Fitting Mixed Effects Models in R

Deepayan Sarkar

18 September 2008

R comes with an extensive collection of packages that extend its functionality. Packages usually provide a coherent collection of functions and datasets (and documentation) geared towards a particular task. A subset of packages, marked as "recommended", are part of a default installation of R. One such package, `nlme`, can be used to fit linear and nonlinear mixed effect models.

For the last few years, Doug Bates, the author of `nlme`, has been writing a completely new implementation of mixed models for R, based on sparse matrix code from Tim Davis. An important feature of this new package, called `lme4`, is that it handles fully crossed and partially crossed random effects gracefully. `lme4` is available at the Comprehensive R Archiving Network (CRAN) [1] and can be installed by calling

```
> install.packages("lme4")
```

in R. This will also install the `Matrix` package, which is a dependency of `lme4`.

Once the package is installed, it can be used in an R session after attaching it with

```
> library(lme4)
> library(lattice)
```

The `lattice` package is another recommended package (hence already installed) that provides Trellis graphics functionality in R. It is useful for the sort of data analyzed using mixed models.

[1] a local mirror is `http://cran.fhcrc.org`

An example: the Pastes data

`Pastes` is a dataset available in the `lme4` package.

Exercise 1 *Read the documentation for the dataset, which can be accessed by typing* `?Pastes` *or* `help(Pastes)`*. Attach the data (see the Usage section on the help page).*

Structure of the Pastes data

```
> str(Pastes)

'data.frame':        60 obs. of  4 variables:
 $ strength: num  62.8 62.6 60.1 62.3 62.7 63.1 60 61.4 57.5 56.9 ...
 $ batch   : Factor w/ 10 levels "A","B","C","D",..: 1 1 1 1 1 1 2 2 2 2 ...
 $ cask    : Factor w/ 3 levels "a","b","c": 1 1 2 2 3 3 1 1 2 2 ...
 $ sample  : Factor w/ 30 levels "A:a","A:b","A:c",..: 1 1 2 2 3 3 4 4 5 5 ...

> xtabs(~ batch + cask, Pastes, sparse = TRUE)

10 x 3 sparse Matrix of class "dgCMatrix"
  a b c
A 2 2 2
B 2 2 2
C 2 2 2
D 2 2 2
E 2 2 2
F 2 2 2
G 2 2 2
H 2 2 2
I 2 2 2
J 2 2 2
```

Exercise 2 *This tabulation suggests that the* `batch` *and* `cask` *variables are crossed. Is this an accurate reflection of the structure of the data? If not, obtain a tabulation that is.*

Although `cask` has 3 levels, there are actually 30 distinct samples. We can label the casks as 'a', 'b' and 'c' but then the `cask` factor by itself is meaningless (because cask 'a' in batch 'A' is unrelated to cask 'a'in batches 'B', 'C', ...). The `cask` factor is only meaningful within a `batch`. Only the `batch` and `cask` factors, which are apparently crossed, were present in the original data set. `cask` may be described as being nested within `batch` but that is not reflected in the data. It is *implicitly nested*, not explicitly nested.

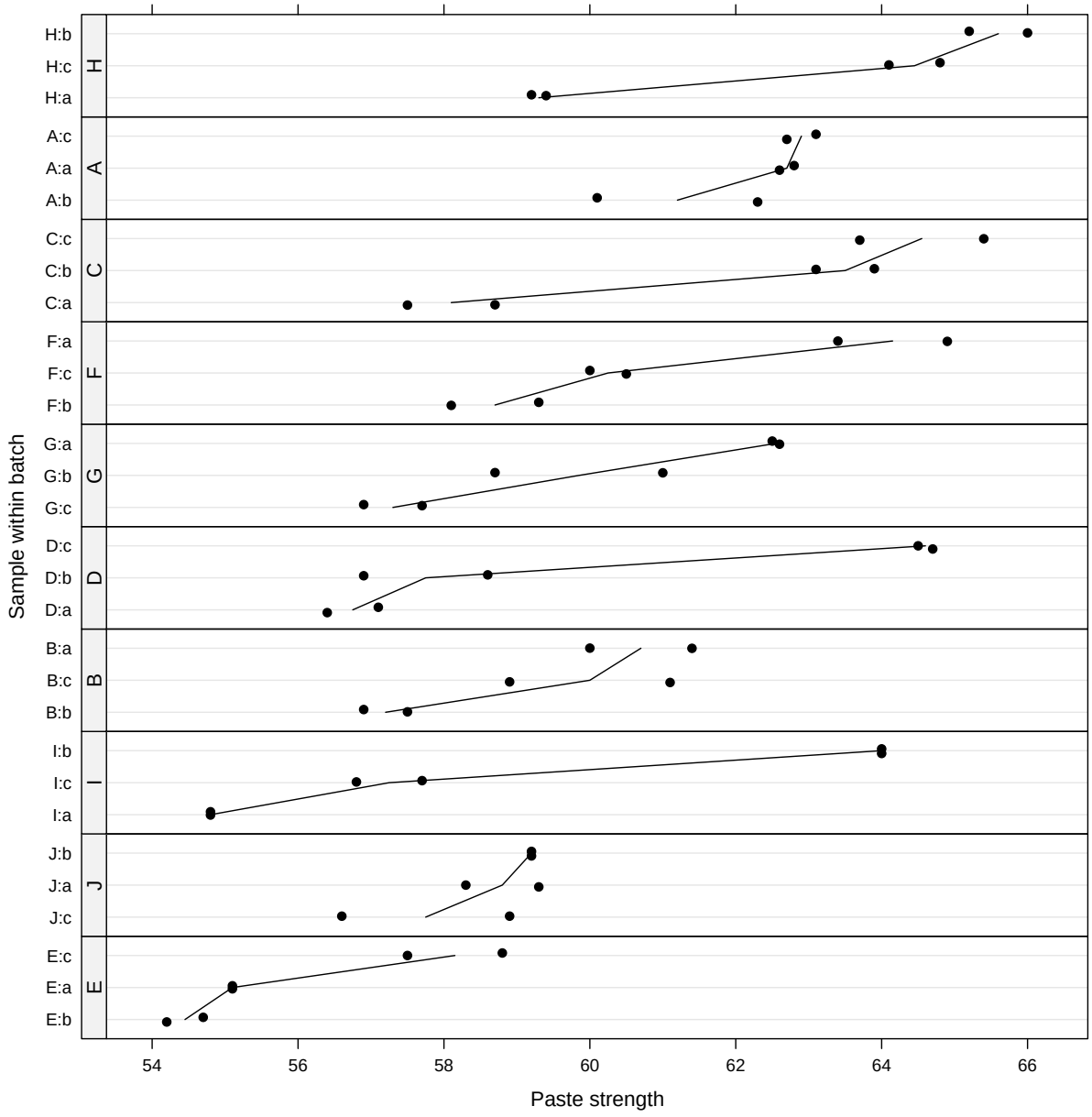
The `lme4` package allows for very general model specifications. It does not require that factors associated with random effects be hierarchical or "multilevel" factors in the design. The same model specification can be used for data with nested or crossed or partially crossed factors. Nesting or crossing is determined from the structure of the factors in the data, not the model specification. You can avoid confusion by immediately creating the explicitly nested factor. The recipe is

```
> Pastes <- transform(Pastes, sample = (batch:cask)[drop = TRUE])
```

This is already done in the `Pastes` dataset for convenience. We should specify models using the `sample` factor with 30 levels, not the `cask` factor with 3 levels.

A plot of the Pastes data is produced by

```
> Pastes$bb <- with(Pastes, reorder(batch, strength))
> Pastes$ss <- with(Pastes, reorder(reorder(sample, strength), as.numeric(batch)))
> dotplot(ss ~ strength | bb, Pastes,
          strip = FALSE, strip.left = TRUE, layout = c(1, 10),
          scales = list(y = list(relation = "free")),
          ylab = "Sample within batch", type = c("p", "a"),
          xlab = "Paste strength", jitter.y = TRUE)
```



A model with nested random effects is fit by

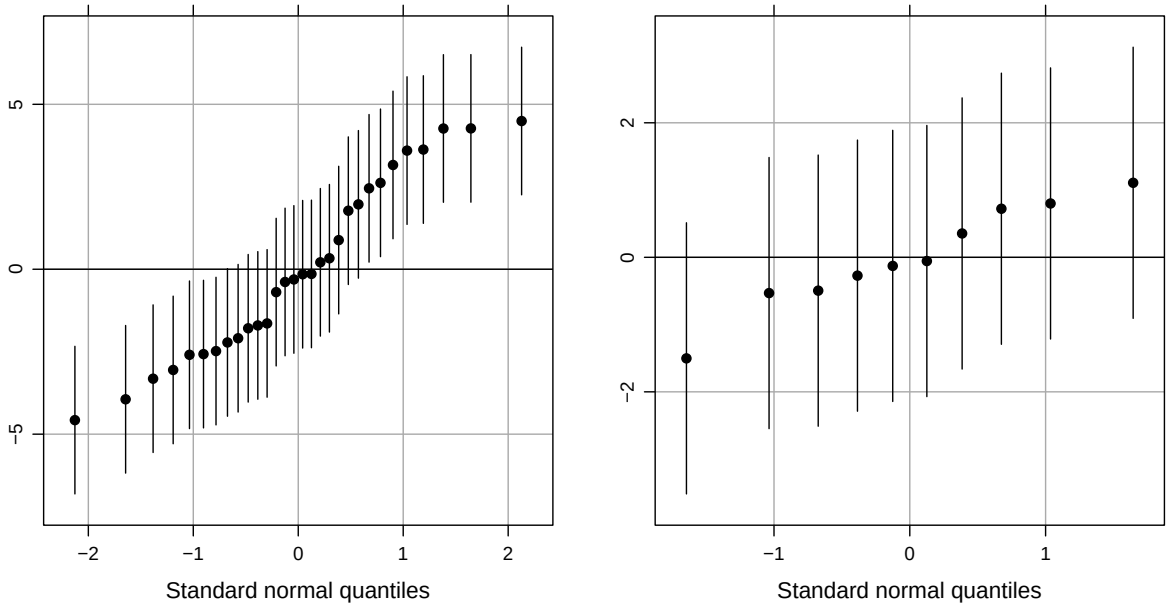
```
> fm3 <- lmer(strength ~ 1 + (1|batch) + (1|sample), Pastes)
> fm3
```

```
Linear mixed model fit by REML
Formula: strength ~ 1 + (1 | batch) + (1 | sample)
   Data: Pastes
 AIC   BIC logLik deviance REMLdev
 255 263.4 -123.5    248.0     247
Random effects:
 Groups   Name        Variance Std.Dev.
 sample   (Intercept) 8.43378  2.90410
 batch    (Intercept) 1.65691  1.28721
 Residual             0.67801  0.82341
Number of obs: 60, groups: sample, 30; batch, 10

Fixed effects:
            Estimate Std. Error t value
(Intercept)  60.0533     0.6768   88.73
```

The random effect "estimates" from model `fm3` can be visualized by

```
> qrr3 <- qqmath(ranef(fm3, postVar = TRUE), strip = FALSE)
> plot(qrr3[[1]], pos = c(0,0,0.5,1), more = TRUE)
> plot(qrr3[[2]], pos = c(0.5,0,1,1))
```



This plot and the data plot both show that the sample-to-sample variability dominates the batch-to-batch variability.

Exercise 3 *We have seen that there is little batch-to-batch variability beyond that induced by the variability of samples within batches. Can we eliminate the random-effects term for* `batch`*? Fit a reduced model without that term and compare it to the original model using the* `anova` *function (put the simpler model first in the call). Sometimes likelihood ratio tests can be evaluated using the REML criterion and sometimes they can't. Instead of learning the rules of when you can and when you can't, it is easiest always to refit the models with* `REML = FALSE` *before comparing. Make sure you fit the models using ML.*

Solution:

```
> fm3M <- update(fm3, REML = FALSE)
> fm4M <- lmer(strength ~ 1 + (1|sample), Pastes, REML = FALSE)
> anova(fm4M, fm3M)
```

p-values of LR tests on variance components

The likelihood ratio is a reasonable criterion for comparing these two models. However, the theory behind using a χ^2 distribution with 1 degree of freedom as a reference distribution for this test statistic does not apply in this case. The null hypothesis is on the boundary of the alternative hypothesis.

Exercise 4 *Why? Explicitly write down the model and the null hypothesis. Note that even at the best of times, the p-values for such tests are only approximate because they are based on the asymptotic behavior of the test statistic.*

In this case the problem with the boundary condition results in a p-value that is larger than it would be if, say, you compared this likelihood ratio to values obtained for data simulated from the null hypothesis model. We say these results are "conservative". As a rule of thumb, the p-value for a simple, scalar term is roughly twice as large as it should be.

Exercise 5 *In this case, does dividing the p-value in half affect our conclusion?*

Session information

- R version 2.7.0 Patched (2008-05-04 r45620), `i686-pc-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8;LC_NUMERIC=C;LC_TIME=en_US.UTF-8;LC_COLLATE=en_US.UTF-8;LC_MONETARY=C;LC_MESSAGES=en_US.UTF-8;LC_PAPER=en_US.UTF-8;LC_NAME=C;LC_ADDRESS=C;LC_TELEPHONE=C;LC_MEASUREMENT=en_US.UTF-8;LC_IDENTIFICATION=C`
- Base packages: base, datasets, graphics, grDevices, methods, stats, tools, utils
- Other packages: lattice 0.17-14, lme4 0.999375-26, Matrix 0.999375-14
- Loaded via a namespace (and not attached): grid 2.7.0