`% Ben Bolker`

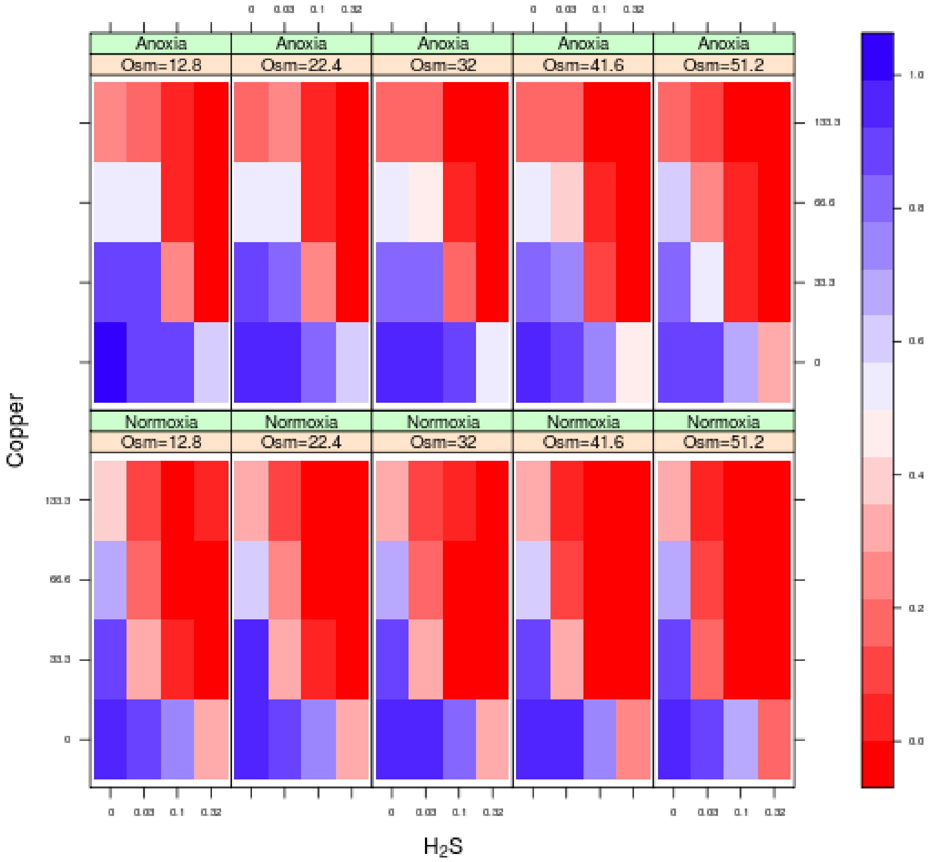
% Wed Jul 3 11:17:28 2013

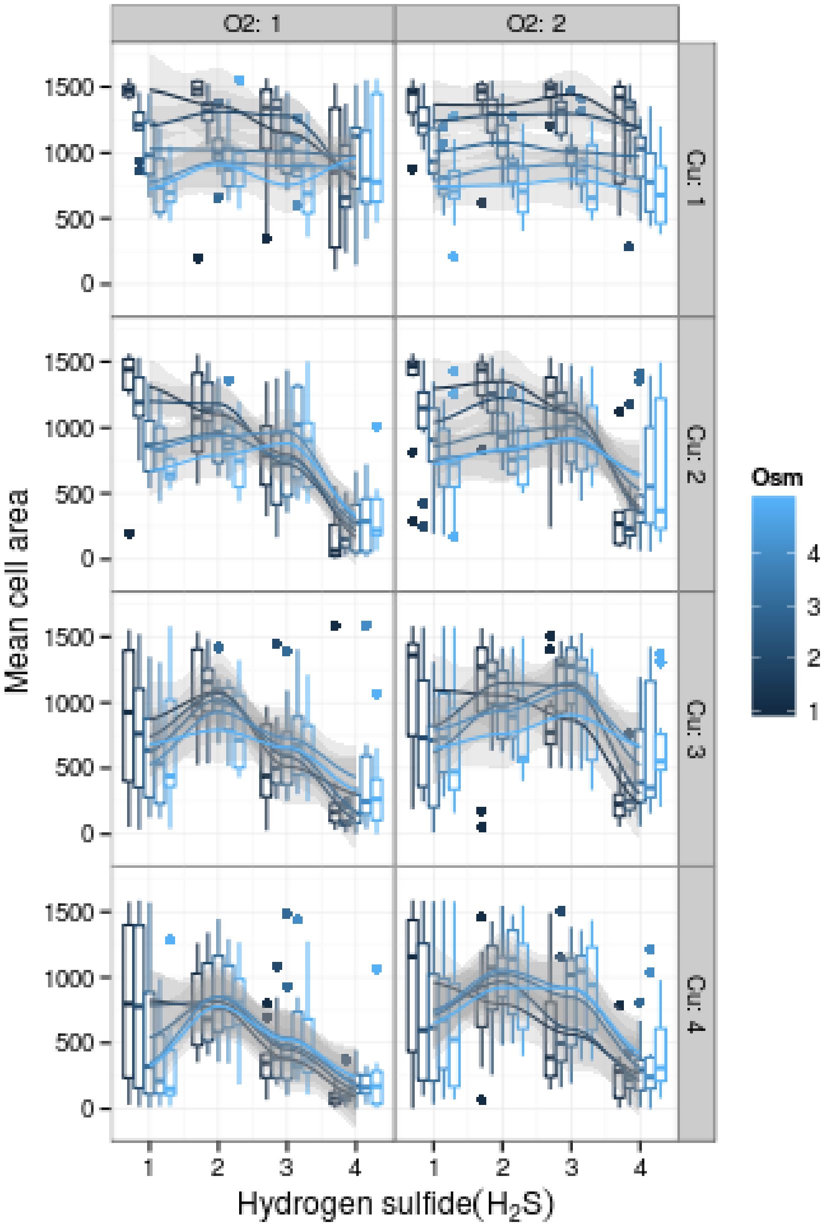
Mixed models

Examples

Glycera

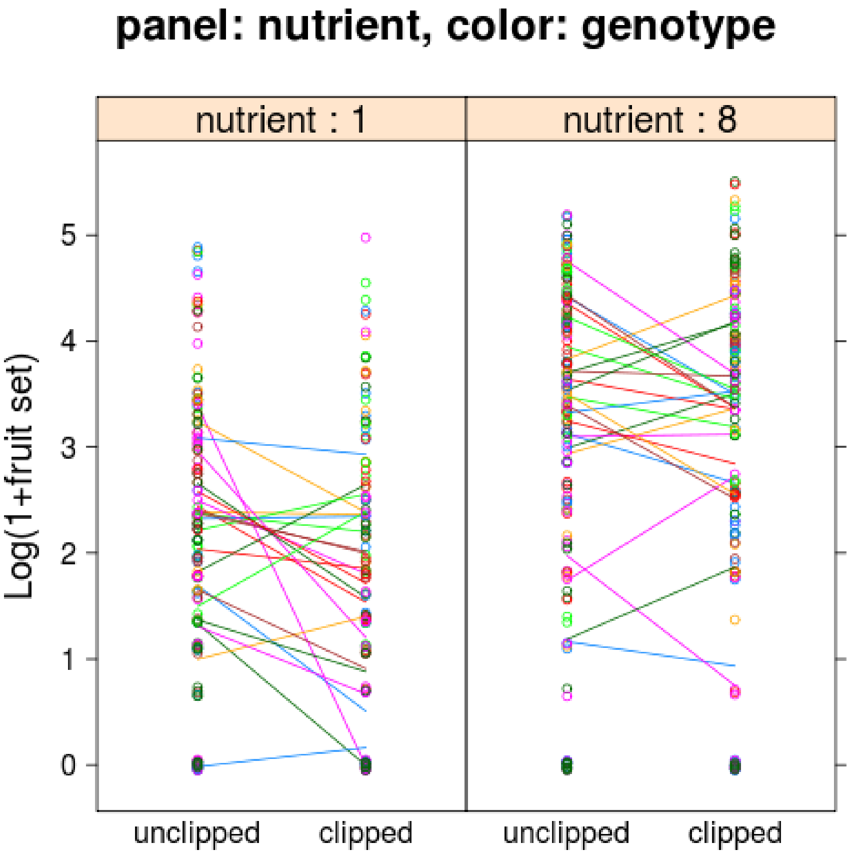
Various responses (survival, cell area) of *Glycera* (marine worm) cells in response to 4-way factorial combinations of stressors (anoxia, copper, salinity, hydrogen sulfide). Random effect of individual (only 10 individuals)!


Anoxia
Osm=12.8
Anoxia
Osm=22.4
Anoxia
Osm=32
Anoxia
Osm=41.6
Anoxia
Osm=51.2
Normoxia
Osm=12.8
Normoxia
Osm=22.4
Normoxia
Osm=32
Normoxia
Osm=41.6
Normoxia
Osm=51.2
0
0.03
0.1
0.32
0
33.3
66.6
133.3
Copper
H$_2$S
1.0
0.8
0.6
0.4
0.2
0.0



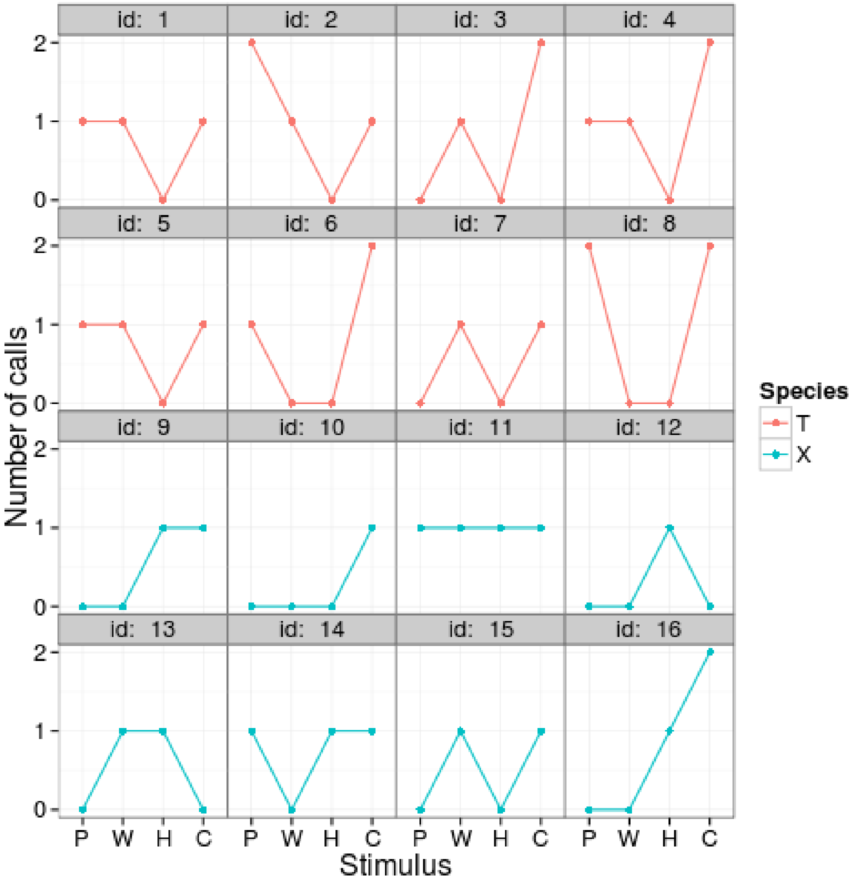
Arabidopsis

Response of *Arabidopsis* fruiting success to nutrients and simulated herbivory (clipping) across genotypes/populations/regions. Nested random effects of every level (in principle). (Banta, Stevens, and Pigliucci 2010; B. M. Bolker et al. 2009)



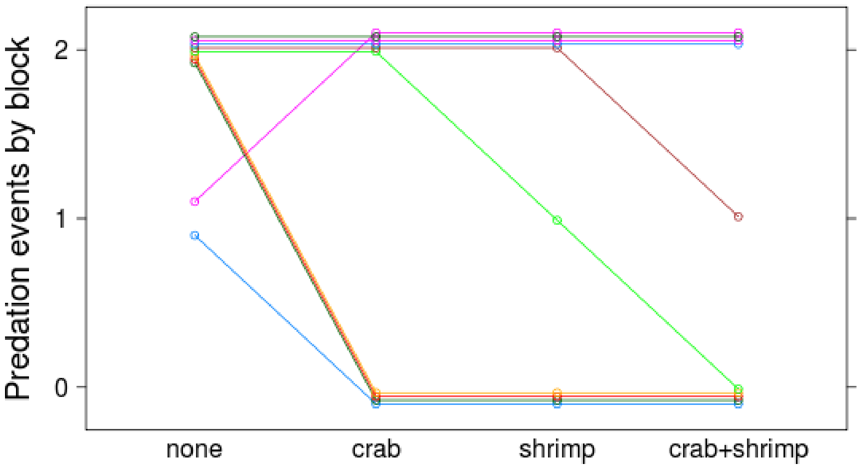
Singing mouse

Response of singing mice to conspecific and heterospecific playback experiments [@pasch_interspecific_IP]



others

- *Culcita*: protection of corals from sea-star predation by invertebrate symbionts (crabs and shrimp) (McKeon et al. 2012)



- coral/farmerfish: mortality and growth of coral in response to removal of farmerfish (before-after/control-impact design)
- owls: variation in owlet vocalization (“sibling negotiation”) as a function of satiation, sex of parent, arrival time, brood size

etc. etc. etc.

“classical” vs “modern” approaches to ANOVA

Classical: “method of moments” estimation

see e.g. (Underwood 1996; Quinn and Keough 2002)

- Decompose total sums of squares into components attributable to different effects, with associated degrees of freedom
- figure out which numerator and denominator correspond to testing a particular hypothesis
- or, figure out what class of design you have (nested vs. random effects vs. split-plot vs. ...) and look it up in a book (e.g. Ellison and Gotelli)
- straightforward: always gives an answer
- quick
- in more complicated designs (e.g. classical genetics estimate of additive + dominance variance), estimates may be negative
- hard to apply with experimental complications such as: lack of balance, crossed designs, autocorrelation

- users may become obsessed with “appropriate df” and significance testing rather than parameter estimation, confidence intervals, biological meaning

Modern: computational

- more flexible (see “complications” above)
- can be (much!) slower
- need to worry about computational, numerical problems such as failure to converge to the correct answer etc.
- allows computation of confidence intervals etc.
- mostly handles allocation of effects to levels automatically

What is a random effect?

[@gelman_analysis_2005]

Philosophical:

- we are not interested in the estimates of the random effects
- technically, this should be that we are not interested *in making inferences* about the estimates of the random effects
- in frequentist settings, we don’t even call these “estimates”, but rather “predictions”, as in BLUP (*best linear unbiased predictor*) (this term does not technically apply to GLMMs)
- the levels are drawn from a larger population
- the levels are “exchangeable” (i.e. we could relabel/swap around the levels without any change in meaning)
- the estimates are a random variable

Pragmatic/computational:

- we have lots of levels, with not much information about each individual level, and possibly unbalanced amounts of information
- we don’t want to use up the degrees of freedom associated with fitting one parameter for each level; automatically adjust between “completely pooled” (no effect) and “completely separate” (fixed effect)
- we want to estimate with *shrinkage*
- we have **enough** levels that it is practical to estimate a variance (i.e. at least 5-6, preferably more than that)

To Bayesians, the difference between fixed and random is much simpler and more pragmatic: do we add a hyperparameter to the model to characterize the estimates, or estimate them separately (= infinite variance)?

dimensions of data size

- information per sample: binary vs. binomial/count vs. Gaussian. Would repeated samples of a particular unit be approximately normal (e.g. mean > 5, or min(successes,failures)>5)? Determines whether we can get away with certain GLMM approximations
- samples per block: if small, then we need random effects; if sufficiently large, then fixed- and random-effects will converge
- number of blocks: do we have enough to estimate a variance? do we have so many that we can estimate it very accurately (i.e. we don't need to worry about denominator df?)

Examples

- genomics, proteomics, telemetry studies: few blocks, many samples per block (individual)
- cancer epidemiology: many blocks, varying samples per block, very little information per sample (i.e., cancer is rare)
- small-scale field experiments: moderate number of samples and blocks
- large-scale field experiments (or observations): often very few blocks

Definition of model and (marginal) likelihood

A useful (if somewhat technical) way to define a plain-vanilla linear model (normally distributed residuals, no random effects) is

$$Y_i \sim \text{Normal}(\mu_i, \sigma^2); \mu = X\beta$$

where X is the *design matrix*, β are the coefficients (all fixed effects). This is equivalent to, but more useful than, the other standard notation $Y_i \sim X\beta + \epsilon$, $\epsilon \sim \text{Normal}(0, \sigma^2)$ because some of the distributions we want to work can't simply be added to the expected response …

When we go to random-effects models, we add another level to the model:

$$Y_i \sim \text{Normal}(\mu_i, \sigma^2)$$
$$\mu = X\beta + Zu$$
$$u \sim \text{MVN}(0, \Sigma(\theta))$$

where Z is the random-effects design matrix; u is the vector of random effects; and θ is a vector of parameters that determines the variance-covariance matrix Σ of the random effects. (For example, for a simple intercept-only, model, θ might just be the among-block variance.)

GLMs look like this:

$$Y_i \sim \text{Distrib}(\mu_i, \phi); \mu = g^{-1}(X\beta)$$

where Distrib is some specified distribution (e.g. binomial or Poisson); ϕ is an optional scale parameter; and g^{-1} is the *inverse link function* (e.g. logistic or exponential, discussed further below).

GLMMs combine the inverse-link and arbitrary-distribution stuff with random effects: essentially,

$$\mu = g^{-1}(X\beta + Zu)$$

with Y and u defined as above.

Because random effects are random variables, the likelihood (or more precisely the *marginal likelihood*) of a set of parameters balances the probability of the data (x) given the fixed parameters β and a *particular* value of the random effect (u) with the probability of that value of the random effect given the variance of the random effects (σ), integrated over all possible values of σ:

$$L(x|\beta, \sigma) = \int L(x|\beta, u) \cdot L(u|\sigma) \, d\sigma$$

The Bayesian posterior distribution is basically the same, except that we include a prior distribution (and we rarely have to worry about the doing the integral explicitly)

Restricted maximum likelihood (REML)

While *maximum likelihood estimation* (finding the β, θ (and possibly u) that maximize the expression above) is a very powerful and coherent approach, it turns out to give biased estimates of the variances.

A heuristic explanation is that it estimates the variances *as though the maximum likelihood estimates of the fixed-effect parameters were correct*, which underestimates the variability in the data.

Two useful special cases of REML are (1) dividing sample sums of squares by $n - 1$ rather than n to get an unbiased estimate of the sample variance and (2) doing a paired t test by explicitly computing the means and variances of the differences in the pairs.

Technically, REML consists of finding a linear combination of the data that exactly reduces the fixed-effect estimates to zero, then estimating the random-effects part of the model on those data.

Important things to know about REML:

- it's most important for small data sets, and when the variance estimates themselves are of interest

- **you cannot sensibly compare/test fixed effect estimates between models estimated with REML**; to do this, you have to fall back to ML estimates (this is done automatically by some packages)
- REML has good properties (unbiased estimates of variances) in simple LMMs, but we're not completely sure whether they hold in more complex models. There is some controversy as to whether extending REML to GLMMs makes sense.

Avoiding MM

- average for *nested* designs (when variance within lower levels is not of interest) [@murtaugh_simplicity_2007]
- use fixed effects instead of random effects when (1) samples per block are large (there will be little shrinkage) or (2) number of samples is small (you will save few degrees of freedom, and variance estimates are likely to be bad anyway)

Estimation

Now that we know how to define a mixed model, how do we estimate the parameters (or how do we convince ourselves that we know what's going on with a "black-box" approach?

The fundamental problem is that while LMs and GLMs can be turned into linear algebra problems (for which there are extremely powerful general algorithms), the formal definition of (G)LMMs involves *numerical integration*, which is a very difficult numerical problem. Also, the Z (random effects design) matrix is often extremely large (but sparse): it has a column for every random-effect variable, of which there can be thousands …

Specifying mixed models in R

- Main packages
    - `nlme`: well-documented (Pinheiro and Bates 2000), stable, allows R-side variance and correlation structures; LMMs only, nested random effects (mostly)
    - `lme4`: fast, crossed random effects, allows GLMMs, developing
    - `glmmADMB`: flexible (e.g. negative binomial, zero-inflated, Beta models); slower
    - `MCMCglmm`: Markov chain Monte Carlo (Bayesian); extremely flexible, reasonably fast
    - BUGS interfaces (`R2jags`/`Rjags`/`BRugs`/`OpenBUGS`): even more flexible
    - AD Model Builder interface (`R2admb)`: ditto

- most stuff (fixed effects, choice of family/link for GLMMs) as in `lm()`/`glm()`
- random effects specification: `effect|grouping variable`, e.g. `1|block` for simple intercept variation among blocks

From glmm FAQ:

formula	meaning
`(1|group)`	random group intercept
`(x|group)`	random slope of x within group with correlated intercept
`(0+x|group)`	random slope of x within group: no variation in intercept
`(1|group) + (0+x|group)`	uncorrelated random intercept and random slope within group
`(1|site/block)`	intercept varying among sites and among blocks within sites (nested random ef
`site+(1|site:block)`	fixed effect of sites plus random variation in intercept among blocks within sites
`(x|site/block)`	slope and intercept varying among sites and among blocks within sites
`x*site+(x|site:block)`	fixed effect variation of slope and intercept varying among sites and random va
`(1|group1)+(1|group2)`	intercept varying among crossed random effects (e.g. site, year)

In `lme4` random effects are included in the

Deterministic (frequentist) approaches

Frequentist/likelihood-based frameworks usually use *deterministic* “hill-climbing” algorithms.

- Classical method-of-moments approaches reduce the problem to a (very easy) one of adding and subtracting sums of squares
- For LMMs, enough algebraic cleverness can reduce the problem to an (extended) linear algebra problem for known θ (variance-defining) parameters; then we can do a nonlinear search over this (relatively low-dimensional) space for the best fit. Sparse matrix algorithms handle the large size of Z.
- For GLMMs, we have an additional nasty step of approximating the integral (which can’t be reduced to a linear algebra problem) … More specifically, there are three general approaches to this approximation
- **Penalized quasi-likelihood**: relatively crude, but fast and flexible [@breslow_whither_2004]. Known to be biased for small-unit samples (e.g. binary data), but widely used: SAS PROC GLIMMIX (original), `MASS::glmmPQL`. `glmmPQL` can do any model that `lme` can (including

correlated models, etc.), whether or not they make sense … Better (second-order) versions exist, but not in R

- **Laplace approximation** Better, still reasonably fast and flexible `glmer`, `glmmML`, `glmmADMB`, `R2admb`/AD Model Builder.
- **Gauss-Hermite quadrature**: most accurate, slowest, least flexible (allows at most 2-3 random effects in practice). User chooses number of quadrature points (8-20?): Laplace corresponds to `nAGQ=1` (in `glmer`). `glmer`, `glmmML`, `repeated`.

Most of these packages are black boxes: we generally just have to specify

- fixed effects (according to Wilkinson-Rogers syntax)
- random effects (according to some sort of extended W-R syntax)
- which algorithm to use (if we have a choice)

AD Model Builder is an outlier from this list; in ADMB, you have to code the negative log-likelihood function explicitly, in a variant of C++. This requires a bit more work. The advantages are (1) ADMB is more flexible than any of the R packages mentioned above; you can basically use any linear or nonlinear model you want. (2) You get a better idea of what model you are actually fitting. (3) In general ADMB is much faster than R's optimization tools, although to make mixed models fit faster you really have to know what you're doing. The `R2admb` package (slightly!) simplifies the process of getting up to speed with ADMB.

Crossed vs. *nested* factors/*implicit* vs. *explicit* nesting

two random factors are *crossed* if levels of each factor are represented in multiple levels of the other, nested otherwise. e.g., "subplots within plots" are nested (there is more than one subplot in a plot, but never more than one plot associated with a subplot), but years and plots might well be crossed. If there are "good years" that are consistently good across all plots, and "good plots" that are consistently good across all years, then the factors are nested. Consider the decomposition of variance

$$\sigma^2_T = \sigma^2_Y + \sigma^2_P + \sigma^2_{YP}$$

if (e.g.) σ^2_P is negligible (there are good and bad years, but plots vary randomly from year to year), then we could model plots as nested within years (`~1|year/plot`, equivalent to `~1|(year+year:plot)`). Otherwise, we have to model plots and years as crossed (`(1|year*plot)`, although frequently we leave out the interaction term, or it gets subsumed into the error variance: `(1|year)+(1|plot)`).

This statistical issue is related to a data-coding issue. If we have plots A, B, C, and individuals 1, 2, 3 within each plot, it is a bit safer to code the nesting *explicitly*, labeling those individuals $A1$, $A2$, $A3$, $B1$ ... $C3$. This way the nesting is obvious to the computer. Otherwise (if we code the individuals 1, 2, 3 in each plot), we have to make sure to specify the nesting in the model formula, as `(1|plot/individual)`: if we specify `(1|plot)+(1|individual)`, we will fit a model with crossed random effects, which would only make sense if individuals $A1$, $B1$, $C1$ were somehow similar to each other (e.g. if the individuals were numbered left-to-right within each plot).

Pitfalls

- Quite common to hit convergence problems (difficulty inverting a matrix at some step of the process), especially with small/noisy/wonky data sets
- Variance parameters estimated as zero (ditto); not always clear whether this is bias or just the best estimate under the circumstances (since variances can't be negative). Lots of optimization methods have trouble under these circumstances.
- Complex algorithms, devil is in the details.
- **Remember that systematic errors and "unknown unknowns" are probably much more important than details of which algorithm you use!**

Stochastic (Bayesian) approaches

Bayesians typically use *stochastic*, Markov chain Monte Carlo algorithms, which while often slower than the deterministic methods can easily be extended to more complex models, and allow more general characterization of uncertainties

Not enough time to describe MCMC here, but ... we set up a "jumping rule" for moving around (semi-)randomly in the parameter space, and prove that in the long run the samples taken by this jumping rule will *converge* to the posterior distribution (more on this later). The key ideas of MCMC are

1. *conditional sampling*: sampling from the distribution of a parameter assuming (momentarily) that we know what the others are (e.g., if we knew the values of the random effects, then the rest of the problem would reduce to a simple (G)LM)
2. *rejection sampling*: we pick a value at random, and then we compare it to something (typically the likelihood × prior value of the place we just came from) to see if we should accept it or not. The most common form of this approach is called *Metropolis-Hastings* sampling.

The simpler packages for using stochastic Bayesian sampling are black boxes in the style of the deterministic packages listed above. `MCMCglmm` is probably the first to try. The only noticeable difference is that for models more complex than simple intercept models (in which case the random effect is specified as `~block`), the random-effects model specification is a bit more complex, because it is more flexible than `glmer` and friends. `INLA` is another “black-box” sampler, said to be very powerful for spatial problems, but I haven’t tried it out.

Most Bayesian analysis is done using programs more like ADMB, where you have to write the model definition yourself. This is a bit weird until you get the hang of it, but has the advantage of expressing the model structure very clearly and explicitly: see the examples in the lab. `JAGS`, with the R interface package `R2jags`, is the one to learn first.

`Stan` (with the `rstan` interface package) is a new and supposedly more powerful MCMC sampler, somewhat like `JAGS`.

Pitfalls

- you have to specify priors (although `MCMCglmm` tries to use generic “weak” priors by default – for small, noisy data sets and complex models, it may complain and tell you to specify more informative priors before it can proceed)
- `JAGS` uses different parameterizations of standard distributions from R: in particular, Normal distributions are defined in terms of *precision* ($1/\sigma^2$) rather than variance
- random effects with small numbers of levels can be quite sensitive to priors, especially the Gamma priors that are traditionally used [@gelman_prior_2006].
- it is your responsibility to check for convergence (i.e., that the sampling chains have run long enough to get a representative sample of the period). Trace plots (using `coda::xyplot`) and scatterplot matrices give visual diagnostics: the effective sample size, Raftery-Lewis diagnostic (`raftery.diag`: for single chains), and Gelman-Rubin diagnostic (`gelman.diag`: for multiple chains) give quantitative diagnostics.

inference

Frequentist approaches

There are basically two ways to compute confidence intervals or test inferences (which are variants of the same procedure: if the null value of a parameter is not in the 95% CI, then you can reject the null hypothesis), one based on looking at the local curvature of the log-likelihood surface and the other based on comparing

(Score tests)

Curvature: Wald tests

- These are the typical results shown in `summary()`
- They assume a quadratic log-likelihood surface: generally true for *large enough*, *nice enough* problems.
- They can be really bad in some cases ("Hauck-Donner effects")
- Z, t tests for single parameters: computing estimate/(std. err.) gives a measure of how far the estimate is from the null value (typically 0); $\hat{\beta} \pm 1.96\sigma$, or $\hat{\beta} \pm q_{t,0.975}\sigma$, gives confidence intervals
- χ^2, F tests for multiple parameters: sum of squares of Z or t statistics (often take correlation among parameters into account as well)

Model comparison: likelihood ratio tests and profile likelihood

- Best if you can get them, but slow and maybe not implemented
- Instead of assuming quadratic fits, actually compute the *profile* and follow it out the appropriate distance
- Null-hypothesis testing is (relatively) easy: just fit the nested models (with and without the focal parameter fixed to zero) and find the p (tail) value associated with χ^2_1 for the deviance difference ($-2 \times \Delta \log L$) (remember not to use REML estimates!)

Reminder: marginal vs. sequential tests

R's `anova` function by default does *sequential* tests.

The results of `summary`, and the results of `anova(*,type="marginal")` (works for `lme`) or `drop1(*,.~.)`, are *marginal* tests. Use these with caution! At the very least you should make sure you are using sum-to-zero contrasts when doing marginal tests. (See also `Anova(*,type="III")` from the `car` package.)

Finite-size corrections

But ... the Z, χ^2 variants above are large-sample approximations. For F, t we need to know the "denominator degrees of freedom".

What to do?

- Use a package (such as `lme`) that guesses df based on classical rules
- Guess yourself based on classical rules
- Various adjustments (Satterthwaite, Kenward-Roger): `pbkrtest` package
- if you can guess that the denominator df are $> 40-50$, then don't worry

- bootstrap or parametric bootstrap or MCMC (below)
- Similar "parameter-counting" issues apply if testing random effects (how many 'numerator' df?): can use `RLRsim` for LMMs to do a simulation-based null hypothesis test
- These issues do *not* apply to GLMMs if the scale parameter is fixed (Poisson, binomial), but do if it is estimated (Gamma, overdispersion models)

… **but** GLMMs have their own separate set of finite-size issues (very often ignored) [@cordeiro_improved_1994;@cordeiro_note_1998]. Some more possibilities (*Bartlett corrections*) in `pbkrtest`

… and not just the inferences (confidences intervals etc.), but the *point estimates* of GL(M)Ms, are known to be biased in the small-sample case: `blme`, `brglm` (GLMs only)

Boundary issues

- Most of the theory assumes that the null hypothesis value does not lie on the boundary of the feasible space (e.g. $\sigma^2 = 0$ is the NH, σ^2 can't be < 0) (Goldman and Whelan 2000; Molenberghs and Verbeke 2007)
- Only applies to testing random effects
- Usually conservative (in the simplest case, p value is twice what it should be)
- `RLRsim`

Information-theoretic approaches

Most of these issues apply just as strongly to information-theoretic approaches, although this is not widely appreciated (Greven 2008; Greven and Kneib 2010).

- *How to count numbers of parameters for penalty terms for ICs*? Depends on the *level of focus* – i.e., whether you want to assess prediction accuracy for the individual units (estimated with shrinkage), or for the whole population. If the latter, something like *conditional AIC* [@vaida_conditional_2005], which counts an intermediate number of parameters (between 1 and the number of blocks) depending on how much information the random effects contribute (unfortunately computing CAIC depends on "the trace of the hat matrix", which is not available from current `lme4`)
- *How to compute effective sample size for finite-size corrections*? Finite-size corrections such as AICc are popular, but we know very little about how to count parameters (see above)

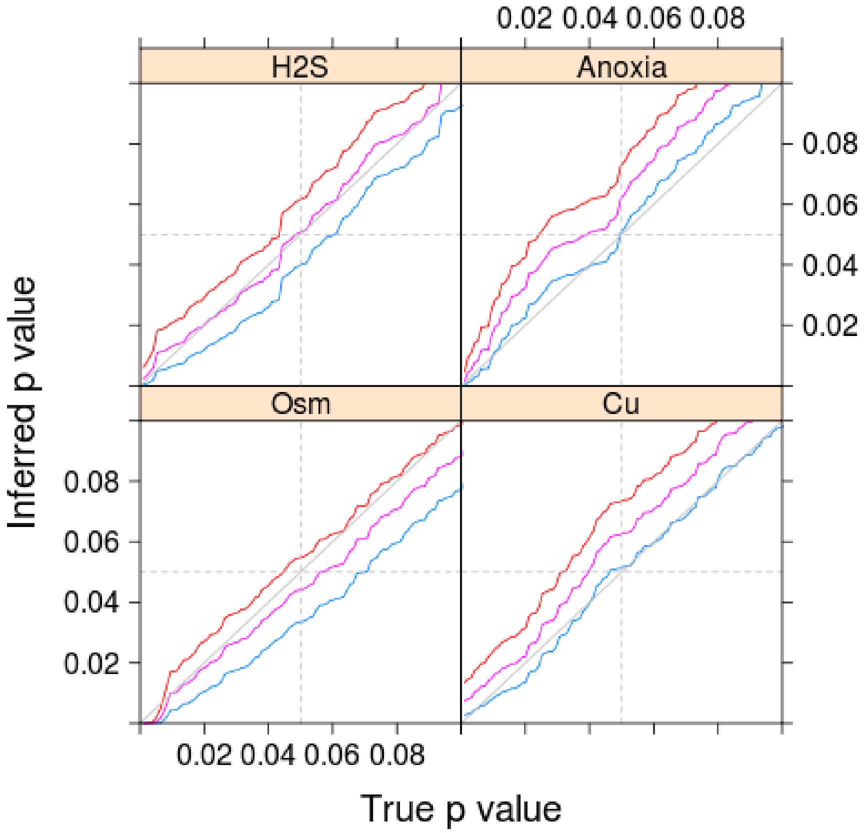
Parametric bootstrapping

A reasonable solution, although very slow and not well characterized … depends on assumption of model correctness (!!); nonparametric bootstrap can be challenging with grouped/correlated data

- fit null model to data
- simulate "data"" from null model
- fit null and working model, compute likelihood difference
- repeat to estimate null distribution

e.g. something like:

```
pboot <- function(m0, m1) {
    s <- simulate(m0)
    L0 <- logLik(refit(m0, s))
    L1 <- logLik(refit(m1, s))
    2 * (L1 - L0)
}
pbdist <- replicate(1000, pboot(fm2, fm1))  ## or plyr::raply for progress bars
obsval <- logLik(m1)
mean(obsval >= c(obsval, pbdist))
```



(Lines represent Normal (blue), t_{14} (magenta), t_7 (red): gray (one-to-one line) is the desired result … different predictors apparently have *different* effective data sizes A similar strategy should work for confidence intervals (simulate the model, save the values of the parameters each time)

post hoc MCMC sampling

Several packages (`glmmADMB`, formerly `lme4`) offer the option to run MCMC once the basic model has been fitted, using information about the location and curvature of the peak to make the MCMC run more efficient

- priors unspecified (flat/uninformative/improper), could be dangerous …
- slow (but usually faster than Bayesian MCMC, since we start from the MLE)
- doesn't give p-values (but see "Bayesian p value" below)

Bayesian approaches

Many of these problems go away in a Bayesian computational framework. *If* we really have a good sample from the (multidimensional) posterior distribution, then we can compute interesting statistics about the *marginal* distributions of the parameters (which is usually what we want to know) simply by looking at the distribution with respect to that parameter.

- *highest posterior density* ("Bayesian credible") intervals: an interval that includes 95% of the posterior distribution, symmetric with respect to probability (draw a cut-off line based on posterior probability): `HPDinterval` in `coda`, `HPDregionplot` in `emdbook` (for 2-D credible regions). Coherent in Bayesian framework
- quantile intervals*: e.g., between 2.5% and 97.5% quantiles of the marginal posterior distribution. Easy to compute, and invariant to parameter transformation, but not sensible according to true Bayesians.
- *Bayesian p-values* (!!!!). `MCMCglmm` gives a p-value. The author of the package, Jarrod Hadfield, says:

  > pMCMC is the two times the smaller of the two quantities: MCMC estimates of i) the probability that a<0 or ii) the probability that $a > 0$, where a is the parameter value. It's not a p-value as such, and better ways of obtaining Bayesian p-values exist.

If you use Bayesian methods, you should probably eschew p values.

Bayesian methods also have a model-selection approach, DIC (Spiegelhalter et al. 2002), which does the same sort of counting as CAIC, and which also depends on the level of focus (and on other assumptions, such as the approximate normality of the posterior distribution)

extras & extensions

Overdispersion

Possibly the most important "extra" topic; **if ignored can lead to severe type I errors**. Assess, crudely, by examining residual deviance/sum of squares of Pearson residuals: should be χ^2_{n-p} distributed … See e.g. Banta example on glmm.wikidot.com examples page

- quasi-likelihood: `glmmPQL`
- conjugate compounding: negative binomial, beta-binomial, etc. (`glmmADMB`)
- observation-level error: induces (e.g.) lognormal-Poisson, logit-normal-binomial (see @elston_analysis_2001, GLMM faq)

correlation (“R-side” effects)

- models for temporal/spatial autocorrelation and heteroscedasticity: sometimes called “R-side” (R for “residual”, G for “grouping”)
- recall the model definition:

  $$Y_i \sim \text{Normal}(\mu_i, \sigma^2)$$

  $$\mu = X\beta + Zu$$

  (etc.). Now we intend to use

  $$Y \sim \text{MVN}(\mu_i, \Sigma_R)$$

  $$\mu = X\beta + Zu$$

  i.e. the residuals can have structure other than independence and constant variance.
- `lme` does this in a reasonably straightforward way by providing a `correlation` argument which can be specified from a variety of temporal and spatial correlation functions: (time-series) `corAR1`, `corCAR1`, `corARMA`, (spatial) `corExp`, `corGaus`, `corSpher`, etc..
- The `ape` package provides correlation structures for phylogenetic trees.
- Using these approaches in GLMMs is a bit tricky*: @dormann_methods_2007 discuss incorporating this in `glmmPQL`, but proceed with caution …
- `MCMCglmm` does provide correlation structures based on pedigrees and phylogenies. There are several R packages providing extensions of `lme4` for pedigree/kinship data.

Non-standard distributions

While almost all GLMMs are Poisson and binomial (and of those most are binary/Bernoulli), it is sometimes nice to be able to use e.g. Gamma distributions (although log-normal often works equally well). In principle this works, but distributions other than Poisson and binomial are much less widely implemented, and much more fragile. `MCMCglmm` and `glmmADMB` are probably where you should look first. Gaussian, Poisson, binomial, Gamma make up (most of the) exponential family. Beyond this you may want to use e.g. a Beta distribution (for proportion data); `glmmADMB`, or rolling your own in `JAGS` or `ADMB`, are your options (although see the `tweedie` package)

Zero-inflation and zero-alteration

May have "too many zeros" (although Poisson and neg binomial with low means, high overdispersion can indeed have lots of zeros). `JAGS`, `MCMCglmm`, `glmmADMB` can handle models of this type

Additional topics

- Multivariate (multi-type) responses (`lme4` by hand, `MCMCglmm`)
- Ordinal responses (`cplmm`)
- Variable importance
- Goodness-of-fit metrics ...
- Complex variance structures (ASREML)
- Penalized models (`glmmLasso`; (Jiang 2008))
- Additive models (`mgcv::gamm`, `gamm4`)

References

Banta, Joshua A., Martin H. H. Stevens, and Massimo Pigliucci. 2010. "A comprehensive test of the 'limiting resources' framework applied to plant tolerance to apical meristem damage." *Oikos* 119 (2) (feb): 359–369. doi:10.1111/j.1600-0706.2009.17726.x. http://onlinelibrary.wiley.com/doi/10.1111/j.1600-0706.2009.17726.x/abstract.

Bolker, Benjamin M., Mollie E. Brooks, Connie J. Clark, Shane W. Geange, John R. Poulsen, M. Henry H. Stevens, and Jada-Simone S. White. 2009. "Generalized linear mixed models: a practical guide for ecology and evolution." *Trends in Ecology & Evolution* 24: 127–135. doi:10.1016/j.tree.2008.10.008. http://www.sciencedirect.com/science/article/B6VJ1-4VGKHJP-1/2/35970065c78c14ad30bf71bd1d5b452e.

Goldman, Nick, and Simon Whelan. 2000. "Statistical Tests of Gamma-Distributed Rate Heterogeneity in Models of Sequence Evolution in Phylogenetics." *Molecular Biology and Evolution* 17 (6): 975–978.

Greven, Sonja. 2008. *Non-Standard Problems in Inference for Additive and Linear Mixed Models*. Göttingen, Germany: Cuvillier Verlag. http://www.cuvillier.de/flycms/en/html/30/-UickI3zKPS,3cEY=/Buchdetails.html?SID=wVZnpL8f0fbc.

Greven, Sonja, and Thomas Kneib. 2010. "On the Behaviour of Marginal and Conditional Akaike Information Criteria in Linear Mixed Models." *Biometrika* 97 (4): 773–789. http://www.bepress.com/jhubiostat/paper202/.

Jiang, Jiming. 2008. "Fence methods for mixed model selection." *The Annals of Statistics* 36 (4) (aug): 1669–1692. doi:10.1214/07-AOS517. http://projecteuclid.org/euclid.aos/1216237296.

McKeon, C. Seabird, Adrian Stier, Shelby McIlroy, and Benjamin Bolker. 2012. "Multiple defender effects: synergistic coral defense by mutualist crustaceans." *Oecologia* 169 (4): 1095–1103. doi:10.1007/s00442-012-2275-2. http://www.springerlink.com/content/nm20758r6557v448/abstract/.

Molenberghs, Geert, and Geert Verbeke. 2007. "Likelihood Ratio, Score, and Wald Tests in a Constrained Parameter Space." *The American Statistician* 61 (1): 22–27. doi:10.1198/000313007X171322.

Pinheiro, José C., and Douglas M. Bates. 2000. *Mixed-effects models in S and S-PLUS*. New York: Springer.

Quinn, Gerry P., and Michael J. Keough. 2002. *Experimental Design and Data Analysis for Biologists*. Cambridge, England: Cambridge University Press.

Spiegelhalter, D. J., N. Best, B. P. Carlin, and A. Van der Linde. 2002. "Bayesian measures of model complexity and fit." *Journal of the Royal Statistical Society B* 64: 583–640.

Underwood, A. J. 1996. *Experiments in Ecology: Their Logical Design and Interpretation Using Analysis of Variance*. Cambridge, England: Cambridge University Press.