Lesson 18 (Actually Lesson 20...)

Clark

Let's consider data from a clinical trial of 59 epileptics. For a baseline, patience were observed for 8 weeks and the number of seizures were recorded. The patience were then randomized to treatment by the drug Progabide (31 patients) or to the placebo group (28 patience). They were then observed for four 2-week periods with the number of seizures recorded.

What are our observational units and what are our experimental units here?

```
library(faraway)
library(tidyverse)

data("epilepsy")
epilepsy$period <- rep(0:4, 59) # Time period observed
epilepsy$drug <- factor(c("placebo","treatment")[epilepsy$treat+1])
epilepsy$phase <- factor(c("baseline", "experiment")[epilepsy$expind+1])
epilepsy %>% head(2)
```

```
##   seizures id treat expind timeadj age period    drug      phase
## 1       11  1     0      0       8  31      0 placebo   baseline
## 2        5  1     0      1       2  31      1 placebo experiment
```

Now, Faraway does the following:

```
ratesum <- epilepsy %>%
  group_by(id, phase, drug) %>%
  summarise(rate=mean(seizures/timeadj))
```

```
## `summarise()` has grouped output by 'id', 'phase'. You can override using the
## `.groups` argument.
```

```
comsum <- spread(ratesum, phase, rate)
#ggplot(comsum, aes(x=baseline, y=experiment, shape=drug)) + geom_point() +
#  scale_x_sqrt() + scale_y_sqrt() + geom_abline(intercept=0, slope=1)+
#  theme(legend.position = "top", legend.direction = "horizontal")
```

and decides to exclude the upper right point from the analysis. Do you concur with this decision?

He next fits the data to a Poisson GLM. Let's write out the model and consider what it means:

Instead, let's consider the uniqueness of each individual in the study. Perhaps we want to consider a mixed model. Now, one option is to assume that that the mechanism of interest varies according to a gamma distribution. That is, we could do:

Now, if we want to maximize the likelihood we could write the likelihood for a single observation as:

In this case, we can think of λ as a nuisance that we need to get rid of. Recall, that the information we really want is contained within the expected value of the Gamma distribution. Currently, our likelihood is a function of λ, μ, and k. So, to get rid of λ we can integrate it out. That is we can do:

Unfortunately, having λ come from a Gamma distribution isn't the most natural way for us to think about the model. If we want to keep the same structure as our linear mixed effects models we could write:

Now, our likelihood could be written as:

And now, unfortunately, we have a problem. If our distribution of our data are normal and the distribution of our random effects are normal we can integrate out our random effects. If not, oftentimes this integral is not solvable through conventional methods. In fact, this can be quite difficult to do if our random effects become more complicated. However, all is not lost. What we will consider today is numerically approximating our integral.

Now, as it turns out, if our random effects are normally distributed, our integral is of the form

$$\int_{-\infty}^{\infty} h(u) \exp(-u^2) du \approx \sum_{k=1}^{q} c_k h(s_k)$$

Integrals of this form can be approximated by finding weights and quadrature points c_k and s_k For more see, eg. https://en.wikipedia.org/wiki/Gaussian_quadrature. The point here here is we can then approximate the integral and then maximize the likelihood (again this is done numerically). As our number of quadrature points q increase, our approximation to the integral gets better. We can start with a single quadrature point, which is called Laplace approximation.

We can do this in R with:

```
library(lme4)

model1 <- glmer(seizures~offset(log(timeadj))+phase*drug +
                  (1|id), nAGQ = 1, family=poisson,
                data=epilepsy)

summary(model1)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: poisson  ( log )
## Formula: seizures ~ offset(log(timeadj)) + phase * drug + (1 | id)
##    Data: epilepsy
##
##      AIC      BIC   logLik deviance df.resid
##   2032.2   2050.7  -1011.1   2022.2      290
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.8657 -0.9374 -0.1689  0.5559  9.8735
##
## Random effects:
```

```
##  Groups Name        Variance Std.Dev.
##  id     (Intercept) 0.6064   0.7787
## Number of obs: 295, groups:  id, 59
##
## Fixed effects:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    1.03274    0.15232   6.780  1.2e-11 ***
## phaseexperiment                0.11183    0.04671   2.394   0.0167 *
## drugtreatment                 -0.02142    0.21015  -0.102   0.9188
## phaseexperiment:drugtreatment -0.10472    0.06480  -1.616   0.1061
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) phsxpr drgtrt
## phasexprmnt -0.162
## drugtretmnt -0.724  0.117
## phsxprmnt:d  0.117 -0.721 -0.159
```

To get a more exact approximation, we can increase the number of quadrature points:

```
model2 <- glmer(seizures~offset(log(timeadj))+phase*drug +
                  (1|id), nAGQ = 25, family=poisson,
                data=epilepsy)

summary(model2)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 25) [glmerMod]
##  Family: poisson  ( log )
## Formula: seizures ~ offset(log(timeadj)) + phase * drug + (1 | id)
##    Data: epilepsy
##
##      AIC      BIC   logLik deviance df.resid
##    970.5    988.9   -480.2    960.5      290
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.8658 -0.9373 -0.1688  0.5558  9.8734
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  id     (Intercept) 0.6071   0.7791
## Number of obs: 295, groups:  id, 59
##
## Fixed effects:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    1.03265    0.15246   6.773 1.26e-11 ***
## phaseexperiment                0.11183    0.04688   2.386   0.0171 *
## drugtreatment                 -0.02140    0.21034  -0.102   0.9190
## phaseexperiment:drugtreatment -0.10472    0.06503  -1.610   0.1073
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
```

```
##             (Intr) phsxpr drgtrt
## phasexprmnt -0.162
## drugtretmnt -0.724  0.118
## phsxprmnt:d  0.117 -0.721 -0.159
```

Let's compare how quickly these execute:

```
start_time <- Sys.time()
model1 <- glmer(seizures~offset(log(timeadj))+phase*drug +
                  (1|id), nAGQ = 1, family=poisson,
                data=epilepsy)
end_time <- Sys.time()
end_time - start_time
```

```
## Time difference of 0.2563159 secs
```

```
start_time <- Sys.time()
model2 <- glmer(seizures~offset(log(timeadj))+phase*drug +
                  (1|id), nAGQ = 25, family=poisson,
                data=epilepsy)
end_time <- Sys.time()
end_time - start_time
```

```
## Time difference of 0.654927 secs
```

Not a huge problem here, but as our number of random effects increase, we may start to run into issues.

Note here that we can't compare model 1 to model 2 using AIC or anything else because they're the same model, just fit two different ways.

Let's look at the Poisson GLMM a bit more. Assuming we have a single random effect we can write:

We can use this to find $E[y_{ij}]$ by finding:

The Variance of y_{ij} in turn is:

If we want to look at the correlation between two observations within the same cluster, we have to do a bit more work than the Normal case where the covariance is just σ_u^2. Here it becomes:

$$\exp[(x_{ij} + x_{ik})\beta][\exp(2\sigma_u^2) - \exp(\sigma_u^2)]$$

From here it's straight forward to find the correlation between two observations. The key takeaway, though, is that the correlation between two observations within a single cluster *must* be positive. We cannot use a Poisson GLMM to model negatively correlated data. Is this significant? Well, maybe, maybe not. Typically we think about data that is clustered together to be more similar, so positively correlated, but it is something to keep in mind.

Let's look at another GLMM, a GLMM for binary responses. From Faraway, an experiment was conducted to study the effects of surface and vision on balance. The balance of subjects was observed for two different surfaces and for restricted and unrestricted vision. Balance was assessed qualitatively on an ordinal four-point scale (which we will turn into a two point scale). The subjects were tested while standing on foam or a normal surface and with their eyes closed or open or with a dome placed over their head. Each subject was tested twice in each of the surface and eye combinations.

Observational units? Experimental units?

```
data(ctsib)
ctsib$stable <- ifelse(ctsib$CTSIB==1,1,0)
ctsib %>% head(4)
```

```
##   Subject  Sex Age Height Weight Surface Vision CTSIB stable
## 1       1 male  22    176   68.2    norm   open     1      1
## 2       1 male  22    176   68.2    norm   open     1      1
## 3       1 male  22    176   68.2    norm closed     2      0
## 4       1 male  22    176   68.2    norm closed     2      0
```

Let's try the MA376 approach:

```
ctsib_glm <- glm(stable~Sex+Age+Height+Weight+
                   Surface+Vision+
                   factor(Subject),
                 family="binomial",
                 data=ctsib)
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

Our model is overspecified. We cannot estimate everything we might want to.

Should we remove Subject?

```
ctsib_test <- glm(stable~as.factor(Subject),family=binomial,data=ctsib)
anova(ctsib_test,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: stable
##
## Terms added sequentially (first to last)
##
##
##                     Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                  479     526.25
## as.factor(Subject) 39   88.925       440     437.33 9.084e-06 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

What does this show?

Here, perhaps, we want to consider a subject effect as a random effect. Or, in other words, we want to fit the model:

Which we do by:

```
library(lme4)
ctsib_glmm <- glmer(stable~Sex + Age + Height +
                      Weight + Surface + Vision +
                      (1|Subject),nAGQ=25,
                    family=binomial,
                    data=ctsib)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.289052 (tol = 0.002, component 1)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unide
##  - Rescale variables?
```

And herein we see an issue. If we look at Faraway's book it looks like everything is fine. However, clearly the `lme4` package has changed since publication and it's now giving a warning saying the model has failed to converge. Or, in otherwords, we can't really trust the likelihood. If we run into this, we likely need to simplify our model in someway, if instead we run a simpler model we are able to converge:

```
ctsib_glmm2 <- glmer(stable~Surface + Vision +
                      (1|Subject),nAGQ=25,
                    family=binomial,
                    data=ctsib)

summary(ctsib_glmm2)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 25) [glmerMod]
```

```
##  Family: binomial  ( logit )
## Formula: stable ~ Surface + Vision + (1 | Subject)
##    Data: ctsib
##
##      AIC      BIC   logLik deviance df.resid
##    247.3    268.2   -118.7    237.3      475
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.7957 -0.1741 -0.0199 -0.0006  5.7423
##
## Random effects:
##  Groups  Name        Variance Std.Dev.
##  Subject (Intercept) 9.556    3.091
## Number of obs: 480, groups:  Subject, 40
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -10.3189     1.5621  -6.606 3.95e-11 ***
## Surfacenorm   7.3938     1.0848   6.816 9.36e-12 ***
## Visiondome    0.6801     0.5296   1.284    0.199
## Visionopen    6.1635     0.9996   6.166 7.02e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Srfcnr Visndm
## Surfacenorm -0.905
## Visiondome  -0.287  0.113
## Visionopen  -0.885  0.834  0.376
```

Let's go through the output.

Now, we haven't really talked about how do we assess whether our model is good. We can extracte residuals and fitted values through (note the difference from Faraway page 281):

```
library(boot)
dd <- fortify.merMod(ctsib_glmm2)

phat <- inv.logit(dd$.fitted)

#plot(phat,ctsib$stable)
```

Let's say you want to model correlated responses differently within clusters. For example, let's say you have count data that you think has an AR(1) type relationship. For our linear model this was easy. For a non-Normal response, as best I know, you have to use Bayesian methods that we will discuss next class.