Generalized Linear Mixed Models

Quantitative Analysis of Vertebrate Populations

Generalize Linear Model

Example with a Poisson distribution for count data. **Why do we use Poisson for counts?**

1. Distribution: $C_i \sim Poisson(\lambda_i)$
2. Link Function: $log(\lambda_i)$
3. Linear Predictor: $log(\lambda_i) = \alpha + \beta X_i$

Generalized Linear Mixed Model

Add submodels for the random effect parameters.

1. Distribution: $C_i \sim Poisson(\lambda_i)$
2. Link Function: $log(\lambda_i)$
3. Linear Predictor: $log(\lambda_i) = \alpha_{j(i)} + \beta X_i$
4. Submodel for random effects parameters

$$\alpha_j \sim \mathcal{N}(\mu_\alpha, \sigma^2_\alpha)$$

Random intercept and slope model

1. Distribution: $C_i \sim Poisson(\lambda_i)$
2. Link Function: $log(\lambda_i)$
3. Linear Predictor: $log(\lambda_i) = \alpha_{j(i)} + \beta_{j(i)} X_i$
4. Submodel for random effects parameters

$$\alpha_j \sim \mathcal{N}(\mu_\alpha, \sigma^2_\alpha)$$
$$\beta_j \sim \mathcal{N}(\mu_\beta, \sigma^2_\beta)$$

Data generation (From Kery 2010)

Random slope and intercepts without correlation

```
n.groups <- 16
n.years <- 30
n <- n.groups * n.years
pop <- gl(n = n.groups, k = n.years)
```

Standardize the year covariate to range from zero to one.

```
original.year <- rep(1:n.years, n.groups)
year <- (original.year-1)/29
```

Build design matrix without intercept

```r
Xmat <- model.matrix(~pop*year-1-year)
print(Xmat[1:21,], dig = 2)          # Print top 21 rows
```

```
##    pop1 pop2 pop3 pop4 pop5 pop6 pop7 pop8 pop9 pop10 pop11 pop12 pop13
## 1     1    0    0    0    0    0    0    0    0     0     0     0     0
## 2     1    0    0    0    0    0    0    0    0     0     0     0     0
## 3     1    0    0    0    0    0    0    0    0     0     0     0     0
## 4     1    0    0    0    0    0    0    0    0     0     0     0     0
## 5     1    0    0    0    0    0    0    0    0     0     0     0     0
## 6     1    0    0    0    0    0    0    0    0     0     0     0     0
## 7     1    0    0    0    0    0    0    0    0     0     0     0     0
## 8     1    0    0    0    0    0    0    0    0     0     0     0     0
## 9     1    0    0    0    0    0    0    0    0     0     0     0     0
## 10    1    0    0    0    0    0    0    0    0     0     0     0     0
## 11    1    0    0    0    0    0    0    0    0     0     0     0     0
## 12    1    0    0    0    0    0    0    0    0     0     0     0     0
## 13    1    0    0    0    0    0    0    0    0     0     0     0     0
## 14    1    0    0    0    0    0    0    0    0     0     0     0     0
## 15    1    0    0    0    0    0    0    0    0     0     0     0     0
## 16    1    0    0    0    0    0    0    0    0     0     0     0     0
## 17    1    0    0    0    0    0    0    0    0     0     0     0     0
## 18    1    0    0    0    0    0    0    0    0     0     0     0     0
## 19    1    0    0    0    0    0    0    0    0     0     0     0     0
## 20    1    0    0    0    0    0    0    0    0     0     0     0     0
## 21    1    0    0    0    0    0    0    0    0     0     0     0     0
##    pop14 pop15 pop16 pop1:year pop2:year pop3:year pop4:year pop5:year
## 1      0     0     0     0.000         0         0         0         0
## 2      0     0     0     0.034         0         0         0         0
## 3      0     0     0     0.069         0         0         0         0
## 4      0     0     0     0.103         0         0         0         0
## 5      0     0     0     0.138         0         0         0         0
## 6      0     0     0     0.172         0         0         0         0
## 7      0     0     0     0.207         0         0         0         0
## 8      0     0     0     0.241         0         0         0         0
## 9      0     0     0     0.276         0         0         0         0
## 10     0     0     0     0.310         0         0         0         0
## 11     0     0     0     0.345         0         0         0         0
## 12     0     0     0     0.379         0         0         0         0
## 13     0     0     0     0.414         0         0         0         0
## 14     0     0     0     0.448         0         0         0         0
## 15     0     0     0     0.483         0         0         0         0
## 16     0     0     0     0.517         0         0         0         0
## 17     0     0     0     0.552         0         0         0         0
## 18     0     0     0     0.586         0         0         0         0
## 19     0     0     0     0.621         0         0         0         0
## 20     0     0     0     0.655         0         0         0         0
## 21     0     0     0     0.690         0         0         0         0
##    pop6:year pop7:year pop8:year pop9:year pop10:year pop11:year
## 1          0         0         0         0          0          0
## 2          0         0         0         0          0          0
## 3          0         0         0         0          0          0
## 4          0         0         0         0          0          0
```

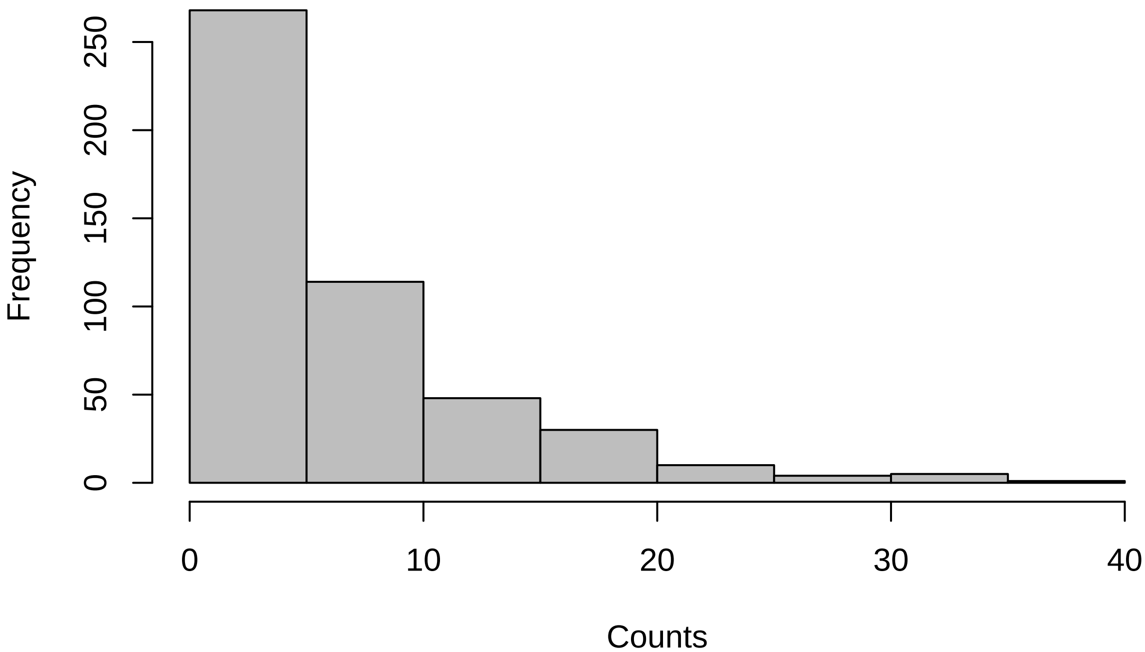
```
## 5          0          0          0          0          0          0
## 6          0          0          0          0          0          0
## 7          0          0          0          0          0          0
## 8          0          0          0          0          0          0
## 9          0          0          0          0          0          0
## 10         0          0          0          0          0          0
## 11         0          0          0          0          0          0
## 12         0          0          0          0          0          0
## 13         0          0          0          0          0          0
## 14         0          0          0          0          0          0
## 15         0          0          0          0          0          0
## 16         0          0          0          0          0          0
## 17         0          0          0          0          0          0
## 18         0          0          0          0          0          0
## 19         0          0          0          0          0          0
## 20         0          0          0          0          0          0
## 21         0          0          0          0          0          0
##    pop12:year pop13:year pop14:year pop15:year pop16:year
## 1           0          0          0          0          0
## 2           0          0          0          0          0
## 3           0          0          0          0          0
## 4           0          0          0          0          0
## 5           0          0          0          0          0
## 6           0          0          0          0          0
## 7           0          0          0          0          0
## 8           0          0          0          0          0
## 9           0          0          0          0          0
## 10          0          0          0          0          0
## 11          0          0          0          0          0
## 12          0          0          0          0          0
## 13          0          0          0          0          0
## 14          0          0          0          0          0
## 15          0          0          0          0          0
## 16          0          0          0          0          0
## 17          0          0          0          0          0
## 18          0          0          0          0          0
## 19          0          0          0          0          0
## 20          0          0          0          0          0
## 21          0          0          0          0          0
```

Draw random intercept and slopes from normal distributions

```r
intercept.mean <- 3         # Choose values for the hyperparams
intercept.sd <- 1
slope.mean <- -2
slope.sd <- 0.6
intercept.effects<-rnorm(n = n.groups, mean = intercept.mean, sd = intercept.sd)
slope.effects <- rnorm(n = n.groups, mean = slope.mean, sd = slope.sd)
all.effects <- c(intercept.effects, slope.effects) # Put them all together
```

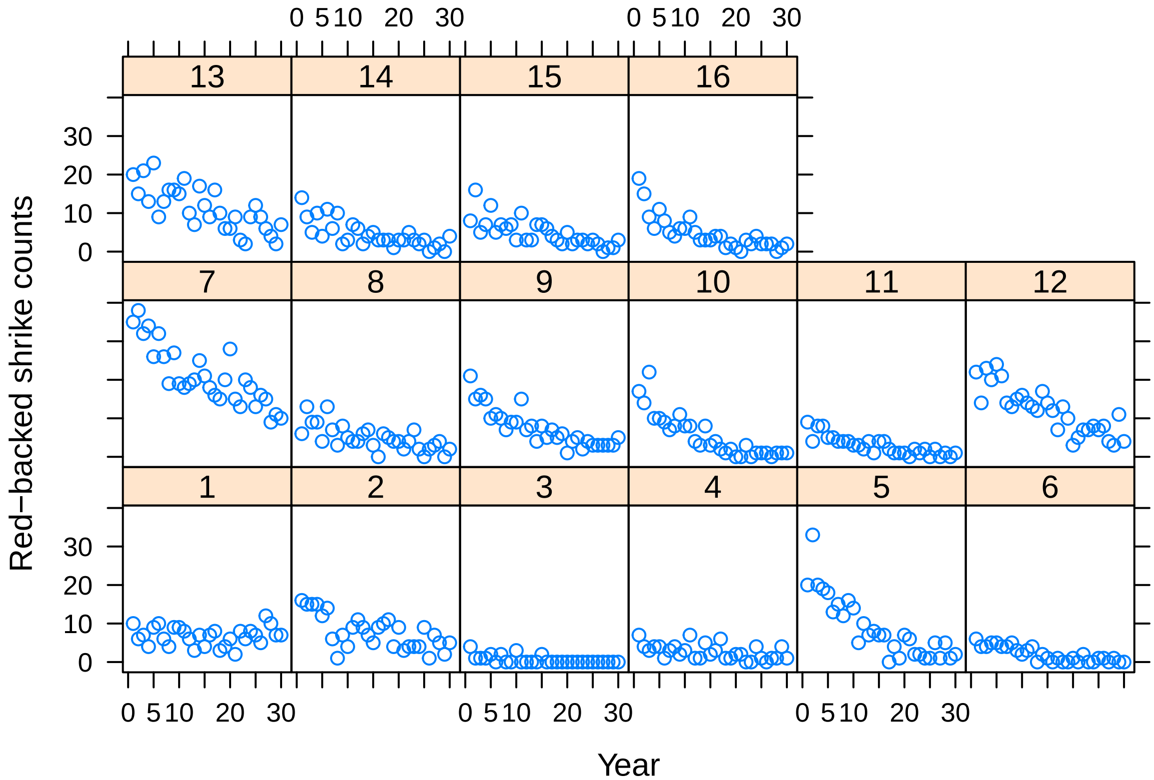
Assemble and inspect results

```r
lin.pred <- Xmat[,] %*% all.effects # Value of lin.predictor
C <- rpois(n = n, lambda = exp(lin.pred)) # Exponentiate and add Poisson noise
hist(C, col = "grey", xlab = "Counts", main = "")          # Inspect what we've created
```



Assemble and inspect results

```r
library("lattice")
xyplot(C ~ original.year | pop, ylab = "Red-backed shrike counts", xlab = "Year")
```



Analysis under a random-coefficients model

```r
library('lme4')
glmm.fit <- glmer(C ~ year + (1 | pop) + ( 0+ year | pop), family = poisson)
summary(glmm.fit)                # Inspect results
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: poisson  ( log )
## Formula: C ~ year + (1 | pop) + (0 + year | pop)
##
##      AIC      BIC   logLik deviance df.resid
##   2132.5   2149.2  -1062.2   2124.5      476
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.8614 -0.6851 -0.1275  0.5581  2.7093
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  pop    (Intercept) 0.4560   0.6753
##  pop.1  year        0.9218   0.9601
## Number of obs: 480, groups:  pop, 16
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.4356     0.1729  14.087  < 2e-16 ***
```

```
## year         -2.0116     0.2582  -7.792 6.62e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr)
## year -0.059
```

Examine Pearson residuals for patterning

```
plot(glmm.fit)
```