results are the same as the text's modulo some inevitable discrepancies due to the Monte Carlo sampling.

12.5 INLA method

INLA is an abreviation for *integrated nested Laplace approximation* and extends the one-dimensional Laplace approximation to compute multiple intergals of the form of (18). It is an alternative to STAN which is faster to run though less accurate. The results are similar to those generated by STAN but I won't say any more about this method.

12.6 GEE method

The integrated likelihood, Bayesian and INLA approaches each rely on the full density function f in (18). They are therefore contrary to the spirit of the quasi-likelihood approach to GLMs which relies only on the specification of the mean and variance (see Chapter 8 of the text or my own "GLM" handout). Is there a version of quasi-likelihood that similarly relies only on means and variances and not the full likelihood specification? The answer is the *generalized estimating equations* approach, usually abbreviated to *GEE*. This approach relies on the means and variances of the observations, and the *correlation structure* within each group. The variable `id` indicates which variable is used for grouping. The possible correlation structures are `corstr='independence'` (ignore intra-group correlation); `corstr='exchangeable'` (all correlations within each group the same; effectively the same as the ICC — see page 2 of this note); `corstr='ar1'` (time series dependence; assumes observations are arranged in time order); `corstr='unstructured'` and `corstr='user-defined'`. For the epilepsy data, the text suggests the code

```
library(geepack)
modgeep=geeglm(seizures~offset(log(timeadj))+expind+treat+I(expind*treat),id=id,
family=poisson,corstr='ar1',data=epilepsy,subset=(id!=49))
summary(modgeep)
```

which again confirms the significance of the treatment × expind interaction, but with a much less statistically significant result (p-value 0.022).

12.7 Other possible analyses

Just to illustrate that non-canonical link functions are possible with `glmer`, here's one example:

```
library(lme4)
modgh1=glmer(seizures~offset(log(timeadj))+expind+treat+I(expind*treat)+(1|id),
family=poisson(link='sqrt'),epilo)
summary(modgh1)
```

I don't recommend this; just wanted to show it was possible.

It's possible to try other correlation structures within `geeglm`, for example

```
modgeep=geeglm(seizures~offset(log(timeadj))+expind+treat+I(expind*treat),id=id,
family=poisson,corstr='exchangeable',data=epilepsy,subset=(id!=49))
summary(modgeep)
#
modgeep=geeglm(seizures~offset(log(timeadj))+expind+treat+I(expind*treat),id=id,
family=poisson,corstr='unstructured',data=epilepsy,subset=(id!=49))
summary(modgeep)
```

The first of these shows a non-significant `expind*treat` interaction term (p=0.077) but the second again shows a significant effect (p=0.039), though the p-value is much larger than in the earlier Poisson analyses.

It's also possible to add terms to the integrated likelihood analysis. Note that when there's more than one random effect, the Gauss-Hermite method in `glmer` cannot be applied; the only possible option is `nAGQ=1` which is equivalent to the Laplace method (default).

```
modgh4=glmer(seizures~offset(log(timeadj))+expind+treat+I(expind*treat)+(1+expind|id),
family=poisson,epilo)
summary(modgh4)
```

This model was suggested by Diggle et al. [1]. It leads to a larger standard error for the `expind*treat` interaction though it is still significant (p=0.019).

We could go beyond this and add further random effect terms, for instance

```
modgh5=glmer(seizures~offset(log(timeadj))+expind+treat+I(expind*treat)+
(1+expind+period|id),family=poisson,epilo)
summary(modgh5)
```

which appears significantly better than `modgh4` as judged by an `anova` test, but the statistical significance of the `expind*treat` is not much changed (p=0.021).

Finally we look at residual plots, see for example Figure 1. It is noticeable that the residuals from `modgh5` are considerably smaller overall than those from `modgh`, which suggests a better fit.

Overall, I'm a little suspicious of the integrated likelihood and Bayesian methods because they don't make any check on the fit of the Poisson distribution (unlike regular GLMs, there's no automatic "quasi" option to check for overdispersion). In this sense, the PQL and GEE methods are more robust. However, putting in additional random effects terms is an alternative method and in some respects more interpretable than overdispersion, and it is reassuring that the final results are quite similar in both methods. On the other hand, the initial results leading to a p-value of the order of 10^{-5} seem to be oversimplified because they ignore the individual-level effects entirely.

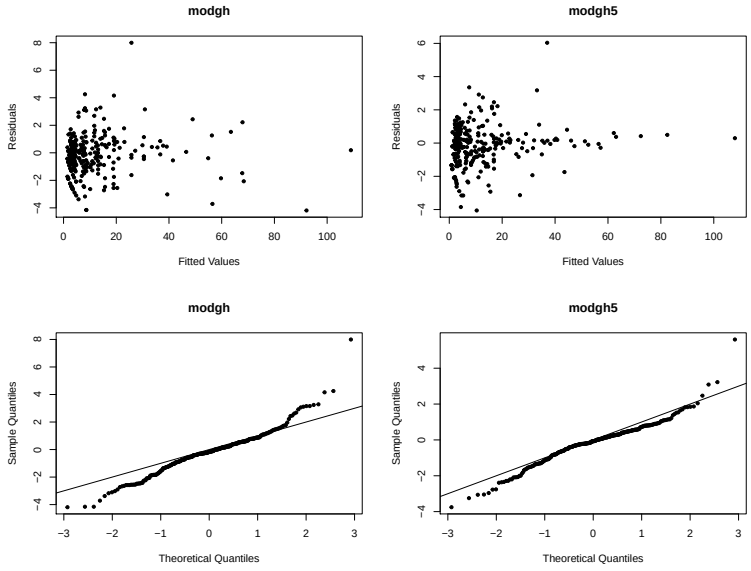

Figure 1: Residual plots from two random effect models

References

[1] P.J. Diggle, P. Heagerty, K.-Y. Liang, and S.L. Zeger. *Analysis of Longitudinal Data, Second Edition*. Oxford University Press, Oxford, 2002.

[2] J.J. Faraway. *Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models. Second Edition*. Chapman and Hall/CRC Press, Boca Raton, Florida, 2016.