# Generalized Linear Mixed Models (GLMM)

Alexander McLain

## Contents

## 1 Packages to fit GLMM in R

Any of the following packages can be used to fit GLMM in R:

- `glmmAK` – Generalized Linear Mixed Models
- `MASS` – Main Package of Venables and Ripley's MASS (see function glmmPQL)
- `MCMCglmm` – MCMC Generalised Linear Mixed Models
- `lme4 (glmer)`
- `glmmML`
- `gamlss.mx`
- `sabreR`

Any of the following packages can be used to fit Hierarchical GLMs:

- `hglm` – hglm is used to fit hierarchical generalized linear models
- `HGLMMM` – Hierarchical Generalized Linear Models

Any of the following packages can be used to fit additive and generalized-additive mixed models:

- `amer` – Additive mixed models with lme4
- `gamm4` – Generalized additive mixed models using mgcv and lme4
- `mgcv` (gamm function, via glmmPQL in MASS package)
- `gamlss.mx`

## 2 Using glmer

To fit the GLMM we're going to use `glmer` in the `lme4` package. We previously used the `lme4` package to fit linear mixed models with `lmer`.

The arguements to `glmer` are:

- `formula`: a two-sided linear formula. Random-effects terms are distinguished by vertical bars ("|") separating expressions for design matrices from grouping factors.
- `data`: an optional data frame.
- `family`: a GLM family, see glm and family.
- `nAGQ`: integer scalar - the number of points per axis for evaluating the adaptive Gauss-Hermite approximation to the log-likelihood. Defaults to 1, corresponding to the Laplace approximation. Values greater than 1 produce greater accuracy in the evaluation of the log-likelihood at the expense of speed. A value of zero uses a faster but less exact form of parameter estimation for GLMMs by optimizing the random effects and the fixed-effects coefficients in the penalized iteratively reweighted least squares step. (See Details.)

In our models below, the computation time is pretty minimal. As a result, we're going to set `nAGQ=20` which is a relatively large value.

# 3 Muscatine Coronary Risk Factor Study

- The Muscatine Coronary Risk Factor Study was a study designed to look at the change in obesity in children.
- Five age cohorts were measured in 1977, 1979 and 1981.
- The study had had five cohorts of children, initially aged 5–7, 7–9, 9–11, 11–13, and 13–15.
- Children were classified as obese or not obese.

```
library(tidyverse)
MCRS_wide <- read_csv("MCRS.csv")
head(MCRS_wide)
```

```
## # A tibble: 6 x 9
##   Gender  OB77  OB79  OB81   CT1   CT2   CT3   CT4   CT5
##    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1      0     1     1     1     8    20    25    16    15
## 2      0     1     1     0     1     7     9    11     6
## 3      0     1     0     1     1     9     7     4     0
## 4      0     1     0     0     0     8     7    13     8
## 5      0     0     1     1     7     8    10     3     7
## 6      0     0     1     0     3     8     8     8     2
```

```
MCRS_kinda_long <- pivot_longer(MCRS_wide, cols = starts_with("CT"), names_to = "Cohort",
                          names_prefix = "CT", values_to = "Count",
                          values_drop_na = TRUE)
head(MCRS_kinda_long,10)
```

```
## # A tibble: 10 x 6
##    Gender  OB77  OB79  OB81 Cohort Count
##     <dbl> <dbl> <dbl> <dbl> <chr>  <dbl>
##  1      0     1     1     1 1          8
##  2      0     1     1     1 2         20
##  3      0     1     1     1 3         25
##  4      0     1     1     1 4         16
##  5      0     1     1     1 5         15
##  6      0     1     1     0 1          1
##  7      0     1     1     0 2          7
##  8      0     1     1     0 3          9
##  9      0     1     1     0 4         11
## 10      0     1     1     0 5          6
```

```
MCRS_longer <- pivot_longer(MCRS_kinda_long, cols = starts_with("OB"), names_to = "Year",
                            names_prefix = "OB", values_to = "OB_ind",
                            values_drop_na = TRUE)
head(MCRS_longer, 10)
```

```
## # A tibble: 10 x 5
##    Gender Cohort Count Year  OB_ind
##     <dbl> <chr>  <dbl> <chr>  <dbl>
##  1      0 1          8 77         1
##  2      0 1          8 79         1
##  3      0 1          8 81         1
##  4      0 2         20 77         1
##  5      0 2         20 79         1
##  6      0 2         20 81         1
##  7      0 3         25 77         1
##  8      0 3         25 79         1
##  9      0 3         25 81         1
## 10      0 4         16 77         1
```

```
tail(MCRS_longer, 5)
```

```
## # A tibble: 5 x 5
##   Gender Cohort Count Year  OB_ind
##    <dbl> <chr>  <dbl> <chr>  <dbl>
## 1      1 4        129 79         0
## 2      1 4        129 81         0
## 3      1 5         91 77         0
## 4      1 5         91 79         0
## 5      1 5         91 81         0
```

We will not “de-aggregate” or “reverse-summarize” the data using the `uncount` function in `tidyverse`.

```
MCRS_longest <- MCRS_longer %>% uncount(weights = Count, .id = "ID")
head(MCRS_longest, 10)
```

```
## # A tibble: 10 x 5
##    Gender Cohort Year  OB_ind    ID
##     <dbl> <chr>  <chr>  <dbl> <int>
##  1      0 1      77         1     1
##  2      0 1      77         1     2
##  3      0 1      77         1     3
##  4      0 1      77         1     4
##  5      0 1      77         1     5
##  6      0 1      77         1     6
##  7      0 1      77         1     7
##  8      0 1      77         1     8
##  9      0 1      79         1     1
## 10      0 1      79         1     2
```

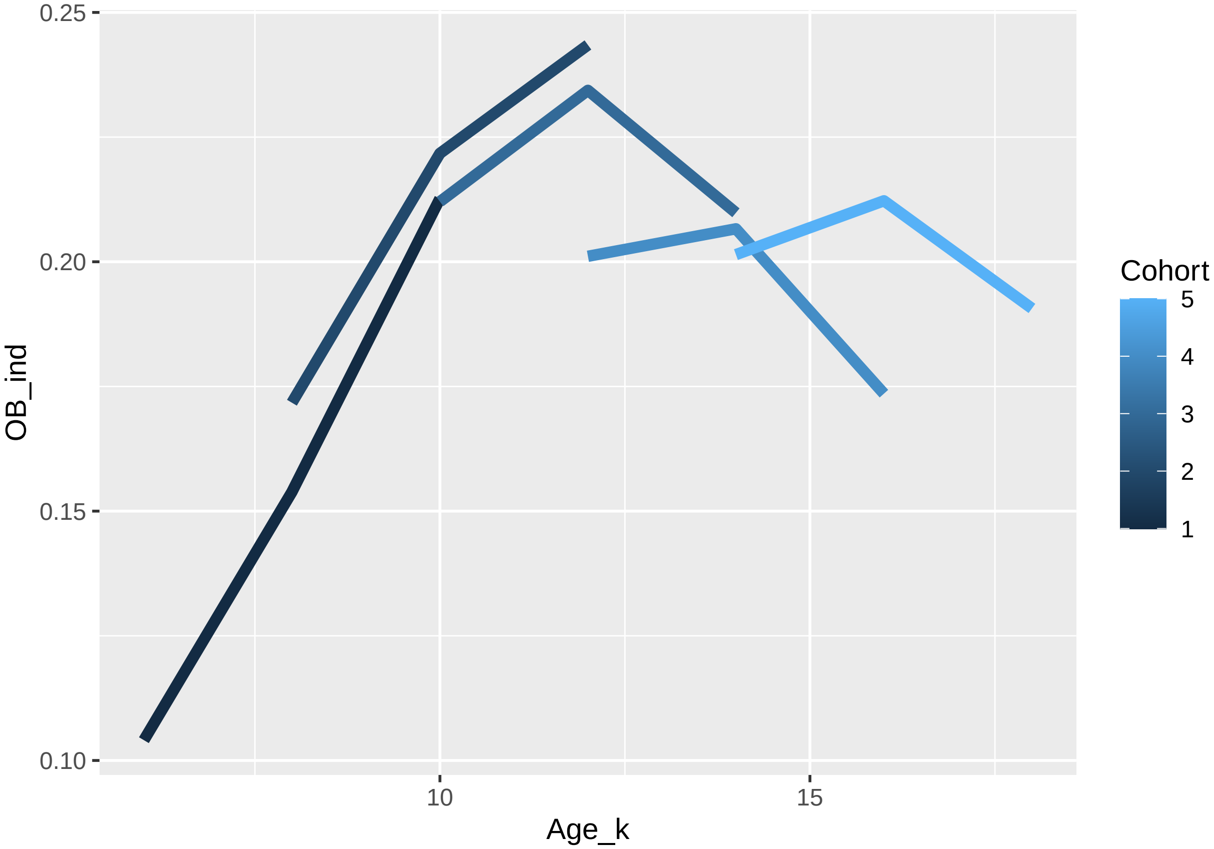
```
tail(MCRS_longest, 10)
```

```
## # A tibble: 10 x 5
##    Gender Cohort Year  OB_ind    ID
##     <dbl> <chr>  <chr>  <dbl> <int>
##  1      1 5      81         0    82
##  2      1 5      81         0    83
## 3        1 5       81          0     84
## 4        1 5       81          0     85
## 5        1 5       81          0     86
## 6        1 5       81          0     87
## 7        1 5       81          0     88
## 8        1 5       81          0     89
## 9        1 5       81          0     90
## 10       1 5       81          0     91
```

Now we can (finally) analyze the data:

```
MCRS_longest <- MCRS_longest %>% mutate(Cohort = as.numeric(Cohort), Year = as.numeric(Year)) %>%
  mutate( Age_k = Cohort*2 + 4 + (Year-77))

p <- ggplot(MCRS_longest, aes(x = Age_k, y = OB_ind) )
p + stat_summary(aes(group = Cohort, col = Cohort), geom = "line",
                 fun = mean, size = 2)
```



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

```
form <- OB_ind ~ log(Age_k) + factor(Cohort) + log(Age_k)*factor(Cohort) + (1|ID)
system.time(log_mod <- glmer(formula = form, data = MCRS_longest, nAGQ = 10, family = binomial))
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 1.83754 (tol = 0.002, component 1)
```

```
##    user  system elapsed
##  22.218   0.048  22.285
```

```r
anova(log_mod)
```

| | npar | Sum Sq | Mean Sq | F value |
|---|---|---|---|---|
| log(Age_k) | 1 | 4.475415 | 4.475415 | 4.475415 |
| factor(Cohort) | 4 | 36.596776 | 9.149194 | 9.149194 |
| log(Age_k):factor(Cohort) | 4 | 11.803761 | 2.950940 | 2.950940 |

```r
form <- OB_ind ~ log(Age_k) + factor(Cohort) + log(Age_k)*factor(Cohort) + (1 + log(Age_k)|ID)
system.time(log_mod_rnd_slp <- glmer(formula = form, data = MCRS_longest, nAGQ = 1, family = binomial))
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## unable to evaluate scaled gradient
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge: degenerate Hessian with 1 negative eigenvalues
```

```
##    user  system elapsed
##  52.468   0.079  52.633
```

# 4 Heart rates of patients on different drug treatments

The HR data frame has 120 rows and 5 columns of the heart rates of patients under one of three possible drug treatments. The variables are - Patient: an ordered factor indicating the patient. - Drug: the drug treatment - a factor with levels a, b and p where p represents the placebo. - baseHR: the patient’s base heart rate - HR: the observed heart rate at different times in the experiment - Time: the time of the observation

Littel, R. C., Milliken, G. A., Stroup, W. W., and Wolfinger, R. D. (1996), SAS System for Mixed Models, SAS Institute (Data Set 3.5).

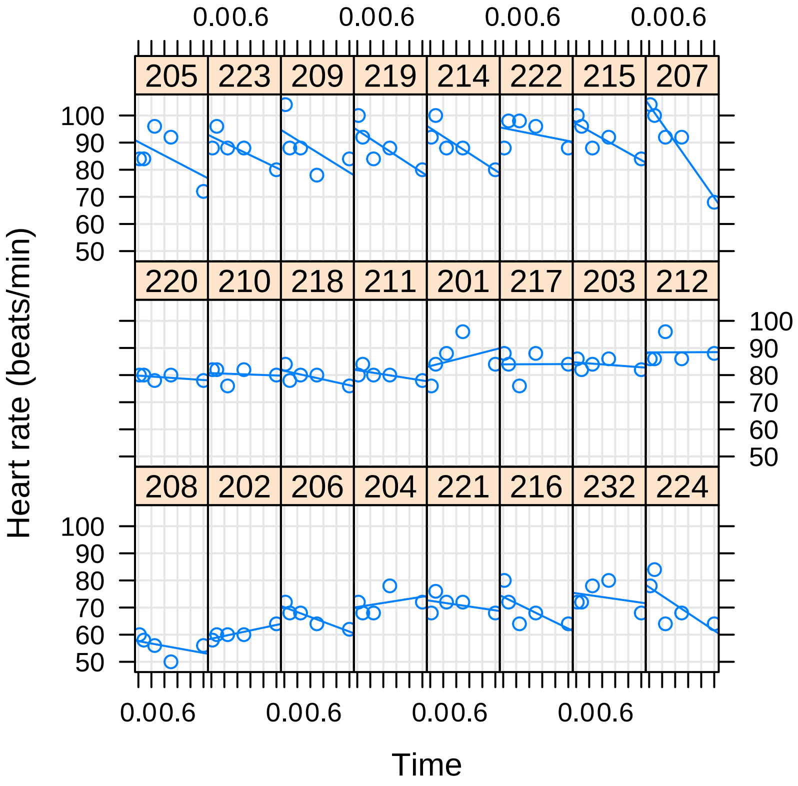
```r
library(SASmixed)
library(lattice)
head(HR)
```

```
##   Patient Drug baseHR HR     Time
## 1     201    p     92 76 0.016667
## 2     201    p     92 84 0.083333
## 3     201    p     92 88 0.250000
## 4     201    p     92 96 0.500000
## 5     201    p     92 84 1.000000
## 6     202    b     54 58 0.016667
```

```r
str(HR)
```

```
## 'data.frame':    120 obs. of  5 variables:
##  $ Patient: Factor w/ 24 levels "201","202","203",..: 1 1 1 1 1 2 2 2 2 2 ...
##  $ Drug   : Factor w/ 3 levels "a","b","p": 3 3 3 3 3 2 2 2 2 2 ...
##  $ baseHR : num  92 92 92 92 92 54 54 54 54 54 ...
##  $ HR     : num  76 84 88 96 84 58 60 60 60 64 ...
##  $ Time   : num  0.0167 0.0833 0.25 0.5 1 ...
##  - attr(*, "ginfo")=List of 7
##   ..$ formula     :Class 'formula'  language HR ~ Time | Patient
##   .. .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
##   ..$ order.groups: logi TRUE
##   ..$ FUN         :function (x)
##   ..$ outer       :Class 'formula'  language ~Drug
##   .. .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
##   ..$ inner       : NULL
##   ..$ labels      :List of 2
##   .. ..$ Time: chr "Time"
##   .. ..$ HR  : chr "Heart Rate"
##   ..$ units       :List of 2
##   .. ..$ Time: chr "(min)"
##   .. ..$ HR  : chr "(beats/min)"
```

```
xyplot(HR ~ Time | Patient, HR, type = c("g", "p", "r"), aspect = "xy",
         index.cond = function(x, y) coef(lm(y ~ x))[1],
         ylab = "Heart rate (beats/min)")
```



```
HR <- HR %>% mutate(HR_incr = 1*I(HR > baseHR))

form <- HR_incr ~ Time + Drug + Time*Drug + (1 |Patient)

system.time( fm1HR <- glmer( form, data = HR, family = binomial, nAGQ = 20) )
```

```
##    user  system elapsed
##   0.383   0.007   0.390
```

```
coef(summary(fm1HR))
```

```
##                   Estimate Std. Error     z value  Pr(>|z|)
## (Intercept) -0.51436799   1.095437 -0.46955501 0.6386730
## Time        -1.95860020   1.495493 -1.30966867 0.1903079
## Drugb        0.88079645   1.581052  0.55709513 0.5774624
## Drugp       -1.83304494   1.648564 -1.11190424 0.2661793
## Time:Drugb  -0.23413302   2.062494 -0.11351936 0.9096188
## Time:Drugp  -0.09539642   2.450641 -0.03892714 0.9689485
```

```
form <- HR_incr ~ Time + Drug + Time*Drug + (1 + Time|Patient)

system.time( fm2HR <- glmer( form, data = HR, family = binomial, nAGQ = 1) )
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.0140551 (tol = 0.002, component 1)
```

```
##    user  system elapsed
##   0.518   0.002   0.520
```

```
fm2HR
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: HR_incr ~ Time + Drug + Time * Drug + (1 + Time | Patient)
##    Data: HR
##      AIC      BIC   logLik deviance df.resid
## 127.0650 152.1524 -54.5325 109.0650      111
## Random effects:
##  Groups  Name        Std.Dev. Corr
##  Patient (Intercept) 2.4424
##          Time        0.4982   -1.00
## Number of obs: 120, groups:  Patient, 24
## Fixed Effects:
## (Intercept)         Time        Drugb        Drugp   Time:Drugb   Time:Drugp
##     -0.5616      -1.7169       0.8915      -1.9529      -0.2438       0.1059
## optimizer (Nelder_Mead) convergence code: 0 (OK) ; 0 optimizer warnings; 1 lme4 warnings
```

Let's compare this with a linear model of the continuous outcome.

```
form <- HR ~ Time + Drug + Time*Drug + (1 + Time|Patient)
system.time( fm3HR <- lmer( form, data = HR) )
```

```
##    user  system elapsed
##   0.048   0.000   0.048
```

```
coef(summary(fm3HR))
```

```
##               Estimate Std. Error       df    t value     Pr(>|t|)
## (Intercept)  81.458391   4.264921 21.00010 19.0996232 9.402450e-15
## Time        -10.698354   3.084920 21.00017 -3.4679522 2.299618e-03
## Drugb         4.850600   6.031510 21.00010  0.8042099 4.302862e-01
## Drugp        -1.425485   6.031510 21.00010 -0.2363396 8.154591e-01
## Time:Drugb    3.511892   4.362735 21.00017  0.8049747 4.298544e-01
## Time:Drugp    7.501310   4.362735 21.00017  1.7194051 1.002473e-01
```

# 5 National Institute of Mental Health shizophrenia study

Schizophrenia data from a randomized controlled trial with patients assigned to either drug or placebo group. "Severity of Illness" was measured, at weeks $0, 1, \ldots 6$, on a four category ordered scale: 1. normal or borderline mentally ill, 2. mildly or moderately ill, 3. markedly ill, and 4. severely or among the most extremely ill. Most of the observations where made on weeks 0,1,3, and 6.

```
library(mixcat)
```

```
## Loading required package: statmod
```

```
data(schizo)
head(schizo)
```

```
##     id y trt wk
## 1 1103 4   1  0
## 2 1103 2   1  1
## 3 1103 2   1  3
## 4 1103 2   1  6
## 5 1104 4   1  0
## 6 1104 2   1  1
```

To fit a GLMM on the housing data we're going to use the `mixcat` package. This function works a little differently than the previous mixed-model functions we've used. Here, the random effects are specified in a separate arguement than the fixed effects.

We'll be using the `npmlt` function. In this function, the response variable $y$ can be binary or multinomial. A binary response should take values 1 and 2, and the function npmlt will model the probability of 1. For an ordinal response, taking values $1, ..., q$, a cumulative logit model can be fit.

```
attach(schizo)
form <- y ~ trt + sqrt(wk) + trt*sqrt(wk)
rand_form <- ~1+trt
cond_log_0 <- npmlt( formula = form, random= rand_form, id = id, k = 2, EB = FALSE)
summary(cond_log_0)
```

```
##
## Call:  npmlt(formula = form, random = rand_form, id = id, k = 2, EB = FALSE)
##
## Coefficients:
##                   Estimate   Std. Error
##   (Intercept) 1    -5.0600      0.2920
##   (Intercept) 2    -2.4162      0.2531
##   (Intercept) 3    -0.5076      0.2425
##   trt              -0.1342      0.2564
##   sqrt(wk)          0.7350      0.1257
##   trt:sqrt(wk)      1.0046      0.1387
##
## Estimated Multivariate NP Dist.:
##
## Mass Points:
##                 (Intercept)      trt  Std. Error (Intercept)  Std. Error trt
##   mass point 1      -2.1008   0.3087                  0.2088          0.1803
##   mass point 2       1.3366  -0.1964                  0.1425          0.1132
##
## Masses:
##            Estimate  Std. Error
##   mass 1   0.38883     0.03077
##   mass 2   0.61117     0.03077
##
## Random effects mean: all dimensions constrained to 0
##
## Random effects covariance (lower) and correlation (upper) matrices:
##
##                (Intercept)       trt
##   (Intercept)      2.80780  -1.00000
##   trt             -0.41257   0.06062
##
##                                              (Intercept)      trt
##   Std. Errors of random effects variances:      0.45028  0.06992
##
##  -2(Log-Likelihood) : 3450
## Number of iterations : 84
```

A nice part about this model is that it's easier to specify which variables have a different effect for each cutpoint. For example,

```
form <- y ~ trt*sqrt(wk)
form_vary <- ~ trt + sqrt(wk)
rand_form <- ~ 1 + trt
cond_log_1 <- npmlt( formula = form ,formula.npo = form_vary, random= rand_form, id = id, k = 2, EB = F
summary(cond_log_1)
```

```
##
## Call:  npmlt(formula = form, formula.npo = form_vary, random = rand_form,      id = id, k = 2, EB = 
##
## Coefficients:
##                  Estimate   Std. Error
##   (Intercept) 1   -5.44418     0.54355
##   (Intercept) 2   -2.55649     0.28926
##   (Intercept) 3   -0.41104     0.25434
##   trt 1            0.12718     0.48796
##   trt 2            0.04345     0.31451
##   trt 3           -0.18535     0.27103
##   sqrt(wk) 1       0.88299     0.21323
##   sqrt(wk) 2       0.78709     0.14428
##   sqrt(wk) 3       0.72731     0.13756
##   trt:sqrt(wk)     0.93402     0.16424
##
## Estimated Multivariate NP Dist.:
##
## Mass Points:
##                 (Intercept)      trt  Std. Error (Intercept)  Std. Error trt
##   mass point 1      -2.1039   0.4108                  0.2104          0.2043
##   mass point 2       1.4214  -0.2775                  0.1614          0.1403
##
## Masses:
##           Estimate  Std. Error
##   mass 1   0.40320     0.03191
##   mass 2   0.59680     0.03191
##
## Random effects mean: all dimensions constrained to 0
##
## Random effects covariance (lower) and correlation (upper) matrices:
##
##                (Intercept)      trt
##   (Intercept)       2.9906  -1.0000
##   trt              -0.5839   0.1140
##
##                                              (Intercept)     trt
##   Std. Errors of random effects variances:       0.5021  0.1133
```

```
##
##  -2(Log-Likelihood) : 3448
## Number of iterations : 103
```

# 6 Housing data

```
library(tidyverse)
library(readxl)
house <- read_excel("Housing.xlsx", na = ".")
## First checked to see if "." was read as NA
str(house)
```

```
## tibble [1,448 x 4] (S3: tbl_df/tbl/data.frame)
##  $ ID     : num [1:1448] 1 1 1 1 2 2 2 2 3 3 ...
##  $ Housing: num [1:1448] 1 2 2 2 1 2 2 1 0 2 ...
##  $ Time   : num [1:1448] 0 6 12 24 0 6 12 24 0 6 ...
##  $ Sec    : num [1:1448] 1 1 1 1 1 1 1 1 1 1 ...
```

```
head(house)
```

```
## # A tibble: 6 x 4
##      ID Housing  Time   Sec
##   <dbl>   <dbl> <dbl> <dbl>
## 1     1       1     0     1
## 2     1       2     6     1
## 3     1       2    12     1
## 4     1       2    24     1
## 5     2       1     0     1
## 6     2       2     6     1
```

For npmlt to work, the outcome needs to be $1, 2, \ldots$. So we'll change that then analyze the data (with no varying effects)

```
house <- house %>% mutate( House = Housing + 1)
form <- House ~ factor(Sec) + factor(Time) + factor(Sec)*factor(Time)
rand_form <- ~ 1

attach(house)
house_glmm <- npmlt( formula = form, random= rand_form, id = ID, k = 3, EB = FALSE)
summary(house_glmm)
```

```
##
## Call:  npmlt(formula = form, random = rand_form, id = ID, k = 3, EB = FALSE)
##
## Coefficients:
##                                  Estimate   Std. Error
##   (Intercept) 1                 -9.959e-01   1.437e-01
##   (Intercept) 2                  1.035e+00   1.407e-01
##   factor(Sec)1                  -1.078e+00   1.481e-01
##   factor(Time)6                 -2.671e-01   1.416e-01
##   factor(Time)12                 5.695e-01   1.316e-01
##   factor(Time)24                -2.591e-02   1.351e-01
##   factor(Sec)1:factor(Time)6     2.327e-01   2.048e-01
##   factor(Sec)1:factor(Time)12   -2.180e-02   1.866e-01
##   factor(Sec)1:factor(Time)24   -8.784e-02   3.581e-15
```

```
##
## Estimated NP Dist.:
##                 Estimate  Std. Error
##   mass point 1  -1.54857     0.99958
##   mass point 2  -0.13298     0.19702
##   mass point 3   2.98487     0.43084
##   mass 1         0.10753     0.16515
##   mass 2         0.80099     0.15873
##   mass 3         0.09147     0.02611
##
## Random effects mean: constrained to 0
##
## Random effects variance:   Estimate  Std. Error
##                              1.0870      0.2433
##
##  -2(Log-Likelihood) : 2598
## Number of iterations : 500
##
## Warning: maximum mumber of iterations was reached
## Warning: eigenvalues of information matrix less than argument 'tol' appear at last iteration
```

```
cbind(house_glmm$coefficients/house_glmm$SE.coefficients)
```

```
##                                      [,1]
## (Intercept) 1               -6.930861e+00
## (Intercept) 2                7.357989e+00
## factor(Sec)1                -7.277448e+00
## factor(Time)6               -1.886408e+00
## factor(Time)12               4.326188e+00
## factor(Time)24              -1.917617e-01
## factor(Sec)1:factor(Time)6   1.136281e+00
## factor(Sec)1:factor(Time)12 -1.168229e-01
## factor(Sec)1:factor(Time)24 -2.452960e+13
```