STAT 528 - Advanced Regression Analysis II

Linear Mixed Models

Daniel J. Eck
Department of Statistics
University of Illinois

Learning Objectives Today

- Linear Mixed Models examples

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.

The pulp data frame has 20 rows and 2 columns. Data comes from an experiment to test the paper brightness depending on a shift operator. The variables are:

- **bright**: Brightness of the pulp as measured by a reflectance meter
- **operator**: Shift operator a-d

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

```r
## (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.

```r
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
## -1.4666 -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
```

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.

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.

LRT

In this example the only fixed effect is the mean and there is not much interest in testing that.

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

```r
## null model fit
nullmod <- lm(bright ~ 1, pulp)

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

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

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

```r
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.

parametric bootstrap

```r
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
##   5.667   0.046   5.714
```

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

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

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

The LRT clearly does not have a χ^2 distribution. See Section 8.2 in Faraway 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. As it happens, this *p*-value is close to the fixed effects *p*-value.

Predicting random effects

Approach this problem from a Bayesian perspective with a prior density for the αs such that $E(\alpha_i) = 0$ 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

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

We will take an empirical Bayesian point of view and substitute the MLEs into D, V, and β to obtain predicted random effects.

These are 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
```

Look what happens when we compute the ratio of fixed effects to the random effects

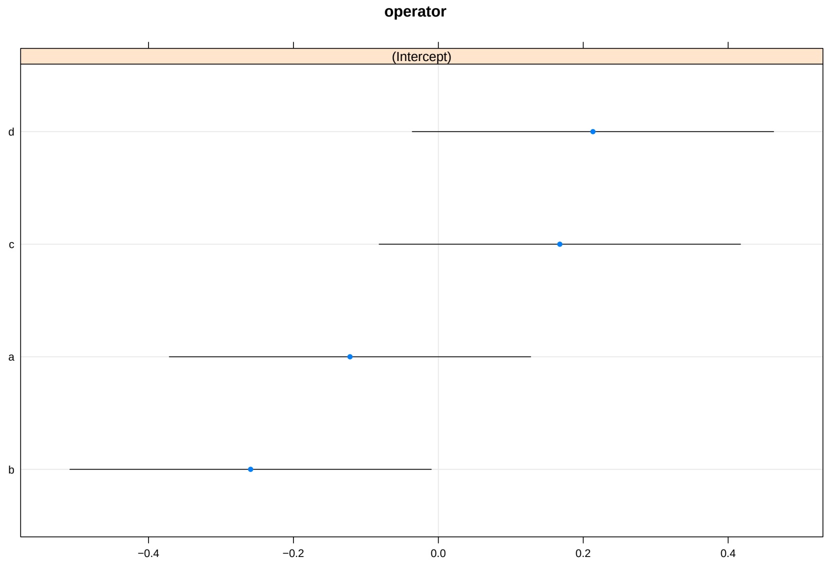
```
## estimated fixed effects divided by
## predicted random effects
cc[[1]]$operator / ranef(mmod)$operator
```

```
##   (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**.

95% confidence intervals for random effects:

```
## $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
```

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:

```r
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
##   6.683   0.051   6.735
```

```r
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
##   9.239   0.052   9.293
```

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

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

Soybean Analysis (see notes)

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.

This data consists of multiple measurements on the same day with only a few days of data collection.

```r
dat <- read.csv("soybean.csv")
dat$ID <- as.factor(dat$ID)
head(dat)
```

```
##      ID plot_number disk       AqE   Precip Precip_7day      days
## 1 NAM10         123   30 0.8237071 2.544962   0.5964009 -1.369516
## 2 NAM10          70   68 1.9022350 2.544962   0.5964009 -1.369516
## 3 NAM10         136   21 0.9653680 2.544962   0.5964009 -1.369516
## 4 NAM10         136   15 0.5140451 2.544962   0.5964009 -1.369516
## 5 NAM10          70   59 1.0306178 2.544962   0.5964009 -1.369516
## 6 NAM10          70   52 1.1022001 2.544962   0.5964009 -1.369516
```

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.

```
m <- lm(AqE ~ ID + days + I(days^2) + Precip + Precip_7day,
  data = dat)

m_re_plot <- lmer(AqE ~ ID + days + I(days^2) + Precip +
  Precip_7day + (1|plot_number), data = dat, REML = FALSE)

m_re_full <- lmer(AqE ~ ID + days + I(days^2) + Precip +
  Precip_7day + (1|plot_number) + (1|disk), data = dat, REML = FALSE)
```

Both AIC and BIC select the largest model with random effects for plots and disk effects.

```r
## AIC
c(AIC(m), AIC(m_re_plot), AIC(m_re_full))
```

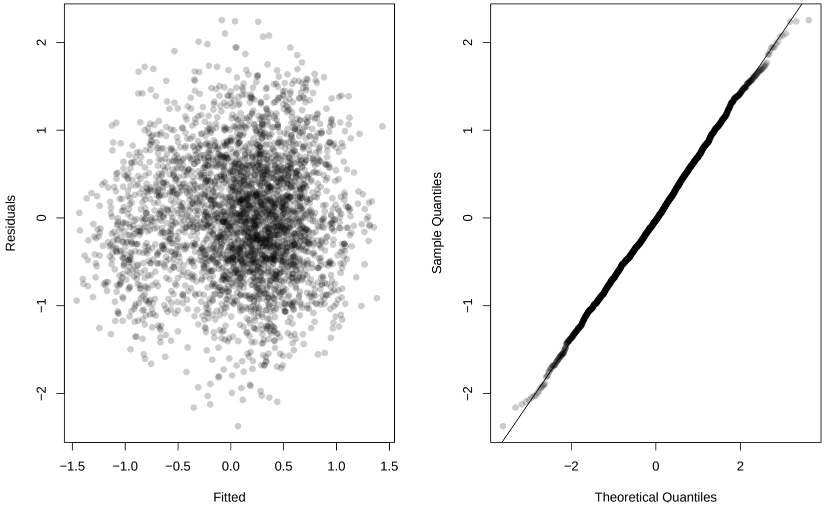
```
## [1] 7532.349 7485.456 7449.532
```

```r
## BIC
c(BIC(m), BIC(m_re_plot), BIC(m_re_full))
```

```
## [1] 7813.152 7772.363 7742.544
```

See the notes for parametric bootstrapping of LRTs.

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



We now investigate which genotypes are associated with AqE values that are different than the RC reference level.

To do this we will consider a simple approach in which we construct a model with a particular genotype level removed, and then compare this smaller model with the final model.

Model comparisons will be made using AIC (the difference of AIC values). We will perform this procedure for each genotype.

Results are displayed on later slides. Negative values indicate that a genotype is associated with AqE values that are different than the RC reference level.

```r
## AIC for each ID variable from full AqE fixed-effects model
M <- model.matrix(AqE ~ ID + days + I(days^2) + Precip +
                    Precip_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(AqE ~ -1 + M1 + (1|plot_number) + (1|disk),
                          data = dat, REML = FALSE)
    AIC(m_re_full) - AIC(foo)
})), ncol = 1)})
```

```
##    user  system elapsed
##   4.470   0.095   4.566
```

```
rownames(AIC_IDs) <- colnames(M)[grep("ID", colnames(M))]
colnames(AIC_IDs) <- c("AqE")
AIC_IDs[AIC_IDs < 0, ]
```

```
##        ID1        ID2        ID3        ID4        ID7        ID9       ID11
## -2.9151635 -0.3967087 -0.8910848 -3.0220489 -5.7221648 -1.8426071 -1.1914115
##       ID12       ID13       ID15       ID16       ID17       ID19       ID24
## -0.6273336 -6.3892507 -4.5287970 -0.3193861 -8.2713125 -1.1833585 -7.6020182
##       ID26       ID27       ID30       ID31       ID32       ID37       ID38
## -5.9966713 -1.0398143 -1.4059465 -0.3284319 -2.4089731 -1.0959652 -3.6448254
##       ID40
## -4.8324469
```