Chapter 10

Random Effects

Grouped data arise in almost all areas of statistical application. Sometimes the grouping structure is simple, where each case belongs to a single group and there is only one grouping factor. More complex datasets have a hierarchical or nested structure or include longitudinal or spatial elements. Sometimes the grouping arises because the same individual is measured repeatedly or sometimes each individual is measured once only but these individuals have some form of group structure. We defer examination of the repeated measurement of individuals to the next chapter, although the statistical methodology used is the same.

All such data share the common feature of correlation of observations within the same group and so analyses that assume independence of the observations will be inappropriate. The use of random effects is one common and convenient way to model such grouping structure.

A *fixed effect* is an unknown constant that we try to estimate from the data. Almost all the parameters used in the linear and generalized linear models we have presented earlier in this book are fixed effects. In contrast, a *random effect* is a random variable. It does not make sense to estimate a random effect; instead, we try to estimate the parameters that describe the distribution of this random effect.

Consider an experiment to investigate the effect of several drug treatments on a sample of patients. Typically, we are interested in specific drug treatments and so we would treat the drug effects as fixed. However, it makes most sense to treat the patient effects as random. It is often reasonable to treat the patients as being randomly selected from a larger collection of patients whose characteristics we would like to estimate. Furthermore, we are not particularly interested in these specific patients, but in the whole population of patients. A random effects approach to modeling effects is more ambitious in the sense that it attempts to say something about the wider population beyond the particular sample. Blocking factors can often be viewed as random effects, because these often arise as a random sample of those blocks potentially available.

There is some judgment required in deciding when to use fixed and when to use random effects. Sometimes the choice is clear, but in other cases, reasonable statisticians may differ. In some analyses, random effects are used simply to induce a certain correlation structure in the data and there is sense in which the chosen levels represent a sample from a population. Gelman (2005) remarks on the variety of definitions for random effects and proposes a particular straightforward solution to the dilemma of whether to use fixed or random effects — he recommends always using random effects.

A *mixed effects* model has both fixed and random effects. A simple example of such a model would be a two-way analysis of variance (ANOVA):

$$y_{ijk} = \mu + \tau_i + v_j + \varepsilon_{ijk}$$

where the μ and τ_i are fixed effects and the error ε_{ijk} and the random effects v_j are independent and identically distributed $N(0,\sigma^2)$ and $N(0,\sigma_v^2)$, respectively.

We would want to estimate the τ_i and test the hypothesis $H_0 : \tau_i = 0\ \forall i$ while we would estimate σ_v^2 and might test $H_0 : \sigma_v^2 = 0$. Notice the difference: we need to estimate and test several fixed effect parameters while we need only estimate and test a single random effect parameter.

In the following sections, we consider estimation and inference for mixed effects models and then illustrate the application to several common designs.

10.1 Estimation

This is not as simple as it was for fixed effects models, where least squares is an easily applied method with many good properties. Let's start with the simplest possible random effects model: a one-way ANOVA design with a factor at a levels:

$$y_{ij} = \mu + \alpha_i + \varepsilon_{ij} \qquad i = 1,\ldots,a \qquad j = 1,\ldots,n$$

where the αs and εs have mean zero, but variances σ_α^2 and σ_ε^2, respectively. These variances are known as the variance components. Notice that this induces a correlation between observations at the same level equal to:

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

This is known as the *intraclass correlation coefficient* (ICC). In the limiting case when there is no variation between the levels, $\sigma_\alpha = 0$ so then $\rho = 0$. Alternatively, when the variation between the levels is much larger than that within the levels, the value of ρ will approach 1. This illustrates how random effects generate correlations between observations.

For simplicity, let there be an equal number of observations n per level. We can decompose the variation as follows (where dot in the subscript indicates the average over that index):

$$\sum_{i=1}^{a}\sum_{j=1}^{n}(y_{ij} - \bar{y}_{\cdot\cdot})^2 = \sum_{i=1}^{a}\sum_{j=1}^{n}(y_{ij} - \bar{y}_{i\cdot})^2 + \sum_{i=1}^{a}\sum_{j=1}^{n}(\bar{y}_{i\cdot} - \bar{y}_{\cdot\cdot})^2$$

or $SST = SSE + SSA$, respectively. SSE is the residual sum of squares, SST is the total sum of squares (corrected for the mean) and SSA is the sum of squares due to α. These quantities are often displayed in an ANOVA table along with the degrees of freedom associated with each sum of squares. Dividing through by the respective degrees of freedom, we obtain the mean squares, MSE and MSA. Now we find that:

$$E(SSE) = a(n-1)\sigma_\varepsilon^2, \qquad E(SSA) = (a-1)(n\sigma_\alpha^2 + \sigma_\varepsilon^2)$$

which suggests using the estimators:

$$\hat{\sigma}^2_\varepsilon = SSE/(a(n-1)) = MSE, \qquad \hat{\sigma}^2_\alpha = \frac{SSA/(a-1) - \hat{\sigma}^2_\varepsilon}{n} = \frac{MSA - MSE}{n}$$

This method of estimating variance components can be used for more complex designs. The ANOVA table is constructed, the expected mean squares calculated and the variance components obtained by solving the resulting equations. These estimators are known as *ANOVA estimators*. These were the first estimators developed for variance components. They have the advantage of taking explicit forms suitable for hand calculation which was important in precomputing days, but they have a number of disadvantages:

1. The estimates can take negative values. For example, in our situation above, if $MSA < MSE$ then $\hat{\sigma}^2_\alpha < 0$. This is rather embarrassing since variances cannot be negative. Various fixes have been proposed, but these all take away from the original simplicity of the estimation method.
2. A balanced design has an equal number of observations per cell, where cell is defined as the finest subdivision of the data according to the factors. In such circumstances, the ANOVA decomposition into sums of squares is unique. For unbalanced data, this is not true and we must choose which ANOVA decomposition to use, which will in turn affect the estimation of the variance components. Various rules have been suggested about how the decomposition should be done, but none of these have universal appeal.
3. The need for complicated algebraic calculations. Formulae for the simpler models are easy to find and coded in software. More complex models will require difficult and opaque constructions.

We would like a method that would avoid negative variances, work unambiguously for unbalanced data and that can be applied in a transparent and straightforward manner. Maximum likelihood (ML) estimation satisfies these requirements. This does require that we assume some distribution for the errors and the random effects. The usual assumption is normality; ML would work for other distributions, but these are rarely considered in this context.

For a fixed effect model with normal errors, we can write:

$$y = X\beta + \varepsilon \qquad \text{or} \qquad y \sim N(X\beta, \sigma^2 I)$$

where X is an $n \times p$ model matrix and β is a vector of length p. We can generalize to a mixed effect model with a vector γ of q random effects with associated model matrix Z which has dimension $n \times q$. Then we can model the response y, given the value of the random effects as:

$$y = X\beta + Z\gamma + \varepsilon \qquad \text{or} \qquad y|\gamma \sim N(X\beta + Z\gamma, \sigma^2 I)$$

If we further assume that the random effects $\gamma \sim N(0, \sigma^2 D)$ then $\text{var}\, y = \text{var}\, Z\gamma + \text{var}\, \varepsilon = \sigma^2 ZDZ^T + \sigma^2 I$ and we can write the unconditional distribution of y as:

$$y \sim N(X\beta, \sigma^2(I + ZDZ^T))$$

If we knew D, then we could estimate β using generalized least squares; see, for example, Chapter 8 in Faraway (2014). However, the estimation of the variance components, D, is often one purpose of the analysis. Standard maximum likelihood is one method of estimation that can be used here. If we let $V = I + ZDZ^T$, then the joint density for the response is:

$$\frac{1}{2\pi^{n/2}|\sigma^2 V|^{1/2}} e^{-\frac{1}{2\sigma^2}(y-X\beta)^T V^{-1}(y-X\beta)}$$

so that the log-likelihood for the data is:

$$l(\beta,\sigma,D|y) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log|\sigma^2 V| - \frac{1}{2\sigma^2}(y-X\beta)^T V^{-1}(y-X\beta)$$

This can be optimized to find maximum likelihood estimates of β, σ^2 and D. This is straightforward in principle, but there may be difficulties in practice. More complex models involving larger numbers of random effects parameters can be difficult to estimate. Sometimes the MLE of a variance parameter may be zero which occurs on the boundary of its domain. The derivative of the likelihood may not be zero in this boundary state which causes problems for many optimization methods.

Standard errors can be obtained using the usual large sample theory for maximum likelihood estimates. The variance can be estimated using the inverse of the negative second derivative of the log-likelihood evaluated at the MLE.

MLEs have some drawbacks. One particular problem is that they are biased. For example, consider an i.i.d. sample of normal data $x_1, \ldots, x_n$, then the MLE is:

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}$$

A denominator of $n-1$ is needed for an unbiased estimator. Similar problems occur with the estimation of variance components. Given that the number of levels of a factor may not be large, the bias of the MLE of the variance component associated with that factor may be quite large. *Restricted maximum likelihood* (REML) estimators are an attempt to get round this problem. The idea is to find all independent linear combinations of the response, k, such that $k^T X = 0$. Form matrix K with columns k, so that:

$$K^T y \sim N(0, K^T V K)$$

We can then proceed to maximize the likelihood based on $K^T y$ which does not involve any of the fixed effect parameters. Once the random effect parameters have been estimated, it is simple enough to obtain the fixed effect parameter estimates. REML generally produces less biased estimates. For balanced data, the REML estimates are usually the same as the ANOVA estimates.

We illustrate the fitting methods using some data from an experiment to test the paper brightness depending on a shift operator described in Sheldon (1960). We start with a fixed effects one-way ANOVA:

```
data(pulp, package="faraway")
op <- options(contrasts=c("contr.sum", "contr.poly"))
lmod <- aov(bright ~ operator, pulp)
summary(lmod)
```

```
            Df Sum Sq Mean Sq F value Pr(>F)
operator     3  1.340   0.447     4.2  0.023
Residuals   16  1.700   0.106
```

We can plot the data as seen in Figure 10.1.

```
library(ggplot2)
ggplot(pulp, aes(x=operator, y=bright))+geom_point(position = position
    ↪ _jitter(width=0.1, height=0.0))
```

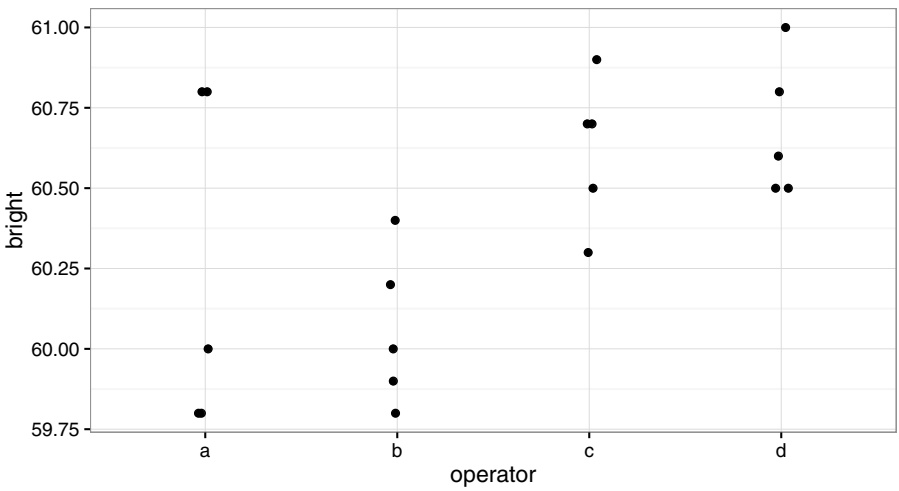

Figure 10.1 *Paper brightness varying by operator. Some jittering has been used to make coincident points apparent.*

```
coef(lmod)
```

```
(Intercept)   operator1   operator2   operator3
      60.40       -0.16       -0.34        0.22
```

```
options(op)
```

We have specified sum contrasts here instead of the default treatment contrasts to make the later connection to the corresponding random effects clearer. The `aov` function is just a wrapper for the standard `lm` function that produces results more appropriate for ANOVA models. We see that the operator effect is significant with a p-value of 0.023. The estimate of σ^2 is 0.106 and the estimated overall mean is 60.4. For sum contrasts, $\sum \alpha_i = 0$, so we can calculate the effect for the fourth operator as $0.16 + 0.34 - 0.22 = 0.28$.

Turning to the random effects model, we can compute the variance of the operator effects, σ^2_α, using the formula above as:

```
(0.447-0.106)/5
```

```
[1] 0.0682
```

Now we demonstrate the maximum likelihood estimators. The original R package for fitting mixed effects models was `nlme` as described in Pinheiro and Bates (2000). Subsequently Bates (2005) introduced the package `lme4`. The syntax for these two packages is somewhat different. The `lme4` package is generally more capable especially for larger datasets. The estimates these two packages produce for smaller, simpler datasets as considered in this chapter will generally be the same. However,

there are some crucial differences in the approach to inference that we will discuss later. We use the `lme4` packages in preference to `nlme`:

```
library(lme4)
mmod <- lmer(bright ~ 1+(1|operator), pulp)
summary(mmod)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: bright ~ 1 + (1 | operator)
   Data: pulp

REML criterion at convergence: 18.6

Scaled residuals:
   Min     1Q Median     3Q    Max
-1.467 -0.759 -0.124  0.628  1.601

Random effects:
 Groups   Name        Variance Std.Dev.
 operator (Intercept) 0.0681   0.261
 Residual             0.1062   0.326
Number of obs: 20, groups:  operator, 4

Fixed effects:
            Estimate Std. Error t value
(Intercept)   60.400      0.149     404
```

The model has fixed and random effects components. The fixed effect here is just the intercept represented by the first `1` in the model formula. The random effect is represented by `(1|operator)` indicating that the data is grouped by `operator` and the `1` indicating that the random effect is constant within each group. The parentheses are necessary to ensure that expression is parsed in the correct order.

The default fitting method is REML. We see that this gives identical estimates to the ANOVA method above — $\hat{\sigma}^2 = 0.106$, $\hat{\sigma}^2_\alpha = 0.068$ and $\hat{\mu} = 60.4$. For unbalanced designs, the REML and ANOVA estimators are not necessarily identical. The standard deviations are simply the square roots of the variances and not estimates of the uncertainty in the variances.

As with the GLM summary output, we find it rather verbose and prefer our own abbreviated version (which is adapted from the `display()` function in the `arm` package of Gelman and Su (2013)):

```
sumary(mmod)
```

```
Fixed Effects:
coef.est  coef.se
   60.40     0.15

Random Effects:
 Groups   Name        Std.Dev.
 operator (Intercept) 0.26
 Residual             0.33
---
number of obs: 20, groups: operator, 4
AIC = 24.6, DIC = 14.4
deviance = 16.5
```

This output contains just the information we need. It is better to use standard devi-

ations rather than variances as the former are measured in the units of the response and so much easier to interpret.

The maximum likelihood estimates may also be computed:

```
smod <- lmer(bright ~ 1+(1|operator), pulp, REML=FALSE)
sumary(smod)
Fixed Effects:
coef.est  coef.se
   60.40     0.13

Random Effects:
 Groups   Name        Std.Dev.
 operator (Intercept) 0.21
 Residual             0.33
---
number of obs: 20, groups: operator, 4
AIC = 22.5, DIC = 16.5
deviance = 16.5
```

The between-subjects SD, 0.21, is smaller than with the REML method as the ML method biases the estimates towards zero. The fixed effects are unchanged.

10.2 Inference

Test Statistic: We follow a general procedure. Decide which component(s) of the model you wish to test. These can be fixed and/or random effects. Specify two models: a null H_0 which does not contain your specified component(s) and an alternative H_1 which does include your component(s). The other terms in the models must be the same. These other terms (usually) make a difference to the result and must be chosen with care.

Using standard likelihood theory, we may derive a test to compare two nested hypotheses, H_0 and H_1, by computing the likelihood ratio test statistic:

$$2(l(\hat{\beta}_1, \hat{\sigma}_1, \hat{D}_1|y) - l(\hat{\beta}_0, \hat{\sigma}_0, \hat{D}_0|y))$$

where $\hat{\beta}_0, \hat{\sigma}_0, \hat{D}_0$ are the MLEs of the parameters under the null hypothesis and $\hat{\beta}_1, \hat{\sigma}_1, \hat{D}_1$ are the MLEs of the parameters under the alternative hypothesis.

If you plan to use the likelihood ratio test to compare two nested models that differ only in their fixed effects, you cannot use the REML estimation method. The reason is that REML estimates the random effects by considering linear combinations of the data that remove the fixed effects. If these fixed effects are changed, the likelihoods of the two models will not be directly comparable. Use ordinary maximum likelihood in this situation if you also wish to use the likelihood ratio test.

Approximate Null Distribution: This test statistic is approximately chi-squared with degrees of freedom equal to the difference in the dimensions of the two parameters spaces (the difference in the number of parameters when the models are identifiable). Unfortunately, this test is not exact and also requires several assumptions — see a text such as Cox and Hinkley (1974) for more details. Serious problems can arise with this approximation.

One crucial assumption is that the parameters under the null are not on the boundary of the parameter space. Since we are often interested in testing hypotheses about

the random effects that take the form $H_0 : \hat{\sigma}^2 = 0$, this is a common problem which makes the asymptotic inference invalid. If you do use the χ^2 distribution with the usual degrees of freedom, then the test will tend to be conservative — the p-values will tend to be larger than they should be. This means that if you observe a significant effect using the χ^2 approximation, you can be fairly confident that it is actually significant. The p-values generated by the likelihood ratio test for fixed effects are also approximate and unfortunately tend to be too small, thereby sometimes overstating the importance of some effects.

Regrettably the p-value based on the χ^2 approximation can either be entirely or just somewhat wrong. Perhaps with sufficient data and favorable models, the approximation may be satisfactory but it is difficult to say exactly when such propitious conditions may arise. Hence the safest advice is to not use this approximation.

Expected mean squares: Another method of hypothesis testing is based on the sums of squares found in the ANOVA decompositions. These tests are sometimes more powerful than their likelihood ratio test equivalents. However, the correct derivation of these tests usually requires extensive tedious algebra that must be recalculated for each type of model. Furthermore, the tests cannot be used (at least without complex and unsatisfactory adjustments) when the experiment is unbalanced. This method only works for simple models and balanced data.

***F*-tests for fixed effects:** We might try to use the F-test used in standard linear models to perform hypothesis tests regarding the fixed effects. The F-statistic is based on residual sums of squares and degrees of freedom as described in Chapter 3 of Faraway (2014). This is the method used in the `nlme` package. In the standard linear model setting, provided the normality assumption is correct, the null distribution has an exact F-distribution. Unfortunately, problems arise in transferring this method to mixed effect models. Firstly, the definition of degrees of freedom becomes murky in the presence of random effect parameters. Secondly, the test statistic is not necessarily F-distributed.

For some simple models with balanced data, the F-test is correct but in other cases with more complex models or unbalanced data, the p-values can be substantially incorrect. It is difficult to specify exactly when this test may be relied upon. For this reason, the `lme4` now declines to state p-values. Furthermore, the t-statistics that one might generate to test or form a confidence interval for a single fixed effect parameter also rely on the same problematic approximations.

Strategies for inference: We have good test statistics in the likelihood ratio test (LRT) or F-statistic but as yet no universally reliable way to obtain a null distribution. One solution would be to ignore the possible problem and use either the `nlme` package or the `lmerTest` package (which restores the questionable p-values to `lme4`). In certain known simple models with balanced data, this will produce accurate results but it would be speculative to report such results in other situations without at least verifying the results using other methods. A number of alternatives exist.

The standard degrees of freedom for the F-statistic in mixed models are not always reliable. Various researchers have developed methods for adjusting these degrees of freedom. One popular method is due to Kenward and Roger (1997). We will illustrate the use of this method later in this chapter. Even if the adjustment is opti-

mal, there remains the problem that the null distribution may not be F. Furthermore, the method is relevant only for the testing of fixed effects.

We can use bootstrap methods to find more accurate p-values for the likelihood ratio test. The usual bootstrap approach is nonparametric in that no distribution is assumed. Since we are willing to assume normality for the errors and the random effects, we can use a technique called the *parametric bootstrap*. We generate data under the null model using the fitted parameter estimates. We compute the likelihood ratio statistic for this generated data. We repeat this many times and use this to judge the significance of the observed test statistic. This approach will be demonstrated below. The problem may also be addressed by using Bayesian methods to fit the models. We discuss these in Chapter 12.

Model Selection: For comparing larger numbers of models, it is unwise to take a testing-based approach to selection. The problems are similar to those encountered in model selection for standard linear models. When the number of models considered becomes more than a handful, the issue of multiple testing arises and p-values lose their normal meaning. Instead it is better to take a criterion-based approach to model selection. Although we can develop the ideas of model selection of linear models and extend them to linear mixed models, there are some important additional difficulties which means that this extension is not straightforward. Firstly, the dependent response means that effective sample size is less than the total number of cases. Secondly, we have two kinds of parameters, some for the fixed effects and some for the random effects. It is not clear how these two types of parameters should be counted together. Thirdly, most criteria are based on the likelihood which does not behave well at the boundary of the parameter space as can occur with variance parameters.

The Akaike Information Criterion (AIC) and its variations are the most popular model selection criterion. In the `lme4` package, AIC is defined as:

$$-2(\texttt{max log likelihood}) + 2p$$

where p is the total number of parameters. We can confidently use this criterion to compare models which differ only in their fixed effects, as the number of random effect parameters will be the same for all models considered. If we compare models where the random effects are also varied, then we must think more carefully about how to count the random effect parameters. This is problematic due to the aforementioned boundary problems.

Other criteria can be considered. The Bayes Information Criterion (BIC) replaces the $2p$ in the AIC with $p \log n$ and tends to prefer smaller models to the AIC. Another popular criterion used with mixed effect models is the Deviance Information Criterion (DIC) of Spiegelhalter et al. (2002). This criterion is more suited to the Bayesian models discussed in Chapter 12. For a discussion of model selection criteria, see Section A.3. For the specific application to linear mixed models, see Müller et al. (2013). For most of the examples considered in this chapter, there are only a few variables so we are able to rely on testing methods to choose between just a few models. We defer an example of using these methods to Section 10.10.

Example: Now let's demonstrate these inferential methods on the `pulp` data. The fixed effect analysis shows that the operator effects are statistically significant

with a p-value of 0.023. A random effects analysis using the expected mean squares approach yields exactly the same F-statistic for the one-way ANOVA. This method works exactly for such a simple model.

We can also employ the likelihood ratio approach to test the null hypothesis that the variance between the operators is zero. In the fixed effects model, we tested the hypothesis that the four operators had the same effect. In the mixed effect model where the operators are treated as random, the hypothesis that this variance is zero claims that there is no differences between operators in the population. This is a stronger claim than the fixed effect model hypothesis about just the four chosen operators.

We first fit the null model:

```
nullmod <- lm(bright ~ 1, pulp)
```

As there are no random effects in this model, we must use `lm`. For models of the same class, we could use `anova` to compute the LRT and its p-value. Here, we need to compute this directly:

```
lrtstat <- as.numeric(2*(logLik(smod)-logLik(nullmod)))
pvalue <- pchisq(lrtstat,1,lower=FALSE)
data.frame(lrtstat, pvalue)
```

```
  lrtstat  pvalue
1  2.5684 0.10902
```

The p-value is now well above the 5% significance level. We cannot say that this result is necessarily wrong, but the use of the χ^2 approximation does cause us to doubt the result.

We can use the parametric bootstrap approach to obtain a more accurate p-value. We need to estimate the probability, given that the null hypothesis is true, of observing an LRT of 2.5684 or greater. Under the null hypothesis, $y \sim N(\mu, \sigma^2)$. A simulation approach generates data under this model, fits the null and alternative models and computes the LRT statistic. The process is repeated a large number of times and the proportion of LRT statistics exceeding the observed value of 2.5684 is used to estimate the p-value. In practice, we do not know the true values of μ and σ, but we can use the estimated values; this distinguishes the parametric bootstrap from the purely simulation approach. The `simulate` function makes it simple to generate a sample from a model:

```
y <- simulate(nullmod)
```

Now taking the data we generate, we fit both the null and alternative models and then compute the LRT. We repeat the process 1000 times:

```
lrstat <- numeric(1000)
set.seed(123)
for(i in 1:1000){
   y <- unlist(simulate(nullmod))
   bnull <- lm(y ~ 1)
   balt <- lmer(y ~ 1 + (1|operator), pulp, REML=FALSE)
   lrstat[i] <- as.numeric(2*(logLik(balt)-logLik(bnull)))
  }
```

We have set the random number seed here so that the results will reproduce exactly if you run the same code. You do not need to set a seed for your own data unless you need to achieve the same reproducibility. Be aware that simulation naturally contains

some variation. If this variation might make a difference to your conclusions, you need to use a larger number of bootstrap samples.

We may examine the distribution of the bootstrapped LRTs. We compute the proportion that are close to zero:

```
mean(lrstat < 0.00001)
```
```
[1] 0.703
```

We see there is a 70% chance that the likelihoods for the null and alternatives are virtually identical giving an LRT statistic of practically zero. The LRT clearly does not have a χ^2 distribution. There is some discussion of this matter in Stram and Lee (1994), who propose a 50:50 mixture of a χ^2 and a mass at zero. Unfortunately, as we can see, the relative proportions of these two components vary from case to case. Crainiceanu and Ruppert (2004) give a more complete solution to the one-way ANOVA problem, but there is no general and exact result for this and more complex problems. The parametric bootstrap may be the simplest approach. The method we have used above is transparent and could be computed much more efficiently if speed is an issue.

Our estimated *p*-value is:

```
mean(lrstat > 2.5684)
```
```
[1] 0.019
```

We can compute the standard error for this estimate by:

```
sqrt(0.019*0.981/1000)
```
```
[1] 0.0043173
```

So we can be fairly sure it is under 5%. If in doubt, do some more replications to make sure; this only costs computer time. As it happens, this *p*-value is close to the fixed effects *p*-value.

The `RLRsim` package of Scheipl et al. (2008) can be used to test random effect terms:

```
library(RLRsim)
exactLRT(smod, nullmod)
```
```
No restrictions on fixed effects. REML-based inference preferable.

        simulated finite sample distribution of LRT. (p-value based
        on 10000 simulated values)

data:
LRT = 2.5684, p-value = 0.0213
```

The result is obtained with less computing time than our explicitly worked example. The difference in the outcomes is within the sampling error. As the output points out, it is slightly better to use REML when testing the random effects (although remember that REML would be invalid for testing fixed effects). We can make this computation:

```
exactRLRT(mmod)
```
```
        simulated finite sample distribution of RLRT.

        (p-value based on 10000 simulated values)

data:
RLRT = 3.4701, p-value = 0.021
```

Notice that the testing function is now `exactRLRT` and that only the alternative model needs to be specified as there is only one random effect component. The outcome is very similar to those obtained previously.

The parametric bootstrap can also be used to construct confidence intervals for the parameters. We simulate data from the chosen model and estimate the parameters. We repeat this process many times, storing the results each time. Quantiles of the bootstrapped estimates are then used to compute the intervals. We need to be able to extract the parameter estimates from the model. We can view the estimates of variance parameters using:

```
VarCorr(mmod)
 Groups   Name        Std.Dev.
 operator (Intercept) 0.261
 Residual             0.326
```

A more convenient form for extracting the values can be obtained as:

```
as.data.frame(VarCorr(mmod))
       grp        var1 var2     vcov   sdcor
1 operator (Intercept) <NA> 0.068083 0.26093
2 Residual        <NA> <NA> 0.106250 0.32596
```

Now we are ready to bootstrap:

```
bsd <- numeric(1000)
for(i in 1:1000){
  y <- unlist(simulate(mmod))
  bmod <- refit(mmod, y)
  bsd[i] <- as.data.frame(VarCorr(bmod))$sdcor[1]
}
```

The `refit` function changes only the response in a model we have already fit. This is significantly faster than fitting the model from scratch as the overhead in setting up the model is avoided. The 95% bootstrap confidence interval for σ_α is:

```
quantile(bsd, c(0.025, 0.975))
   2.5%   97.5%
0.00000 0.51335
```

Essentially the same result can be obtained more directly using the `confint` function:

```
confint(mmod, method="boot")
Computing bootstrap confidence intervals ...
                             2.5 %   97.5 %
sd_(Intercept)|operator  0.00000  0.51539
sigma                    0.21347  0.45522
(Intercept)             60.09417 60.69724
```

Nevertheless, it is worth understanding the detailed method of construction to know how it works and to allow one to modify the method if circumstances require it.

In this case, the lower bound is zero. This is not surprising given our earlier uncertainty over whether there really is a difference between the operators. In simpler circumstances, there is a duality between confidence intervals and hypothesis tests in that the outcome of a test can be determined by whether the point null hypothesis lies within the confidence interval. Unfortunately, this duality does not apply in all circumstances, this being a case in point. If you want to do a hypothesis test, use the method described earlier and not the confidence interval.

In this example, the random and fixed effect tests gave similar outcomes. However, the hypotheses in random and fixed effects are intrinsically different. To generalize somewhat, it is easier to conclude there is an effect in a fixed effects model since the conclusion applies only to the levels of the factor used in the experiment, while for random effects, the conclusion extends to levels of the factor not considered.

Since the range of the random effect conclusions is greater, the evidence necessarily has to be stronger.

10.3 Estimating Random Effects

In a fixed effects model, the effects are represented by parameters and it makes sense to estimate them. For example, in the one-way ANOVA model:

$$y_{ij} = \mu + \alpha_i + \varepsilon_{ij}$$

We can calculate $\hat{\alpha}_i$. We do need to resolve the identifiability problem with the αs and the μ, but once we decide on this, the meaning of the $\hat{\alpha}$s is clear enough. We can then proceed to make further inference such as multiple comparisons of these levels.

In a model with random effects, the αs are no longer parameters, but random variables. Using the standard normal assumption:

$$\alpha_i \sim N(0, \sigma_\alpha^2)$$

It does not make sense to estimate the αs because they are random variables. So instead, we might think about the expected values. However:

$$E\alpha_i = 0 \qquad \forall i$$

which is clearly not very interesting. If one looks at this from a Bayesian point of view, as described in, for example, Gelman et al. (2013), we have a prior density on the αs. The prior mean is $E\alpha_i = 0$. Let f represent density, then the posterior density for α is given by:

$$f(\alpha_i|y) \propto f(y|\alpha_i) f(\alpha_i)$$

We can then find the posterior mean, denoted by $\hat{\alpha}$ as:

$$E(\alpha_i|y) = \int \alpha_i f(\alpha_i|y) d\alpha_i$$

For the general case, this works out to be:

$$\hat{\alpha} = DZ^T V^{-1}(y - X\beta)$$

Now a purely Bayesian approach would specify the parameters of the prior (or specify priors for these) and compute a posterior distribution for α. Here we take an empirical Bayes point of view and substitute the MLEs into D, V and β to obtain the predicted random effects. These may be computed as:

```
ranef(mmod)$operator
```

```
  (Intercept)
a    -0.12194
b    -0.25912
c     0.16767
d     0.21340
```

The predicted random effects are related to the fixed effects. These fixed effects are:

```
(cc <- model.tables(lmod))
```

```
Tables of effects

 operator
operator
    a     b     c     d
-0.16 -0.34  0.22  0.28
```

Let's compute the ratio to the random effects as:

```
cc[[1]]$operator/ranef(mmod)$operator
  X.Intercept.
a       1.3121
b       1.3121
c       1.3121
d       1.3121
```

We see that the predicted random effects are exactly in proportion to the fixed effects. Typically, the predicted random effects are smaller and could be viewed as a type of *shrinkage* estimate.

The 95% confidence intervals for the random effects can be calculated and displayed as seen in Figure 10.2.

```
library(lattice)
dotplot(ranef(mmod, condVar=TRUE))
```

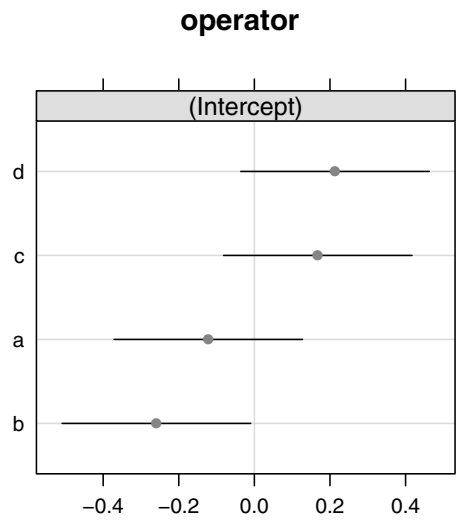

Figure 10.2 *Confidence intervals for the random effects in the pulp data.*

10.4 Prediction

Suppose we wish to predict a new value. If the prediction is to be made for a new operator or unknown operator, the best we can do is give $\hat{\mu} = 60.4$. If we know the operator, then we can combine this with our fixed effects to produce what are known as the *best linear unbiased predictors* (BLUPs) as follows:

```
fixef(mmod)+ranef(mmod)$operator
  (Intercept)
a      60.278
b      60.141
c      60.568
d      60.613
```

We can also use the `predict` function to construct these predictions. First consider the prediction for a new or unknown operator. We specify the random effects part of the prediction as `~0` meaning that this term is not present. In more complex models with more than one random effect, more choices are available. By default this would produce a fitted value for each case in the data but since these are identical we take only the first value:

```
predict(mmod, re.form=~0)[1]
   1
60.4
```

Now we specify that the operator is 'a':

```
predict(mmod, newdata=data.frame(operator="a"))
     1
60.278
```

The `predict` function for mixed model objects does not compute standard errors or prediction intervals. For this simple model, it would be possible to compute these explicitly but for more general models, it becomes much more difficult. For this reason, we present a parametric bootstrap method for computing these as it is clearer how the bands are computed. We start with the unknown operator case:

```
group.sd <- as.data.frame(VarCorr(mmod))$sdcor[1]
resid.sd <- as.data.frame(VarCorr(mmod))$sdcor[2]
pv <- numeric(1000)
for(i in 1:1000){
 y <- unlist(simulate(mmod))
 bmod <- refit(mmod, y)
 pv[i] <- predict(bmod, re.form=~0)[1] + rnorm(n=1,sd=group.sd) +
     ↪ rnorm(n=1,sd=resid.sd)
}
quantile(pv, c(0.025, 0.975))
  2.5%  97.5%
59.535 61.286
```

As in previous bootstraps, the first step is to simulate from the fitted model. We refit the model with the simulated response and generate a predicted value. But there are two additional sources of variation. We have variation due to the new operator and also due to a new observation from that operator. For this reason, we add normal sample values with standard deviations equal to those estimated earlier. If you really want a confidence interval for the mean prediction, you should not add these extra error terms. We repeat this 1000 times and take the appropriate quantiles to get a 95% interval.

Some modification is necessary if we know the operator we are making the prediction interval for. We use the option `use.u=TRUE` in the `simulate` function indicating that we should simulate new values conditional on the estimated random effects. We need to do this because otherwise we would simulate an entirely new 'a' effect in each replication. Instead, we want to preserve the originally generated 'a' effect.

```
for(i in 1:1000){
 y <- unlist(simulate(mmod, use.u=TRUE))
 bmod <- refit(mmod, y)
 pv[i] <- predict(bmod, newdata=data.frame(operator="a")) + rnorm(n=1,
     ↪ sd=resid.sd)
}
quantile(pv, c(0.025, 0.975))
```

```
  2.5%  97.5%
59.606 61.023
```

In a simple model such as this, we could mathematically calculate the standard error formulas and use this to compute these intervals more efficiently. However, the bootstrap is more general and is easier to apply in more complex situations. More bootstrapping functionality can be found in the `lme4::bootMer()` function and also in the `merTools` package. Bootstrapping is fast enough for simple models but greater efficiency is needed in more complex cases.

10.5 Diagnostics

It is important to check the assumptions made in fitting the model. Diagnostic methods available for checking linear mixed models largely mirror those used for linear models but there are some variations. Residuals are commonly defined as the difference between the observed and fitted values. In mixed models, there is more than one kind of fitted (or predicted) value resulting in more than one kind of residual. The default predicted values and residuals use the estimated random effects. This means these residuals can be regarded as estimates of ε which is usually what we want.

As with linear models, this pair of diagnostics plots is most valuable:

```
qqnorm(residuals(mmod),main="")
plot(fitted(mmod),residuals(mmod),xlab="Fitted",ylab="Residuals")
abline(h=0)
```

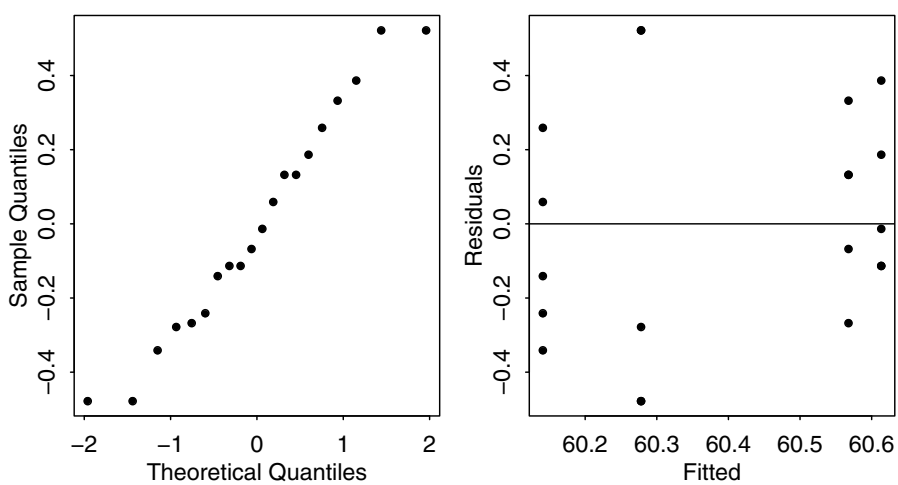

Figure 10.3 *Diagnostic plots for the one-way random effects model.*

The plots are shown in Figure 10.3 and indicate no particular problems. Random effects models are particularly sensitive to outliers, because they depend on variance components that can be substantially inflated by unusual points. The QQ plot is one way to pick out outliers. We also need the normality for the testing. The residual-fitted plot is also important because we made the assumption that the error variance was constant.

If we had more than four groups, we could also look at the normality of the group level effects and check for constant variance also. With so few groups, it is not sensible to do this. Also note that there is no particular reason to think about multiple comparisons. These are for comparing selected levels of a factor. For a random effect, the levels were randomly selected, so such comparisons have less motivation.

10.6 Blocks as Random Effects

Blocks are properties of the experimental units. The blocks are either clearly defined by the conditions of the experiment or they are formed with the judgment of the experimenter. Sometimes, blocks represent groups of runs completed in the same period of time. Typically, we are not interested in the block effects specifically, but must account for their effect. It is therefore natural to treat blocks as random effects.

We illustrate with an experiment to compare four processes, A, B, C and D, for the production of penicillin. These are the treatments. The raw material, corn steep liquor, is quite variable and can only be made in blends sufficient for four runs. Thus a randomized complete block design is suggested by the nature of the experimental units. The data comes from Box et al. (1978). We start with the fixed effects analysis:

```
data(penicillin, package="faraway")
summary(penicillin)
```

```
 treat    blend       yield
 A:5   Blend1:4   Min.   :77
 B:5   Blend2:4   1st Qu.:81
 C:5   Blend3:4   Median :87
 D:5   Blend4:4   Mean   :86
       Blend5:4   3rd Qu.:89
                  Max.   :97
```

We plot the data as seen in Figure 10.4. We create a version of the blend variable to get neater labeling.

```
penicillin$Blend <- gl(5,4)
ggplot(penicillin, aes(y=yield, x=treat, shape=Blend))+geom_point()+
    ↪ xlab("Treatment")
ggplot(penicillin, aes(y=yield, x=Blend, shape=treat))+geom_point()
```

It is convenient to use sum contrasts rather than the default treatment contrasts for the purpose of comparison to the mixed effect modeling to come.

```
op <- options(contrasts=c("contr.sum", "contr.poly"))
lmod <- aov(yield ~ blend + treat, penicillin)
summary(lmod)
```

```
            Df Sum Sq Mean Sq F value Pr(>F)
blend        4  264.0    66.0    3.50  0.041
treat        3   70.0    23.3    1.24  0.339
Residuals   12  226.0    18.8
```

```
coef(lmod)
```

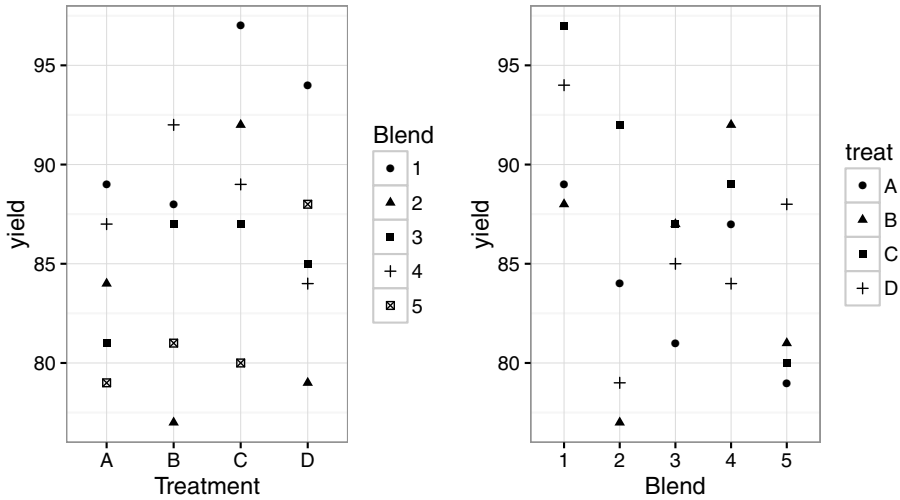

Figure 10.4 *Yield from penicillin blends varying by treatment.*

```
(Intercept)      blend1      blend2      blend3      blend4
         86           6          -3          -1           2
     treat1      treat2      treat3
         -2          -1           3
```

From this we see that there is no significant difference between the treatments, but there is between the blends. Now let's fit the data with a mixed model, where we have fixed treatment effects, but random blend effects. This seems natural since the blends we use can be viewed as having been selected from some notional population of blends.

```
mmod <- lmer(yield ~ treat + (1|blend), penicillin)
sumary(mmod)
Fixed Effects:
            coef.est coef.se
(Intercept) 86.00     1.82
treat1      -2.00     1.68
treat2      -1.00     1.68
treat3       3.00     1.68

Random Effects:
 Groups   Name        Std.Dev.
 blend    (Intercept) 3.43
 Residual             4.34
---
number of obs: 20, groups: blend, 5
AIC = 118.6, DIC = 128
deviance = 117.3
options(op)
```

We notice a few connections. The residual variance is the same in both cases: $18.8 = 4.34^2$. This is because we have a balanced design and so REML is equivalent to the

ANOVA estimator. The treatment effects are also the same as is the overall mean. The BLUPs for the random effects are:

```
ranef(mmod)$blend
       (Intercept)
Blend1     4.28788
Blend2    -2.14394
Blend3    -0.71465
Blend4     1.42929
Blend5    -2.85859
```

which, as with the one-way ANOVA, are a shrunken version of the corresponding fixed effects. The usual diagnostics show nothing amiss.

We have a number of options in testing the fixed effects in this example. For this simple balanced model, the `aov` function can be used:

```
amod <- aov(yield ~ treat + Error(blend), penicillin)
summary(amod)
Error: blend
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals  4    264      66

Error: Within
          Df Sum Sq Mean Sq F value Pr(>F)
treat      3     70    23.3    1.24   0.34
Residuals 12    226    18.8
```

Notice how the random effects term for `blend` is specified. The p-value for testing the treatment effects is 0.34 indicating no significant effect. This test is exact and works well here but only works for simple balanced models. For example, a single missing value would invalidate this test so we have good reason to explore more general methods.

We might try to base a test on the F-statistic which can be obtained like this:

```
anova(mmod)
Analysis of Variance Table
      Df Sum Sq Mean Sq F value
treat  3     70    23.3    1.24
```

For this simple balanced case, it can be shown that this F-statistic has a true F null distribution with usual degrees of freedom from which we could derive a p-value. Unfortunately, as with the `aov` method, this result does not generalize well.

More reliable F-tests can be achieved by using adjusted degrees of freedom. The `pbkrtest` package (Halekoh and Højsgaard (2014)) implements the Kenward-Roger (Kenward and Roger (1997)) method:

```
library(pbkrtest)
amod <- lmer(yield ~ treat + (1|blend), penicillin, REML=FALSE)
nmod <- lmer(yield ~ 1 + (1|blend), penicillin, REML=FALSE)
KRmodcomp(amod, nmod)
F-test with Kenward-Roger approximation; computing time: 0.14 sec.
large : yield ~ treat + (1 | blend)
small : yield ~ 1 + (1 | blend)
       stat   ndf   ddf F.scaling p.value
Ftest  1.24  3.00 12.00         1    0.34
```

It is essential to use the ML method of estimation when testing fixed effects. Since we wish to test the treatment effect, we fit the model with this term and the same model but without this term. As can be seen, it produces an identical result to the `aov`

output with the same degrees of freedom and p-value. The advantage of this method is that it can be generalized to a much wider class of problems.

We can also use the parametric bootstrap. First we compute the LRT statistic:

```
as.numeric(2*(logLik(amod)-logLik(nmod)))
```

```
[1] 4.0474
```

Just for reference, we could use the χ^2 approximation to quickly compute a p-value:

```
1-pchisq(4.0474,3)
```

```
[1] 0.25639
```

This is just an approximation of unknown quality. We aim to do better than this.

We can improve the accuracy with the parametric bootstrap approach. We can generate a response from the null model and use this to compute the LRT. We repeat this 1000 times, saving the LRT each time:

```
lrstat <- numeric(1000)
for(i in 1:1000){
ryield <- unlist(simulate(nmod))
nmodr <- refit(nmod, ryield)
amodr <- refit(amod, ryield)
lrstat[i] <- 2*(logLik(amodr)-logLik(nmodr))
}
```

Notice how we have used `refit` to speed up the computation. Under the standard likelihood theory, the LRT statistic here should have a χ^2_3 distribution. A QQ plot of these simulated LRT values indicates that this is a poor approximation. We can compute our estimated p-value as:

```
mean(lrstat > 4.0474)
```

```
[1] 0.353
```

which is much closer to the F-test result than the χ^2_3-based approximation.

The `pbkrtest` package offers a convenient way to perform the parametric bootstrap for fixed effect terms:

```
pmod <- PBmodcomp(amod, nmod)
summary(pmod)
```

```
Parametric bootstrap test; time: 32.22 sec; samples: 1000 extremes: 333;
large : yield ~ treat + (1 | blend)
small : yield ~ 1 + (1 | blend)
         stat   df  ddf p.value
PBtest   4.05              0.33
Gamma    4.05              0.33
Bartlett 3.42 3.00         0.33
F        1.35 3.00 12.9    0.30
LRT      4.05 3.00         0.26
```

The parametric bootstrap p-value is 0.33, which is similar to our previous results. Remember that bootstrap is based on random sampling so if you repeat this, you will get slightly different results. Since this p-value is not close to significance, we have no worries about this. Notice that the output also produces the χ^2-based LRT result along with three other versions that are explained in the documentation for the `pbkrtest` package. The package also offers the possibility of using the multiple cores available now on most computers. This parallel computing can be helpful as the parametric bootstrap is computationally expensive.

We can also test the significance of the blends. As with a fixed effects analysis, we are typically not directly interested in size of the blocking effects. Once having decided to design the experiment with blocks, we must retain them in the model.

However, we may wish to examine the blocking effects for information useful for the design of future experiments. We can fit the model with and without random effects and compute the LRT:

```
rmod <- lmer(yield ~ treat + (1|blend), penicillin)
nlmod <- lm(yield ~ treat, penicillin)
as.numeric(2*(logLik(rmod)-logLik(nlmod,REML=TRUE)))
```

```
[1] 2.7629
```

We need to specify the nondefault REML option for null model to ensure that the LRT is computed correctly. Now we perform the parametric bootstrap much as before:

```
lrstatf <- numeric(1000)
for(i in 1:1000){
   ryield <-  unlist(simulate(nlmod))
   nlmodr <- lm(ryield ~ treat, penicillin)
   rmodr <- refit(rmod, ryield)
   lrstatf[i] <- 2*(logLik(rmodr)-logLik(nlmodr,REML=TRUE))
  }
```

Again, the distribution is far from χ^2_1 which is clear when we examine the proportion of generated LRTs which are close to zero:

```
mean(lrstatf < 0.00001)
```

```
[1] 0.551
```

We can see from this that the LRT is clearly not χ^2_1 distributed. Even the nonzero values seem to have some other distribution. This makes it clear that asymptotic approximations cannot be relied on these circumstances.

We can compute the estimated p-value as:

```
mean(lrstatf > 2.7629)
```

```
[1] 0.043
```

So we find a significant blend effect. The p-value is close to that observed for the fixed effects analysis. Given that the p-value is close to 5%, we might wish to increase the number of bootstrap samples to increase our confidence in the result.

We can also use `RLRsim` to obtain a p-value.

```
library(RLRsim)
exactRLRT(rmod)
```

```
        simulated finite sample distribution of RLRT.

        (p-value based on 10000 simulated values)

data:
RLRT = 2.7629, p-value = 0.0406
```

In this example, we saw no major advantage in modeling the blocks as random effects, so we might prefer to use the fixed effects analysis as it is simpler to execute. However, in subsequent analyses, we shall see that the use of random effects will be mandatory as equivalent results may not be obtained from a purely fixed effects analysis.

10.7 Split Plots

Split plot designs originated in agriculture, but occur frequently in other settings. As the name implies, main plots are split into several subplots. The main plot is treated with a level of one factor while the levels of some other factor are allowed to vary

with the subplots. The design arises as a result of restrictions on a full randomization. For example, a field may be divided into four subplots. It may be possible to plant different varieties in the subplots, but only one type of irrigation may be used for the whole field. Note the distinction between split plots and blocks. Blocks are features of the experimental units which we have the option to take advantage of in the experimental design. Split plots impose restrictions on what assignments of factors are possible. They impose requirements on the design that prevent a complete randomization. Split plots often arise in nonagricultural settings when one factor is easy to change while another factor takes much more time to change. If the experimenter must do all runs for each level of the hard-to-change factor consecutively, a split-plot design results with the hard-to-change factor representing the whole plot factor.

Consider the following example. In an agricultural field trial, the objective was to determine the effects of two crop varieties and four different irrigation methods. Eight fields were available, but only one type of irrigation may be applied to each field. The fields may be divided into two parts with a different variety planted in each half. The whole plot factor is the method of irrigation, which should be randomly assigned to the fields. Within each field, the variety is randomly assigned. Here is a summary of the data:

```
data(irrigation,package="faraway")
summary(irrigation)
```

```
     field   irrigation variety     yield
 f1     :2   i1:4       v1:8    Min.   :34.8
 f2     :2   i2:4       v2:8    1st Qu.:37.6
 f3     :2   i3:4               Median :40.1
 f4     :2   i4:4               Mean   :40.2
 f5     :2                      3rd Qu.:42.7
 f6     :2                      Max.   :47.6
 (Other):4
```

We can plot the data as seen in Figure 10.5.

```
ggplot(irrigation, aes(y=yield, x=field, shape=irrigation, color=
    ↪ variety)) + geom_point()
```

The irrigation and variety are fixed effects, but the field is clearly a random effect. We must also consider the interaction between field and variety, which is necessarily also a random effect because one of the two components is random. The fullest model that we might consider is:

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

$\mu, i_i, v_j, (iv)_{ij}$ are fixed effects; the rest are random having variances σ^2_f, σ^2_{vf} and σ^2_ε. Note that we have no $(if)_{ik}$ term in this model. It would not be possible to estimate such an effect since only one type of irrigation is used on a given field; the factors are not crossed. We would fit such a model using the expression

```
lmer(yield ~ irrigation * variety + (1|field) +(1|field:variety),
    ↪ irrigation)
```

However, if you try to fit such a model, it will fail because it is not possible to distinguish the variety within the field variation. We would need more than one observation per variety within each field for us to separate the two variabilities. We resort to a simpler model that omits the variety by field interaction random effect:

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

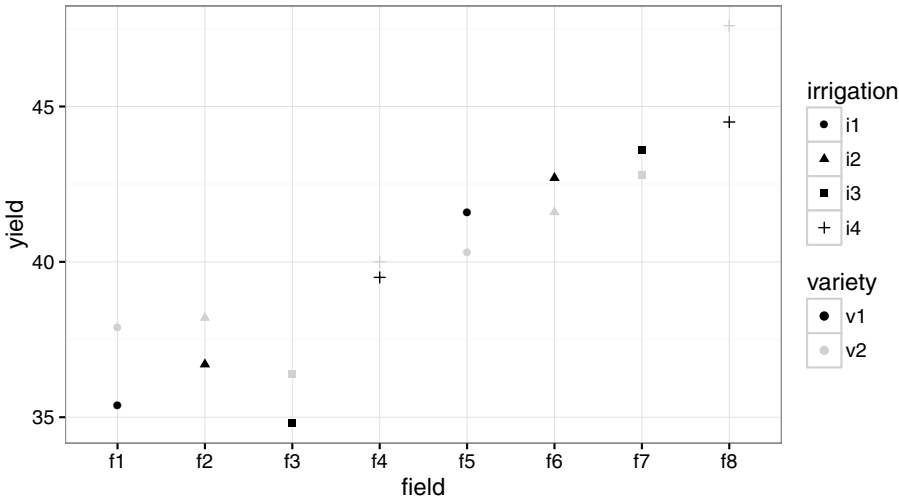

Figure 10.5 *Yield on fields with different irrigation methods.*

```
lmod <- lmer(yield ~ irrigation * variety + (1|field), irrigation)
sumary(lmod)
Fixed Effects:
                      coef.est coef.se
(Intercept)            38.50     3.03
irrigationi2            1.20     4.28
irrigationi3            0.70     4.28
irrigationi4            3.50     4.28
varietyv2               0.60     1.45
irrigationi2:varietyv2 -0.40     2.05
irrigationi3:varietyv2 -0.20     2.05
irrigationi4:varietyv2  1.20     2.05

Random Effects:
 Groups   Name        Std.Dev.
 field    (Intercept) 4.02
 Residual             1.45
---
number of obs: 16, groups: field, 8
AIC = 65.4, DIC = 91.8
deviance = 68.6
```

We can see that the largest variance component is that due to the field effect: $\hat{\sigma}_f = 4.02$ with $\hat{\sigma}_\varepsilon = 1.45$.

The relatively large standard errors compared to the fixed effect estimates suggest that there may be no significant fixed effects. We can check this sequentially using F-tests with adjusted degrees of freedom:

```
library(pbkrtest)
lmoda <- lmer(yield ~ irrigation + variety + (1|field),data=irrigation
    ↪ )
KRmodcomp(lmod, lmoda)
F-test with Kenward-Roger approximation; computing time: 0.07 sec.
```

```
large : yield ~ irrigation * variety + (1 | field)
small : yield ~ irrigation + variety + (1 | field)
      stat  ndf  ddf F.scaling p.value
Ftest 0.25 3.00 4.00         1    0.86
```

We find there is no significant interaction term. We can now test each of the main effects starting with the variety:

```
lmodi <- lmer(yield ~ irrigation + (1|field), irrigation)
KRmodcomp(lmoda, lmodi)
F-test with Kenward-Roger approximation; computing time: 0.06 sec.
large : yield ~ irrigation + variety + (1 | field)
small : yield ~ irrigation + (1 | field)
      stat  ndf  ddf F.scaling p.value
Ftest 1.58 1.00 7.00         1    0.25
```

Dropping variety from the model seems reasonable since the p-value of 0.25 is large. We can test irrigation in a similar manner:

```
lmodv <- lmer(yield ~  variety + (1|field), irrigation)
KRmodcomp(lmoda, lmodv)
F-test with Kenward-Roger approximation; computing time: 0.06 sec.
large : yield ~ irrigation + variety + (1 | field)
small : yield ~ variety + (1 | field)
      stat  ndf  ddf F.scaling p.value
Ftest 0.39 3.00 4.00         1    0.77
```

Irrigation also fails to be significant.

We should check the diagnostic plots to make sure there is nothing amiss:

```
plot(fitted(lmod),residuals(lmod),xlab="Fitted",ylab="Residuals")
qqnorm(residuals(lmod),main="")
```

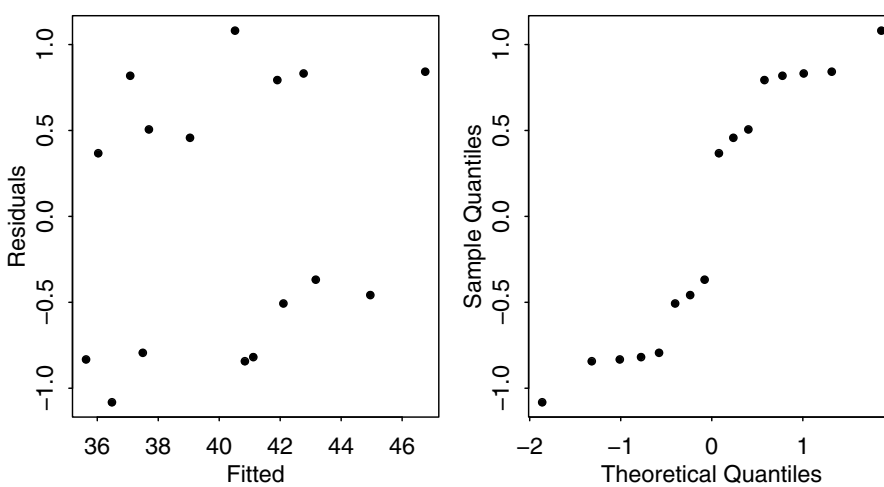

Figure 10.6 *Diagnostic plots for the split plot example.*

We can see in Figure 10.6 that there is no problem with the nonconstant variance, but that the residuals indicate a bimodal distribution caused by the pairs of observations in each field. This type of divergence from normality is unlikely to cause any major problems with the estimation and inference.

We can test the random effects term like this:

```
library(RLRsim)
exactRLRT(lmod)
```

```
        simulated finite sample distribution of RLRT.

        (p-value based on 10000 simulated values)

data:
RLRT = 6.1118, p-value = 0.0098
```

We see that the fields do seem to vary as the result is clearly significant.

Sometimes analysts ignore the split-plot variable as in:

```
mod <- lm(yield ~ irrigation * variety, data=irrigation)
anova(mod)
```

```
Analysis of Variance Table

Response: yield
                  Df Sum Sq Mean Sq F value Pr(>F)
irrigation         3   40.2    13.4    0.73   0.56
variety            1    2.2     2.2    0.12   0.73
irrigation:variety  3    1.6     0.5    0.03   0.99
Residuals          8  146.5    18.3
```

The results will not be the same. This last model is incorrect because it fails to take into account the restrictions on the randomization introduced by the fields and the additional variability thereby induced.

10.8 Nested Effects

When the levels of one factor vary only within the levels of another factor, that factor is said to be *nested*. For example, when measuring the performance of workers at several different job locations, if the workers only work at one location, the workers are nested within the locations. If the workers work at more than one location, then the workers are *crossed* with locations.

Here is an example to illustrate nesting. Consistency between laboratory tests is important and yet the results may depend on who did the test and where the test was performed. In an experiment to test levels of consistency, a large jar of dried egg powder was divided up into a number of samples. Because the powder was homogenized, the fat content of the samples is the same, but this fact is withheld from the laboratories. Four samples were sent to each of six laboratories. Two of the samples were labeled as G and two as H, although in fact they were identical. The laboratories were instructed to give two samples to two different technicians. The technicians were then instructed to divide their samples into two parts and measure the fat content of each. So each laboratory reported eight measures, each technician four measures, that is, two replicated measures on each of two samples. The data comes from Bliss (1967):

```
data(eggs, package="faraway")
summary(eggs)
```

```
      Fat         Lab    Technician Sample
 Min.   :0.060   I  :8   one:24     G:24
 1st Qu.:0.307   II :8   two:24     H:24
 Median :0.370   III:8
```

```
 Mean   :0.388   IV :8
 3rd Qu.:0.430   V  :8
 Max.   :0.800   VI :8
```

We can plot the data as seen in Figure 10.7.

```
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()
```

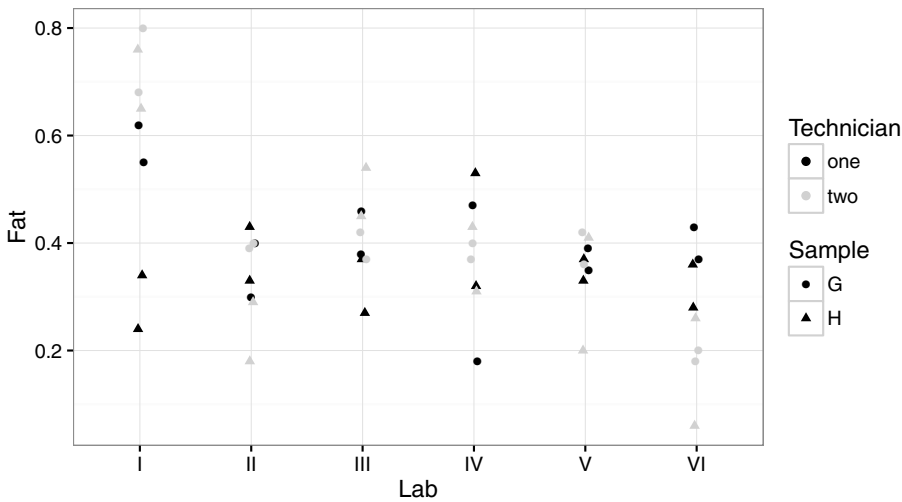

Figure 10.7 *Fat content of homogenous powdered egg as tested by different laboratories, technicians and samples.*

Although the technicians have been labeled "one" and "two," they are two different people in each lab. Thus the technician factor is nested within laboratories. Furthermore, even though the samples are labeled "H" and "G," these are not the same samples across the technicians and the laboratories. Hence we have samples nested within technicians. Technicians and samples should be treated as random effects since we may consider these as randomly sampled. If the labs were specifically selected, then they should be taken as fixed effects. If, however, they were randomly selected from those available, then they should be treated as random effects. If the purpose of the study is to come to some conclusion about consistency across laboratories, the latter approach is advisable.

For the purposes of this analysis, we will treat labs as random. So all our effects (except the grand mean) are random. The model is:

$$y_{ijkl} = \mu + L_i + T_{ij} + S_{ijk} + \varepsilon_{ijkl}$$

This can be fit using:

```
cmod <- lmer(Fat ~ 1 + (1|Lab) + (1|Lab:Technician) +  (1|Lab:
    ↪ Technician:Sample), data=eggs)
sumary(cmod)
```

```
Fixed Effects:
coef.est  coef.se
    0.39     0.04

Random Effects:
 Groups                Name        Std.Dev.
 Lab:Technician:Sample (Intercept) 0.06
 Lab:Technician        (Intercept) 0.08
 Lab                   (Intercept) 0.08
 Residual                          0.08
---
number of obs: 48, groups: Lab:Technician:Sample, 24; Lab:Technician, 12; Lab, 6
AIC = -54.2, DIC = -73.3
deviance = -68.8
```

So we have $\hat{\sigma}_L = 0.08$, $\hat{\sigma}_T = 0.08$, $\hat{\sigma}_S = 0.06$ and $\hat{\sigma}_\varepsilon = 0.08$. So all four variance components are of a similar magnitude. The lack of consistency in measures of fat content can be ascribed to variance between labs, variance between technicians, variance in measurement due to different labeling and just plain measurement error. We can see if the model can be simplified by removing the lowest level of the variance components. Again the parametric bootstrap can be used:

```
cmodr <- lmer(Fat ~ 1 + (1|Lab) + (1|Lab:Technician), data=eggs)
lrstat <- numeric(1000)
for(i in 1:1000){
   rFat <- unlist(simulate(cmodr))
   nmod <- lmer(rFat ~ 1 + (1|Lab) + (1|Lab:Technician), data=eggs)
   amod <- lmer(rFat ~ 1 + (1|Lab) + (1|Lab:Technician) +
   (1|Lab:Technician:Sample), data=eggs)
   lrstat[i] <- 2*(logLik(amod)-logLik(nmod))
  }
mean(lrstat > 2*(logLik(cmod)-logLik(cmodr)))
```

```
[1] 0.092
```

We do not reject $H_0 : \sigma_S^2 = 0$. A similar computation may be made using the `RLRsim` package. This requires us to specify another model where only the tested random effect is included:

```
library(RLRsim)
cmods <- lmer(Fat ~ 1 + (1|Lab:Technician:Sample), data=eggs)
exactRLRT(cmods, cmod, cmodr)
```

```
        simulated finite sample distribution of RLRT.

        (p-value based on 10000 simulated values)

data:
RLRT = 1.6034, p-value = 0.1056
```

An examination of the reduced model is interesting:

```
VarCorr(cmodr)
```

```
 Groups         Name        Std.Dev.
 Lab:Technician (Intercept) 0.0895
 Lab            (Intercept) 0.0769
 Residual                   0.0961
```

The variation due to samples has been absorbed into the other components.

So we can reasonably say that the variation due to samples can be ignored. We may now test the significance of the variation between technicians. Using the same method above, this is found to be significant.

Although the data has a natural hierarchical structure which suggests a particular order of testing, we might reasonably wonder which of the components contribute substantially to the overall variation. Why test the sample effect first? A look at the confidence intervals reveals the problem:

```
confint(cmod, method="boot")
```

```
                                        2.5 %   97.5 %
sd_(Intercept)|Lab:Technician:Sample 0.000000 0.097527
sd_(Intercept)|Lab:Technician        0.000000 0.136021
sd_(Intercept)|Lab                   0.000000 0.152663
sigma                                0.058872 0.107040
(Intercept)                          0.299666 0.473920
```

We might drop any of the three random effect terms but it is not possible to be sure which is best to go. It is safest to conclude there is some variation in the fat measurement coming from all three sources.

10.9 Crossed Effects

Effects are said to be crossed when they are not nested. In full factorial designs, effects are completely crossed because every level of one factor occurs with every level of another factor. However, in some other designs, crossing is not complete. An example of less than complete crossing is a latin square design, where there is one treatment factor and two blocking factors. Although not all combinations of factors occur, the blocking factors are not nested. When at least some crossing occurs, methods for nested designs cannot be used. We consider a latin square example.

In an experiment reported by Davies (1954), four materials, A, B, C and D, were fed into a wear-testing machine. The response is the loss of weight in 0.1 mm over the testing period. The machine could process four samples at a time and past experience indicated that there were some differences due to the position of these four samples. Also some differences were suspected from run to run. A fixed effects analysis of this dataset may be found in Faraway (2014). Four runs were made. The latin square structure of the design may be observed:

```
data(abrasion, package="faraway")
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"
```

We can plot the data as seen in Figure 10.8.

```
library(ggplot2)
ggplot(abrasion,aes(x=material, y=wear, shape=run, color=position))+
    ↪ geom_point(position = position_jitter(width=0.1, height=0.0))+
    ↪ scale_color_grey()
```

A fixed effects analysis of the data reveals:

```
lmod <- aov(wear ~ material + run + position, abrasion)
summary(lmod)
```

```
            Df Sum Sq Mean Sq F value  Pr(>F)
material     3   4622    1540   25.15 0.00085
run          3    986     329    5.37 0.03901
```

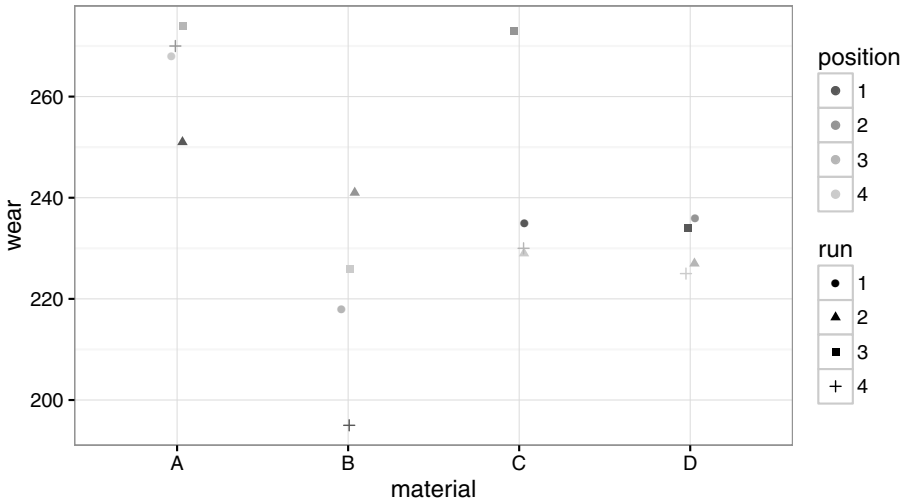

Figure 10.8 *Abrasion wear on material according to run and position.*

```
position     3   1468     489    7.99 0.01617
Residuals    6    367      61
```

All the effects are significant. However, we might regard the run and position as random effects. The appropriate model is then:

```
mmod <- lmer(wear ~ material + (1|run) + (1|position), abrasion)
sumary(mmod)
Fixed Effects:
            coef.est coef.se
(Intercept) 265.75     7.67
materialB   -45.75     5.53
materialC   -24.00     5.53
materialD   -35.25     5.53

Random Effects:
 Groups   Name        Std.Dev.
 run      (Intercept)  8.18
 position (Intercept) 10.35
 Residual              7.83
---
number of obs: 16, groups: run, 4; position, 4
AIC = 114.3, DIC = 140.4
deviance = 120.3
```

The `lmer` function is able to recognize that the run and position effects are crossed and fit the model appropriately. We can test the random effects using the `RLRsim` package. We need to fit both models that use just one random effect:

```
library(RLRsim)
mmodp <- lmer(wear ~ material + (1|position), abrasion)
mmodr <- lmer(wear ~ material + (1|run), abrasion)
exactRLRT(mmodp, mmod, mmodr)
        simulated finite sample distribution of RLRT.
```

```
        (p-value based on 10000 simulated values)

data:
RLRT = 4.5931, p-value = 0.0139
```

This first comparison tests the significance of the `position` term. The first model in the `exactRLRT` specifies the model with only that random effect term being tested. The second and third terms specify the alternative and null models under the hypothesis being tested. We see that the position variance is statistically significant. We can also test the run term:

```
exactRLRT(mmodr, mmod, mmodp)
        simulated finite sample distribution of RLRT.

        (p-value based on 10000 simulated values)

data:
RLRT = 3.0459, p-value = 0.0345
```

We see that the run variation is also statistically significant. Since the design of this experiment has already restricted the randomization to allow for these effects, we would keep these terms in the model even if they were found not to be significant. This information would only be valuable for future experiments.

The fixed effect term can be tested using the `pbkrtest` package. Given the small balanced nature of the experiment, we can feel confident in using the Kenward-Roger adjustment. Note that we need to use ML estimation for the fixed effect comparison.

```
library(pbkrtest)
mmod <- lmer(wear ~ material + (1|run) + (1|position), abrasion,REML=
    ↪ FALSE)
nmod <- lmer(wear ~ 1+ (1|run) + (1|position), abrasion,REML=FALSE)
KRmodcomp(mmod, nmod)
F-test with Kenward-Roger approximation; computing time: 0.15 sec.
large : wear ~ material + (1 | run) + (1 | position)
small : wear ~ 1 + (1 | run) + (1 | position)
      stat  ndf  ddf F.scaling p.value
Ftest 25.1  3.0  6.0         1 0.00085
```

We find that there is a clearly significant difference in the materials.

The fixed effects analysis was somewhat easier to execute, but the random effects analysis has the advantage of producing estimates of the variation in the blocking factors which will be more useful in future studies. Fixed effects estimates of the run effect for this experiment are only useful for the current study.

10.10 Multilevel Models

Multilevel models is a term used for models for data with hierarchical structure. The term is most commonly used in the social sciences. We can use the methodology we have already developed to fit some of these models.

We take as our example some data from the Junior School Project collected from primary (U.S. term is elementary) schools in inner London. The data is described in detail in Mortimore et al. (1988) and a subset is analyzed extensively in Goldstein (1995).

The variables in the data are the `school`, the `class` within the school (up to

four), `gender`, `social` class of the father (I=1; II=2; III nonmanual=3; III manual=4; IV=5; V=6; Long-term unemployed=7; Not currently employed=8; Father absent=9), `raven`'s test in year 1, student `id` number, `english` test score, `math`ematics test score and school `year` (coded 0, 1 and 2 for years one, two and three). So there are up to three measures per student. The data was obtained from the *Multilevel Models project*.

We shall take as our response the math test score result from the final year and try to model this as a function of gender, social class and the Raven's test score from the first year which might be taken as a measure of ability when entering the school. We subset the data to ignore the math scores from the first two years:

```
data(jsp, package="faraway")
jspr <- jsp[jsp$year==2,]
```

We start with two plots of the data. Due to the discreteness of the score results, it is helpful to *jitter* (add small random perturbations) the scores to avoid overprinting. The use of transparency, specified using the `alpha` parameter, also helps with dense data.

```
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()
```

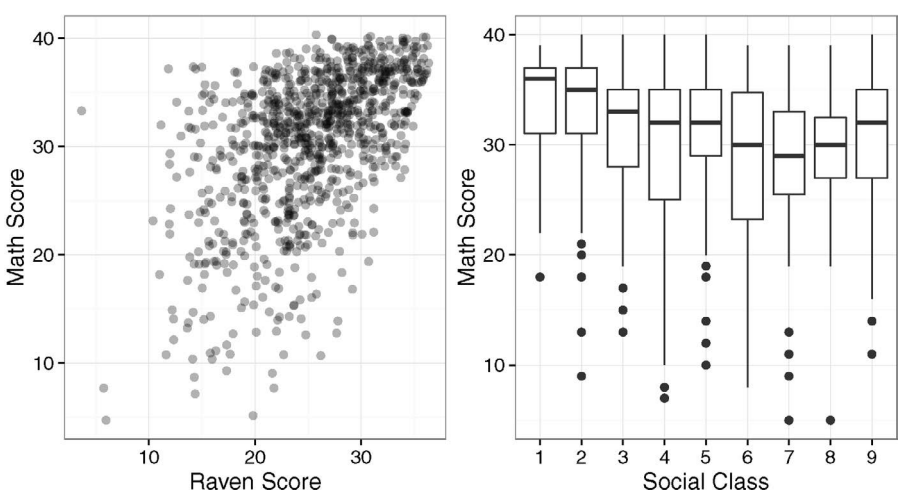

Figure 10.9 *Plots of the Junior School Project data.*

In Figure 10.9, we can see the positive correlation between the Raven's test score and the final math score. The maximum math score was 40, which reduces the variability at the upper end of the scale. We also see how the math scores tend to decline with social class.

One possible approach to analyzing these data is multiple regression. For example, we could fit:

```
glin <- lm(math ~ raven*gender*social,jspr)
anova(glin)
```

```
Analysis of Variance Table

Response: math
                    Df  Sum Sq Mean Sq F value Pr(>F)
raven                1   11481   11481  368.06 <2e-16
gender               1      44      44    1.41 0.2347
social               8     779      97    3.12 0.0017
raven:gender         1 0.01145 0.01145 0.00037 0.9847
raven:social         8     583      73    2.33 0.0175
gender:social        8     450      56    1.80 0.0727
raven:gender:social  8     235      29    0.94 0.4824
Residuals          917   28603      31
```

It would seem that gender effects can be removed entirely, giving us:

```
glin <- lm(math ~ raven*social,jspr)
anova(glin)
Analysis of Variance Table

Response: math
             Df Sum Sq Mean Sq F value Pr(>F)
raven         1  11481   11481  365.72 <2e-16
social        8    778      97    3.10 0.0019
raven:social  8    564      71    2.25 0.0222
Residuals   935  29351      31
```

This is a fairly large dataset, so even small effects can be significant. Even though the `raven:social` term is significant at the 5% level, we remove it to simplify interpretation:

```
glin <- lm(math ~ raven+social,jspr)
summary(glin)
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  17.0248     1.3745   12.39   <2e-16
raven         0.5804     0.0326   17.83   <2e-16
social2       0.0495     1.1294    0.04    0.965
social3      -0.4289     1.1957   -0.36    0.720
social4      -1.7745     1.0599   -1.67    0.094
social5      -0.7823     1.1892   -0.66    0.511
social6      -2.4937     1.2609   -1.98    0.048
social7      -3.0485     1.2907   -2.36    0.018
social8      -3.1175     1.7749   -1.76    0.079
social9      -0.6328     1.1273   -0.56    0.575

n = 953, p = 10, Residual SE = 5.632, R-Squared = 0.29
```

We see that the final math score is strongly related to the entering Raven score and that the math scores of the lower social classes are lower, even after adjustment for the entering score. Of course, any regression analysis requires more investigation than this; there are diagnostics and transformations to be considered and more. However, even if we were to do this, there would still be a problem with this analysis. We are assuming that the 953 students in the dataset are independent observations. This is not a tenable assumption as the students come from 50 different schools. The number coming from each school varies:

```
table(jspr$school)
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
26 11 14 24 26 18 11 27 21  0 11 23 22 13  7 16  6 18 14 13 28
22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
```

```
14 18 21 14 20 22 15 13 27 35 23 44 27 16 28 17 12 14 10 10 41
44 45 46 47 48 49 50
 5 11 15 33 63 22 14
```

It is highly likely that students in the same school (and perhaps class) will show some dependence. So we have somewhat less than 953 independent cases worth of information. Any analysis that pretends these are independent is likely to overstate the significance of the results. Furthermore, the analysis above tells us nothing about the variation between and within schools. People will certainly be interested in this. We could aggregate the results across schools but this would lose information and expose us to the dangers of an ecological regression.

We need an analysis that uses the individual-level information, but also reflects the grouping in the data. Our first model has fixed effects representing all interactions between `raven`, `social` and `gender` with random effects for the school and the class nested within the school:

```
mmod <- lmer(math ~ raven*social*gender+(1|school)+(1|school:class),
    ↪ data=jspr)
```

A look at the summary output from this model suggests that gender may not be significant. We can test this using the Kenward-Roger adjusted F-test from the `pbkrtest` package:

```
mmodr <- lmer(math ~ raven*social+(1|school)+(1|school:class),  data=
    ↪ jspr)
KRmodcomp(mmod, mmodr)
```

```
F-test with Kenward-Roger approximation; computing time: 0.39 sec.
large : math ~ raven * social * gender + (1 | school) + (1 | school:class)
small : math ~ raven * social + (1 | school) + (1 | school:class)
        stat    ndf    ddf F.scaling p.value
Ftest   1.01  18.00 892.94         1    0.44
```

This can be verified using the parametric bootstrap although with a dataset of this size, it does take some time to run. The size of the dataset means that we can be quite confident about the adjusted F-test in any case.

In this example, we have more than a handful of potential models we might consider even if we vary only the fixed effect part of the model. In such circumstances, we might prefer to take a criterion-based approach to model selection. One approach is to specify all the models we wish to consider:

```
all3 <- lmer(math ~ raven*social*gender+(1|school)+(1|school:class),
    ↪ data=jspr, REML=FALSE)
all2 <- update(all3, . ~ . - raven:social:gender)
notrs <- update(all2, . ~ . -raven:social)
notrg <- update(all2, . ~ . -raven:gender)
notsg <- update(all2, . ~ . -social:gender)
onlyrs <- update(all2, . ~ . -social:gender - raven:gender)
all1 <-  update(all2, . ~ . -social:gender - raven:gender - social:
    ↪ raven)
nogen <- update(all1, . ~ . -gender)
```

It is important to use the ML method for constructing the AICs. As explained previously, it is not sensible to use the REML method when comparing models with different fixed effects. We have specified models with a three-way interaction, all two-way interactions, models leaving out each two-way interaction, a model excluding any interaction involving gender, a model with just main effects and finally a

model without gender entirely. Now we can create a table showing the AIC and BIC values:

```
anova(all3, all2, notrs, notrg, notsg, onlyrs, all1, nogen)[,1:4]
```

```
       Df  AIC  BIC logLik
all1   14 5956 6024  -2964
nogen  21 5949 6051  -2954
onlyrs 22 5950 6057  -2953
notrs  23 5962 6073  -2958
notsg  23 5952 6064  -2953
notrg  30 5956 6102  -2948
all2   31 5958 6108  -2948
all3   39 5967 6156  -2944
```

The `anova` output produces chi-squared tests for comparing the models. This is not correct here as the sequence of models is not nested and furthermore, these tests are inaccurate for reasons previously explained. We exclude this part of the output using `[,1:4]`. We can see that the AIC is minimized by the model that removes gender entirely. This confirms our hypothesis-testing based approach to selecting the model but rather more thoroughly by also considering the intermediate models.

The BIC criterion commonly prefers models that are smaller than the AIC. We see that illustrated in this example as BIC picks the model with only the main effects. We might reasonably add other models to the comparison. It becomes tedious to list all the possibilities when there are more variables but it requires some more complex R code to generate these automatically.

Given that we have decided that gender is not important, we simplify to:

```
jspr$craven <- jspr$raven-mean(jspr$raven)
mmod <- lmer(math ~ craven*social+(1|school)+(1|school:class),jspr)
sumary(mmod)
```

```
Fixed Effects:
              coef.est coef.se
(Intercept)    31.91     1.20
craven          0.61     0.19
social2         0.02     1.27
social3        -0.63     1.31
social4        -1.97     1.20
social5        -1.36     1.30
social6        -2.27     1.37
social7        -2.55     1.41
social8        -3.39     1.80
social9        -0.83     1.25
craven:social2 -0.13     0.21
craven:social3 -0.22     0.22
craven:social4  0.04     0.19
craven:social5 -0.15     0.21
craven:social6 -0.04     0.23
craven:social7  0.40     0.23
craven:social8  0.26     0.26
craven:social9 -0.08     0.21

Random Effects:
 Groups       Name        Std.Dev.
 school:class (Intercept) 1.08
 school       (Intercept) 1.77
 Residual                 5.21
```

```
---
number of obs: 953, groups: school:class, 90; school, 48
AIC = 5963.2, DIC = 5893.6
deviance = 5907.4
```

We centered the Raven score about its overall mean. This means that we can interpret the social effects as the predicted differences from social class one at the mean Raven score. If we did not do this, these parameter estimates would represent differences for `raven`=0 which is not very useful. We can see the math score is strongly related to the entering Raven score. We see that for the same entering score, the final math score tends to be lower as social class goes down. Note that class 9 here is when the father is absent and class 8 is not necessarily worse than 7, so this factor is not entirely ordinal. We also see the most substantial variation at the individual level with smaller amounts of variation at the school and class level.

We check the standard diagnostics first:

```
diagd <- fortify(mmod)
ggplot(diagd,aes(sample=.resid))+stat_qq()
ggplot(diagd,aes(x=.fitted,y=.resid)) +geom_point(alpha=0.3) +geom_
    ↪ hline(yintercept=0) +xlab("Fitted") +ylab("Residuals")
```

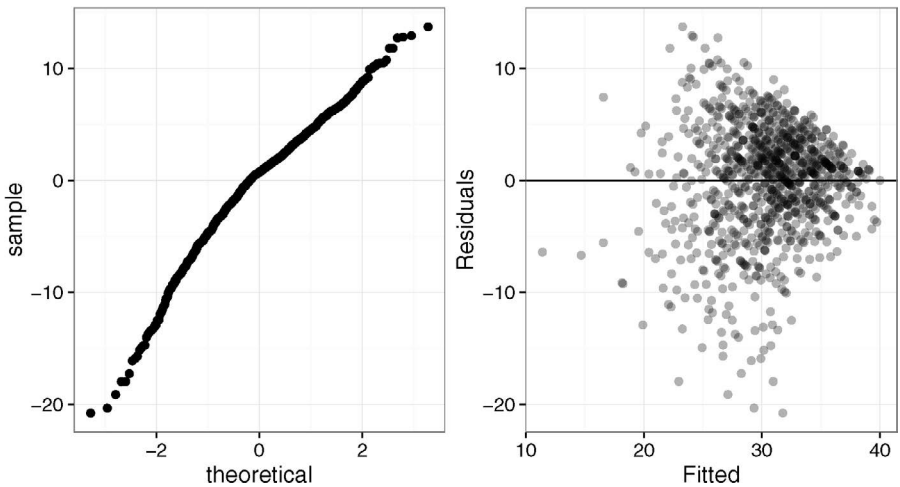

Figure 10.10 *Diagnostic plots for the Junior Schools Project model.*

In Figure 10.10, we see that the residuals are close to normal, but there is a clear decrease in the variance with an increase in the fitted values. This is due to the reduced variation in higher scores already observed. We might consider a transformation of the response to remove this effect.

We can also check the assumption of normally distributed random effects. We can do this at the school and class level:

```
qqnorm(ranef(mmod)$school[[1]],main="School effects")
qqnorm(ranef(mmod)$"school:class"[[1]],main="Class effects")
```

We see in Figure 10.11 that there is approximate normality in both cases with some

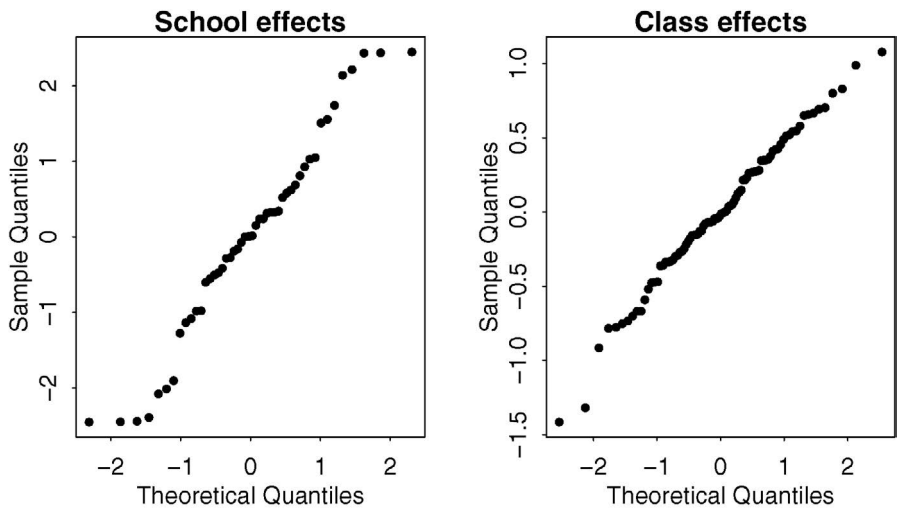

Figure 10.11 *QQ plots of the random effects at the school and class levels.*

evidence of short tails for the school effects. It is interesting to look at the sorted school effects:

```
adjscores <- ranef(mmod)$school[[1]]
```

These represent a ranking of the schools adjusted for the quality of the intake and the social class of the students. The difference between the best and the worst is about five points on the math test. Of course, we must recognize that there is variability in these estimated effects before making any decisions about the relative strengths of these schools. Compare this with an unadjusted ranking that simply takes the average score achieved by the school, centered by the overall average:

```
rawscores <- coef(lm(math ~ school-1,jspr))
rawscores <- rawscores-mean(rawscores)
```

We compare these two measures of school quality in Figure 10.12:

```
plot(rawscores,adjscores)
sint <- c(9,14,29)
text(rawscores[sint],adjscores[sint]+0.2,c("9","15","30"))
```

School 10 is listed but has no students, hence the need to adjust the labeling. There are some interesting differences. School 15 looks best on the raw scores but after adjustment, it drops to 15th place. This is a school that apparently performs well, but when the quality of the incoming students is considered, its performance is not so impressive. School 30 illustrates the other side of the coin. This school looks average on the raw scores, but is doing quite well given the ability of the incoming students. School 9 is actually doing a poor job despite raw scores that look quite good.

It is also worth plotting the residuals and the random effects against the predictors. We would be interested in finding any inhomogeneity or signs of structure that might lead to an improved model.

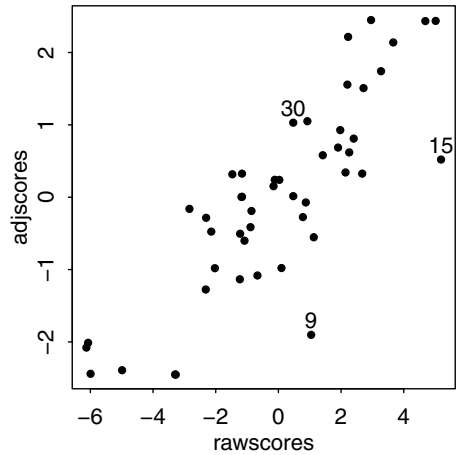

Figure 10.12 *Raw and adjusted school-quality measures. Three selected schools are marked.*

We may also be interested to know whether there really is much variation between schools or classes within schools. We can investigate this by testing the random effect terms using the `RLRsim` package. We need to fit models without each of the random effect terms.

```
library(RLRsim)
mmodc <- lmer(math ~ craven*social+(1|school:class),jspr)
mmods <- lmer(math ~ craven*social+(1|school),jspr)
```

We can test the class effect:

```
exactRLRT(mmodc, mmod, mmods)
        simulated finite sample distribution of RLRT.

        (p-value based on 10000 simulated values)

data:
RLRT = 2.3903, p-value = 0.0549
```

The evidence for a class effect is quite marginal. We would certainly choose to include it for testing fixed effect terms as we would rather be sure that it had been taken account of. Even so we can see that the class effect may be quite small. In contrast, we can test for a school effect:

```
exactRLRT(mmods, mmod, mmodc)
        simulated finite sample distribution of RLRT.

        (p-value based on 10000 simulated values)

data:
RLRT = 7.1403, p-value = 0.0033
```

The school effect comes through strongly. It seems schools matter more than specific teachers.

Compositional Effects: Fixed effect predictors in this example so far have been at the lowest level, the student, but it is not improbable that factors at the school or

class level might be important predictors of success in the math test. We can construct some such predictors from the individual-level information; such factors are called *compositional effects*. For example, the average entering score for a school might be an important predictor. The ability of one's fellow students may have an impact on future achievement. We construct this variable:

```
schraven <- lm(raven ~ school, jspr)$fit
```

and insert it into our model:

```
mmodc <- lmer(math ~ craven*social+schraven*social+(1|school)+  (1|
    ↪ school:class),jspr)
KRmodcomp(mmod, mmodc)
```

```
F-test with Kenward-Roger approximation; computing time: 0.16 sec.
large : math ~ craven * social + schraven * social + (1 | school) + (1 |
    school:class)
small : math ~ craven * social + (1 | school) + (1 | school:class)
        stat    ndf    ddf F.scaling p.value
Ftest   0.68   9.00 640.14     0.997    0.73
```

We see that this new effect is not significant. We are not constrained to taking means. We might consider various quantiles or measures of spread as potential compositional variables.

Much remains to be investigated with this dataset. We have only used the simplest of error structures and we should investigate whether the random effects may also depend on some of the other covariates.

Further Reading: The classical approach to random effects can be found in many older books such as Snedecor and Cochran (1989) or Scheffé (1959). More recent books such as Searle et al. (1992) also focus on the ANOVA approach. A wide range of models are explicitly considered in Milliken and Johnson (1992). Multilevel models are covered in Goldstein (1995), Raudenbush and Bryk (2002) and Gelman and Hill (2006). The predecessor to the `lme4` package was `nlme` which is described in Pinheiro and Bates (2000), but the book still contains much general material of interest.

Exercises

1. The `denim` dataset concerns the amount of waste in material cutting for a jeans manufacturer due to five suppliers.
   (a) Plot the data and comment.
   (b) Fit the linear fixed effects model. Is the operator significant?
   (c) Make a useful diagnostic plot for this model and comment.
   (d) Analyze the data with supplier as a random effect. What are the estimated standard deviations of the effects?
   (e) Test the significance of the supplier term.
   (f) Compute confidence intervals for the random effect SDs.
   (g) Locate two outliers and remove them from the data. Repeat the fitting, testing and computation of the confidence intervals, commenting on the differences you see from the complete data.

(h) Estimate the effect of each supplier. If only one supplier will be used, choose the best.

2. The `coagulation` dataset comes from a study of blood coagulation times. Twenty-four animals were randomly assigned to four different diets and the samples were taken in a random order.
   (a) Plot the data and comment.
   (b) Fit a fixed effects model and construct a prediction together with a 95% prediction interval for the response of a new animal assigned to diet D.
   (c) Now fit a random effects model using REML. A new animal is assigned to diet D. Predict the blood coagulation time for this animal along with a 95% prediction interval.
   (d) A new diet is given to a new animal. Predict the blood coagulation time for this animal along with a 95% prediction interval
   (e) A new diet is given to the first animal in the dataset. Predict the blood coagulation time for this animal with a prediction interval. You may assume that the effects of the initial diet for this animal have washed out.
3. The `eggprod` dataset concerns an experiment where six pullets were placed into each of 12 pens. Four blocks were formed from groups of three pens based on location. Three treatments were applied. The number of eggs produced was recorded.
   (a) Make suitable plots of the data and comment.
   (b) Fit a fixed effects model for the number of eggs produced with the treatments and blocks as predictors. Determine the significance of the two predictors and perform a basic diagnostic check.
   (c) Fit a model for the number of eggs produced with the treatments as fixed effects and the blocks as random effects. Which treatment is best in terms of maximizing production according to the model? Are you sure it is better than other two treatments?
   (d) Use the Kenward-Roger approximation for an F-test to check for differences between the treatments. How does the result compare to the fixed effects result?
   (e) Perform the same test but using a bootstrap method. How do the results compare?
   (f) Test for the significance of the blocks. Does the outcome agree with the fixed effects result?
4. Data on the cutoff times of lawnmowers may be found in the dataset `lawn`. Three machines were randomly selected from those produced by manufacturers A and B. Each machine was tested twice at low speed and high speed.
   (a) Make plots of the data and comment.
   (b) Fit a fixed effects model for the cutoff time response using just the main effects of the three predictors. Explain why not all effects can be estimated.

(c) Fit a mixed effects model with manufacturer and speed as main effects along with their interaction and machine as a random effect. If the same machine were tested at the same speed, what would be the SD of the times observed? If different machines were sampled from the same manufacturer and tested at the same speed once only, what would be the SD of the times observed?

(d) Test whether the interaction term of the model can be removed. If so, go on to test the two main fixed effects terms.

(e) Check whether there is any variation between machines.

(f) Fit a model with speed as the only fixed effect and manufacturer as a random effect with machines also as a random effect nested within manufacturer. Compare the variability between machines with the variability between manufacturers.

(g) Construct bootstrap confidence intervals for the terms of the previous model. Discuss whether the variability can be ascribed solely to manufacturers or to machines.

5. A number of growers supply broccoli to a food processing plant. The plant instructs the growers to pack the broccoli into standard-size boxes. There should be 18 clusters of broccoli per box. Because the growers use different varieties and methods of cultivation, there is some variation in the cluster weights. The plant manager selected three growers at random and then four boxes at random supplied by these growers. Three clusters were selected from each box. The data may be found in the `broccoli` dataset. The weight in grams of the cluster is given.

(a) Plot the data and comment on the nature of the variation seen.

(b) Compute the mean weights within growers. Compute the mean weights within boxes.

(c) Fit an appropriate mixed effects model. Comment on how the variation is assigned to the possible sources.

(d) Test whether there may be no variation attributable to growers.

(e) Test whether there may be no variation attributable to boxes.

(f) Compute confidence intervals for the SD components in your full model.

6. An experiment was conducted to select the supplier of raw materials for production of a component. The breaking strength of the component was the objective of interest. Four suppliers were considered. The four operators can only produce one component each per day. A latin square design is used and the data is presented in `breaking`.

(a) Plot the data and interpret.

(b) Fit a fixed effects model for the main effects. Determine which factors are significant.

(c) Fit a mixed effects model with operators and days as random effects but the suppliers as fixed effects. Why is this a natural choice of fixed and random effects? Which supplier results in the highest breaking point? What is the nature of the variation between operators and days?

(d) Test the operator and days effects.

(e) Test the significance of the supplier effect.

(f) For the best choice of supplier, predict the proportion of components produced in the future that will have a breaking strength less than 1000.

7. An experiment was conducted to optimize the manufacture of semiconductors. The `semicond` data has the resistance recorded on the wafer as the response. The experiment was conducted during four different time periods denoted by ET and three different wafers during each period. The position on the wafer is a factor with levels 1 to 4. The `Grp` variable is a combination of ET and wafer. Analyze the data as a split plot experiment where ET and position are considered as fixed effects. Since the wafers are different in experimental time periods, the `Grp` variable should be regarded as the block or group variable.

   (a) Plot the data appropriately and comment.

   (b) Fit a fixed effects model with an interaction between `ET` and `position` (no other predictors). What terms are significant? What is wrong with using this model to make inference about these predictors?

   (c) Fit a model appropriate to the split plot design used here. Comment on the relative variation between and within the groups (`Grp`).

   (d) Test for the effect of position.

   (e) Which level of `ET` results in the highest resistance? Can we be sure that this is really better than the second highest level?

   (f) Make a plot of the residuals and fitted values and interpret. Make a QQ plot and comment.

8. Redo the Junior Schools Project data analysis in the text with the final year English score as the response. Highlight any differences from the analysis of the final year Math scores.

9. An experiment was conducted to determine the effect of recipe and baking temperature on chocolate cake quality. Fifteen batches of cake mix for each recipe were prepared. Each batch was sufficient for six cakes. Each of the six cakes was baked at a different temperature which was randomly assigned. Several measures of cake quality were recorded of which breaking angle was just one. The dataset is presented as `choccake`.

   (a) Plot the data and comment.

   (b) Fit linear model with an interaction between recipe and temperature as fixed effects and no random effects. Which terms are significant? Why is this analysis unreliable?

   (c) Fit a mixed effects model that takes account of the batch structure, identifying the design type. Compare the temperature effect (minimum to maximum) with the likely difference between batches. How do they compare?

   (d) Test for a recipe effect.

   (e) Check the following diagnostic plots and comment.

      i. The residuals against fitted values.

ii. A QQ plot of the residuals.
iii. A QQ plot of the batch random effects.