<http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture23.htm>

**Lecture 23—Monday, November 15, 2010**

**Topics**

* [Count regression models (continued)](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture23.htm#count)
* [Accounting for observational heterogeneity](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture23.htm#accounting)
  + [Including a blocking variable to account for data structure](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture23.htm#including)
  + [The random intercepts model](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture23.htm#ranints)
* [Fitting a random intercepts model in R](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture23.htm#fitting)
* [Random intercepts models with patch-level predictors](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture23.htm#random)
* [Comparing fixed effects intercept estimates with random intercept predictions](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture23.htm#comparing)
* [R code](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture23.htm#Rcode)

**Count regression models (continued)**

Last time we considered alternative approaches to modeling density and settled on including log(area) as a covariate in an ordinary count regression model. We have not yet formally investigated the choice of probability model for these data. A graph of the gross distribution of the data revealed that the count distribution is sufficiently removed from zero so that a normal model might be a reasonable choice. I load the data and fit four different models: a Poisson distribution with a log link, a Poisson distribution with an identity link, a normal distribution, and a log-normal distribution (normal distribution with a log-transformed response).

birds <- read.table( 'http://www.unc.edu/courses/2010fall/ecol/563/001/data/lectures/birds.csv', header=T, sep=',')

birds.short <- birds[!is.na(birds$S),]

**#fit four distributions: Poisson with log and identity links, normal, and lognormal**

out1 <- glm(S~landscape+factor(year)+log(area), data=birds.short, family=poisson)

out2 <- glm(S~landscape+factor(year)+log(area), data=birds.short, family=poisson(link=identity))

out3 <- glm(S~landscape+factor(year)+log(area), data=birds.short, family=gaussian)

out4 <- glm(log(S)~landscape+factor(year)+log(area), data=birds.short, family=gaussian)

We can use AIC to compare the first three models, but the last model uses a transformed response and its log-likelihood is not commensurate with the others without first making some adjustments. We used a function in [lecture 15](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture15.htm#Rformula) for this purpose, so I reload it and use it on the log-transformed model.

**#function to extract logL of y for log-transformed model**

norm.log <- function(model,y) {

t.y <- log(y)

s <- sqrt(sum(residuals(model)^2)/length(residuals(model)))

LL <- sum(log(dnorm(t.y,mean=predict(model), sd=s)\*1/y))

LL

}

**#examine log-likelihoods**

sapply(list(out1,out2,out3),logLik)

[1] -782.3082 -761.0007 -764.3715

norm.log(out4,birds.short$S)

[1] -793.013

sapply(list(out1,out2,out3),AIC)

[1] 1578.616 1536.001 1544.743

-2\*norm.log(out4,birds.short$S)+2\*(length(coef(out3))+1)

[1] 1602.026

So, the log-transformed model performs the worst followed by the Poisson model with a log link. In contrast the normal model and Poisson model with identity link are fairly close. One advantage of fitting the Poisson model is that it has a built-in goodness of fit test in the residual deviance. We previously saw that the predicted counts are all fairly large so that the chi-squared distribution of the residual deviance is probably reasonable.

sum(fitted(out2)<5)

[1] 0

out2$deviance/out2$df.residual

[1] 1.019659

1-pchisq(out2$deviance,out2$df.residual)

[1] 0.4020139

The residual deviance provides no evidence of lack of fit. The ratio of the deviance to its degrees of freedom is very close to one so there is no suggestion of over- or underdispersion. I examine the coefficient estimates from this model and the normal model.

round(summary(out2)$coefficients,4)

Estimate Std. Error z value Pr(>|z|)  
(Intercept) 22.2227 0.8545 26.0066 0.0000  
landscapeBauxite -6.0596 0.8544 -7.0924 0.0000  
landscapeForest -2.5358 0.8899 -2.8496 0.0044  
landscapeUrban -7.0908 0.8825 -8.0351 0.0000  
factor(year)2006 -2.7670 0.7408 -3.7351 0.0002  
factor(year)2007 -2.6721 0.6928 -3.8569 0.0001  
log(area) 5.9571 0.3512 16.9621 0.0000

round(summary(out3)$coefficients,4)

Estimate Std. Error t value Pr(>|t|)  
(Intercept) 22.5414 0.8547 26.3739 0.0000  
landscapeBauxite -5.7663 0.8617 -6.6915 0.0000  
landscapeForest -2.3879 0.8467 -2.8204 0.0052  
landscapeUrban -6.9356 0.8953 -7.7468 0.0000  
factor(year)2006 -2.7441 0.7642 -3.5910 0.0004  
factor(year)2007 -2.5367 0.7043 -3.6019 0.0004  
log(area) 5.4854 0.3387 16.1976 0.0000

The coefficient estimates from the two models are very close with similar standard errors. Our conclusions and interpretations are the same no matter which of these models we choose. We will work initially with the normal model because the methods we consider next are easier to implement for normal models.

**Accounting for observational heterogeneity**

Although the final model appears to fit the data quite well, there is one aspect of these data that we have not yet addressed—observational heterogeneity arising from a structured data set. By observational heterogeneity I mean that the observations occur in groups where we expect the observations within a group to exhibit less variability than observations coming from different groups. This means that observations within a group are more similar to one another than they are to observations from different groups. The result is that observations coming from the same group are correlated. This has important ramifications for data analysis.

It is often the case that the "structure" of structured data assumes a hierarchical form. Observations are members of groups that in turn are nested in still larger groups. Structured data of this sort are also called hierarchical or multilevel data.

The existence of heterogeneity in the birds data set should be suspected from the way the data were collected. The sampling units were patches, not patch-years. A set of patches from a population of patches was selected for study. Once these patches were selected the plan was to visit each of these patches in all three years. (The fact that this did not happen in all cases was not part of the original design.) Thus the sample design here is similar to what is called a one-stage cluster sample in the field of survey sampling. The primary sampling unit was the patch and once a patch was selected all observations associated with that patch (all three years of data) were included. Fig. 1 illustrates a one-stage cluster sample for a population of 20 patches from which a random sample of 7 patches was selected. The sample unit is the entire block of three years of data yielding a total of 21 observations.

|  |
| --- |
| Fig. 1 |
| **Fig. 1** A repeated measures design viewed as a one-stage cluster sample ([R code](http://www.unc.edu/courses/2010fall/ecol/563/001/notes/lec23fig1and2Rcode.txt)) |

This is readily distinguishable from a simple random sample of patch-years. A practical way such a sample could arise is if in each of the three years a different simple random sample of patches was selected. This is called a cross-sectional design. With such a design we'd expect many patches to provide one year of data, some to provide two years, and hardly any to provide all three years of data (Fig. 2). With a very large population, observational heterogeneity would be largely non-existent because most patches would occur as singletons.

|  |
| --- |
| Fig. 2 |
| **Fig. 2** A simple random sample of patches within each year ([R code](http://www.unc.edu/courses/2010fall/ecol/563/001/notes/lec23fig1and2Rcode.txt)) |

The birds data set is an example of a repeated measures design in which patches were repeatedly visited over time. Observational heterogeneity is likely to exist because we have observations made on the same patch (the repeated measures) as well as observations made on different patches. We would expect observations from the same patch to be more similar (vary less) than observations coming from different patches.

**Including a blocking variable to account for data structure**

If we examine the raw data we find considerable evidence that observations from the same patch in different years are more similar to each other than to observations in different patches. Fig. 3 displays the raw counts for patches with data from all three years.

good.names2 <- names(table(birds.short$patch))[table(birds.short$patch)>2]

birds.short3 <- birds.short[birds.short$patch %in% good.names2,]

my.col <- c('dodgerblue3','tomato')

library(lattice)

dotplot(patch~S, data=birds.short3, xlab='Richness', panel=function(x,y){

panel.dotplot(jitter(x), y, col=my.col[as.numeric(y)%%2+1], pch=as.numeric(y)%%4+15)

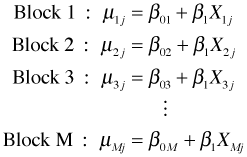
}, scales=list(y=list(cex=.7)))

|  |
| --- |
| fig 3 |
| **Fig. 3** Dot plot of richness by patch for those patches with three years of data |

One explanation for why observations coming from the same patch are similar is that they share characteristics in common with each other that they don't share with observations coming from different patches. A model-based way to account for shared characteristics is to include a term in the model that takes a different value on each patch but has the same value for all observations coming from the same patch. An obvious way to accomplish this is to treat patch as a factor and include it in the model as a predictor thus generating a set of dummy regressors.

A comparable situation arises in analyzing data from an experiment called a randomized block design. As an illustration of such a design consider the following example from agriculture. To test the effect of four different diets on the growth of pigs we might select multiple litters (farrows) of pigs and from each litter choose four animals to which the four diets are randomly assigned. The rationale for this design is to reduce extraneous variability that is unrelated to the effect of diet. Because an animal's genetics influences its growth, we are more likely to detect differences in growth due to diet if we confine our comparisons to piglets from the same litter than if diets were randomly assigned to animals without regard to family. Analytically we include litter in an analysis of variance model as a factor. Because litter is being use as a statistical control we refer to litter as a blocking variable rather than a treatment (also a categorical variable). A block is the categorical equivalent of a covariate (which is continuous).

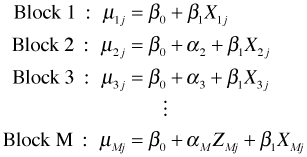
Suppose *X* is a dummy variable indicating the level of the treatment of interest (for simplicity let it have just two levels) and there are *M* blocks, then the randomized block design can be written as follows.



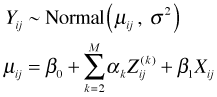
The first subscript denotes the block and the second subscript denotes the observation from that block. In this hypothetical example there are only two observations per block so *j* = 1, 2. Each block has a different intercept and the two observations within a block have the same value of this intercept. The randomized block design can be expressed more succinctly as follows.

|  |  |
| --- | --- |
| random block model | (1) |

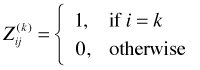
Here *i* = 1, 2, … , *M* denotes the block and more generally *j* = 1, 2, … , *ni* denotes the observations on that block. Typically we treat the blocking variable as a factor so that an alternative representation of the randomized block design is the following.



In this notation β0 is the intercept for block 1. For any block *j* with *j* ≠ 1, the intercept is β0 + αj so that αj is the amount the intercept in block *j* differs from the intercept of block 1. By constructing *M* – 1 dummy variables *Z*(2), *Z*(3), … , *Z*(M) for the blocks the full model can be written succinctly as follows.



where



In the birds data set patch plays the role of a block and year plays the role of a "treatment". Each "block" contains between 1 and 3 observations depending upon how many times that patch was visited. Thus we can account for the structure of the data by fitting the following randomized block design.

**#treat patches as blocks and estimate a randomized block design**

out1.lm <- lm(S~factor(patch)+factor(year), data=birds.short)

I extract the estimates that correspond to the year effects.

summary(out1.lm)$coefficients[102:103,]

Estimate Std. Error t value Pr(>|t|)  
factor(year)2006 -3.483249 0.5457852 -6.382089 1.949181e-09  
factor(year)2007 -2.841858 0.4992840 -5.691867 6.170423e-08

It's worthwhile comparing these estimates to what we would get if we had ignored the structure of the data altogether.

out0.lm <- lm(S~factor(year), data=birds.short)

summary(out0.lm)$coefficients

Estimate Std. Error t value Pr(>|t|)  
(Intercept) 25.484211 0.7828133 32.554648 1.275833e-92  
factor(year)2006 -4.055639 1.2018524 -3.374490 8.551756e-04  
factor(year)2007 -3.092906 1.1160536 -2.771288 5.996428e-03

The magnitudes of the effects are similar but there is a striking difference in the precision of the estimates reported for the two models. The standard errors in the randomized block design are less than half of what they are when blocking is ignored. This is the primary reason for blocking in a randomized block design, to increase the precision of the estimates of the treatment effect.

A major nuisance in the fixed effects approach to the randomized block design is that we're forced to estimate the separate block effects, in this case 100 of them. This is actually more than just a nuisance. Recall that one of the motivations in collecting the bird data was to determine how landscape type affects bird richness. Suppose we add landscape and patch area as predictors to the randomized block design and examine the estimates we obtain.

out3.lm <- lm(S~factor(patch)+factor(year)+log(area)+landscape, data=birds.short)

coef(out3.lm)

(Intercept) factor(patch)ag1b factor(patch)ag1c factor(patch)ag1d factor(patch)ag2a   
24.0000000 -9.0000000 1.0000000 11.0000000 4.7750357   
factor(patch)ag2b factor(patch)ag2c factor(patch)ag2d factor(patch)ag3a factor(patch)ag3b   
16.9209289 1.1083691 10.4417024 13.7750357 11.4417024   
factor(patch)ag3c factor(patch)ag3d factor(patch)ag3e factor(patch)ag3f factor(patch)ag4a   
0.4417024 3.7750357 1.9209289 -6.8916309 -6.5582976   
factor(patch)ag4b factor(patch)ag4c factor(patch)ag4d factor(patch)ag4e factor(patch)ag4f   
12.7416247 10.4417024 14.9209289 11.0000000 6.7750357   
factor(patch)ag5a factor(patch)ag5b factor(patch)ag5c factor(patch)ag5d factor(patch)ag5e   
11.7750357 -1.2249643 13.9209289 8.1083691 10.7750357   
factor(patch)b1a factor(patch)b1b factor(patch)b1c factor(patch)b1d factor(patch)b1e   
10.7750357 -1.8916309 -10.8916309 3.1083691 3.7750357   
factor(patch)b1f factor(patch)b1g factor(patch)b2a factor(patch)b2b factor(patch)b2c   
-8.8374464 -4.3374464 6.1083691 -6.2249643 2.7750357   
factor(patch)b2d factor(patch)b2e factor(patch)b3a factor(patch)b3b factor(patch)b3c   
-5.5582976 -9.2249643 1.4417024 2.7750357 -11.5582976   
factor(patch)b3d factor(patch)b3e factor(patch)b3f factor(patch)b4a factor(patch)b4b   
0.7750357 2.4417024 -11.8916309 1.4417024 5.7750357   
factor(patch)b4c factor(patch)b4d factor(patch)b4e factor(patch)b5a factor(patch)b5b   
-4.5582976 -12.0000000 -8.8916309 0.8418578 4.8418578   
factor(patch)b5c factor(patch)b5d factor(patch)Ref1a factor(patch)Ref1b factor(patch)Ref1c   
1.8418578 -3.1581422 12.1083691 -6.5582976 2.7750357   
factor(patch)Ref1d factor(patch)Ref1e factor(patch)Ref1g factor(patch)Ref2a factor(patch)Ref2b   
-4.0000000 8.0000000 -7.5790711 12.7750357 8.1083691   
factor(patch)Ref2c factor(patch)Ref2d factor(patch)Ref2e factor(patch)Ref3a factor(patch)Ref3b   
15.1083691 1.7750357 0.9209289 -10.2249643 0.1083691   
factor(patch)Ref3c factor(patch)Ref3d factor(patch)Ref4a factor(patch)Ref4b factor(patch)Ref4c   
-1.2249643 1.7750357 -3.0790711 4.9209289 1.9209289   
factor(patch)Ref4d factor(patch)Ref5a factor(patch)Ref5b factor(patch)Ref5c factor(patch)Ref5d   
2.4209289 -7.0790711 6.4209289 3.4209289 1.4209289   
factor(patch)Ref5e factor(patch)Ref6a factor(patch)Ref6b factor(patch)Ref6c factor(patch)Ref6d   
-5.0790711 -2.5790711 2.4209289 5.4209289 6.4209289   
factor(patch)Ref6e factor(patch)Ref6f factor(patch)u1a factor(patch)u1b factor(patch)u1c   
12.4417024 9.1083691 -5.8916309 0.4417024 -5.5582976   
factor(patch)u1d factor(patch)u1e factor(patch)u2a factor(patch)u2b factor(patch)u2c   
-10.2249643 -4.2249643 7.7750357 -6.5582976 -0.5582976   
factor(patch)u2d factor(patch)u2e factor(patch)u3a factor(patch)u3b factor(patch)u3c   
-5.8916309 1.1083691 8.1083691 -2.5582976 -0.2249643   
factor(patch)u3d factor(patch)u4a factor(patch)u4b factor(patch)u4c factor(patch)u4d   
-7.2249643 -2.2249643 -7.8916309 -2.2249643 -1.5582976   
factor(patch)u4e factor(year)2006 factor(year)2007 log(area) landscapeBauxite   
2.1083691 -3.4832494 -2.8418578 NA NA   
landscapeForest landscapeUrban   
NA NA

Observe that no estimates are reported for log(area) and landscape. Area and landscape are constant on a patch and hence they are subsumed by the individual patch effects. From a statistical point of view landscape and area are completely collinear with patch. If we choose to estimate the individual patch effects, then we cannot also estimate the effect of landscape.

**The random intercepts model**

The inability to simultaneously include block predictors in the randomized block design coupled with the overall cumbersomeness of having to estimate so many additional parameters are two drawbacks of treating the randomized block design as an ordinary regression model. In contrast to this fixed effects approach, it is often preferable to fit what's called a mixed effects model. Rather than formally estimating the individual intercepts of the randomized block design we instead assume they come from a single probability distribution and we estimate the parameters of that distribution instead. The usual assumption is that the distribution of the intercepts is normal. The mixed effects formulation of the randomized block design can be written as follows.

|  |  |
| --- | --- |
| random intercepts | (2a) |

The first subscript *i* denotes the patch and the second subscript *j* denotes the individual observation on that patch. Because the β0i are random variables we refer to the β0i as random intercepts and so this model is called a random intercepts model. A second way the random intercepts model is often written is as follows.

|  |  |
| --- | --- |
| random intercepts | (2b) |

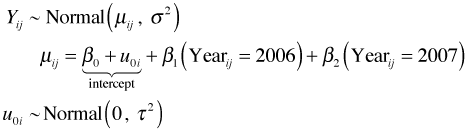
In this formulation the *u*0i represent the deviations from the population mean intercept value β0 and so are called the random intercept effects or, when there is no chance of confusion, simply random effects.

In a regression model with a normally distributed response we can view the response as having a normal distribution with mean given by the regression model, or we can view the deviations from that regression line, the residuals, as being normally distributed with mean zero. Thus a third way of formulating the random intercepts model with a normally distributed response is the following.

|  |  |
| --- | --- |
| random intercepts | (2c) |

**Fitting a random intercepts model in R**

Two packages in R can be used to fit random intercepts models with a normally distributed response, **nlme** and **lme4**. The **nlme** package is part of the standard installation of R, so we'll work with it first. The function that fits the random intercepts model is **lme**. We start by fitting a model in which mean richness varies by year.



So, in this model we assume that the patch richness values in 2005 are drawn from a common population with mean β0. These patch means all then shift by an amount β1 in 2006 and by an amount β2 in year 2007 relative to their values in 2005. This model is fit in R with the **lme** function as follows.

library(nlme)

out1.lme <- lme(S~factor(year), data=birds.short, random=~1|patch, method="ML")

* The notation S~factor(year) specifies what's called the fixed effects portion of the model. It indicates that mean richness is determined as follows.

fixed effects

The coefficients β0, β1, and β2 are referred to as fixed effects in the model.

* The notation random=~1|patch defines the random effects portion of the model. The numeral 1 is R's notation for the intercept and thus this indicates we want random intercepts. The notation |patch means that the intercepts should vary by patch. The default assumption is that these random intercepts are independent and come from a normal distribution with mean 0 and variance τ2.
* The argument method="ML" is needed in order to obtain maximum likelihood estimates. The default estimation method is something called REML.

The summary table of the **lme** model resembles the summary tables of **lm** and **glm** that we've seen before.

summary(out1.lme)

Linear mixed-effects model fit by maximum likelihood  
Data: birds.short   
AIC BIC logLik  
1594.871 1612.616 -792.4354

Random effects:  
Formula: ~1 | patch  
(Intercept) Residual  
StdDev: 6.816287 3.264958

Fixed effects: S ~ factor(year)   
Value Std.Error DF t-value p-value  
(Intercept) 25.328225 0.7639798 154 33.15300 0  
factor(year)2006 -3.540433 0.5414112 154 -6.53927 0  
factor(year)2007 -2.866248 0.4952889 154 -5.78702 0  
Correlation:   
(Intr) f()2006  
factor(year)2006 -0.279   
factor(year)2007 -0.315 0.460

Standardized Within-Group Residuals:  
Min Q1 Med Q3 Max   
-2.04592545 -0.56583341 -0.01271766 0.56325043 2.48919162

Number of Observations: 257  
Number of Groups: 101

The fixed effects estimates are quite close to what we obtained with the separate intercepts (randomized block design) model.

**#estimates from random intercepts model**

summary(out1.lme)$tTable

Value Std.Error DF t-value p-value  
(Intercept) 25.328225 0.7639798 154 33.153004 5.360123e-72  
factor(year)2006 -3.540433 0.5414112 154 -6.539267 8.610398e-10  
factor(year)2007 -2.866248 0.4952889 154 -5.787022 3.884851e-08

**#estimates from separate intercepts model**

summary(out1.lm)$coefficients[c(102:103),]

Estimate Std. Error t value Pr(>|t|)  
factor(year)2006 -3.483249 0.5457852 -6.382089 1.949181e-09  
factor(year)2007 -2.841858 0.4992840 -5.691867 6.170423e-08

The estimates of the term labeled (Intercept) in the two models are not comparable. In the random intercepts model the fixed effects intercept estimate is the population average intercept. In the separate intercepts model the intercept represents the intercept of the reference patch, "ag1a".

The novel portion of the **lme** output is the random effects section. We can extract this information from the model with the **VarCorr** function.

VarCorr(out1.lme)

patch = pdLogChol(1)   
Variance StdDev   
(Intercept) 46.46178 6.816287  
Residual 10.65995 3.264958

The notation in the output matches the formulation of the random intercepts model in eqn (2c). There we specified that the residuals have a normal distribution, residuals, and that the random intercept effects have a normal distribution, intercepts. From the output we see that the residual variance σ2 = 10.66 and the intercept variance τ2 = 46.46.

Because we specified **method="ML"** we can extract a log-likelihood and AIC from the model for use in comparing models. If we compare the separate intercepts and random intercepts model we find, not surprisingly, that the separate intercepts model has a much larger log-likelihood.

logLik(out1.lm)

'log Lik.' -604.7782 (df=104)

logLik(out1.lme)

'log Lik.' -792.4354 (df=5)

If we use AIC which penalizes the separate intercepts model for the 99 additional parameters that it estimates, AIC still ranks the separate intercepts model as the better of the two models.

sapply(list(out1.lm, out1.lme), AIC)

[1] 1417.556 1594.871

This is not an unusual result and it is not a sufficient reason to reject the random intercepts model in favor of the separate intercepts model. AIC chooses the separate intercepts model because it does fit the data better, although overfit might be a better description. The objective in using the random intercepts model is to account for the structure of the data while still making it possible to assess the manner in which patch-level predictors affect richness. This is impossible to do with the separate intercepts model.

**Random intercepts models with patch-level predictors**

The inclusion of patch-level predictors in a random intercepts model poses no special problems for the **lme** function. We just include them as predictors in the fixed effects portion of the model specification.

out2.lme <- lme(S~factor(year)+log(area), data=birds.short, random=~1|patch, method="ML")

out3.lme <- lme(S~factor(year)+log(area)+landscape, data=birds.short, random=~1|patch, method="ML")

sapply(list(out1.lm, out1.lme, out2.lme, out3.lme), AIC)

[1] 1417.556 1594.871 1513.075 1487.639

Based on AIC we conclude that log(area) and landscape are important predictors of mean richness. Notice that the AIC of these models still can't compete with the separate intercepts model, but that model tells us nothing about why patches differ just that they do. The models with patch-level predictors on the other hand suggest possible reasons for the patch differences.

In the language of multilevel models we would call any of the random intercepts models we've been fitting a two-level model. The patches define the level-2 units while the individual observations on that patch (the patch-year observations) define the level-1 units. A variable such as year that varies among the level-1 units within a level-2 unit is called a level-1 variable whereas variables like area and landscape that vary between patches (but are constant within a patch) are called level-2 variables. This distinction manifests itself in the coefficient table of the parameter estimates.

round(summary(out3.lme)$tTable,4)

Value Std.Error DF t-value p-value  
(Intercept) 22.5324 1.0761 154 20.9395 0.0000  
factor(year)2006 -3.2236 0.5403 154 -5.9668 0.0000  
factor(year)2007 -2.6961 0.4940 154 -5.4576 0.0000  
log(area) 5.3747 0.4535 96 11.8511 0.0000  
landscapeBauxite -5.4288 1.1717 96 -4.6333 0.0000  
landscapeForest -2.2564 1.1292 96 -1.9983 0.0485  
landscapeUrban -6.5973 1.2456 96 -5.2967 0.0000

The level-1 and level-2 variables are assigned different degrees of freedom and the fact that this has happened tells us that we've formulated the model correctly. The effects associated with the two patch-level predictors, area and landscape, are listed as having 96 degrees of freedom, while the effects associated with the individual observations within a patch are listed as having 154 degrees of freedom. The 96 degrees of freedom derives from the number of patches less than the number of estimated patch-related parameters whereas the 154 degrees of freedom derives from the number of observations less the number of patches less the number of estimated level-1 coefficients.

The **fixef** function applied to the model extracts the coefficient estimates of the fixed effects terms.

fixef(out3.lme)

(Intercept) factor(year)2006 factor(year)2007 log(area) landscapeBauxite   
22.532391 -3.223622 -2.696105 5.374737 -5.428829   
landscapeForest landscapeUrban   
-2.256370 -6.597316

Even though the random intercepts are not estimated in fitting the random effects model it is possible to obtain predictions of the random intercepts in the model after the fact. To see these predictions use the **ranef** function applied to the model. The reported values correspond to the terms denoted *u*0i in eqns (2b) and (2c).

dim(ranef(out3.lme))

[1] 101 1

ranef(out3.lme)[1:101,]

[1] -2.78295013 -2.59471511 -0.97753261 0.58298047 3.38749799 -2.50227695 4.29151270  
[8] 3.02831485 -0.68893887 2.78643368 0.63783600 -0.73845982 -0.14674056 -4.22928174  
[15] -3.90244776 3.40382545 2.89071456 -3.54708099 -1.73237856 2.13988571 0.93393248  
[22] -0.69733917 -2.34788845 1.64763369 1.15746314 2.78422372 1.48255224 -0.93409843  
[29] 4.27220401 2.96057712 -0.65642833 1.07253226 0.92969023 -2.14594320 0.57459653  
[36] -2.20908644 -3.20546701 0.21833909 1.67031339 -4.08324725 2.71916929 -0.06702808  
[43] -5.02999389 0.28620677 0.09638525 -0.22675310 -3.77535381 -0.73733920 1.00991344  
[50] 2.08975071 1.39469202 -0.49040733 5.02958656 -3.03975423 -0.28430664 -0.66181929  
[57] 0.47130427 -12.75691903 7.41248981 5.74140265 7.33862550 0.86676490 0.23071941  
[64] -5.86191293 -4.20656918 -3.36302522 0.86676490 -0.28153046 1.26910603 0.92091591  
[71] 1.26601416 -3.04231644 0.62770577 -1.44288371 -1.14658170 -1.66192345 0.06356779  
[78] 1.26601416 1.61420427 2.30440077 4.00308969 -3.54313427 -2.56249738 0.05940408  
[85] -0.36448272 -3.19887359 -5.12464804 5.72936602 -0.43195480 0.74722925 0.61692770  
[92] 0.74222765 4.42396079 -2.66582004 1.19870825 -0.80526052 1.92703958 -0.05010292  
[99] -0.03196336 0.84020354 -1.04946350

The **coef** function combines the random and fixed effects together and lists the results separately for each of the 101 level-2 units (patches). In the column labeled (Intercept) are reported β0i = β0 + *u*0i. In the rest of the columns we see just the usual fixed effect estimates—the coefficients of the remaining regressors in the model.

dim(coef(out3.lme))

[1] 101 7

coef(out3.lme)[1:6,]

(Intercept) factor(year)2006 factor(year)2007 log(area) landscapeBauxite landscapeForest  
ag1a 19.74944 -3.223622 -2.696105 5.374737 -5.428829 -2.256370  
ag1b 19.93768 -3.223622 -2.696105 5.374737 -5.428829 -2.256370  
ag1c 21.55486 -3.223622 -2.696105 5.374737 -5.428829 -2.256370  
ag1d 23.11537 -3.223622 -2.696105 5.374737 -5.428829 -2.256370  
ag2a 25.91989 -3.223622 -2.696105 5.374737 -5.428829 -2.256370  
ag2b 20.03011 -3.223622 -2.696105 5.374737 -5.428829 -2.256370  
landscapeUrban  
ag1a -6.597316  
ag1b -6.597316  
ag1c -6.597316  
ag1d -6.597316  
ag2a -6.597316  
ag2b -6.597316

**Comparing fixed effects intercept estimates with random intercept predictions**

It's useful to compare the intercepts from the separate intercepts model with the predictions of the random intercepts obtained from the random intercepts model. To facilitate this comparison I first refit the separate intercepts model but explicitly remove the intercept term so that the **lm** function returns an intercept estimate for each patch, rather than reporting the intercept for the reference patch plus 100 effect estimates.

out1a.lm <- lm(S~factor(patch)+factor(year)-1, data=birds.short)

fix.ints <- coef(out1a.lm)[1:101]

coef(out1a.lm)[102:103]

factor(year)2006 factor(year)2007   
-3.483249 -2.841858

ran.ints <- coef(out1.lme)[,1]

length(ran.ints)

[1] 101

mydata <- data.frame(patch=rownames(coef(out1.lme)), fixed=fix.ints, random=ran.ints)

mydata[1:10,]

patch fixed random  
factor(patch)ag1a ag1a 24.00000 24.24787  
factor(patch)ag1b ag1b 15.00000 16.92743  
factor(patch)ag1c ag1c 25.00000 25.06125  
factor(patch)ag1d ag1d 35.00000 33.19507  
factor(patch)ag2a ag2a 28.77504 28.55542  
factor(patch)ag2b ag2b 40.92093 39.32720  
factor(patch)ag2c ag2c 25.10837 25.14925  
factor(patch)ag2d ag2d 34.44170 33.81950  
factor(patch)ag3a ag3a 37.77504 36.91601  
factor(patch)ag3b ag3b 35.44170 34.74845

Fig. 4 displays the two groups of intercepts together in a dot plot.

**#display both together in same graph: random ints are open circles**

dotplot(patch~fixed, data=mydata ,xlab='intercept', panel=function(x,y) {

panel.dotplot(x,y)

panel.points(mydata$random, as.numeric(y), pch=1, cex=.7)

panel.abline(v=fixef(out1.lme)[1], lty=2, col=2)

},scales=list(y=list(cex=.7)), key=list(x=.75, y=.925, corner=c(0,0), text=list(c('random intercepts', 'fixed intercepts'), cex=.85), points=list(pch=c(1,16), col='dodgerblue', cex=.8)))

|  |
| --- |
| fig 4 |
| **Fig. 4** Dot plot of separate intercept estimates and predictions of the random intercepts |

The vertical line in the picture denotes the population average estimate of the intercept, β0, from the random intercepts model. What is striking in Fig. 4 is that in all cases the random intercept prediction in a patch is closer to the population average estimate than is the separate intercepts estimate in that patch. This phenomenon is called shrinkage and the predictions of the random intercepts are sometimes called shrinkage estimates. Other terms that are used are empirical Bayes estimates and BLUPs (best linear unbiased predictors).

The shrinkage arises because the intercepts are constrained to arise from a single distribution of intercepts. Thus they are not as free to vary as are the intercepts in the separate intercepts model. This constraint makes the intercepts less sensitive to the vagaries of the data in individual patches and prevents overfitting. In simple situations the shrinkage estimate for a patch can be viewed as a weighted mean of the individual least squares estimate (obtained from the separate intercepts model) and the population average estimate of the intercept, β0. Patches with more patch-level observations will show less shrinkage. Shrinkage estimates of the intercepts for patches with very little data will exhibit greater shrinkage to the population average value.

**R Code**

A compact collection of most of the R code displayed in this document appears [here](http://www.unc.edu/courses/2010fall/ecol/563/001/notes/lecture23%20Rcode.html).

[Course Home Page](http://www.unc.edu/courses/2010fall/ecol/563/001/index.html)

|  |
| --- |
| Jack Weiss *Phone:* (919) 962-5930 *E-Mail:* jack\_weiss@unc.edu *Address:* Curriculum in Ecology, Box 3275, University of North Carolina, Chapel Hill, 27599 Copyright © 2010 Last Revised--November 28, 2010 URL: [http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture23.htm](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture23.htm#lecture23) |