Generalized Linear Mixed Models (GLMM) II

Alexander McLain

Contents

1 Epilepsy randomized clinical trial

Taken from: Hothorn, T., & Everitt, B. S. (2014). **A handbook of statistical analyses using R.** CRC press.

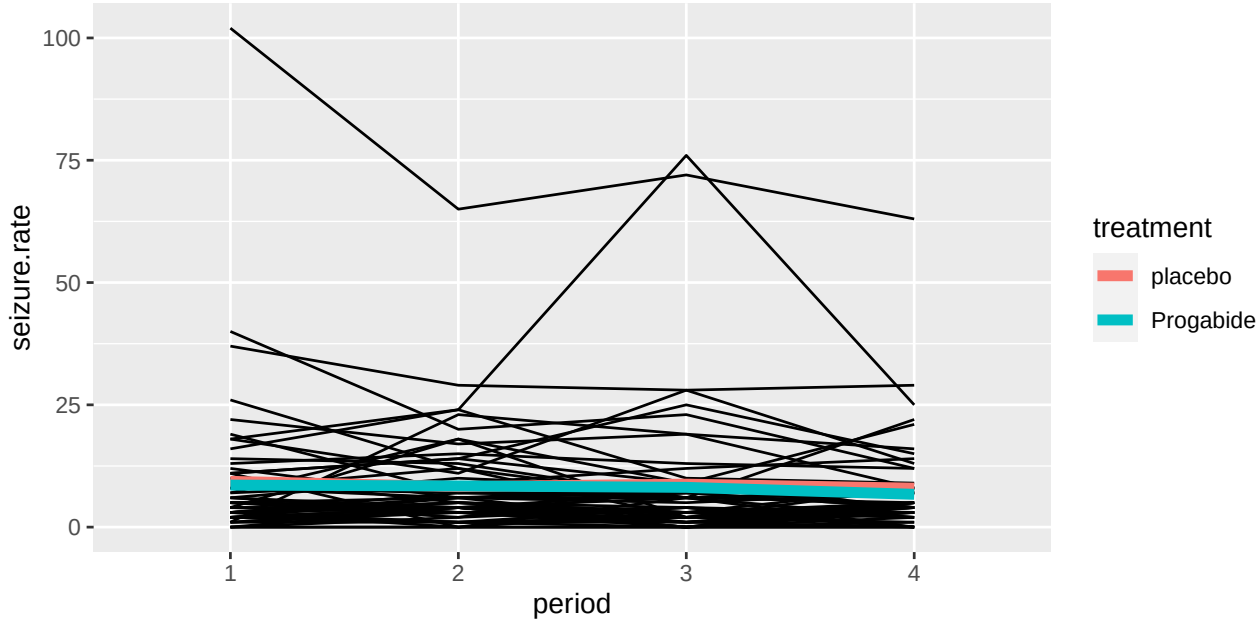
```
library(tidyverse)
data("epilepsy", package = "HSAUR2")
head(epilepsy)
```

	treatment	base	age	seizure.rate	period	subject
1	placebo	11	31	5	1	1
110	placebo	11	31	3	2	1
112	placebo	11	31	3	3	1
114	placebo	11	31	3	4	1
2	placebo	11	30	3	1	2
210	placebo	11	30	5	2	2

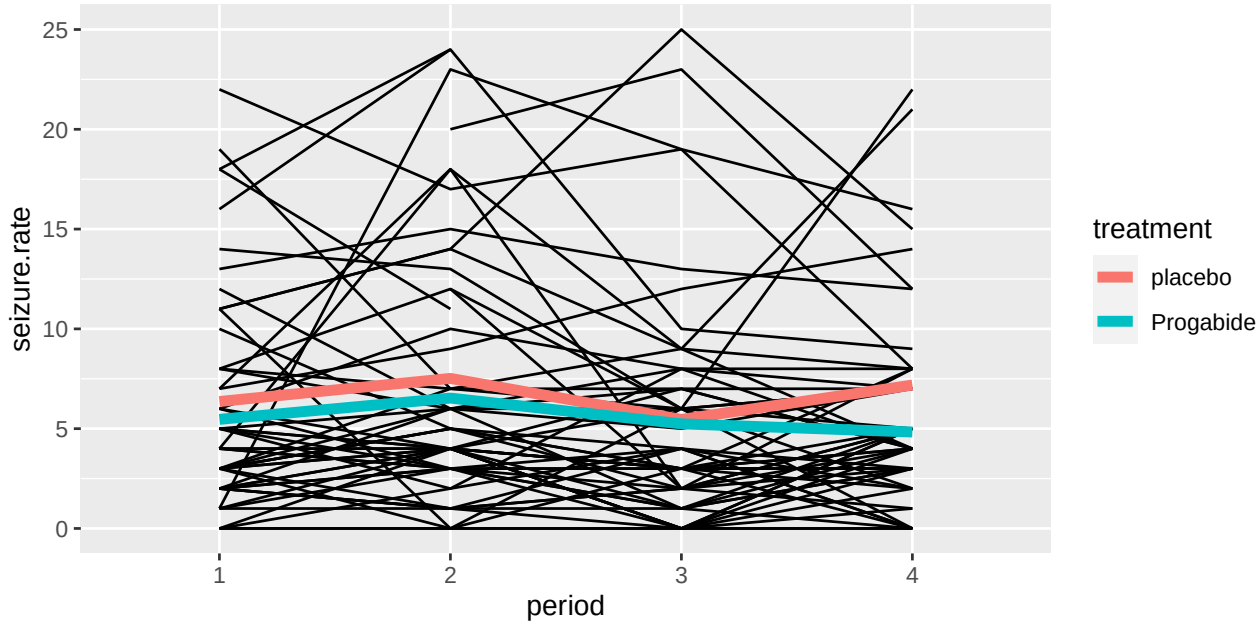
```
str(epilepsy)
```

```
## 'data.frame':    236 obs. of  6 variables:
##  $ treatment   : Factor w/ 2 levels "placebo","Progabide": 1 1 1 1 1 1 1 1 1 1 ...
##  $ base        : int  11 11 11 11 11 11 11 11 6 6 ...
##  $ age         : int  31 31 31 31 30 30 30 30 25 25 ...
##  $ seizure.rate: int  5 3 3 3 3 5 3 3 2 4 ...
##  $ period      : Ord.factor w/ 4 levels "1"<"2"<"3"<"4": 1 2 3 4 1 2 3 4 1 2 ...
##  $ subject     : Factor w/ 59 levels "1","2","3","4",..: 1 1 1 1 2 2 2 2 3 3 ...
```

```
p <- ggplot(epilepsy, aes(x = period, y = seizure.rate, group = subject))
p + geom_line() + stat_summary(aes(group = treatment, color = treatment), geom = "line",
                  fun = mean, size = 2)
```




```
p + geom_line() + stat_summary(aes(group = treatment, color = treatment), geom = "line",
                  fun = mean, size = 2) + ylim(0,25)
```



‘‘‘

We fit this data before, but this time we're going to do a better job controling for the number of seizures in the 5-week baseline period. The model were going to fit is

$$\log\left\{E\left(\frac{Y_{ij}}{2}\middle|b_{i0}\right)\right\} = \beta_0 + b_{i0} + \beta_1 \log(Base_i/5) + \beta_2 TRT_i + \beta_3 t_{ij} + \beta_4 TRT_i t_{ij}$$

Note that the 2 is there because each observation is over a 2 week period. This term will be put as an `offest` to the model. The expectation corresponding to this model is

$$E\left(\frac{Y_{ij}}{2}\middle| b_{i0}\right) = \left(\frac{Base_i}{5}\right)^{\beta_1} e^{\beta_0 + b_{i0} + \beta_2 TRT_i + \beta_3 t_{ij} + \beta_4 TRT_i t_{ij}}$$

```
library(lme4)
library(lmerTest)
```

```
epilepsy <- epilepsy %>% mutate( per = log(2), base_sc = log(base/5) ) %>%
  rename( trt = treatment)
fm <- seizure.rate ~ base_sc + age + trt + period + trt*period + (1|subject)
system.time(glmm_int_1 <- glmer(formula = fm, data = epilepsy, nAGQ = 1, family = "poisson",
                  offset = per))
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.0234883 (tol = 0.002, component 1)
```

```
##    user  system elapsed
##   0.500   0.008   0.517
```

```
system.time(glmm_int <- glmer(formula = fm, data = epilepsy, nAGQ = 20, family = "poisson",
                  offset = per))
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.0400499 (tol = 0.002, component 1)
```

```
##    user  system elapsed
##   0.726   0.002   0.728
```

```
logLik(glmm_int_1)
```

```
## 'log Lik.' -665.0871 (df=11)
```

```
logLik(glmm_int)
```

```
## 'log Lik.' -282.0144 (df=11)
```

```
VarCorr(glmm_int_1)
```

```
##  Groups  Name        Std.Dev.
##  subject (Intercept) 0.51771
```

```
VarCorr(glmm_int)
```

```
##  Groups  Name        Std.Dev.
##  subject (Intercept) 0.51957
```

```
round( coef( summary( glmm_int_1)), 3)
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-0.787	0.424	-1.857	0.063
base_sc	1.025	0.101	10.136	0.000
age	0.011	0.012	0.884	0.377
trtProgabide	-0.322	0.151	-2.140	0.032
period.L	-0.095	0.064	-1.481	0.139
period.Q	0.012	0.064	0.181	0.856
period.C	-0.075	0.064	-1.171	0.242
trtProgabide:period.L	-0.078	0.091	-0.853	0.394
trtProgabide:period.Q	-0.098	0.090	-1.084	0.278
trtProgabide:period.C	0.044	0.090	0.491	0.623

```
round( coef( summary( glmm_int)), 3)
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-0.788	0.426	-1.850	0.064
base_sc	1.025	0.102	10.086	0.000
age	0.011	0.012	0.882	0.378
trtProgabide	-0.323	0.151	-2.133	0.033
period.L	-0.096	0.064	-1.484	0.138
period.Q	0.011	0.065	0.177	0.860
period.C	-0.075	0.065	-1.158	0.247
trtProgabide:period.L	-0.077	0.092	-0.839	0.401
trtProgabide:period.Q	-0.098	0.091	-1.073	0.283
trtProgabide:period.C	0.043	0.090	0.481	0.631

Let's add a random slope to the model.

```
fm <- seizure.rate ~ base_sc + age + trt + period + trt*period +
                     (1 + as.numeric(period)|subject)

system.time(glmm_int_slp <- glmer(formula = fm, data = epilepsy, nAGQ = 1, family = "poisson",
                  offset = per))
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.0277794 (tol = 0.002, component 1)
```

```
##    user  system elapsed
##   0.583   0.003   0.587
```

```
VarCorr(glmm_int_slp)
```

```
##  Groups  Name                Std.Dev. Corr
##  subject (Intercept)         0.64706
##          as.numeric(period) 0.14996  -0.605
```

```
anova(glmm_int_1,glmm_int_slp)
```

	npar	AIC	BIC	logLik	deviance	Chisq	Df	Pr(>Chisq)
glmm_int_1	11	1352.174	1390.276	-665.0871	1330.174	NA	NA	NA
glmm_int_slp	13	1336.214	1381.243	-655.1068	1310.214	19.96065	2	4.63e-05

```
anova(glmm_int,glmm_int_slp)
```

	npar	AIC	BIC	logLik	deviance	Chisq	Df	Pr(>Chisq)
glmm_int	11	586.0288	624.131	-282.0144	564.0288	NA	NA	NA
glmm_int_slp	13	1336.2135	1381.243	-655.1068	1310.2135	0	2	1

1.1 Over-dispersed Poisson model

Now, were going to look into the possibility of having *over-dispersion* with our Poisson model. Recall that the Poisson distribution assumes that the mean and variance are the same or that the standard deviation is the square root of the mean. This may not be true and the data may have variance > mean. When this is

the case we fit an **over-dispersed Poisson model**.

We can fit an over-dispersed Poisson model using the negative binomial distribution. The negative binomial distribution is similar to the Poisson model, but if the mean is equal to μ the variance is equal to $\mu + \theta\mu^2$ where $\theta > 0$. As a result, the variance > mean for this model.

The formula and link will be the same. We'll fit the model using the `glmer.nb` function which is a GLMM especially for the negative binomial distribution.

```
fm <- seizure.rate ~ base_sc + age + trt + period + trt*period + (1|subject)
system.time(glmm_int_nb <- glmer.nb(formula = fm, data = epilepsy, nAGQ = 20,
                         offset = per))
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.0400499 (tol = 0.002, component 1)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.0125332 (tol = 0.002, component 1)
```

```
## boundary (singular) fit: see ?isSingular
```

```
##    user  system elapsed
##  22.716   0.037  22.815
```

```
anova( glmm_int, glmm_int_nb)
```

	npar	AIC	BIC	logLik	deviance	Chisq	Df	Pr(>Chisq)
glmm_int	11	586.0288	624.1310	-282.01442	564.0288	NA	NA	NA
glmm_int_nb	12	134.3283	175.8943	-55.16415	110.3283	453.7005	1	0

```
## The neg.binomial theta parameter:
getME( glmm_int_nb, "glmer.nb.theta")
```

```
## [1] 0.6724278
```

```
summary( glmm_int_nb)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 20) [glmerMod]
##  Family: Negative Binomial(0.6724)  ( log )
## Formula: seizure.rate ~ base_sc + age + trt + period + trt * period +
##     (1 | subject)
##    Data: epilepsy
##  Offset: per
##
##      AIC      BIC   logLik deviance df.resid
##    134.3    175.9    -55.2    110.3      224
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.8021 -0.3847 -0.0871  0.1951  3.1779
##
## Random effects:
##  Groups  Name        Variance  Std.Dev.
##  subject (Intercept) 3.257e-11 5.707e-06
## Number of obs: 236, groups:  subject, 59
##
```

```
## Fixed effects:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -0.653581   0.484444  -1.349    0.177
## base_sc                 1.036746   0.108766   9.532   <2e-16 ***
## age                     0.009354   0.014407   0.649    0.516
## trtProgabide           -0.278260   0.173103  -1.607    0.108
## period.L               -0.138560   0.246956  -0.561    0.575
## period.Q                0.043469   0.246708   0.176    0.860
## period.C                0.035850   0.247092   0.145    0.885
## trtProgabide:period.L   0.016211   0.341212   0.048    0.962
## trtProgabide:period.Q  -0.223354   0.340498  -0.656    0.512
## trtProgabide:period.C   0.050294   0.339539   0.148    0.882
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) bas_sc age    trtPrg perd.L perd.Q perd.C trP:.L trP:.Q
## base_sc     -0.442
## age         -0.908  0.124
## trtProgabid -0.233 -0.133  0.111
## period.L    -0.013 -0.031  0.028  0.005
## period.Q     0.006  0.017 -0.013 -0.005  0.008
## period.C    -0.034  0.064  0.017 -0.013 -0.001 -0.003
## trtPrgbd:.L -0.013  0.032  0.001  0.002 -0.723 -0.006  0.002
## trtPrgbd:.Q  0.026 -0.034 -0.017  0.008 -0.006 -0.725  0.000  0.007
## trtPrgbd:.C  0.005 -0.012 -0.003  0.006  0.000  0.002 -0.725  0.003  0.001
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular
```

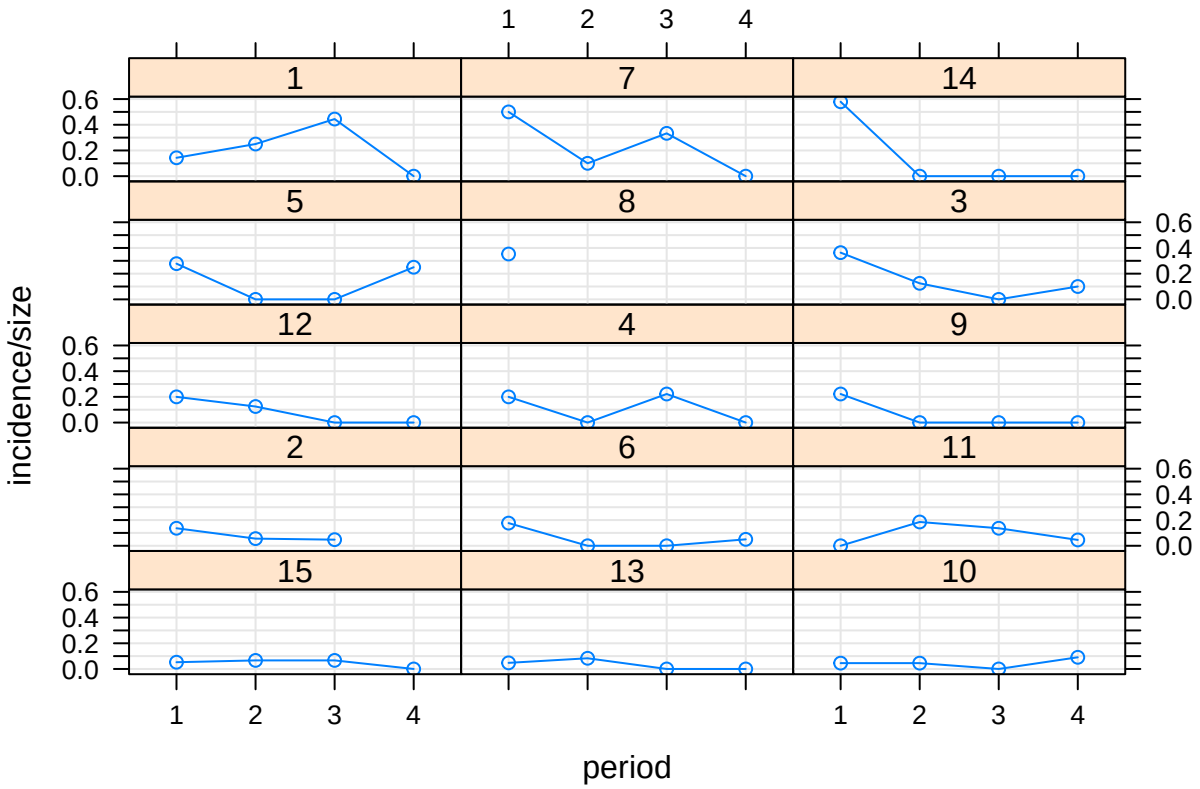
Note that this suggest we do not need a subject level random effect.

You could also fit an over-dispersed Poisson model by including a observation level random effect, but for reasons that are beyond our scope this is not as good of an option. We'll explore this idea below.

2 Contagious bovine pleuropneumonia

Contagious bovine pleuropneumonia (CBPP) is a major disease of cattle in Africa, caused by a mycoplasma. This dataset describes the serological incidence of CBPP in zebu cattle during a follow-up survey implemented in 15 commercial herds located in the Boji district of Ethiopia. The goal of the survey was to study the within-herd spread of CBPP in newly infected herds. Blood samples were quarterly collected from all animals of these herds to determine their CBPP status. These data were used to compute the serological incidence of CBPP (new cases occurring during a given time period). Some data are missing (lost to follow-up).

```
library(lattice)
xyplot(incidence/size ~ period|herd, cbpp, type=c('g','p','l'),
       layout=c(3,5), index.cond = function(x,y)max(y))
```




```
(gm1 <- glmer(cbind(incidence, size - incidence) ~ period + (1 | herd),
              data = cbpp, family = binomial))
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: cbind(incidence, size - incidence) ~ period + (1 | herd)
##    Data: cbpp
##       AIC      BIC   logLik deviance df.resid
## 194.0531 204.1799 -92.0266 184.0531       51
## Random effects:
##  Groups Name        Std.Dev.
##  herd   (Intercept) 0.6421
## Number of obs: 56, groups:  herd, 15
## Fixed Effects:
## (Intercept)      period2      period3      period4
##     -1.3983      -0.9919      -1.1282      -1.5797
```

Using nAGQ=0 only gets close to the optimum

```
(gm1a <- glmer(cbind(incidence, size - incidence) ~ period + (1 | herd),
               cbpp, binomial, nAGQ = 0))
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 0) [glmerMod]
##  Family: binomial  ( logit )
## Formula: cbind(incidence, size - incidence) ~ period + (1 | herd)
##    Data: cbpp
##       AIC      BIC   logLik deviance df.resid
## 194.1087 204.2355 -92.0543 184.1087       51
```

```
## Random effects:
##  Groups Name        Std.Dev.
##  herd   (Intercept) 0.6418
## Number of obs: 56, groups:  herd, 15
## Fixed Effects:
## (Intercept)      period2      period3      period4
##     -1.3605      -0.9762      -1.1111      -1.5597
```

Using `nAGQ = 20` provides a better evaluation of the deviance. Currently the internal calculations use the sum of deviance residuals, which is not directly comparable with the nAGQ=0 or nAGQ=1 result.

```
form_int <- cbind( incidence, size - incidence) ~ period +  (1 | herd)
(gm1a <- glmer( form_int, cbpp, binomial, nAGQ = 20))
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 20) [glmerMod]
##  Family: binomial  ( logit )
## Formula: cbind(incidence, size - incidence) ~ period + (1 | herd)
##    Data: cbpp
##       AIC      BIC   logLik deviance df.resid
## 110.0100 120.1368 -50.0050 100.0100       51
## Random effects:
##  Groups Name        Std.Dev.
##  herd   (Intercept) 0.6475
## Number of obs: 56, groups:  herd, 15
## Fixed Effects:
## (Intercept)      period2      period3      period4
##     -1.3992      -0.9914      -1.1278      -1.5795
```

GLMM with individual-level variability. For this data set the model is the same as one allowing for a period:herd interaction, which the plot indicates could be needed.

```
cbpp$obs <- 1:nrow(cbpp)
form_OD <- cbind( incidence, size - incidence) ~ period +
    (1 | herd) +  (1|obs)
gm2 <- glmer(form_OD, family = binomial, data = cbpp)
anova(gm1,gm2)
```

	npar	AIC	BIC	logLik	deviance	Chisq	Df	Pr(>Chisq)
gm1	5	194.0531	204.1799	-92.02657	184.0531	NA	NA	NA
gm2	6	186.6383	198.7904	-87.31916	174.6383	9.414806	1	0.0021524

```
summary(gm2)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: cbind(incidence, size - incidence) ~ period + (1 | herd) + (1 |
##     obs)
##    Data: cbpp
##
##       AIC      BIC   logLik deviance df.resid
##    186.6    198.8    -87.3    174.6       50
##
## Scaled residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -1.2866 -0.5989 -0.1181  0.3575  1.6216
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  obs    (Intercept) 0.79400  0.8911
##  herd   (Intercept) 0.03384  0.1840
## Number of obs: 56, groups:  obs, 56; herd, 15
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.5003     0.2967  -5.056 4.27e-07 ***
## period2      -1.2265     0.4803  -2.554  0.01066 *
## period3      -1.3288     0.4939  -2.690  0.00713 **
## period4      -1.8662     0.5936  -3.144  0.00167 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##         (Intr) perid2 perid3
## period2 -0.559
## period3 -0.537  0.373
## period4 -0.441  0.327  0.314
```

3 GEE versus GLMM

The data are from a randomized, double-blind, parallel-group, multicenter study comparing two oral treatments (denoted A and B) for toe-nail infection. Patients were evaluated for the degree of onycholysis (the degree of separation of the nail plate from the nail-bed) at baseline (week 0) and at weeks 4, 8, 12, 24, 36, and 48 thereafter. The onycholysis outcome variable is binary (none or mild versus moderate or severe). The binary outcome was evaluated on 294 patients comprising a total of 1908 measurements.

```
library(tidyverse)
toenail <- read.csv("toenail.csv")
head(toenail)
```

ID	Response	Treatment	Month	Visit
1	1	1	0.0000000	1
1	1	1	0.8571429	2
1	1	1	3.5357143	3
1	0	1	4.5357143	4
1	0	1	7.5357143	5
1	0	1	10.0357143	6

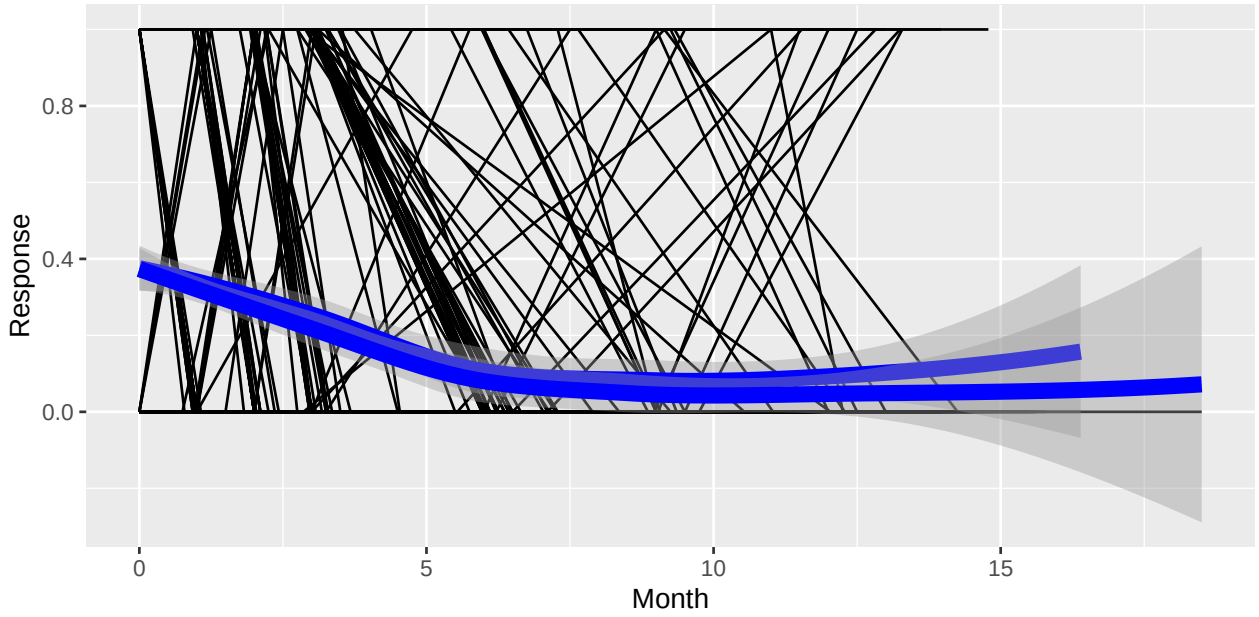
```
str(toenail)
```

```
## 'data.frame':    1908 obs. of  5 variables:
##  $ ID       : int  1 1 1 1 1 1 1 2 2 2 ...
##  $ Response : int  1 1 1 0 0 0 0 0 0 1 ...
##  $ Treatment: int  1 1 1 1 1 1 1 0 0 0 ...
##  $ Month    : num  0 0.857 3.536 4.536 7.536 ...
##  $ Visit    : int  1 2 3 4 5 6 7 1 2 3 ...
```

```
table( toenail$Treatment )
```

0	1
937	971

```
p <- ggplot(toenail, aes(x = Month, y = Response, group = ID) )
p + geom_line() + geom_smooth(aes(group = Treatment), method = "loess",
                 color = "blue", size = 3)
```




```
library(lme4)
library(lmerTest)
```

```
fm <- Response ~ Treatment*(Month + log(Month+1)) + (1|ID)

system.time(glmm_int_50 <- glmer(formula = fm, data = toenail, nAGQ = 50,
                                 family = "binomial"))
```

```
##    user  system elapsed
##   4.630   0.009   4.647
```

```
system.time(glmm_int_20 <- glmer(formula = fm, data = toenail, nAGQ = 20,
                                 family = "binomial"))
```

```
##    user  system elapsed
##   3.249   0.007   3.257
```

```
system.time(glmm_int_10 <- glmer(formula = fm, data = toenail, nAGQ = 10,
                                 family = "binomial"))
```

```
##    user  system elapsed
##   2.298   0.006   2.325
```

```
system.time(glmm_int_5  <- glmer(formula = fm, data = toenail, nAGQ = 5,
                                 family = "binomial"))
```

```
##    user  system elapsed
##   2.302   0.004   2.313
```

```
system.time(glmm_int_1  <- glmer(formula = fm, data = toenail, nAGQ = 1,
                                 family = "binomial"))
```

```
##    user  system elapsed
##   2.609   0.003   2.618
```

```
system.time(glmm_int_0  <- glmer(formula = fm, data = toenail, nAGQ = 0,
                                 family = "binomial"))
```

```
##    user  system elapsed
##    0.09    0.00    0.09
```

```
rbind( logLik(glmm_int_50), logLik(glmm_int_20), logLik(glmm_int_10),
       logLik(glmm_int_5) , logLik(glmm_int_1) , logLik(glmm_int_0))
```

```
## Warning in kable_pipe(x = structure(c("-621.3481", "-621.3342", "-621.4979", :
## The table should have a header (column names)
```

-621.3481
-621.3342
-621.4979
-625.7075
-624.2390
-635.8513

```
VarCorr(glmm_int_50)
```

```
##  Groups Name        Std.Dev.
##  ID     (Intercept) 4.089
```

```
VarCorr(glmm_int_5)
```

```
##  Groups Name        Std.Dev.
##  ID     (Intercept) 3.7821
```

```
VarCorr(glmm_int_1)
```

```
##  Groups Name        Std.Dev.
##  ID     (Intercept) 4.6598
```

```
VarCorr(glmm_int_0)
```

```
##  Groups Name        Std.Dev.
##  ID     (Intercept) 3.6036
```

```
round( coef( summary( glmm_int_50)), 3)
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-1.296	0.472	-2.742	0.006
Treatment	-0.150	0.644	-0.232	0.816
Month	-0.209	0.100	-2.091	0.037
log(Month + 1)	-0.882	0.445	-1.983	0.047
Treatment:Month	-0.098	0.154	-0.633	0.527
Treatment:log(Month + 1)	-0.115	0.660	-0.174	0.862

```
round( coef( summary( glmm_int_1)), 3)
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-2.257	0.838	-2.693	0.007
Treatment	-0.320	0.750	-0.426	0.670
Month	-0.226	0.101	-2.246	0.025
log(Month + 1)	-0.834	0.443	-1.885	0.059
Treatment:Month	-0.106	0.155	-0.687	0.492
Treatment:log(Month + 1)	-0.078	0.655	-0.119	0.905

```
round( coef( summary( glmm_int_0)), 3)
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-0.529	0.390	-1.357	0.175
Treatment	-0.046	0.553	-0.084	0.933
Month	-0.182	0.092	-1.981	0.048
log(Month + 1)	-0.763	0.410	-1.859	0.063
Treatment:Month	-0.077	0.142	-0.541	0.589
Treatment:log(Month + 1)	-0.109	0.609	-0.178	0.858

Now let's fit the same model with a gee

```
library(geepack)
fm_gee <- Response ~ Treatment*(Month + log(Month+1))

gee_int <- geeglm(formula = fm_gee, data = toenail, family = binomial,
                  id = ID, corstr = "exchangeable")
round( coef(summary(gee_int)) , 3)
```

	Estimate	Std.err	Wald	Pr(>|W|)
(Intercept)	-0.435	0.178	5.985	0.014
Treatment	-0.049	0.248	0.038	0.845
Month	-0.108	0.054	4.044	0.044
log(Month + 1)	-0.290	0.189	2.347	0.126
Treatment:Month	-0.096	0.103	0.872	0.350
Treatment:log(Month + 1)	0.108	0.311	0.120	0.729

```
round( coef( summary( glmm_int_50)), 3)
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-1.296	0.472	-2.742	0.006
Treatment	-0.150	0.644	-0.232	0.816
Month	-0.209	0.100	-2.091	0.037
log(Month + 1)	-0.882	0.445	-1.983	0.047
Treatment:Month	-0.098	0.154	-0.633	0.527
Treatment:log(Month + 1)	-0.115	0.660	-0.174	0.862

The parameter vector in the GEE model needs to be interpreted completely different from the parameter vector in the GLMM:

- GEE: marginal interpretation
- GLMM: conditional interpretation, conditionally upon level of random effects

In general, the model for the marginal average is not of the same parametric form as the conditional average in the GLMM.

For logistic mixed models, with normally distributed random intercepts, it can be shown that the marginal model can be well approximated by again a logistic model, but with parameters approximately satisfying

$$\frac{\beta_{GLMM}}{\beta_{GEE}} = \sqrt{c^2\sigma^2 + 1} > 1$$

where σ is the standard deviation of the random intercepts and $c = 0.5881$. In the toenail example we had random intercept with standard deviation:

```
VarCorr(glmm_int_50)
```

```
##  Groups Name        Std.Dev.
##  ID     (Intercept) 4.089
```

```
( (0.5881^2) * (4.089^2) + 1)^(1/2)
```

```
## [1] 2.604377
```

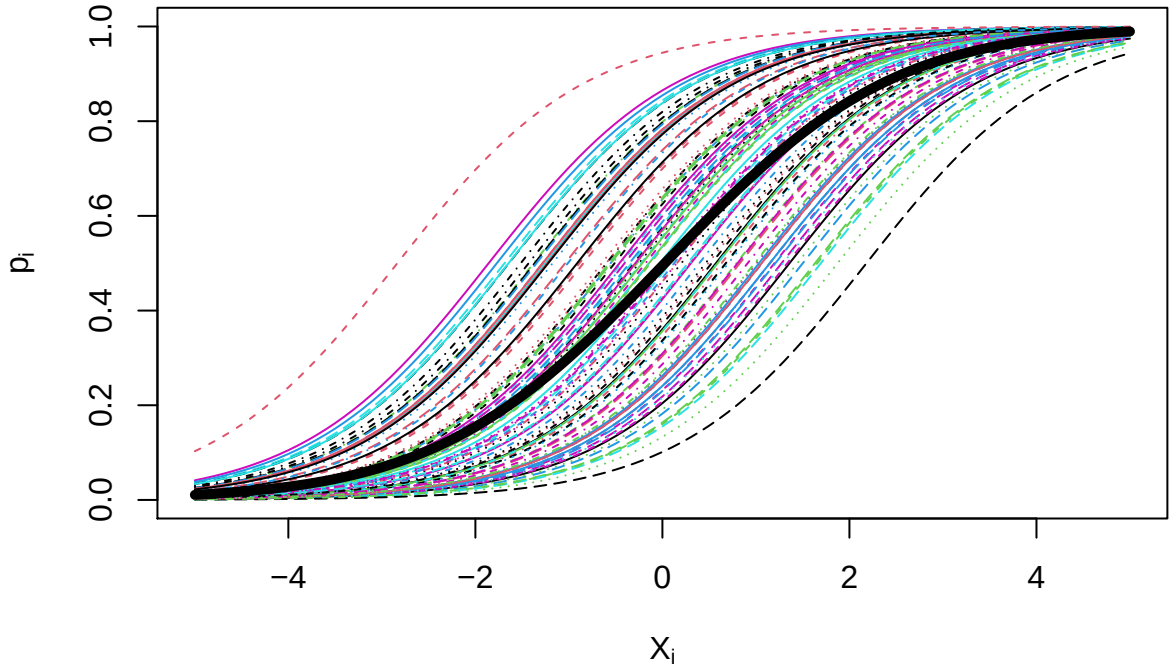
the actual ratios were

```
res <- cbind(fixef(glmm_int_50), fixef(glmm_int_50)/2.604, gee_int$coefficients)
colnames(res) <- c("GLMM", "GLMM/2.604", "GEE")
res
```

	GLMM	GLMM/2.604	GEE
(Intercept)	-1.2955792	-0.4975343	-0.4349642
Treatment	-0.1496465	-0.0574679	-0.0485553
Month	-0.2085953	-0.0801057	-0.1079477
log(Month + 1)	-0.8821684	-0.3387744	-0.2896097
Treatment:Month	-0.0976014	-0.0374813	-0.0958057
Treatment:log(Month + 1)	-0.1145764	-0.0440002	0.1078359

```
rand_ef <- rnorm(100, 0 , 1)
Xi <- seq(-5, 5, 0.1)
p_mat <- NULL
for(i in 1:100 ){
  pi = exp( Xi + rand_ef[i])/( 1 + exp( Xi + rand_ef[i]) )
  p_mat <- rbind( p_mat, pi)
}
p_avg <- apply( p_mat, 2, mean)

matplot(Xi,t(p_mat), type = "l", xlab = expression(X[i]),
        ylab = expression(p[i]))
lines(Xi,p_avg,lwd=5)
```


0.0
0.2
0.4
0.6
0.8
1.0
p_i
−4
−2
0
2
4
X_i