Linear Mixed Models (LMMs) Notes

Daniel J. Eck

Contents

Many common statistical models can be expressed as linear models that incorporate both *fixed effects*, which are parameters associated with an entire population or with certain repeatable levels of experimental factors, and *random effects*, which are associated with individual units drawn at random from a population.

A model with both fixed effects and random effects is called a *mixed-effects model*. The random effects in a mixed-effects model can either be *nested* or *crossed*. 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.

So far in this class we have discussed models with fixed effects only. In this sense, regression model parameters are population level parameters or sub-population parameters. Inference for these parameters is at the population level. We will now introduce a simple example of a mixed-effects model.

We will consider a two-way analysis of variance (ANOVA) as a simple example of a mixed-effects model with both fixed and random effects:

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

where μ a population mean fixed effect, τ_i is a sub-population fixed-effect, v_j is a random effect, and ε_{ijk} is an error term. We will assume that $v_j \overset{iid}{\sim} N(0, \sigma_v^2)$ and $\varepsilon_{ijk} \overset{iid}{\sim} N(0, \sigma^2)$. In this model we may want to

- estimate τ_i and test the hypothesis $H_0 : \tau_i = 0$,
- estimate σ_v^2 and test $H_0 : \sigma_v^2 = 0$.

Notice that we want to estimate and test all of the fixed-effects τ_i while we only need to estimate and test a single random effects parameter.

Basic linear mixed-effects model example

This is not as simple for fixed effects models. We will start with the simplest possible example random effects model: a one-way ANOVA design with a factor at a levels:

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

where the αs and the εs have zero mean and respective variances σ_α^2 and σ_ε^2. These variances are known as the **variance components**. Note that this formulation induces an intraclass correlation between observations at the same level that is formally expressed as

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

For simplicity, let there be an equal number of observations n per level. Recall that we can decompose the total variation as SST = SSE + SSA,

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

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 use the estimators:

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

These estimators are known as the ANOVA estimators, and this method of estimating the variance components can be used for more complicated designs. These estimators have the advantage that they take explicit form (a major advantage in the precomputing days), but they have a number of disadvantages including:

1. the possibility of negative variance estimates
2. not easily adapted to unbalanced data
3. the need for complex algebraic complications in more complex models

One technique for estimating parameters in mixed-effect regression models that alleviates these concerns is the familiar method of maximum likelihood estimation. Of course, this requires that we assume distributions for the random effects and the errors.

More general LMM setup

We will now consider models with normal modeling assumptions:

$$Y = X\beta + Zb + \varepsilon \quad \text{or} \quad Y|b \sim N(X\beta + Zb, \sigma^2 I),$$

where $Y \in \mathbb{R}^n$ is the response vector, $X \in \mathbb{R}^{n\times p}$ is a fixed model matrix, $\beta \in \mathbb{R}^p$ is a coefficient vector and is considered a fixed effect, $Z \in \mathbb{R}^{n\times q}$ is a fixed model matrix for the random effects, $b \in \mathbb{R}^q$ is a vector of random effects, and σ^2 is the variance of the error distribution. If we further assume that the random effects satisfy $b \sim N(0, \sigma^2 D)$, then the unconditional response Y has the distribution

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

We can estimate β using generalized least squares if we know D. However, knowledge of D is usually lacking. Then, letting $V = I + ZDZ^T$, the log likelihood for this model is

$$l(\beta, \sigma, D|y) \propto \frac{\log|\sigma^2 V|}{2} + \frac{(Y - X\beta)^T V^{-1}(Y - X\beta)}{2\sigma^2}.$$

In principle the above can be estimated by maximum likelihood estimation, but it is difficult in practice. One difficulty is that the final unrestricted estimate may estimate some of the variances in D to be negative, and such estimates must be constrained to be zero. First order optimization conditions may therefore not be satisfied.

MLE also suffers bias when estimating random effect parameters. Recall a simple example where $x_1, \ldots, x_n \overset{iid}{\sim} N(\mu, \sigma^2)$, then the MLE of σ^2 is $\hat{\sigma}^2 = n^{-1}\sum_i (x_i - \bar{x})^2$. This estimate is biased, an unbiased estimator is leading term should be $(n-1)^{-1}$ instead of n^{-1}. Conceptually, these problems are driven by estimation of μ. If μ were known then $n^{-1}\sum_i (x_i - \mu)^2$ is an unbiased estimator for σ^2. Similar problems occur with the estimation of variance components. Given, that the number of factors may not be large, the bias of the MLE for the variance component associated with that factor may be quite large.

Restricted maximum likelihood estimation (REML) estimators are an attempt to get around problems with standard unrestricted maximum likelihood estimation. The idea of REML is to find all linear combinations of the response, denoted k, such that $k^T X = 0$. Form a matrix K with columns k, so that

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

This removes the effect of estimating the mean (fixed effects) from the problem. We can then proceed to maximize the likelihood based on $K^T Y$. Once the random effect parameters have been estimated, it is easy enough to estimate the fixed effect parameters via plug-in. Plug-in gives,

$$Y \approx N(X\beta, \sigma^2 \widehat{V}_{\text{REML}}),$$

and maximum likelihood then gives

$$\hat{\beta}_{\text{REML}} = \left(X^T \widehat{V}_{\text{REML}}^{-1} X\right)^{-1} X^T \widehat{V}_{\text{REML}}^{-1} Y.$$

REML generally produces less biased estimates than unconditional MLE. For balanced data, REML estimates are usually the same as ANOVA estimates.

Example: pulp data set

We illustrate the fitting methods using some data from an experiment which tests how the paper brightness is influenced by the shift operator on duty. See Section 8.1 in Faraway [2016] for more details. We start with a fixed effects one-way ANOVA.

```
library(ggplot2)
library(faraway)
data(pulp)
## note that we are changing the contrasts to sum contrasts
op = options(contrasts = c("contr.sum", "contr.poly"))
## aov is another wrapper for lm; results more appropriate for ANOVA models
lmod = aov(bright ~ operator, pulp)
summary(lmod)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## operator     3   1.34  0.4467   4.204 0.0226 *
## Residuals   16   1.70  0.1062
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coef(lmod)
```

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

We see that the p-value for the operator effect is roughly 0.023. We have $\hat{\sigma}^2 = .106$ and the overall mean is 60.40. For sum contrasts, we have that $\sum_i \alpha_i = 0$, so we can calculate the effect for the fourth operator as 0.16 + 0.34 - 0.22 = 0.28. Turing to the random effects model, we can calculate the variance of the operator effects $\hat{\sigma}^2_\alpha$ using the formula above as

```
## (MSA - MSE) / n
(0.4467 - 0.1062)/5
```

```
## [1] 0.0681
```

We now demonstrate the MLE approach using REML. We will use the `lme4` package to accomplish this task.

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
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
## -1.4666 -0.7595 -0.1244  0.6281  1.6012
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  operator (Intercept) 0.06808  0.2609
##  Residual             0.10625  0.3260
## Number of obs: 20, groups:  operator, 4
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  60.4000     0.1494   404.2
```

In the above, the model fitted has both 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 the `(1|operator)` indicating that the data is grouped by the `operator` variable, and the `1` indicates that the random effect is constant within each group.

The default fitting method is the REML procedure mentioned above. We see that this method gives identical estimates to the ANOVA results above. For unbalanced designs, the REML and ANOVA estimators are not necessarily identical.

Inference for nested models

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:

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

where $\hat{\beta}_0, \hat{\sigma}_0^2, \hat{D}_0$ are the MLEs of the parameters under the null hypothesis and $\hat{\beta}_1, \hat{\sigma}_1^2, \hat{D}_1$ are the MLEs of the parameters under the alternative hypothesis. The distribution of the above test statistic

$$2\{l(\hat{\beta}_1, \hat{\sigma}_1^2, \hat{D}_1|Y) - l(\hat{\beta}_0, \hat{\sigma}_0^2, \hat{D}_0|Y)\} \approx \chi^2_{\text{df}}$$

where the degrees of freedom is equal to difference in the dimensions of the two parameter spaces (the difference in the number of parameters when both models are identifiable). Note that testing in this manner is approximate and that the parameters under the null hypothesis are on the boundary of the parameter space. This is a real concern when testing the hypothesis about the random effects that take the form $H_0 : \hat{\sigma}^2 = 0$. Furthermore, the χ^2 approximation can be poor even when the conditions are met. The p-values generated by the LRT for fixed effects are approximate and unfortunately tend to be too small, thereby sometimes overstating the importance of some effects.

Note that we may use bootstrap methods to find more accurate p-values for the LRT. The most popular bootstrap approach is nonparametric in that no distributional assumptions are made. Since we are willing to assume normality for the errors and the random effects in our modeling formulation, we can use a technique called the **parametric bootstrap**. In this technique, we generate data under the null model using the fitted parameter estimates as guesses for the true parameter values. We compute the LRT test statistic for this generated data. We repeat this scheme many times and compare the observed LRT test statistic to the distribution of LRT test statistics obtained from repeatedly generation of samples. We will discuss this approach below. Here is a useful concise reference on bootstrap procedures.

An alternative approach is to condition on the estimated values of the random effect parameters and then use standard F- or t-tests. This assumes that the covariance of the random part of the model D is equal to its estimated value.

- **Testing random effects**: In most cases testing will be of the form $H_0 : \sigma^2 = 0$. This breaks the assumption that the parameter value is in the interior of its parameter space. The null distribution is unknown in general and precise methods therefore requires numerical methods. Use of the χ^2 distribution with the usual dfs tends to inflate the p-values. Thus you can be confident in significant findings. More time consuming bootstrapping methods are needed for small, not significant p-values. However, careful coding using parallel methods can greatly quicken these computations.
- **Expected mean squares**: These are more powerful than the LRT but they often require tedious algebra that is dependent on the model under study. They also require extensive adjustments for unbalanced designs.
- **Testing fixed effects**: For testing the fixed effects we must use unrestricted maximum likelihood estimation. One can use the usual LRT, but this test may be anti-conservative. The parametric bootstrap is recommended here.

Example: pulp data set (continued)

In this example the only fixed effect is the mean and there is not much interest in testing that. We first fit the null model

```
## null model fit
nullmod = lm(bright ~ 1, pulp)

## alternative model fit
smod = lmer(bright ~ 1 + (1|operator), pulp, REML = FALSE)
```

As there are no random effects in the null model we must use `lm`. For models of the same class, we could use built in software to perform a statistical test. Here we have to do the testing manually

```
as.numeric(2*(logLik(smod) - logLik(nullmod)))
```

```
## [1] 2.568371
```

```
pchisq(2.568371, df = 1, lower = FALSE)
```

```
## [1] 0.1090199
```

The p-value for this test is well above a nominal 5% significance level (assuming that testing is desired at this level). The use of the χ^2 test with its noted shortcomings allows for some doubt in this finding.

We will now try the parametric bootstrap procedure to obtain a more accurate p-value. We need to estimate the probability, given the null hypothesis is true, of observation an LRT of 2.568371 or greater. We have that

$$y \overset{H_0}{\sim} N(\mu, \sigma^2).$$

A simulation approach would generate data under this model, fit the null and alternative models and then compute the LRT. The process would be repeated a large number of times and the proportion of estimated LRTs exceeding the the observed LRT value of 2.568371 would be used to estimate the p-value. In practice we do not know the true values of μ and σ, but we can use plug-in.

Let's try it out. Fix the bootstrap sample size to be $B = 1000$ and then continue with the parametric bootstrap procedure

```
set.seed(13)
B = 1000
lrtstat = numeric(B)
system.time(for(b in 1:B){
  y = unlist(simulate(nullmod))
  bnull = lm(y ~ 1)
  balt = lmer(y ~ 1 + (1|operator), pulp, REML = FALSE)
  lrtstat[b] = as.numeric(2*(logLik(balt) - logLik(bnull)))
})
```

```
##    user  system elapsed
##   5.525   0.040   5.564
```

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

```
mean(lrtstat < 1e-5)
```

```
## [1] 0.69
```

The LRT clearly does not have a χ^2 distribution. See Section 8.2 in Faraway [2016] for discussion on this. An interesting refinement can be seen **here**. The parametric bootstrap may be the simplest approach, and the method we have used above is transparent and could be computed much more efficiently if needed. With all of this in mind the estimated p-value is

```
## p-value
pval = mean(lrtstat > 2.568371)
pval
```

```
## [1] 0.026
```

```
## simple standard error of the above
sqrt(pval*(1-pval)/B)
```

```
## [1] 0.005032296
```

We can be fairly sure that the estimated p-value is under a 0.05 nominal level. If in doubt, do some more replications to make sure; this only costs computer time (One can perform the above analysis with $B = 1e4$ and make it as fast as possible using parallel programming with either `mclapply` and/or `foreach` and whatever accompanying software packages are needed.) As it happens, this p-value is close to the fixed effects p-value.

In this example, the random and fixed effect tests gave similar outcomes. However, the hypotheses in random and fixed effects are intrinsically different. The conclusions from testing hypotheses involving fixed effects are only relevant to the levels under study. On the other hand, tests of hypotheses involving random effects are relevant of levels of the effect not considered. They are based upon a random generative mechanism that, in this case, are indexed by a single parameter. The conclusions of such test are more meaningful, more general, and should require more care than their fixed effects counterparts.

Predicting random effects

In a model with random effects, the *alpha*s are no longer parameters, but are realizations from a random generative process. Using the standard normal assumption we have that

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

It does not make much sense to estimate the αs because they are random variables from a common distribution under this setup. So instead we may think of interesting probabilities and expected values corresponding to the random effect distribution. However, $E(\alpha_i) = 0$ for all i which is clearly not very interesting.

If however, one approaches this problem from a Bayesian perspective then we have a prior density for the αs and $E(\alpha_i) = 0$ is just the prior mean. Let f represent a density function, 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 as $\hat{\alpha}$ as

$$E(\alpha_i|Y) = \int \alpha_i f(\alpha_i|Y)d\alpha_i.$$

For the general case, this works out to be

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

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

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

```
##   (Intercept)
## a  -0.1219403
## b  -0.2591231
## c   0.1676679
## d   0.2133955
```

The predicted random effects are related to the fixed effects. We can show these for all operators

```
(cc = model.tables(lmod))
```

```
## Tables of effects
##
##  operator
## operator
##     a     b     c     d
## -0.16 -0.34  0.22  0.28
```

and then compute the ratio to the random effects as

```
## estimated fixed effects divided by predicted random effects
cc[[1]]$operator / ranef(mmod)$operator
```

```
##   (Intercept)
## a    1.312117
## b    1.312117
## c    1.312117
## d    1.312117
```

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 are displayed below:

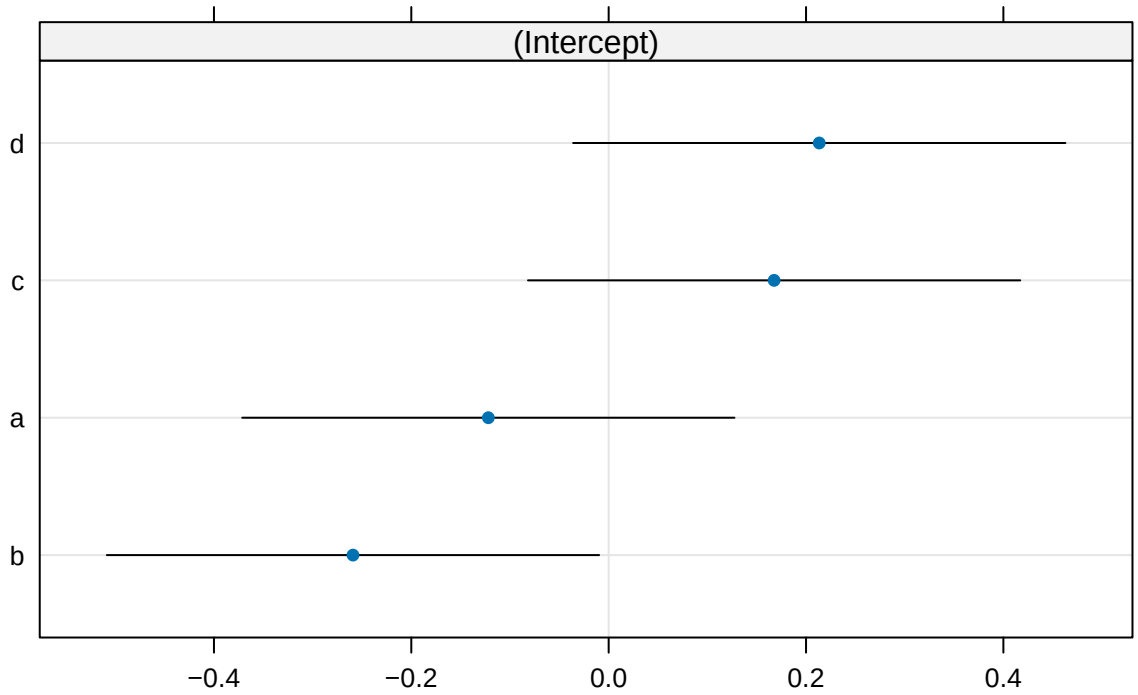
```
library(lattice)
```

```
##
## Attaching package: 'lattice'
```

```
## The following object is masked from 'package:faraway':
##
##     melanoma
```

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

```
## $operator
```



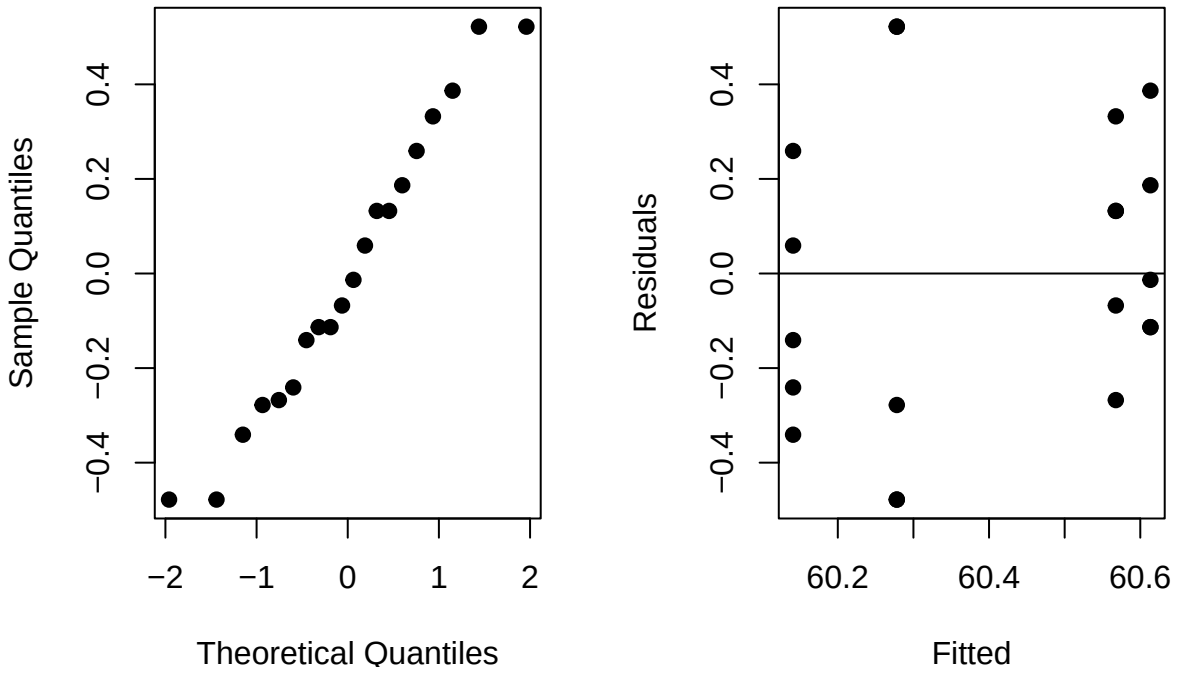
Suppose we wish to predict a new value. If the prediction is to be made for a new operator or unknown operator, the best we can do is give $\hat{\mu} = 60.4$. If we know the operator, then we can combine this with our fixed effects to give the best linear unbiased predictors (BLUPs) as follows

```
fixef(mmod) + ranef(mmod)$operator
```

```
##   (Intercept)
## a    60.27806
## b    60.14088
## c    60.56767
## d    60.61340
```

The implication of this is that we have more than one type of residual depending on what fitted values we use for inference. The default predicted values and residuals are from the outermost level of grouping. Thus, the usual diagnostic plots are still worthwhile.

```
par(mfrow = c(1,2))
qqnorm(resid(mmod), main="", pch = 19)
plot(fitted(mmod),resid(mmod), xlab="Fitted", ylab="Residuals", pch = 19)
abline(0,0)
```



Returning to the random effect predictions, we present a parametric bootstrap method for computing standard errors of the predicted random effects. As in previous bootstrap, 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. 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]
B = 1000
pv = numeric(B)
system.time(for(i in 1:B){
  y = unlist(simulate(mmod, use.u = TRUE))
  bmod = suppressWarnings(refit(mmod, y))
  pv[i] = predict(bmod, re.form=~0)[1] + rnorm(n=1,sd=group.sd) +  rnorm(n=1,sd=resid.sd)
})
```

```
##    user  system elapsed
##   6.344   0.030   6.374
```

```
quantile(pv, c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 59.62780 61.24264
```

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.

```
system.time(for(i in 1:B){
 y = unlist(simulate(mmod, use.u=TRUE))
```

```
  bmod = suppressWarnings(refit(mmod, y))
  pv[i] = predict(bmod, newdata=data.frame(operator="a")) + rnorm(n=1,sd=resid.sd)
})
```

```
##    user  system elapsed
##   9.136   0.056   9.192
```

```
quantile(pv, c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 59.60897 60.97931
```

Example: Soybean analysis

The researchers are interested in determining which genotypes are associated with a different expression of apparent quantum efficiency (AqE) from the RC reference genotype. AqE is a part of the photosynthetic process in plants where improvements in AqE allow for improvements in yield. The researchers have collected data across multiple plots for multiple days over two years. The plot effects are not of interest.

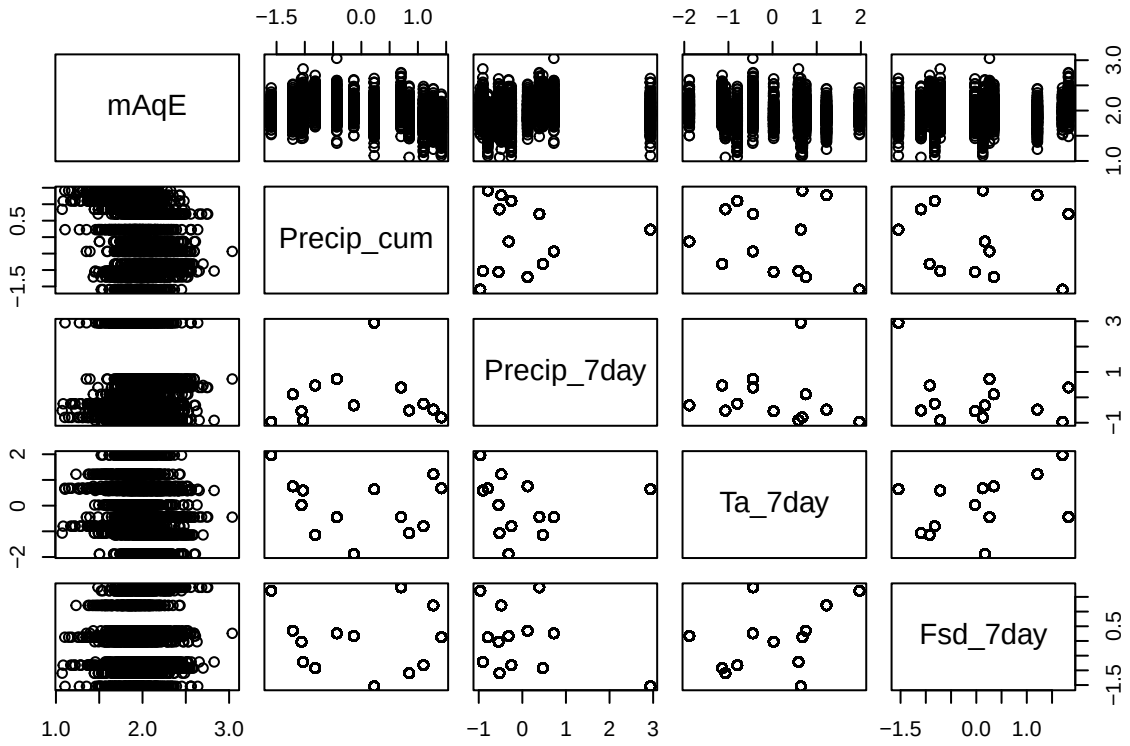
It is known that AqE will have a quadratic effect in date, and that AqE may depend on weather conditions. Some weather variables are also collected (Precip_cum, Precip_7day, Ta_7day, and Fsd_7day).

```
dat = read.csv("soybean.csv")
dat$ID = as.factor(dat$ID)
head(dat)
```

```
##         Date year    ID plot_number plot_number_year     mAqE Precip_cum
## 1 2021-06-24 2021 NAM46           1           1_2021 2.585728 -0.8163990
## 2 2021-06-30 2021 NAM46           1           1_2021 2.453331  0.2272382
## 3 2022-07-12 2022 NAM26           1           1_2022 2.295015 -1.2118618
## 4 2021-07-13 2021 NAM46           1           1_2021 2.011619  0.8453337
## 5 2022-07-19 2022 NAM26           1           1_2022 1.983389 -1.0587556
## 6 2021-07-20 2021 NAM46           1           1_2021 1.781536  1.0987849
##   Precip_7day     Ta_7day    Fsd_7day   Date_num
## 1   0.4695263 -1.14071540 -0.92257332 -1.0278192
## 2   2.9365607  0.63872728 -1.54297943 -0.9956941
## 3   0.1215929  0.75291580  0.34722996  1.0228330
## 4  -0.5227283 -1.06866861 -1.09565527 -0.9260897
## 5  -0.5442057  0.02483327 -0.02693322  1.0603123
## 6  -0.2528293 -0.79961846 -0.82287859 -0.8886104
```

This data consists of multiple measurements on the same day with only a few days of data collection. Here is a depiction of the relationships between the response and the weather variables.

```
pairs(dat[,6:10])
```



We now fit three models in increasing complexity. The first is a standard linear regression model with linear terms for precipitation variables and day, and a quadratic term for days. The next model includes a random effect for plots. The full model includes an additional random effect for disk.

```
m1_mAqE_lm = lm(mAqE ~ ID + Date_num + I(Date_num^2),
  data = dat)

m1_mAqE = lmer(mAqE ~ ID + Date_num + I(Date_num^2) +
  (1|plot_number_year),
  data = dat, REML = FALSE,
  control = lmerControl(optimizer ="Nelder_Mead"))

m1_mAqE_full = lmer(mAqE ~ ID + Date_num + I(Date_num^2) +
  Precip_cum + Precip_7day + Ta_7day + Fsd_7day + (1|plot_number_year),
  data = dat, REML = FALSE,
  control = lmerControl(optimizer ="Nelder_Mead"))
```

Just as before we can use a parametric bootstrap to compare nested models. In this case we compare the linear regression model to a mixed-effects model containing a random effect for plots. In this example we can see that the parametric bootstrap is fairly time-consuming.

```
set.seed(13)
B = 1000
lrtstat = numeric(B)

# parametric bootstrap
system.time(for(b in 1:B){
  y = unlist(simulate(m1_mAqE_lm))
  bnull = lm(y ~ ID + Date_num + I(Date_num^2), data = dat)
  balt = lmer(y ~ ID + Date_num + I(Date_num^2) +
    (1|plot_number_year),
```

```
    data = dat, REML = FALSE,
    control = lmerControl(optimizer ="Nelder_Mead"))
  lrtstat[b] = as.numeric(2*(logLik(balt) - logLik(bnull)))
})
```

```
##    user  system elapsed
##  56.691   1.584  58.312
```

We examine the distribution of the bootstrapped LRTs. We observe a large mass at zero and only observe a few non-zero bootstrap observations.

```
mean(lrtstat < 1e-5)
```

```
## [1] 0.9
```

The bootstrapped p-value is 0.

```
## p-value
pval = mean(lrtstat >
  as.numeric(2*(logLik(m1_mAqE) - logLik(m1_mAqE_lm))))
pval
```

```
## [1] 0
```

We now compare the mixed-effects model containing a random effect for plots with a larger mixed-effects model containing fixed effects for weather variables. Again, the parametric bootstrap is fairly time-consuming.

```
set.seed(13)
B = 1000
lrtstat = numeric(B)

# parametric bootstrap
system.time(for(b in 1:B){
  y = unlist(simulate(m1_mAqE))
  bnull = lmer(y ~ ID + Date_num + I(Date_num^2) +
    (1|plot_number_year),
    data = dat, REML = FALSE,
    control = lmerControl(optimizer ="Nelder_Mead"))
  balt = lmer(y ~ ID + Date_num + I(Date_num^2) +
    Precip_cum + Precip_7day + Ta_7day + Fsd_7day +
      (1|plot_number_year),
    data = dat, REML = FALSE,
    control = lmerControl(optimizer ="Nelder_Mead"))
  lrtstat[b] = as.numeric(2*(logLik(balt) - logLik(bnull)))
})
```

```
##    user  system elapsed
## 168.911   3.957 172.943
```

The bootstrapped p-value is 0.

```
## p-value
pval = mean(lrtstat > as.numeric(2*(logLik(m1_mAqE_full) - logLik(m1_mAqE))))
pval
```

```
## [1] 0
```

A quicker decision can be made using model selection criteria. Both AIC and BIC select the largest model with random effects for plots and fixed effects for weather variables.

```
## AIC
c(AIC(m1_mAqE_lm), AIC(m1_mAqE), AIC(m1_mAqE_full))
```

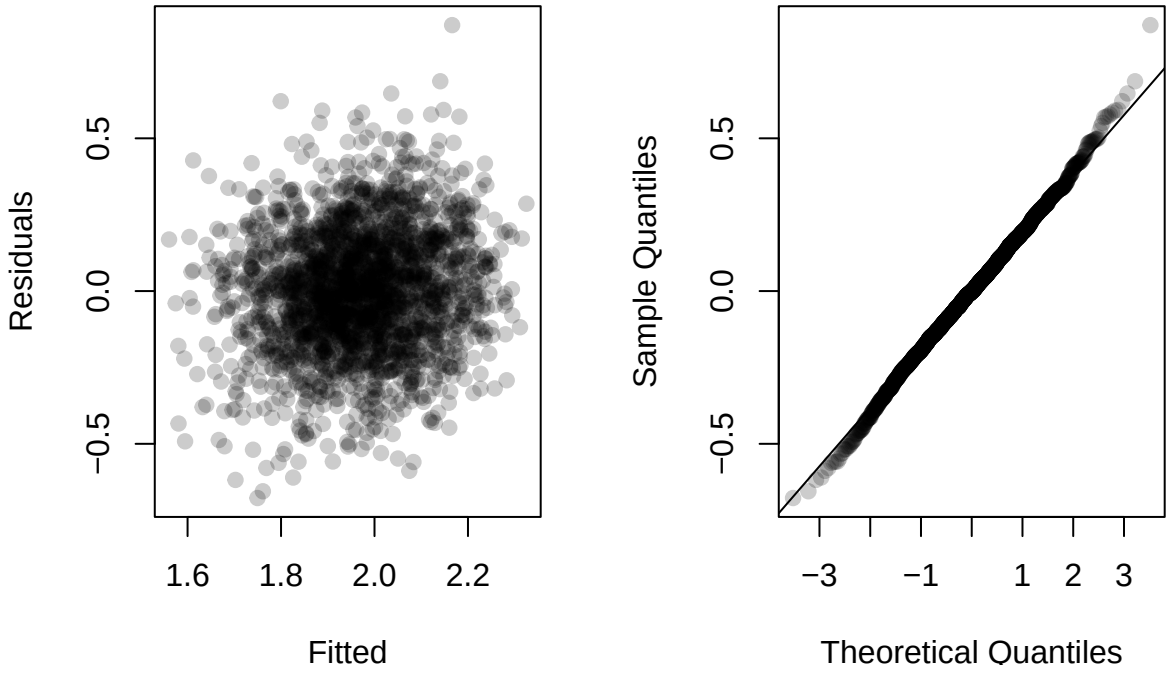
```
## [1] -190.9327 -239.5221 -415.3697
```

```
## BIC
c(BIC(m1_mAqE_lm), BIC(m1_mAqE), BIC(m1_mAqE_full))
```

```
## [1]   62.05648   19.21680 -133.63177
```

We see that the modeling assumptions of constant variance and normality of residuals are satisfied.

```
par(mfrow = c(1,2))
plot(fitted(m1_mAqE_full), residuals(m1_mAqE_full),
         xlab="Fitted", ylab="Residuals", pch = 19,
         col = rgb(0,0,0,alpha=0.2))
a = qqnorm(residuals(m1_mAqE_full), main="", pch = 19,
                        col = rgb(0,0,0,alpha=0.2))
qqline(residuals(m1_mAqE_full))
```



We now investigate which genotypes are associated with AqE values that are different than the RC reference level. To do this we will construct a model with a particular genotype level removed, and then compare the smaller model with the full model selected above. Model comparisons will be made using AIC. We will perform this procedure for each genotype. Results are displayed below, negative values indicate that a genotype is associated with AqE values that are different than the RC reference level.

```
## AIC for each ID variable from full AqE fixed-effects model
M = model.matrix(mAqE ~ ID + Date_num + I(Date_num^2) +
  Precip_cum + Precip_7day + Ta_7day + Fsd_7day,
                   data = dat)

# Note that likelihood ratios are asymptotic, i.e. don't account for
# uncertainty in the estimate of the residual variance
library(parallel)
ncores = detectCores() - 2
```

```
system.time({AIC_IDs = matrix(unlist(lapply(
    grep("ID", colnames(M)), function(j){
    M1 = M[, -j]
    foo = lmer(mAqE ~ -1 + M1 + (1|plot_number_year),
                              data = dat, REML = FALSE,
  control = lmerControl(optimizer ="Nelder_Mead"))
    AIC(m1_mAqE_full) - AIC(foo)
})), ncol = 1)})
```

```
##    user  system elapsed
##   3.247   0.110   3.357
```

```
rownames(AIC_IDs) = colnames(M)[grep("ID", colnames(M))]
colnames(AIC_IDs) = c("AqE")
cbind(round(AIC_IDs,2), ifelse(AIC_IDs < 0, 1, 0))
```

```
##         AqE AqE
## ID1   -0.32   1
## ID2   -0.13   1
## ID3    0.87   0
## ID4    1.17   0
## ID5    1.86   0
## ID6   -0.53   1
## ID7    0.69   0
## ID8   -2.70   1
## ID9   -0.70   1
## ID10   1.66   0
## ID11   1.98   0
## ID12   0.99   0
## ID13  -9.03   1
## ID14   1.73   0
## ID15  -1.44   1
## ID16   1.21   0
## ID17 -12.40   1
## ID18   1.80   0
## ID19  -0.52   1
## ID20   1.90   0
## ID21  -2.86   1
## ID22   1.61   0
## ID23   1.99   0
## ID24   0.86   0
## ID25   1.99   0
## ID26  -4.28   1
## ID27  -3.22   1
## ID28   0.93   0
## ID29  -1.48   1
## ID30  -0.30   1
## ID31   0.43   0
## ID32  -2.84   1
## ID33   1.11   0
## ID34   1.85   0
## ID35   1.90   0
## ID36   1.99   0
## ID37  -1.06   1
```

```
## ID38   0.76   0
## ID39   0.67   0
## ID40  -5.50   1
```

Split plots

Split plot designs originated in agriculture, but also occur in many other settings. A split plot design is as follows:

- One starts with a main whole plot in which a level of a factor variable is applied.
- The main plot is then divided into sections (split plots) and the levels of another factor variable is applied to each of these split plots.

The design arises as a result of restrictions on full randomization. For example, consider an agricultural experiment that studies the effects of plant varieties and irrigation techniques. It may be difficult to vary irrigation techniques over a field while it is easy to plant different plant varieties across a field. To overcome this challenge we consider a split plot design in which a sample of main plots (fields) each receive one irrigation technique at random. Each main plot is then divided into a number of split plots equal to the number of levels of plant varieties under study. Then each plant variety is randomly assigned to each split plot at random. Generally speaking, split plot designs arise when one factor is easy to change and another factor takes much more effort to manipulate. Here is a simple conceptual visual for split plot designs.

Example: irrigation split plot design

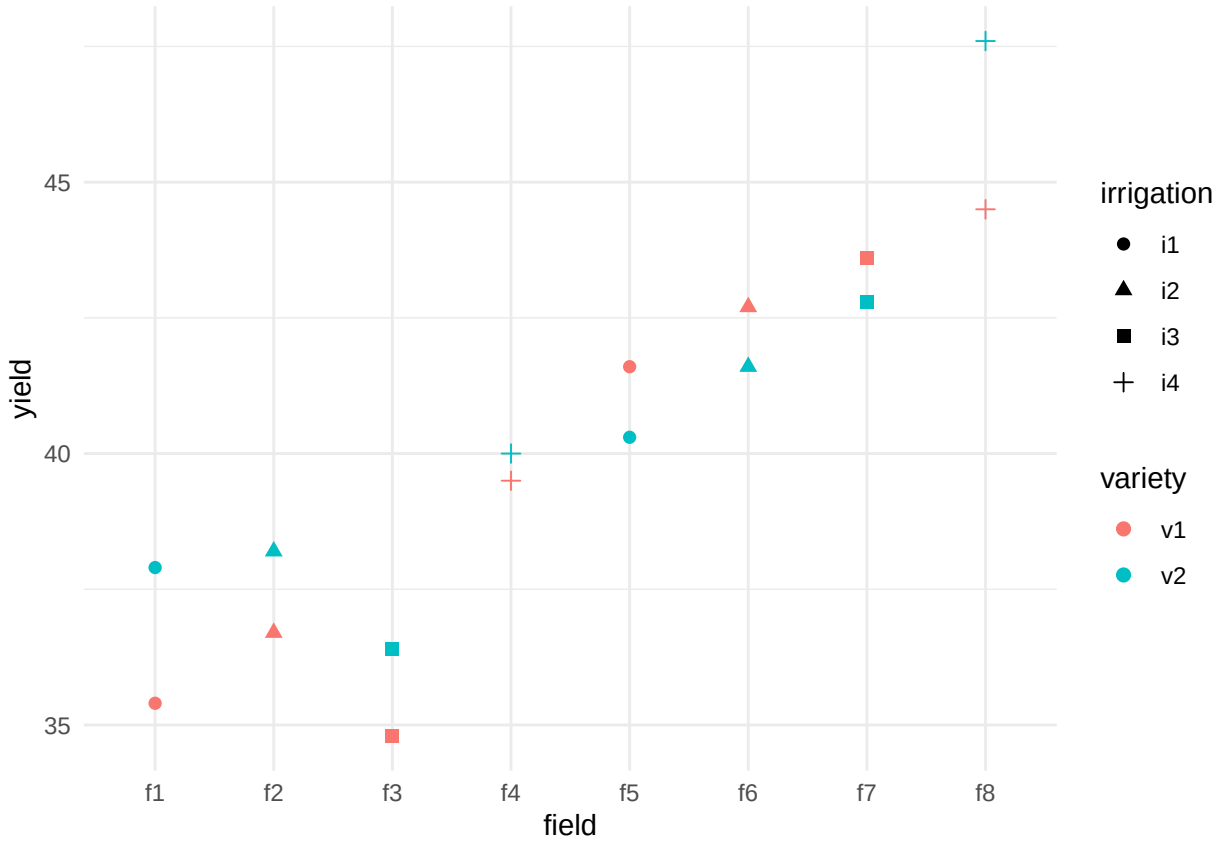
We now perform a data analysis corresponding to the motivating irrigation example. In this example we want to determine which irrigation and plant variety combination produces the highest yield. There are four irrigation methods and two plant varieties under consideration. Eight fields were available, but only one type of irrigation may be applied to each field. The whole plot factor is the irrigation method which should be randomly assigned to each of the fields. The fields may be divided into two halves with each plant variety randomly assigned to each of the halves. A summary and visualization of the data are included below:

```
data(irrigation)

## data summary
summary(irrigation)
```

```
##      field   irrigation variety     yield
##  f1     :2   i1:4       v1:8    Min.   :34.80
##  f2     :2   i2:4       v2:8    1st Qu.:37.60
##  f3     :2   i3:4               Median :40.15
##  f4     :2   i4:4               Mean   :40.23
##  f5     :2                      3rd Qu.:42.73
##  f6     :2                      Max.   :47.60
##  (Other):4
```

```
## data visualization
ggplot(irrigation, aes(y=yield, x=field, shape=irrigation, color= variety)) +
  geom_point(size = 2) +
  theme_minimal()
```



Both irrigation and plant variety are fixed effects, but the field is clearly a random effect (**Why so?**). 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:

$$y_{ijk} = \mu + r_i + v_j + (rv)_{ij} + f_k + (vf)_{jk} + \varepsilon_{ijk},$$

where

- μ is a fixed model intercept,
- r_i is the fixed effect for irrigation that ranges over $i \in \{1, \ldots, 4\}$ levels,
- v_j is the fixed effect for plant variety that ranges over $j \in \{1, 2\}$ levels,
- $(rv)_{ij}$ is the fixed effect interaction between irrigation and plant variety with an index ranging over i and j,
- f_k is the random effect for field for which there are 8 realizations each indexed by k,
- $(vf)_{jk}$ is the random effect interaction between plant variety and field with an index ranging over j and k, and
- ε is an error term for each observation.

We consider normal distributions for the random effect terms that are indexed by single variance parameters σ^2_f, σ^2_{vf}, and σ^2_ε. Note that there is no irrigation and field interaction terms in this model. This is because it would not be possible to estimate such an effect since only one type of irrigation method is assigned to each field; these factors are not crossed. We try to fit this model as follows:

```
lmod = lmer(yield ~ irrigation * variety + (1|field) + (1|field:variety),
            data = irrigation)
```

However, if you try to fit such a model, it will fail because it is not possible to distinguish the variety within 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:

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

```
lmod = lmer(yield ~ irrigation * variety + (1|field),
             data = irrigation)
sumary(lmod)
```

```
## Fixed Effects:
##                       coef.est coef.se
## (Intercept)           40.23     1.47
## irrigation1           -1.42     2.54
## irrigation2           -0.42     2.54
## irrigation3           -0.83     2.54
## variety1              -0.37     0.36
## irrigation1:variety1   0.08     0.63
## irrigation2:variety1   0.27     0.63
## irrigation3:variety1   0.18     0.63
##
## Random Effects:
##  Groups   Name        Std.Dev.
##  field    (Intercept) 4.02
##  Residual             1.45
## ---
## number of obs: 16, groups: field, 8
## AIC = 76.5, DIC = 80.7
## deviance = 68.6
```

We can see that the largest variance component is that due to the field effect, with $\hat{\sigma}_f = 4.03$ compared to $\hat{\sigma}_\varepsilon = 1.45$. 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 a backwards model selection procedure with F-tests with adjusted degrees of freedom:

```
library(pbkrtest)
lmoda = lmer(yield ~ irrigation + variety + (1|field), data=irrigation)
KRmodcomp(lmod, lmoda)
```

```
## large : yield ~ irrigation * variety + (1 | field)
## small : yield ~ irrigation + variety + (1 | field)
##         stat    ndf    ddf F.scaling p.value
## Ftest 0.2452 3.0000 4.0000         1  0.8612
```

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

```
lmodi = lmer(yield ~ irrigation + (1|field), irrigation)
KRmodcomp(lmoda, lmodi)
```

```
## large : yield ~ irrigation + variety + (1 | field)
## small : yield ~ irrigation + (1 | field)
##         stat    ndf    ddf F.scaling p.value
## Ftest 1.5782 1.0000 7.0000         1  0.2493
```

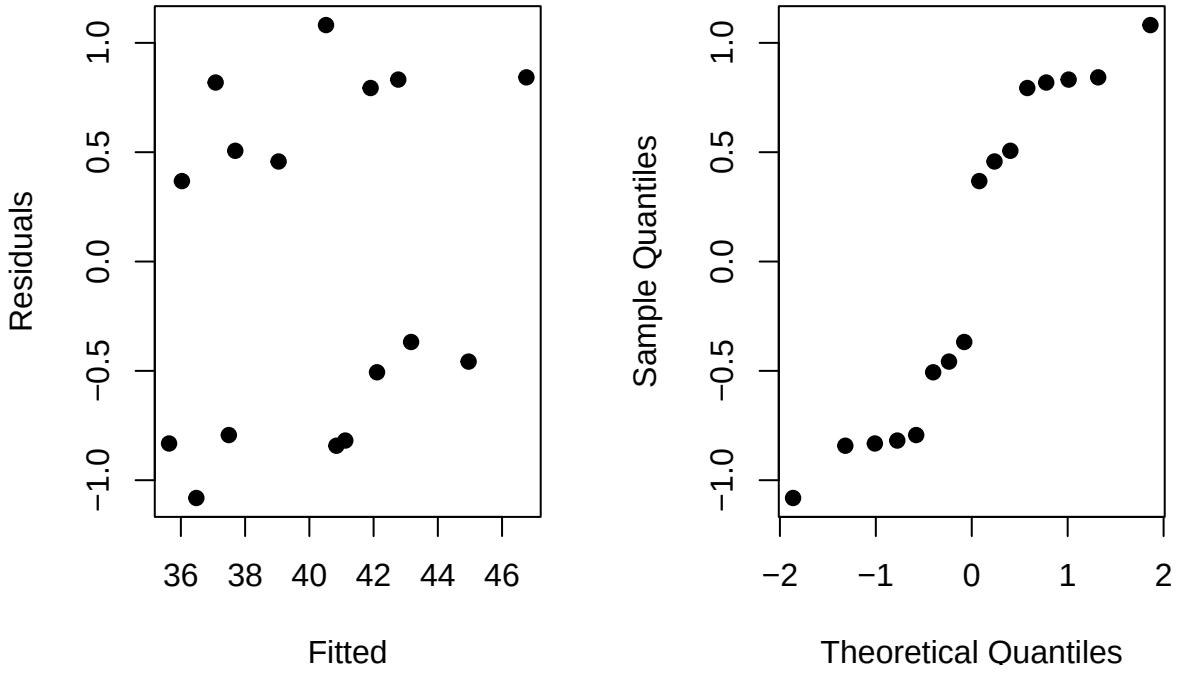
Dropping variety from the model seems reasonable since the p-value of 0.25 is large. We can test irrigation in a similar manner:

```
lmodv = lmer(yield ~ variety + (1|field), irrigation)
KRmodcomp(lmoda, lmodv)
```

```
## large : yield ~ irrigation + variety + (1 | field)
## small : yield ~ variety + (1 | field)
##         stat    ndf    ddf F.scaling p.value
## Ftest 0.3882 3.0000 4.0000         1  0.7685
```

Irrigation also fails to be significant. We should check the diagnostic plots to make sure there is nothing amiss:

```
par(mfrow = c(1,2))
plot(fitted(lmod), residuals(lmod), xlab="Fitted", ylab="Residuals", pch = 19)
qqnorm(residuals(lmod), main="", pch = 19)
```



We can see 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 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 = 0.0094
```

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

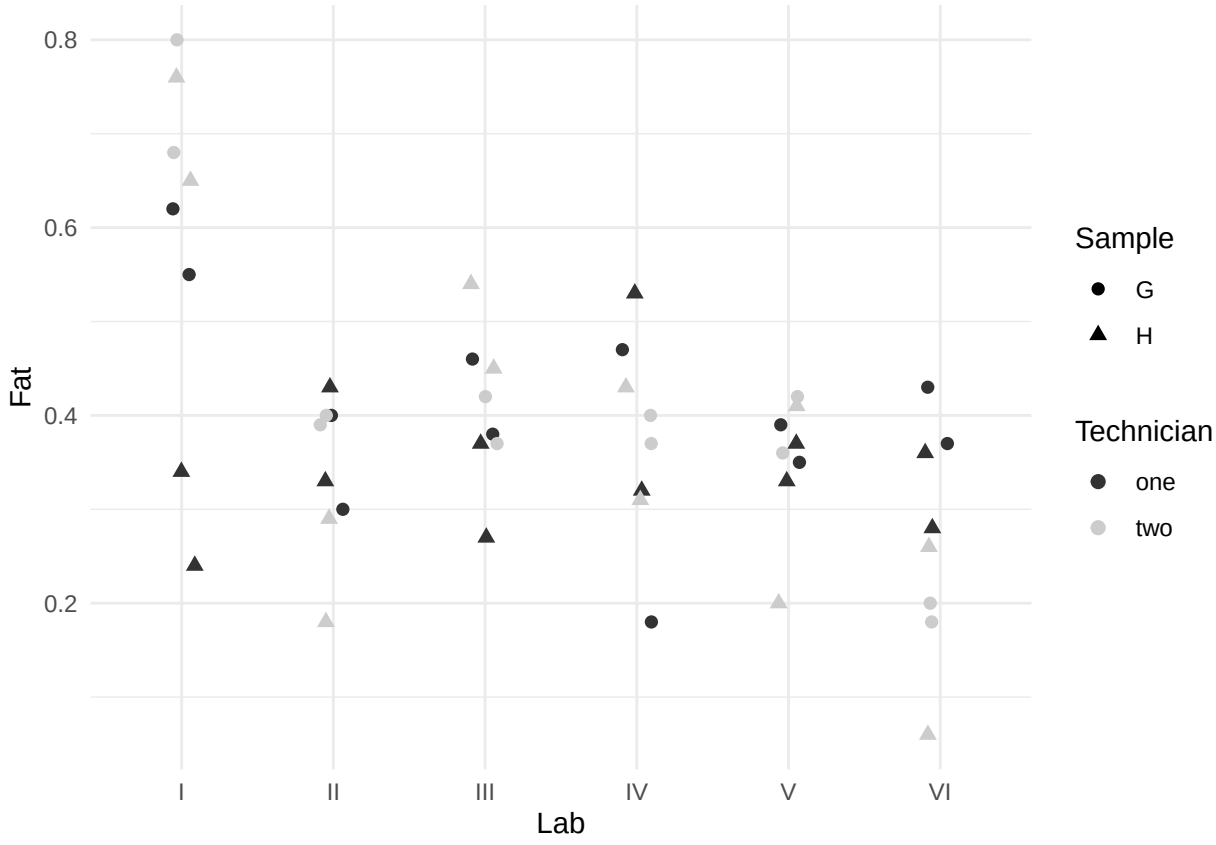
Nested random effects via an example

In this Section we provide an example of 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)
summary(eggs)
```

```
##       Fat          Lab    Technician Sample
##  Min.   :0.0600   I  :8   one:24     G:24
##  1st Qu.:0.3075   II :8   two:24     H:24
##  Median :0.3700   III:8
##  Mean   :0.3875   IV :8
##  3rd Qu.:0.4300   V  :8
##  Max.   :0.8000   VI :8
```

```
ggplot(eggs, aes(y=Fat, x=Lab, color=Technician, shape=Sample)) +
  geom_point(size = 2, position = position_jitter(width=0.1, height=0.0)) +
  scale_color_grey() +
  theme_minimal()
```



Note that 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, note that 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:

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

where,

- μ is a fixed model intercept,
- L_i is the random effect for lab for which there are 6 realizations each indexed by i,
- T_{ij} is the random effect for technician nested within lab for which there are 2 realizations each indexed by j within each lab i,
- S_{ijk} is the random effect for sample nested within technician nested within lab. There are four realizations each indexed by k by each technician j within each lab i, and
- ε is an error term for each observation.

This can be fit with:

```
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 that the estimated variance components are all similar in magnitude. The lack of consistency in measures of fat content can be ascribed to variance between labs, technicians, measurement due to different labeling, and measurement error.

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 %
## .sig01      0.00000000 0.09564357
## .sig02      0.00000000 0.13957263
```

```
## .sig03      0.00000000 0.14948989
## .sigma      0.06081876 0.10674937
## (Intercept) 0.30653979 0.47883284
```

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.

Repeated measures and longitudinal data

In repeated measures designs there are several individuals (or units) under study and multiple measurements are taken repeatedly on each individual. When these repeated measurements are taken over time, it is called a longitudinal study or, in some applications, a panel study. Typically there are various covariates concerning the individual that are also recorded and interest centers on how the response depends on these covariates over time. Often it is reasonable to believe that the response of each individual has several components:

- a fixed effect, which is a function of the covariates;
- a random effect, which expresses the variation between individuals;
- and an error, which is due to measurement or unrecorded variables.

In a repeated measures design we will suppose that each individual has a response y_i which is now a vector of length n_i and is modeled conditionally on the random effect b_i as

$$y_i|b_i \sim N(X_i\beta + Z_ib_i, \sigma^2\Lambda_i).$$

Note that this is a very similar model used in the linear mixed-effects model construction above, with the exception that we now allow for the individuals to have a more general covariance structure Λ_i. As before, we will assume that $b_i \sim N(0, \sigma^2 D)$ so that

$$y_i \sim N(X_i\beta, \Sigma_i)$$

where $\Sigma_i = \sigma^2(\Lambda_i + Z_iDZ_i^T)$. Now suppose that we have N individuals and assume that the measurement errors and random effects between individuals are uncorrelated. Then we can combine the data as:

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \qquad X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix}, \qquad B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix},$$

and

- $\tilde{D} = \text{diag}(D, D, \ldots, D)$,
- $Z = \text{diag}(Z_1, Z_2, \ldots, Z_N)$,
- $\Sigma = \text{diag}(\Sigma_1, \Sigma_2, \ldots, \Sigma_N)$, and
- $\Lambda = \text{diag}(\Lambda_1, \Lambda_2, \ldots, \Lambda_N)$, where

$y_i \in \mathbb{R}^{n_i}$, $X_i \in \mathbb{R}^{n_i \times p}$, $\beta \in \mathbb{R}^p$, $b_i \in \mathbb{R}^q$, $Z_i \in \mathbb{R}^{n_i \times q}$, and the rest of the quantities in the above follow from these specifications. With this setup we can write the model as

$$Y \sim N(X\beta, \Sigma) \qquad \text{where} \qquad \Sigma = \sigma^2(\Lambda + Z\tilde{D}Z^T).$$

The log-likelihood for the data is then computed as previously and estimation, testing, standard errors and confidence intervals all follow using standard likelihood theory as before. There is no strong distinction between repeated measures methodology and the methodology of linear mixed-effects models.

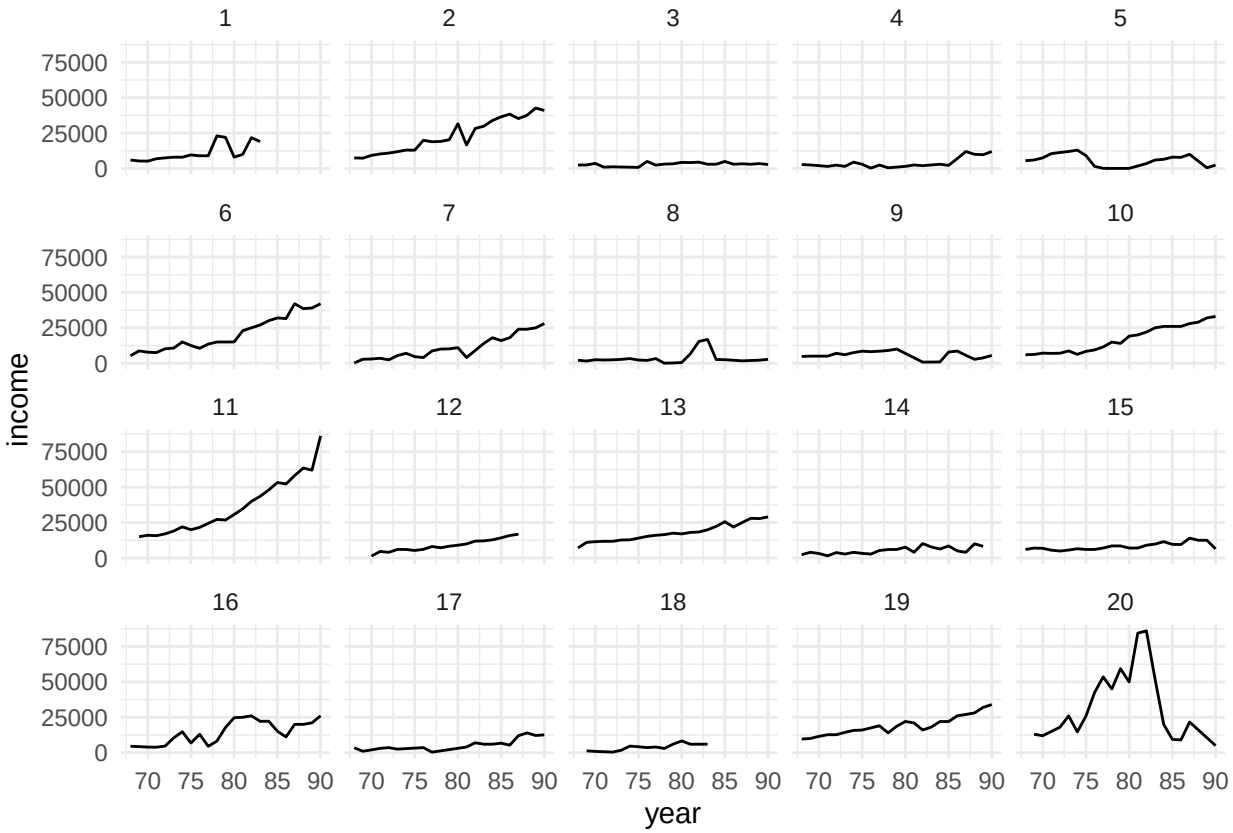
Example: Panel Study of Income Dynamics

The Panel Study of Income Dynamics (PSID), begun in 1968, is a longitudinal study of a representative sample of U.S. individuals described in Hill (1992). The study is conducted at the Survey Research Center, Institute for Social Research, University of Michigan, and is still continuing. There are currently 8700 households in the study and many variables are measured. We chose to analyze a random subset of this data, consisting of 85 heads of household who were aged 25–39 in 1968 and had complete data for at least 11 of the years between 1968 and 1990. The variables included were annual income, sex, years of education and age in 1968:

```
library(tidyverse)
data(psid)
```

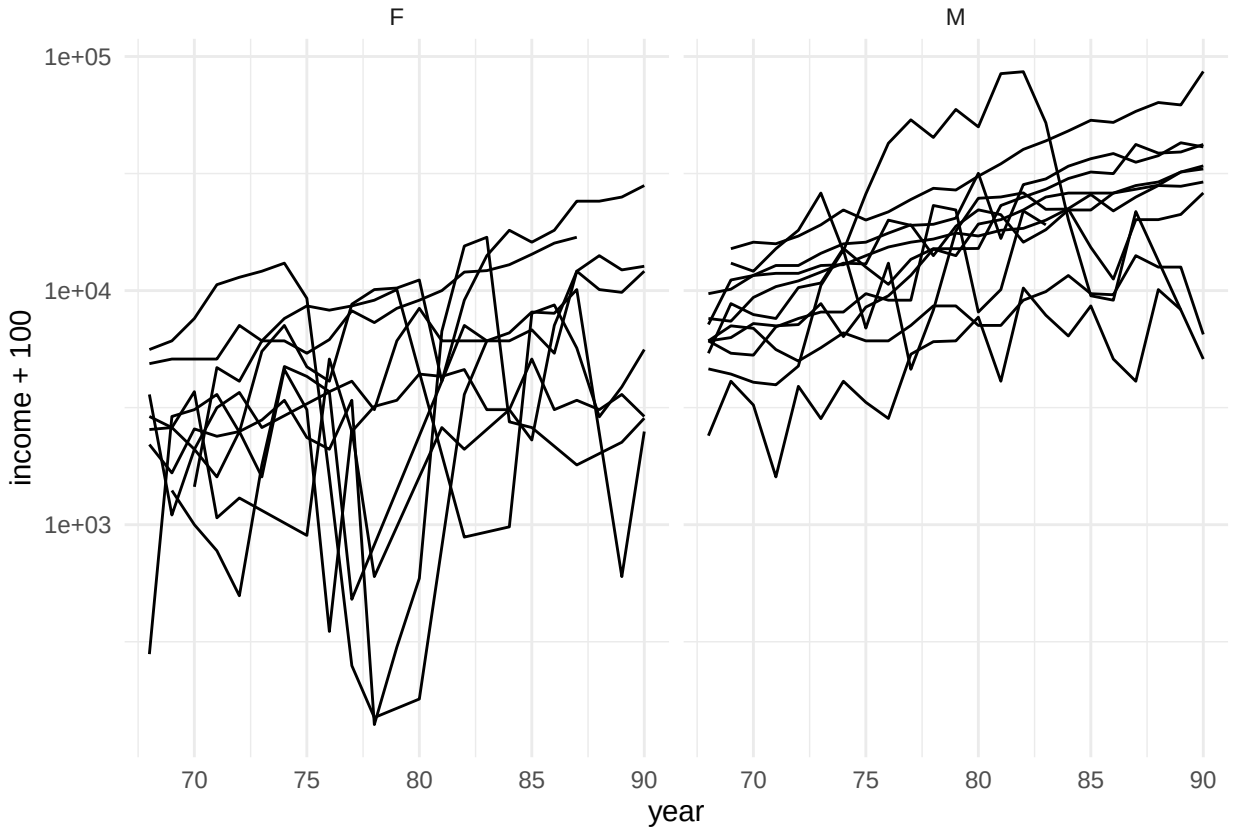
Now plot the data for the first 20 individuals:

```
psid20 = filter(psid, person <= 20)
ggplot(psid20, aes(x=year, y=income)) +
  geom_line() +
  facet_wrap(~ person) +
  theme_minimal()
```



We see that some individuals have a slowly increasing income, typical of someone in steady employment in the same job. Other individuals have more erratic incomes. We can also show how the incomes vary by sex. Income is more naturally considered on a log-scale:

```
ggplot(psid20, aes(x=year, y=income+100, group=person)) +
  geom_line() +
  facet_wrap(~ sex) +
  scale_y_log10() +
  theme_minimal()
```



We added \$100 to the income of each subject to remove the effect of some subjects having very low incomes for short periods of time. These cases distorted the plots without the adjustment. We see that men's incomes are generally higher and less variable while women's incomes are more variable, but are perhaps increasing more quickly. We could fit a line to each subject starting with the first:

```
lmod = lm(log(income) ~ I(year-78), subset=(person==1), psid)
coef(lmod)
```

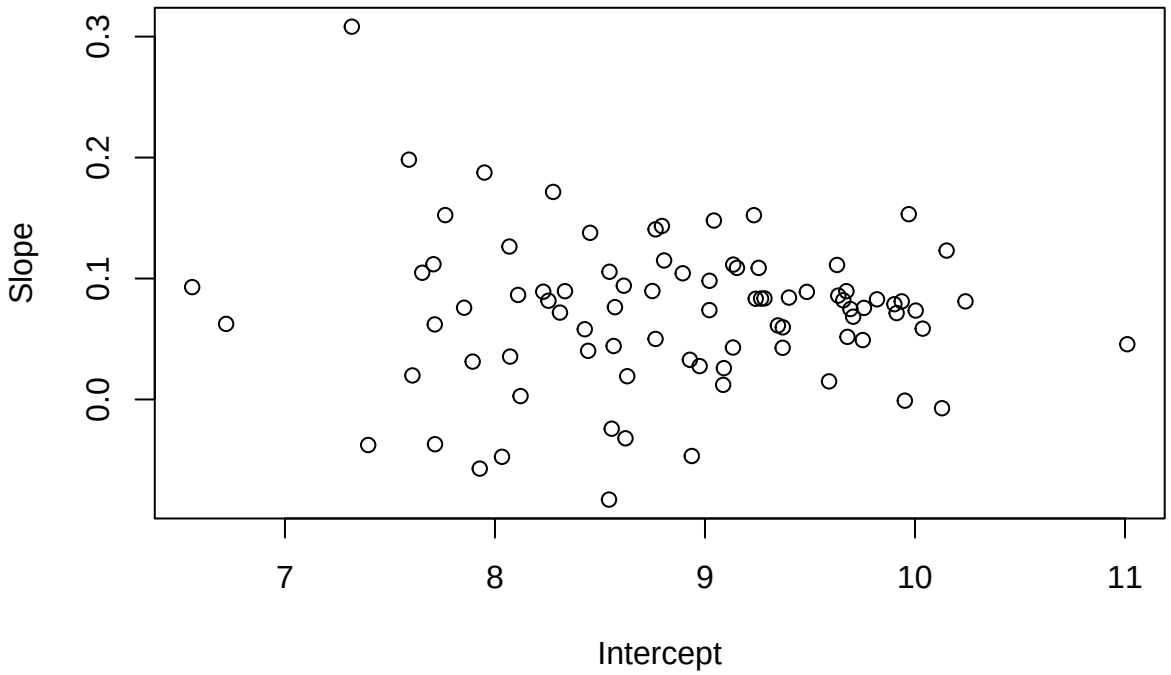
```
##  (Intercept) I(year - 78)
##    9.3999568    0.0842667
```

Note that we have centered the predictor year at its median value so that the intercept will represent the predicted log income in 1978 and not the year 1900 which is nonsense (any sensible location transformation would be appropriate, but centering with respect to the median is perhaps most appropriate). We now fit a line for all the subjects and plot the results:

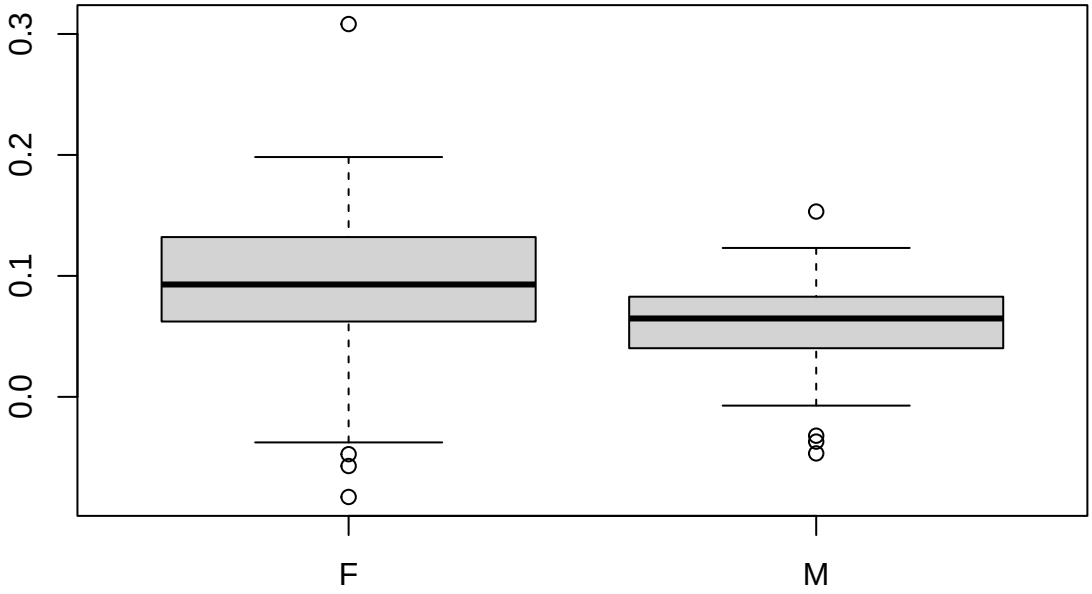
```
ml = lmList(log(income) ~ I(year-78) | person, psid)
intercepts = sapply(ml,coef)[1,]
slopes = sapply(ml,coef)[2,]
```

The `lmList` command fits a linear model to each group within the data, here specified by person. A list of linear models, one for each group, is returned from which we extract the intercepts and slopes.

```
plot(intercepts,slopes,xlab="Intercept",ylab="Slope")
```




```
psex = psid$sex[match(1:85,psid$person)]
boxplot(split(slopes,psex))
```



In the first panel, we see how the slopes relate to the intercepts — there is little correlation. This means we can test incomes and income growths separately. In the second panel, we compare the income growth rates where we see these as higher and more variable for women compared to men. We can test the difference in income growth rates for men and women:

```
t.test(slopes[psex=="M"],slopes[psex=="F"])
```

```
##
##  Welch Two Sample t-test
##
## data:  slopes[psex == "M"] and slopes[psex == "F"]
```

```
## t = -2.3786, df = 56.736, p-value = 0.02077
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.05916871 -0.00507729
## sample estimates:
##  mean of x  mean of y
## 0.05691046 0.08903346
```

We see that women have a significantly higher growth rate than men. We can also compare the incomes at the intercept (which is 1978):

```
t.test(intercepts[psex=="M"],intercepts[psex=="F"])
```

```
##
##  Welch Two Sample t-test
##
## data:  intercepts[psex == "M"] and intercepts[psex == "F"]
## t = 8.2199, df = 79.719, p-value = 3.065e-12
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.8738792 1.4322218
## sample estimates:
## mean of x mean of y
##  9.382325  8.229275
```

We see that men have significantly higher incomes.

This is an example of a response feature analysis. It requires choosing an important characteristic. We have chosen two here: the slope and the intercept. For many datasets, this is not an easy choice and at least some information is lost by doing this. Response feature analysis is attractive because of its simplicity. By extracting a univariate response for each individual, we are able to use a wide array of well-known statistical techniques. However, it is not the most efficient use of the data as all the additional information besides the chosen response feature is discarded. Notice that having additional data on each subject would be of limited value.

Suppose that the income change over time can be partly predicted by the subject's age, sex and educational level. We do not expect a perfect fit. The variation may be partitioned into two components. Clearly there are other factors that will affect a subject's income. These factors may cause the income to be generally higher or lower or they may cause the income to grow at a faster or slower rate. We can model this variation with a random intercept and slope, respectively, for each subject. We also expect that there will be some year-to-year variation within each subject. For simplicity, let us initially assume that this error is homogeneous and uncorrelated, that is, $\Lambda_i = I$. We also center the year to aid interpretation as before. We may express these notions in the model:

```
psid$cyear = psid$year - 78
mmod = lmer(log(income) ~ cyear*sex + age + educ + (cyear|person), psid)
```

This model can be written as:

$$\log(\text{income})_{ij} = \mu + \beta_y \text{year}_j + \beta_g \text{sex}_i + \beta_{yg} \text{sex}_i * \text{year}_j + \beta_e \text{education}_i + \beta_a \text{age}_i + \gamma_i^0 + \gamma_i^1 \text{year}_j + \varepsilon_{ij}$$

where i indexes the individuals and j indexes the years, and log is the natural logarithm. We have:

$$\begin{pmatrix} \gamma_j^0 \\ \gamma_j^1 \end{pmatrix} \sim N(0, \sigma^2 D)$$

The model summary is:

```
sumary(mmod, digits=3)
```

```
## Fixed Effects:
##             coef.est coef.se
## (Intercept)  7.249    0.540
## cyear        0.072    0.006
## sex1        -0.575    0.061
## age          0.011    0.014
## educ         0.104    0.021
## cyear:sex1   0.013    0.006
##
## Random Effects:
##  Groups   Name        Std.Dev. Corr
##  person   (Intercept) 0.531
##           cyear       0.049    0.187
##  Residual             0.684
## ---
## number of obs: 1661, groups: person, 85
## AIC = 3842.5, DIC = 3748.4
## deviance = 3785.5
```

We now analyze this summary table. Lets start with the fixed effects. We see that income increases about 10% for each additional year of education ($\exp(0.104) \approx 0.104$). We see that age does not appear to be significant. For females, the reference level in this example, income increases about 8.5% a year, while for men, it increases about 8.5% - 2.9% = 5.9% a year. We see that, for this data, the incomes of women are $\exp(1.150) = 3.16$ times higher (far higher!). Now the random effects. We know the mean for males and females, but individuals will vary about this. The standard deviation for the intercept and slope are 0.531 and 0.049 ($\sigma\sqrt{D_{11}}$ and $\sigma\sqrt{D_{22}}$), respectively. These have a correlation of 0.187 ($\text{cor}(\gamma^0, \gamma^1)$). Finally, there is some additional variation in the measurement not so far accounted for having standard deviation of 0.684 ($\text{sd}(\varepsilon_{ij})$). We see that the variation in increase in income is relatively small while the variation in overall income between individuals is quite large. Furthermore, given the large residual variation, there is a large year-to-year variation in incomes.

How to fit a mixed-effects model using `lme4`

This Section will follow the package vignette corresponding to the `lme4` package. This vignette can be seen here. We will consider the simple mixed-effects structure

$$\begin{aligned} Y|b &\sim N(X\beta + Zb, \sigma^2 I), \\ b &\sim N(0, \Sigma), \end{aligned}$$

although it should be noted that the vignette considers a slightly more general formulation involving known mean offsets o and known variance weights W, and that much of the materials presented here are aimed at convenience, computational efficiency, and robustness.

It is convenient to express Σ as

$$\Sigma_\theta = \sigma^2 \Lambda_\theta \Lambda_\theta^T$$

where $\Lambda_\theta \in \mathbb{R}^{q \times q}$ is known as a *relative covariance factor* depending on a *variance-component parameter* vector θ.

The following table indexes the dimensions of linear mixed-effects models, the subscript $i = 1, \ldots, k$ denotes a specific random-effects term:

Symbol	Size
n	Length of the response vector Y
p	Number of columns of fixed-effects model matrix X
k	Number of random-effects terms
p_i	Number of columns of the raw model matrix X_i
l_i	Number of levels of the grouping factor indices i_i
$q_i = p_i l_i$	Number of columns of the term-wise model matrix Z_i
$q = \sum_i^k q_i$	Number of columns of random-effects model matrix Z
$m_i = \binom{p_i+1}{2}$	Number of covariance parameters for term i
$m = \sum_i^k m_i$	Total number of covariance parameters

The following table denotes the symbols used to describe the structure of the random-effects model matrix and the relative covariance factor. The subscript $i = 1, \ldots, k$ denotes a specific random-effects term:

Symbol	Size	Description
X_i	$n \times p_i$	Raw random-effects model matrix
J_i	$n \times l_i$	Indicator matrix of grouping factor indices
X_{ij}	$p_i \times 1$	Column vector containing jth row of X_i
J_{ij}	$l_i \times 1$	Column vector containing jth row of J_i
i_i	n	Vector of grouping factor indices
Z_i	$n \times q_i$	Term-wise random-effects model matrix
θ	m	Number of covariance parameters
T_i	$p_i \times p_i$	Lower triangular template matrix
Λ_i	$q_i \times q_i$	Term-wise relative covariance factor

It is important to note that X_i is a raw-random effects matrix and **NOT** a subpart of the fixed-effects model matrix X. This is a notational choice used in the lme4 vignette.

Algebraic and computational account of mixed-model formulas

The fixed-effects terms of a mixed-model formula are parsed to produce the fixed-effects model matrix, X, in the same way that the R `lm` function generates model matrices.

However, a mixed-model formula incorporates $k \geq 1$ random-effects terms $(r|f)$ as well. These k terms are used to produce the random-effects model matrix Z, and the structure of the relative covariance factor Λ_θ, which are matrices that typically have a sparse structure.

We now describe how one might construct these matrices from the random-effects terms, considering first a single term (written as $(r|f)$ in a formula for a linear mixed model) and then generalizing to multiple terms. The above tables summarize the matrices and vectors that determine the structure of Z and Λ_θ.

The expression r in $(r|f)$ is a linear model formula that evaluates to an R model matrix X_i of size $n \times p_i$, called the raw random-effects model matrix for term i. A term is said to be a *scalar* random-effects term when $p_i = 1$, otherwise it is *vector-valued*. For a simple, scalar random-effects term of the form $(1|f)$, X_i is the $n \times 1$ matrix of ones, which implies a random intercept model.

The expression f evaluates to an R factor, called the grouping factor, for the term. For the ith term, we represent this factor mathematically with a vector i_i of factor indices, which is an n-vector of values from $1, \ldots, l_i$ where l_i is the number of levels of the grouping factor variable f in the ith random effects term. Let J_i be the $n \times l_i$ matrix of indicator columns for i_i. Using the `Matrix` package (Bates and Maechler 2015) in R, we may construct the transpose of J_i from a factor vector, f, by coercing f to a 'sparseMatrix' object. For example,

```
library(Matrix)
# first argument is the number of levels
# second argument is the number of replications
(f = gl(3, 2))
```

```
## [1] 1 1 2 2 3 3
## Levels: 1 2 3
```

```
(Ji = t(as(f, Class = "sparseMatrix")))
```

```
## 6 x 3 sparse Matrix of class "dgCMatrix"
##      1 2 3
## [1,] 1 . .
## [2,] 1 . .
## [3,] . 1 .
## [4,] . 1 .
## [5,] . . 1
## [6,] . . 1
```

When $k > 1$ we order the random-effects terms so that $l_1 \geq l_2 \geq \cdots \geq l_k$; in general, this ordering reduces "fill-in" (i.e., the proportion of elements that are zero in the lower triangle of $\Lambda_\theta^T Z^T Z \Lambda_\theta + I$ but not in the lower triangle of its left Cholesky factor L_θ). This reduction in fill-in provides more efficient matrix operations within the penalized least squares algorithm described below.

Examples: The PSID example above considered $k = 1$ term `(cyear|person)` which corresponded to a random intercept and a random slope for an individual's income over time. In this example $X_i, i = k = 1$, with $p_i = 2$ corresponding to a model matrix consisting of an intercept and a column for years centered around 1978, the median value, and J_i provides indexes for each individual in the study. The Bliss (1967) example above concerning consistency of measurements considered $k = 3$ terms `(1|Lab)`,`(1|Lab:Technician)`, and `(1|Lab:Technician:Sample)`. Here, $X_i, i = 1, 2, 3$, with $p_i = 1, i = 1, 2, 3$, corresponding to intercept vectors and the J_i matrices consider separate grouping factors which reflect the nested random effects.

Constructing the random-effects model matrix

The ith random-effects term contributes $q_i = l_i p_i$ columns to the model matrix Z. These columns are grouped into a matrix Z_i, which is referred to as the *term-wise model matrix* for the ith term. Thus q, the number of columns in Z and the dimension of the random variable b is

$$q = \sum_{i=1}^{k} q_i = \sum_{i=1}^{k} l_i p_i.$$

Creating the matrix Z_i from X_i and J_i is straightforward but is somewhat awkward to describe. Consider Z_i as being further decomposed into l_i blocks of p_i columns. The rows in the first block are the rows of X_i multiplied by the 0/1 values in the first column of J_i and similarly for the subsequent blocks. With these definitions we may define the term-wise random-effects model matrix Z_i for the ith term as a transposed Khatri-Rao product,

$$Z_i = (J_i^T * X_i^T)^T = \begin{bmatrix} J_{i1}^T \otimes X_{i1}^T \\ J_{i2}^T \otimes X_{i2}^T \\ \vdots \\ J_{in}^T \otimes X_{in}^T \end{bmatrix},$$

where $*$ and $\otimes$ are, respectively, the Khatri-Rao and Kronecker products, and J_{ij}^T and X_{ij}^T are row vectors of the the jth rows of J_i and X_i. These rows correspond to the jth sample in the response vector Y and j runs from 1 to n. The `Matrix` package in R contains a `KhatriRao` function, which can be used to form Z_i. For example, if we begin with a raw model matrix,

```
(Xi = cbind(1, rep.int(c(-1, 1), 3L)))
```

```
##      [,1] [,2]
## [1,]    1   -1
```

```
## [2,]    1    1
## [3,]    1   -1
## [4,]    1    1
## [5,]    1   -1
## [6,]    1    1
```

```
(Zi = t(KhatriRao(t(Ji), t(Xi))))
```

```
## 6 x 6 sparse Matrix of class "dgCMatrix"
##
## [1,] 1 -1 .  . .  .
## [2,] 1  1 .  . .  .
## [3,] .  . 1 -1 .  .
## [4,] .  . 1  1 .  .
## [5,] .  . .  . 1 -1
## [6,] .  . .  . 1  1
```

In particular, for a simple, scalar term, Z_i is formed by element-wise multiplication of the single column of X_i by each of the columns of J_i. Because each Z_i is generated from indicator columns, its cross-product, $Z_i^T Z_i$ is block diagonal consisting of l_i diagonal blocks each of size p_i. Note that this means that when $k = 1$, $Z^T Z$ will be block diagonal. These block-diagonal properties allow for more efficient sparse matrix computations.

The full random-effects model matrix Z is constructed from $k \geq 1$ blocks,

$$Z = [Z_1 \; Z_2 \; \cdots \; Z_k].$$

A more detailed look at Z is gvien through its transpose where the columns represent samples and the rows represent random effect terms:

$$Z^T = \begin{bmatrix} J_{11} \otimes X_{11} & J_{12} \otimes X_{12} & \cdots & J_{1n} \otimes X_{1n} \\ J_{21} \otimes X_{21} & J_{22} \otimes X_{22} & \cdots & J_{2n} \otimes X_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ J_{k1} \otimes X_{k1} & J_{k2} \otimes X_{k2} & \cdots & J_{kn} \otimes X_{kn} \end{bmatrix}.$$

Note that the proportion of elements of Z^T that are structural zeros is

$$\frac{\sum_{i=1}^{k} p_i(l_i - 1)}{\sum_{i=1}^{k} p_i}.$$

Therefore, the sparsity of Z^T increases with the number of grouping factor levels. As the number of levels is often large in practice, it is essential for speed and efficiency to take account of this sparsity, for example by using sparse matrix methods, when fitting mixed models.

Constructing the relative covariance factor

The $q \times q$ covariance factor Λ_θ is a block diagonal matrix whose ith diagonal block Λ_i is of size q_i, $i = 1, \ldots, k$. We refer to Λ_i as the *term-wise relative covariance factor*. Furthermore, Λ_i is a homogeneous block diagonal matrix with each of the l_i lower-triangular blocks on the diagonal being a copy of a $p_i \times p_i$ lower-triangular template matrix T_i. The covariance parameter vector θ of length $m_i = \binom{p_i+1}{2}$, $i = 1, \ldots, k$, consists of the elements in the lower triangle of T_i. To provide a unique representation we require that the diagonal elements of the T_i terms to be non-negative.

More details are given in the lme4 vignette.

Objective function

In the initial parameterization of the linear mixed-effects model, the covariance parameter θ appears only in the marginal distribution of the random effects through the variance matrix Σ_θ. However, from the

perspective of computational stability and efficiency, it is advantageous to reformulate the model such that θ appears only in the conditional distribution for the response vector given the random effects. Such a reparameterization allows us to work with singular covariance matrices, which regularly arise in practice.

This reparameterization is made by defining a *spherical random-effects* variable U with distribution

$$U \sim N(0, \sigma^2 I_q).$$

If we set

$$b = \Lambda_\theta U,$$

then b will have the assumed marginal random-effects distribution in our setup. The conditional distribution of the response vector given the random effects may now be reformulated as,

$$Y|U = u \ \sim \ N(\mu_{Y|U=u}, \sigma^2),$$

where

$$\mu_{Y|U=u} = X\beta + Z\Lambda_\theta u$$

is a vector of linear predictors, which can be interpreted as a conditional mean (or mode in the case of normality). Similarly, we also define $\mu_{U|Y=y_{\text{obs}}}$ as the conditional mean (or mode in the case of normality) of the spherical random effects given the observed response vector. Note also that we use the u symbol throughout to represent a specific value of the random variable U.

Penalized least squares (PLS)

The computational methods for maximum likelihood estimation of the mixed-effects model involve repeated applications of the PLS method. In particular, the PLS problem is to minimize the penalized residual sum-of-squares,

$$r^2(\theta, \beta, u) = \rho^2(\theta, \beta, u) + \|u\|^2$$

over $[u^T, \beta^T]^T$, where

$$\rho^2(\theta, \beta, u) = \|y_{\text{obs}} - \mu_{Y|U=u}\|^2.$$

This notation makes explicit that the minimization depends on θ, β, and u. The penalization term $\|u\|^2$ penalizes models with larger magnitude values of u. In the so-called "pseudo-data" approach we write the penalized weighted residual sum-of-squares as the squared length of a block matrix equation,

$$r^2(\theta, \beta, u) = \left\| \begin{bmatrix} y_{\text{obs}} \\ 0 \end{bmatrix} - \begin{bmatrix} Z\Lambda_\theta & X \\ I_q & 0 \end{bmatrix} \begin{bmatrix} u \\ \beta \end{bmatrix} \right\|^2.$$

This pseudo-data approach shows that the PLS problem may also be thought of as a standard least squares problem for an extended response vector, which implies that the minimizing value $(\mu^T_{U|Y=y_{\text{obs}}}, \ \hat{\beta}^T_\theta)^T$ satisfies the normal equations,

$$\begin{bmatrix} \Lambda_\theta^T Z^T y_{\text{obs}} \\ X^T y_{\text{obs}} \end{bmatrix} = \begin{bmatrix} \Lambda_\theta^T Z^T Z \Lambda_\theta + I & \Lambda_\theta^T Z^T X \\ X^T Z \Lambda_\theta & X^T X \end{bmatrix} \begin{bmatrix} \mu_{U|Y=y_{\text{obs}}} \\ \hat{\beta}_\theta \end{bmatrix},$$

where it is again noted that $\mu_{U|Y=y_{\text{obs}}}$ is the conditional mean of the random effects U given $Y = y_{\text{obs}}$. Note that this conditional mean also depends on θ, although is not made explicit. We can perform a Cholesky decomposition on the above cross-product matrix, so that

$$\begin{bmatrix} \Lambda_\theta^T Z^T Z \Lambda_\theta + I & \Lambda_\theta^T Z^T X \\ X^T Z \Lambda_\theta & X^T X \end{bmatrix} = \begin{bmatrix} L_\theta & 0 \\ R_{ZX}^T & R_X^T \end{bmatrix} \begin{bmatrix} L_\theta^T & R_{ZX} \\ 0 & R_X \end{bmatrix}.$$

We may use this decomposition to rewrite the penalized weighted residual sum-of-squares as,

$$r^2(\theta, \beta, u) = r^2(\theta) + \|L_\theta^T(u - \mu_{U|Y=y_{\text{obs}}}) + R_{ZX}(\beta - \hat{\beta}_\theta)\|^2 + \|R_X(\beta - \hat{\beta}_\theta)\|^2.$$

The above derivation is an important expression in the underlying `lme4`. It relates the penalized residuals sum-of-squares with its minimum value and is extremely useful when we integrate out the random effects, which is required for maximum likelihood estimation. Note that we have simplified notation by writing $r^2(\theta, \hat{\beta}_\theta, \mu_{U|Y=y_{\text{obs}}})$ as $r^2(\theta)$.

Probability densities

The residual sums-of-squares discussed in the previous section can be used to express various probability densities, which are required for maximum likelihood estimation,

$$\begin{aligned} f_{Y|U}(y_{\text{obs}}|u) &= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{\rho^2(\theta,\beta,u)}{2\sigma^2}\right), \\ f_U(u) &= \frac{1}{(2\pi\sigma^2)^{q/2}} \exp\left(-\frac{\|u\|^2}{2\sigma^2}\right), \\ f_{Y,U}(y_{\text{obs}},u) &= \frac{1}{(2\pi\sigma^2)^{(n+q)/2}} \exp\left(-\frac{r^2(\theta,\beta,u)}{2\sigma^2}\right), \\ f_{U|Y}(u|y_{\text{obs}}) &= \frac{f_{Y,U}(y_{\text{obs}},u)}{f_Y(y_{\text{obs}})} \end{aligned}$$

where

$$f_Y(y_{\text{obs}}) = \int f_{Y,U}(y_{\text{obs}},u)du.$$

The log-likelihood to be maximized can therefore be written as,

$$\mathcal{L}(\theta,\beta,\sigma^2|y_{\text{obs}}) = \log f_Y(y_{\text{obs}}).$$

The integral representation of $f_Y(y_{\text{obs}})$ is explicitly written as,

$$\begin{aligned} f_Y(y_{\text{obs}}) = \frac{1}{(2\pi\sigma^2)^{(n+q)/2}} \exp\left[\frac{-r^2(\theta) - \|R_X(\beta-\hat{\beta}_\theta)\|^2}{2\sigma^2}\right] \\ \times \int \exp\left[-\frac{\|L_\theta^T(u-\mu_{U|Y=y_{\text{obs}}}) + R_{ZX}(\beta-\hat{\beta}_\theta)\|^2}{2\sigma^2}\right] du. \end{aligned}$$

The term inside the norm of the exponent being integrated is affine in u. Thus, a simple change of variables can ease the computation of the above integral,

$$v = L_\theta^T(u-\mu_{U|Y=y_{\text{obs}}}) + R_{ZX}(\beta-\hat{\beta}_\theta),$$

where the Jacobian of this affine transformation from u to v is $|L_\theta|$. Therefore we simply the integration problem as,

$$\frac{1}{(2\pi\sigma^2)^{(n+q)/2}} \exp\left[-\frac{r^2(\theta) + \|R_X(\beta-\hat{\beta}_\theta)\|^2}{2\sigma^2}\right] \int \exp\left[-\frac{\|v\|^2}{2\sigma^2}\right] |L_\theta|^{-1} dv,$$

which by properties of exponential integrands becomes,

$$\exp\mathcal{L}(\theta,\beta,\sigma^2|y_{\text{obs}}) = f_Y(y_{\text{obs}}) = \frac{|L_\theta|^{-1}}{(2\pi\sigma^2)^{n/2}} \exp\left[-\frac{r^2(\theta) + \|R_X(\beta-\hat{\beta}_\theta)\|^2}{2\sigma^2}\right].$$

Evaluating and profiling the deviance

We are able to explicitly profile β and σ out of the log-likelihood, to find a compact expression for the profiled deviance (negative twice the profiled log-likelihood) and the profiled REML criterion as a function of the relative covariance parameters, θ, only. Furthermore these criteria can be evaluated quickly and accurately. To estimate the parameters, θ, β, and σ^2, we minimize negative twice the log-likelihood, which can be written as,

$$-2\mathcal{L}(\theta,\beta,\sigma^2|y_{\text{obs}}) = \log|L_\theta|^2 + n\log(2\pi\sigma^2) + \frac{r^2(\theta)}{\sigma^2} + \frac{\|R_X(\beta-\hat{\beta}_\theta)\|^2}{\sigma^2}.$$

It is very easy to profile out β, it only enters the ML criterion through the final term, which is zero when $\beta = \hat{\beta}_\theta$ where $\hat{\beta}_\theta$ is found by solving the previous penalized least-squares problem. Therefore, we can write a partially profiled ML criterion as,

$$-2\mathcal{L}(\theta, \sigma^2|y_{\text{obs}}) = \log|L_\theta|^2 + n\log(2\pi\sigma^2) + \frac{r^2(\theta)}{\sigma^2}.$$

This criterion is only partially profiled because it still depends on σ^2. Differentiating this criterion with respect to σ^2 and setting the result equal to zero yields,

$$0 = \frac{n}{\hat{\sigma}^2_\theta} - \frac{r^2(\theta)}{\hat{\sigma}^4_\theta},$$

which leads to a maximum profiled likelihood estimate,

$$\hat{\sigma}^2_\theta = \frac{r^2(\theta)}{n}.$$

This estimate can be substituted into the partially profiled criterion to yield the fully profiled ML criterion,

$$-2\mathcal{L}(\theta|y_{\text{obs}}) = \log|L_\theta|^2 + n\left[1 + \log\left(\frac{2\pi r^2(\theta)}{n}\right)\right].$$

This expression for the profiled deviance depends only on θ. Although q, the number of columns in Z and the size of Σ_θ, can be very large indeed, the dimension of θ is small, frequently less than 10. The `lme4` package uses generic nonlinear optimizers to optimize this expression over θ to find its maximum likelihood estimate.

The REML criterion

The REML criterion can be obtained by integrating the marginal density for Y with respect to the fixed effects,

$$\int f_Y(y_{\text{obs}})d\beta = \frac{|L_\theta|^{-1}}{(2\pi\sigma^2)^{n/2}} \exp\left[-\frac{r^2(\theta)}{2\sigma^2}\right] \int \exp\left[-\frac{\|R_X(\beta - \hat{\beta}_\theta)\|^2}{2\sigma^2}\right] d\beta,$$

which can be evaluated with the change of variables $v = R_X(\beta - \hat{\beta}_\theta)$, where the Jacobian determinant of the transformation from β to v is $|R_X|$. Therefore, we are able to write the integral as,

$$\begin{aligned}\int f_Y(y_{\text{obs}})d\beta &= \frac{|L_\theta|^{-1}}{(2\pi\sigma^2)^{n/2}} \exp\left[-\frac{r^2(\theta)}{2\sigma^2}\right] \int \exp\left[-\frac{\|v\|^2}{2\sigma^2}\right] |R_X|^{-1} d\beta \\ &= \frac{|L_\theta|^{-1}|R_X|^{-1}}{(2\pi\sigma^2)^{(n-p)/2}} \exp\left[-\frac{r^2(\theta)}{2\sigma^2}\right].\end{aligned}$$

A similar profile likelihood argument as the previous Section leads to the profiled REML criterion,

$$-2\mathcal{L}(\theta|y_{\text{obs}}) = \log\left(|L_\theta|^2|R_X|^2\right) + (n-p)\left[1 + \log\left(\frac{2\pi r^2(\theta)}{n-p}\right)\right].$$

Acknowledgments

We borrow materials from Faraway [2016], Pinheiro and Bates [2006], Agresti [2013], and the vignette for the `lem4` package

References

Alan Agresti. *Categorical Data Analysis*. John Wiley & Sons, 2013.

Julian J Faraway. *Extending the linear model with R: generalized linear, mixed effects and nonparametric regression models*. CRC press, 2016.

José Pinheiro and Douglas Bates. *Mixed-effects models in S and S-PLUS*. Springer Science & Business Media, 2006.