The Anatomy of a Mixed Model Analysis, with R's *lme4* Package

John Maindonald, Centre for Mathematics & Its Applications,
Australian National University

January 24, 2009

Abstract

This talk makes brief summary comments on abilities, in R's *lme4* package, for analysis of mixed models, i.e., models that have multiple superposed levels of variation.

There are normality and independence assumptions for each of the sets of random effects in the model. What should the statistical sleuth make of the anatomical details, once they are on show? Simulation seems a pretty much indispensible tool. One or other set of assumptions may be of greater or lesser consequence, depending on the relative magnitudes of the relevant effects and on the inferences that are intended.

The function `lme()` in the *nlme* package has extensive abilities for handling repeated measures models, while `lmer()` (in *lme4*) is able to fit generalized linear mixed models.

R Packages for Mixed Models

1. *nlme*: function `lme()`, for hierarchical models (+?). Development has pretty much ceased.
2. *lme4*: – both hierarchical & crossed models. Use `lmer()` for linear mixed models and (maybe) `glmer()` for generalized linear mixed models. *It is important when discussing the behavior of lmer and other functions in the lme4 package to state the version of the package that you are using. The package changes as I experiment with the computational methods. Douglas Bates, 5 Nov 2008.*
3. Note `anova()` for balanced designs. Beware however of output from the function `model.tables()`!
4. Other R packages for working with GLMMs include *glmmAK*, *glmmBUGS* (an interface to WinBugs) and *glmmML*.
5. *lmeSplines*: adds spline modelling capability to *nlme*.

Classes of Models

- Crossed versus nested models.
- Notions of balance:
  - Complete balanced designs.
  - Balanced incomplete block designs.
  - Generally balanced designs (SEs of treatment differences all equal; this is a superclass of generally balanced designs a/c Genstat)
  - Unbalanced designs.
- ANOVA, or Multi-level modeling (e.g. Shading data)
  - ANOVA: Stratum mean squares are a big part of the story.
  - Multi-level models: Know your components of variance!

Kiwifruit Shading Trial (`kiwishade` from *DAAG* package)

Blocks (3)/plots (4 per block)/vines (4 per plot)

4 shading treatments, randomized to whole plots within blocks.

Kiwifruit shading trial

Fit to vines:

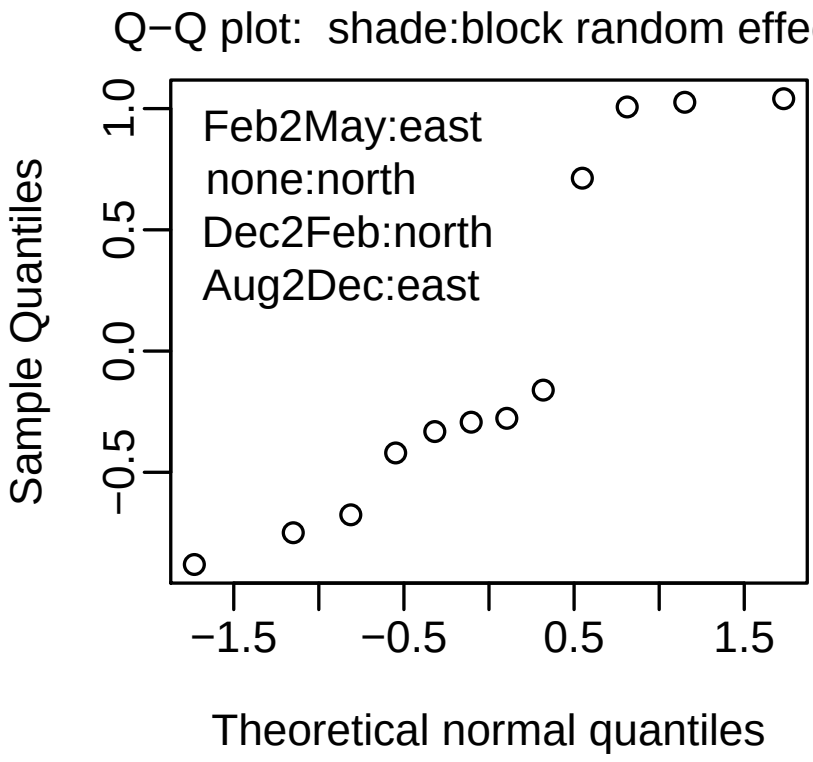
$$y_{vine} = \text{GM} + \text{eff}_{block} + \text{eff}_{trt} + \text{eff}_{plot} + \text{resid}_{vine}$$

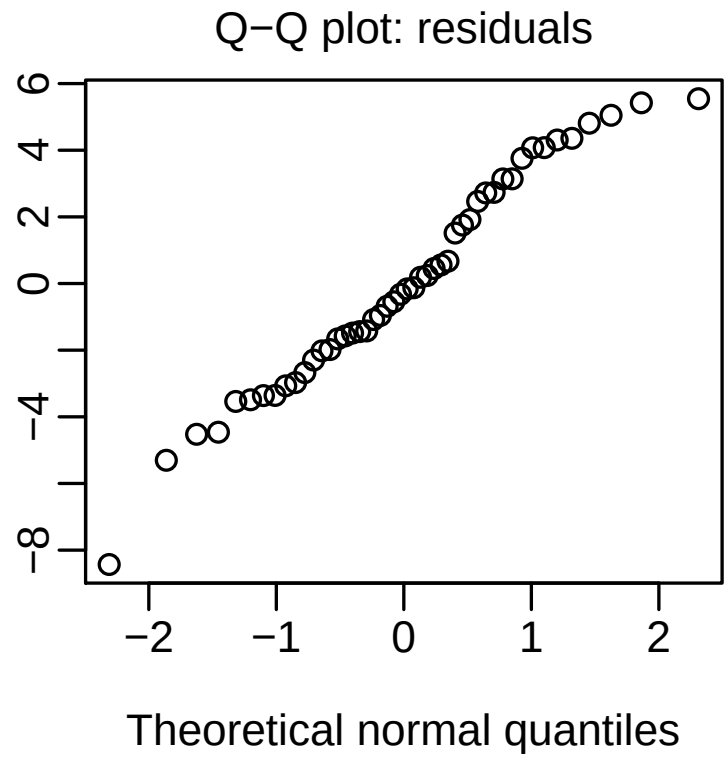
```
vfit <- lmer(yield ~ shade + (1 | block/shade), kiwishade
```

- We can equally well calculate plot means, and fit to plots.
- The analysis provides estimates of the effects that appear in the equation above, both the `shade:block` (i.e., plot) effects and (if of interest) the `block` effects.
- In this instance, estimated plot effects provide a check on normality, at the level where normality may be important for inferences about treatment effects. (In general, with more complex models, averaging and/or tradeoffs between estimates of different effects may have the result that the plots of effect estimates are not very informative.)

lmer(yield ~ shade + (1 | block/shade), kiwishade)

$\sigma^2_B = 4.08$	Var(mean)	$= 4.08 + \frac{2.19}{4} + \frac{12.2}{16}$
$\sigma^2_p = 2.19$	Var(mean)	$= 2.19 + \frac{12.2}{4}$
$\sigma^2 = 12.2$		





Conventional ANOVA Table

	Df	Sum of Sq	Mean sq	E[Mean sq]
`block` level	2	172.35	86.17	$16\sigma_B^2 + 4\sigma_P^2 + \sigma_V^2$
`plot:block` level				
`shade`	3	1394.51	464.84	$4\sigma_P^2 + \sigma_V^2 +$ trt ms
`residual`	6	125.57	20.93	$4\sigma_P^2 + \sigma_V^2$
Units (Residual)	36	438.58	12.18	σ_V^2

A muli-level model perspective

	Df	Var cpt	Var[mean]	
`block` level	2	4.08	$4.08 + \frac{2.19}{4} + \frac{12.18}{16}$	$= \frac{86.17}{16}$
`plot:block` level				
`shade`	3			
`residual`	6	2.19	$2.19 + \frac{12.18}{4}$	$= \frac{20.93}{4}$
Units (Residual)	36	12.18	12.18	

Random or Fixed Blocks; Vine or Plot; *lme()* or *lme4()*

For estimating treatment effects, the following are equivalent:

```
lmer(yield ~ shade + (1 | block/shade), data=kiwishade)
lmer(yield ~ block + shade + (1 | block:shade),
     data=kiwishade)  # fixed blocks; unchanged trt SEDs
```

Random or Fixed Blocks; Vine or Plot; *lme()* or *lme4()*

For estimating treatment effects, the following are equivalent:

```
lmer(yield ~ shade + (1 | block/shade), data=kiwishade)
lmer(yield ~ block + shade + (1 | block:shade),
     data=kiwishade)  # fixed blocks; unchanged trt SEDs
## -------------- Analysis of plot means --------------
ksPlot <- with(kiwishade,
     aggregate(yield, by=list(block=block, shade=shade),
               mean))
names(ksPlot)[3] <- "avyield"
```

Random or Fixed Blocks; Vine or Plot; *lme()* or *lme4()*

For estimating treatment effects, the following are equivalent:

```
lmer(yield ~ shade + (1 | block/shade), data=kiwishade)
lmer(yield ~ block + shade + (1 | block:shade),
     data=kiwishade)  # fixed blocks; unchanged trt SEDs
## -------------- Analysis of plot means --------------
ksPlot <- with(kiwishade,
     aggregate(yield, by=list(block=block, shade=shade),
               mean))
names(ksPlot)[3] <- "avyield"
## Analysis based on plot means
lmer(avyield ~ shade + (1 | block), data=ksPlot)
```

Random or Fixed Blocks; Vine or Plot; *lme()* or *lme4()*

For estimating treatment effects, the following are equivalent:

```
lmer(yield ~ shade + (1 | block/shade), data=kiwishade)
lmer(yield ~ block + shade + (1 | block:shade),
     data=kiwishade)  # fixed blocks; unchanged trt SEDs
## -------------- Analysis of plot means --------------
ksPlot <- with(kiwishade,
     aggregate(yield, by=list(block=block, shade=shade),
               mean))
names(ksPlot)[3] <- "avyield"
## Analysis based on plot means
lmer(avyield ~ shade + (1 | block), data=ksPlot)
- - - - - - - - - Note also the lme() (from nlme) syntax: -
library(nlme)
lme(yield ~ shade, random = ~1 | block/shade,
    data = kiwishade)
## Modify similarly for other possibilities
```

Different Paths to Enlightenment; Vine or Plot as Unit

```
vfit <- lmer(yield ~ shade + (1 | block/shade), kiwishade
pfit <- lmer(avyield ~ shade + (1 | block), data=ksPlot)
plot(resid(pfit) ~ ranef(vfit, drop=TRUE)[["shade:block"]
```

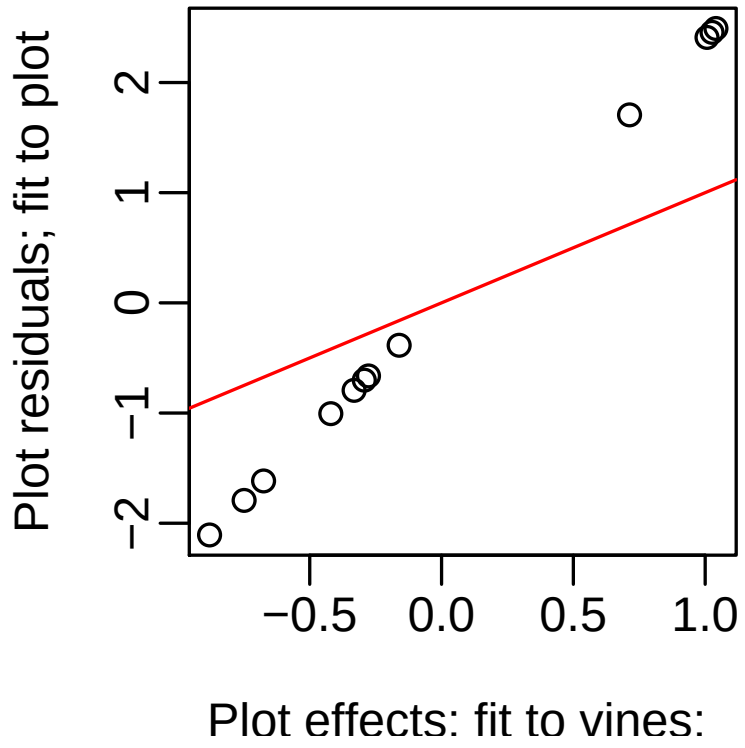
Fit to vines: SD[plot effects] = 1.48

$$y_{vine} = \text{GM} + \text{eff}_{block} + \text{eff}_{trt} + \text{eff}_{plot} + \text{resid}_{vine}$$

Fit to plots: SD[plot means] = 2.29

$$y_{plot} = \text{GM} + \text{eff}_{block} + \text{eff}_{trt} + \text{resid}_{plot}$$

The red line is the line $y = x$.



Degrees of Freedom and *p*-value Issues

- Unless designs are suitably balanced, *p*-values are more than ordinarily suspect.
  - In general, for tests in a multilevel model where the denominator SE or variance is a linear combination of the components of variance, there is no uniquely defined *p*-value!
  - For hierarchical designs that are in a suitable sense close to balance *p*-values, based on *t*- and *F*-statistics as for balanced designs, may not be too bad. (Users of *lme4* will however have to work out the details for themselves.)
  - Kenward-Roger approximations (not available in *lme4*), can in principle cater for both hierachical and crossed designs.
- *lme4* offers MCMC estimation of Bayesian credible intervals.
  - This seems better in principle than distributional approximations. Ideally, users would be encouraged to think about the priors for the random effects, and experiment with different choices of priors.
- Degrees of freedom information has other and more legitimate uses than for calculating *p*-values. They measure (roughly) the amount of information on which one or other SE may be based.

The Management of the Mixed Effects Menagerie

	Var est	d.f.	
Block	2.19	2	p=0.076
Plot	4.08	6 (3 for shade)	p=0.14
Residual	12.18	36	

- Random effects that are not at the level of treatment units can often be removed with relative impunity.
- "Remove if not significant", prior to testing for fixed effects, is (except perhaps as above) an abuse of testing Consider, e.g.:
  - With plot effect: 6 d.f. for trts: SED = 1.87; 6 d.f.
  - Without effect: 42 d.f. for trts: SED = 1.50; 42 d.f.

  NB also: With small d.f., non-normality is more serious.
- Historical experience can be a good guide.
- Simplify crossed design to hierarchical? (When/how?)

Generalization, not Inference

‘Generalization” stimulates the pertinent brain juices.
“Inference” may stimulate the wrong juices, or none at all!

- Treatment differences, many paths lead to the desired end. (NB: the three contiguous blocks assess the variability of the treatment effect under a limited range of soil and climate conditions. Is this caveat fatal to useful generalization?)

$$\text{SED} = \sqrt{\frac{2}{3}(2.19 + \frac{12.2}{4})}$$

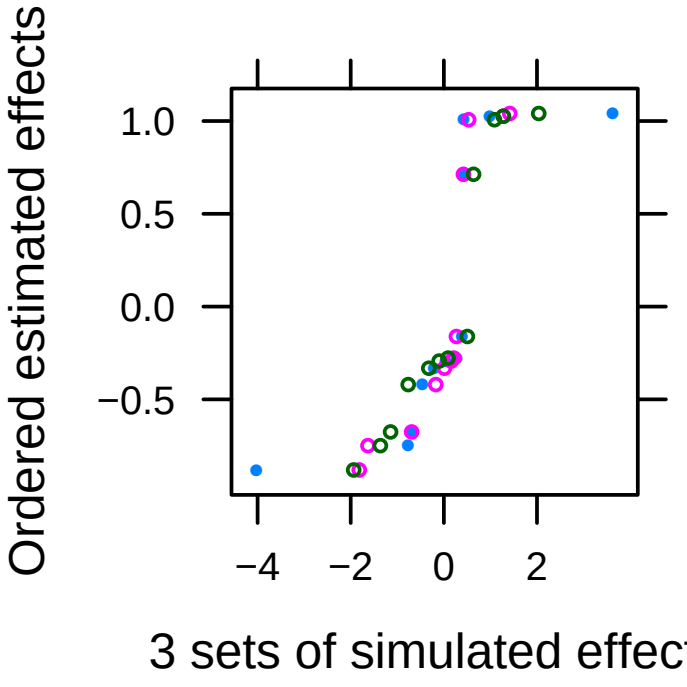
- For treatment means (average of 3 plots)

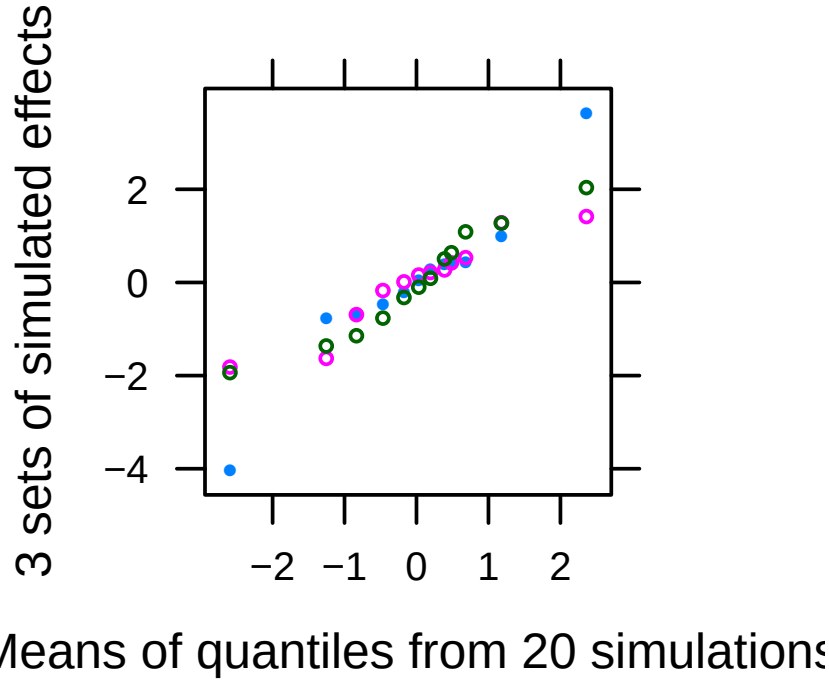
$$\text{SE} = \sqrt{\frac{1}{3}(2.19 + \frac{12.2}{4})} \qquad \text{(generalize to different plots)}$$

$$\text{SE} = \sqrt{4.08 + \frac{1}{3}(2.19 + \frac{12.2}{4})} \qquad \text{(generalize to different block)}$$

Checking Normality Assumptions

```
obj <- lmer(yield ~ 0 + shade + (1 | block/shade), kiwishade)
eff <- "shade:block"
obseff <- ranef(obj, drop=TRUE)[[eff]]
simvals <- simulate(obj, nsim = 3)
ranef(refit(kiwi.lmer, data.frame(simvals)[,1]), drop=TRUE)[[eff]
```





Checking Normality Assumptions – Notes

The variance of the plot random effects is estimated with low accuracy. The data are consistent with a zero variance. Thus, the slope in the Q-Q plot of simulated effects against means of quantiles from 20 simulations varies widely. For a substantial fraction of simulations, points lie on a line that is very nearly vertical. My function `diagmer()` can be used to do repeated simulations.

SEs for Fixed Effect Estimates

```
Random effects:
 Groups      Name        Variance Std.Dev.
 shade:block (Intercept)  2.19    1.48
 block       (Intercept)  4.08    2.02
 Residual                 12.18    3.49
Number of obs: 48, groups: shade:block, 12; block, 3

Fixed effects:
             Estimate Std. Error t value
(Intercept)    100.20       1.76    56.9
shadeAug2Dec     3.03       1.87     1.6
shadeDec2Feb   -10.28       1.87    -5.5
shadeFeb2May    -7.43       1.87    -4.0
```

NB: `(Intercept)` is the mean for `shadenone`. Other treatment effects are differences from `shadenone` as reference.

Marvellous `mcmcsamp()` (MCMC)

```
> kiwi.mcmc <- mcmcsamp(kiwi.lmer, n=1000)
> HPDinterval(kiwi.mcmc, prob=0.95)
. . . .
                lower  upper
(Intercept)    95.924 104.79    #[c.f. (95.9, 106.1)]
shadeAug2Dec   -0.432   6.28    #[c.f. (-1.5, 7.6)]
shadeDec2Feb -13.955  -6.89    #[c.f. (-14.9, -5.7)]
shadeFeb2May -10.983  -3.93    #[c.f. (-12.0, -2.9)]
. . . .
$ST                    #[multipliers for sigma]
     lower upper
[1,]     0  0.61
[2,]     0  1.86
. . . .
$sigma
     lower upper
[1,]  2.89  4.51
```

Alternative Formulation – one fixed factor only

```
> (kiwi.lmer <- lmer(yield ~ 0+shade + (1|block/shade),
                     data=kiwishade))
. . . .
Random effects:
 Groups      Name        Variance Std.Dev.
 shade:block (Intercept)  2.1863  1.4786
 block       (Intercept)  4.0779  2.0194
 Residual                12.1828  3.4904
Number of obs: 48, groups: shade:block, 12; block, 3

Fixed effects:
             Estimate Std. Error t value
shadenone     100.203      1.762   56.88
shadeAug2Dec  103.233      1.762   58.61
shadeDec2Feb   89.921      1.762   51.05
shadeFeb2May   92.774      1.762   52.67
```

Slots and extractor functions

```
> slotNames(kiwi.lmer)
 [1] "env"      "nlmodel"  "frame"    "call"  . . . .
> kiwi.lmer@call
lmer(formula = yield ~ 0 + shade + (1 | block/shade),
     data = kiwishade)
```

Mostly, use extractor functions:

`resid()`, `fitted()`, `VarCorr()`, `ranef()`, `mcmcsamp()`, `print()`, `summary()`

Heterogeneous Variances? – Machines & Operators

```
> form1 <- score ~ Machine + (1 | Worker)
> lmer(lmer(form1, data=Machines)
. . . .
Random effects:
 Groups   Name        Variance Std.Dev.
 Worker   (Intercept) 26.5     5.15
 Residual             10.0     3.16
Number of obs: 54, groups: Worker, 6

Fixed effects:
            Estimate Std. Error t value
(Intercept)    52.36       2.23   23.49
MachineB        7.97       1.05    7.56
MachineC       13.92       1.05   13.21
```

`Machines` is from the *MEMSS* package.

Machines – General random var-cov structure

```
> form2 <- score ~ Machine + (Machine | Worker)
> lmer(form2, data=Machines)
. . . .
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 Worker   (Intercept) 16.641   4.079
          MachineB    34.547   5.878     0.484
          MachineC    13.615   3.690    -0.365  0.297
 Residual              0.925   0.962
Number of obs: 54, groups: Worker, 6

Fixed effects:
            Estimate Std. Error t value
(Intercept)    52.36       1.68   31.15
MachineB        7.97       2.42    3.29
MachineC       13.92       1.54    9.04
```

```
> form3 <- score ~ Machine + (1 | Worker) +
+                        (0+as.numeric(Machine=="B") | Worker)
+                        (0+as.numeric(Machine=="C") | Worker)
> lmer(form3, data=Machines)
. . . .
Random effects:
 Groups   Name                           Variance Std.Dev.
 Worker   (Intercept)                    16.600   4.074
 Worker   as.numeric(Machine == "B") 34.684   5.889
 Worker   as.numeric(Machine == "C") 13.301   3.647
 Residual                                 0.926   0.962
Number of obs: 54, groups: Worker, 6

Fixed effects:
            Estimate Std. Error t value
(Intercept)    52.36       1.68   31.19
MachineB        7.97       2.43    3.28
MachineC       13.92       1.52    9.14
```

Comparison with over-dispersed GLM (quasibinomial)

Here, use the "sum" contrasts, and compare with the overall mean.

	glmer			quasibinomial		
	Est	SE	z	Est	SE (binomial SE)	t
(Intercept)	-2.32	0.22	-10.5	-2.33	0.21 (.14)	-11.3
Period1	-0.66	0.32	-2.1	-0.72	0.45 (.31)	-1.6
Period2	0.93	0.18	5.0	1.06	0.26 (.17)	4.2
Period3	-0.07	0.23	-0.3	-0.11	0.34 (.23)	-0.3
Period4	-0.20	0.25	-0.8	-0.24	0.36 (.24)	-0.7

The SEs (really SEDs) are not much increased from the binomial model. The estimates of treatment effects (differences from the overall mean) are however substantially reduced (pulled in towards the overall mean). The net effect is that the z-statistic is smaller for the `glmer` model than the t for the quasibinomial model.