STAT 528 HW 6

2023-05-02

##**Problem 1**: Do the following:

a) Explain why a GLMM with a Bernoulli response may have biased estimates of regression coefficients.

b) We have the same reasons as with LMMs to view χ^2 testing with some skepticism. Elaborate on these reasons.

c) Construct a bootstrap procedure that estimates the p-value corresponding to the deviance based nested model hypothesis test for the `ctsid` data analysis in the GLMM notes.

Solution:

a)

The idea of PQL is to transform the original response to produce a pseudo response. The transformed psuedo response will approximately behave like normally distributed random variables, for which optimization algorithms for linear mixed effects models can be used. The transformation works well when the original response is already close to being normally distributed, like poisson with high counts and binomial with large number of trials. But in the other case, when the original response is highly discrete, hence far from being normally distributed, like Bernoulli response, the approximation is poor and will result bias.

b)

For testing fixed effects, the LRT is only approximately χ^2 distributed under null hypothesis and tend to give smaller p-values, overestimating significance of variables. For testing random effects, the problem is more serious then the null hypothesis is of the form $H_0 : \hat{\sigma_0} = 0$. Because in this case, the parameter is on the boundary of parameter space, making the χ^2 approximation completely fail. Even when the null hypothesis is of other forms, LRT tends to give larger p-values, underestimating significance of variables.

c)

```
library(faraway)
library(tidyverse)
library(lme4)
library(ggplot2)
data(ctsib)
ctsib$stable <- ifelse(ctsib$CTSIB==1,1,0)
ctsib <- ctsib %>%
mutate(Age = scale(Age), Height = scale(Height), Weight = scale(Weight))
```

```
system.time(modgh <- glmer(stable ~ Sex + Age + Height + Weight + Surface +
Vision + (1|Subject), nAGQ=25, family=binomial, data=ctsib,
control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5))))
```

```
##    user  system elapsed
##   0.472   0.008   0.481
```

```
summary(modgh)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 25) [glmerMod]
##  Family: binomial  ( logit )
## Formula: stable ~ Sex + Age + Height + Weight + Surface + Vision + (1 |
##     Subject)
##    Data: ctsib
## Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05))
##
##      AIC      BIC   logLik deviance df.resid
##    247.9    285.5   -114.9    229.9      471
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.0175 -0.1334 -0.0185 -0.0006  4.9238
##
## Random effects:
##  Groups  Name        Variance Std.Dev.
##  Subject (Intercept) 7.657    2.767
## Number of obs: 480, groups:  Subject, 40
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -11.94595    1.97470  -6.049 1.45e-09 ***
## Sexmale       3.18721    1.74637   1.825   0.0680 .
## Age          -0.02587    0.49899  -0.052   0.9587
## Height       -1.96489    0.95011  -2.068   0.0386 *
## Weight        1.08598    0.93138   1.166   0.2436
## Surfacenorm   7.40243    1.08471   6.824 8.83e-12 ***
## Visiondome    0.68230    0.53048   1.286   0.1984
## Visionopen    6.19147    1.00072   6.187 6.13e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Sexmal Age    Height Weight Srfcnr Visndm
## Sexmale     -0.657
## Age         -0.038  0.097
## Height       0.400 -0.426  0.043
## Weight       0.050 -0.332 -0.167 -0.564
## Surfacenorm -0.833  0.263 -0.005 -0.263  0.125
## Visiondome  -0.242  0.035  0.001 -0.041  0.023  0.114
## Visionopen  -0.821  0.271  0.000 -0.262  0.116  0.836  0.376
```

```
modgh2 <- glmer(stable ~ Surface + Vision + (1|Subject), nAGQ=25,
family=binomial, data=ctsib)
anova(modgh, modgh2)
```

```
## Data: ctsib
## Models:
## modgh2: stable ~ Surface + Vision + (1 | Subject)
## modgh: stable ~ Sex + Age + Height + Weight + Surface + Vision + (1 | Subject)
##        npar    AIC    BIC  logLik deviance Chisq Df Pr(>Chisq)
```

```
## modgh2    5 247.30 268.17 -118.65   237.30
## modgh     9 247.89 285.46 -114.95   229.89 7.407  4     0.1159
```

```
library(doParallel)
set.seed(42)
B <- 1e3

myCluster <- makeCluster(detectCores()-2, type='PSOCK')
registerDoParallel(myCluster)

system.time(lrtstat <- foreach(1:B) %dopar% {
  y <- unlist(simulate(modgh2))
  bnull <- lme4::glmer(y ~ Surface + Vision + (1|Subject), nAGQ=25,
              family=binomial, data=ctsib)
  balt <- lme4::glmer(y ~ Sex + Age + Height + Weight + Surface +
            Vision + (1|Subject), nAGQ=25, family=binomial, data=ctsib,
            control=lme4::glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5)))
  as.numeric(2*(logLik(balt) - logLik(bnull)))
})
```

```
##    user  system elapsed
##   0.250   0.055  89.632
```

```
pval <- mean(lrtstat > as.numeric(2*(logLik(modgh) - logLik(modgh2))))
pval
```

```
## [1] 0.165
```

Again, the simplified model is justified.

Problem 2: Show that a GLM in canonical form can be cast as a GEE. Explain your work in detail.

Solution:

GLM framework:

$$f(y;\theta,\phi) = \exp\left\{\frac{y\theta - c(\theta)}{a_i(\phi)} + b(y,\phi)\right\},$$

where θ is the location parameter, $a(\phi)$ is a scale parameter, $b(y,\phi)$ is the normalizing term, and ϕ is the dispersion parameter.

Thus, we have

$$E(y) = c'(\theta), V(y) = Var(y) = c''(\theta)a(\phi)$$

And the log-likelihood for the exponential family is,

$$L(\theta, \phi | y_1, ..., y_n) = \sum_{i=1}^{n} \left\{ \frac{y_i \theta_i - c(\theta_i)}{a_i(\phi)} + c(y_i, \phi) \right\}$$

then we can get

$$\frac{\partial L}{\partial \theta_i} = \left\{ \frac{y_i - c'(\theta_i)}{a_i(\phi)} \right\} = \left\{ \frac{y_i - \mu_i}{a_i(\phi)} \right\}$$

holds for $i = 1, .., n$.

Also, we obtain $\frac{\partial \theta_i}{\partial \mu_i} = \frac{a_i(\phi)}{V(\mu_i)}$ by the following,

$$\mu_i = c'(\theta_i) \iff \frac{\partial}{\partial \theta_i} \mu_i = \frac{\partial}{\partial \theta_i} c'(\theta_i) \iff \frac{\partial \mu_i}{\partial \theta_i} = c''(\theta_i) \iff \frac{\partial \theta_i}{\partial \mu_i} = \frac{a_i(\phi)}{V(\mu_i)},$$

Thus, if we set $\frac{\partial L}{\partial \beta} = 0$, we will obtain

$$\begin{aligned} 0 = \frac{\partial L}{\partial \beta} &= \sum_{i=1}^{n} \frac{\partial L}{\partial \theta_i} \frac{\partial \theta_i}{\partial \beta} \\ &= \sum_{i=1}^{n} \frac{\partial L}{\partial \theta_i} \frac{\partial \theta_i}{\partial \mu_i} \frac{\partial \mu_i}{\partial \beta} \\ &= \sum_{i=1}^{n} \frac{y_i - \mu_i}{a_i(\phi)} \frac{a_i(\phi)}{V(\mu_i)} \frac{\partial \mu_i}{\partial \beta} \\ &= \sum_{i=1}^{n} \frac{y_i - \mu_i}{V(\mu_i)} \frac{\partial \mu_i}{\partial \beta} \\ &= \sum_{i=1} D_i^T V_i^{-1} (Y_i - \mu_i) \end{aligned}$$

Which is equivalent to the GEE identity,

where $D_i = \frac{\partial \mu_i}{\partial \beta}$, V_i refers to the variance of Y_i. And this is just the same as the setting in GEE.

Problem 3: The National Youth Survey collected a sample of 11–17 year olds, 117 boys and 120 girls, asking questions about marijuana usage. The data is presented in the `potuse` dataset int he `faraway` package. Do the following:

a) Plot the total number of people falling into each usage category as it varies over time separately for each sex.

b) Condense the levels of the response into whether the person did or did not use marijuana that year. Turn the year into a numerical variable. Fit a GLMM for the now binary response with an interaction between sex and year as a predictor using Gauss-Hermite quadrature. Comment on the effect of sex.

c) Fit a reduced model without sex and use it to test for the significance of sex in the larger model.

d) Fit a model with year as a factor. Should this model be preferred to the model with year as just a linear term? Interpret the estimated effects in the year as a factor version of the model.

e) Fit your final model using PQL, Bayesian methods, and MCLA. Compare the results and discuss any inconsistencies if any arise.

f) Fit GEE version of the model and compare it to the analogous GLMM fits.

Solution:

a)

```
library(reshape2)
library(lme4)
library(MASS)
```

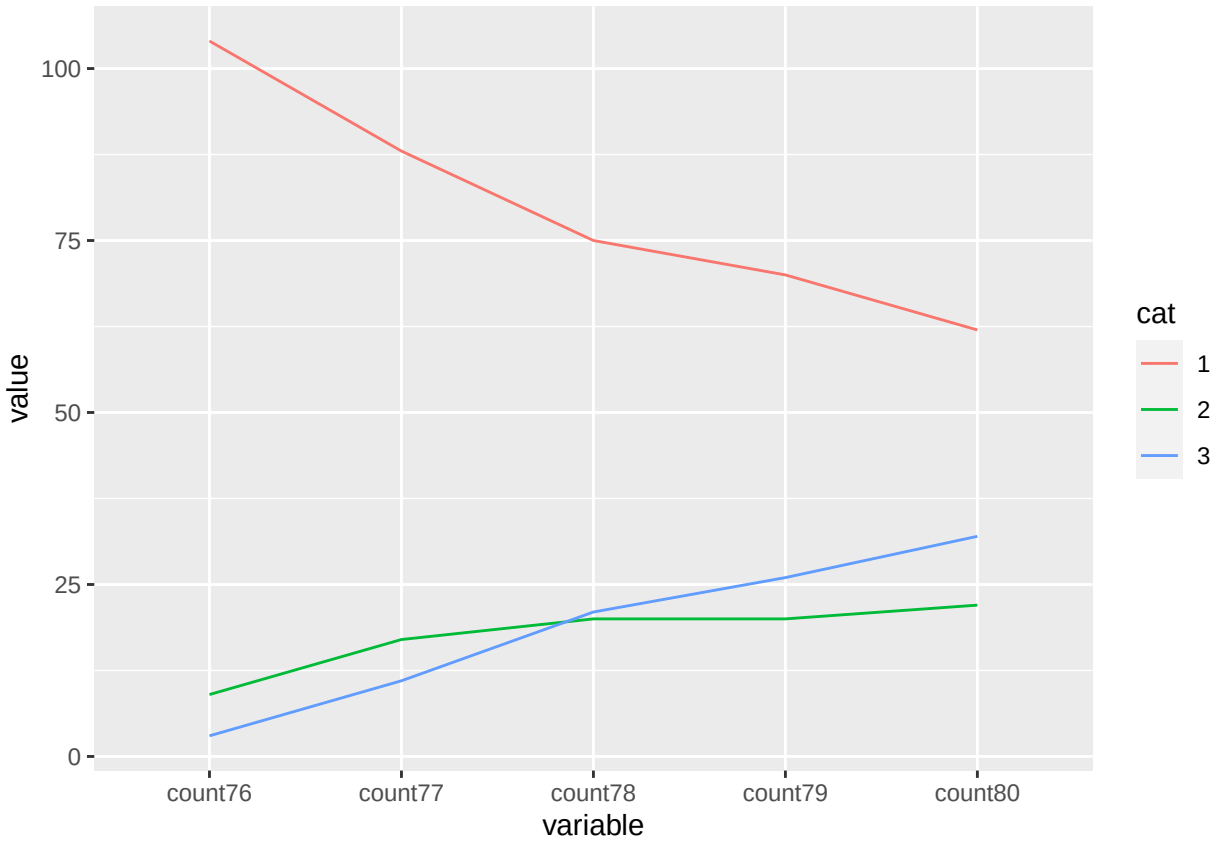
```
head(potuse, 5)
```

```
##   sex year.76 year.77 year.78 year.79 year.80 count
## 1   1       1       1       1       1       1    48
## 2   1       1       1       1       1       2     8
## 3   1       1       1       1       1       3     4
## 4   1       1       1       1       2       1     2
## 5   1       1       1       1       2       2     4
```

```
y76_m <- potuse %>% filter(sex == 1) %>% group_by(year.76) %>%
  summarise(count76 = sum(count)) %>% rename(cat=year.76)
y77_m <- potuse %>% filter(sex == 1)%>% group_by(year.77) %>%
  summarise(count77 = sum(count))%>% mutate(cat=year.77)
y78_m <- potuse %>% filter(sex == 1)%>% group_by(year.78) %>%
  summarise(count78 = sum(count))%>% mutate(cat=year.78)
y79_m <- potuse %>% filter(sex == 1)%>% group_by(year.79) %>%
  summarise(count79 = sum(count))%>% mutate(cat=year.79)
y80_m <- potuse %>% filter(sex == 1)%>% group_by(year.80) %>%
  summarise(count80 = sum(count))%>% mutate(cat=year.80)
count_male <- y76_m %>% left_join(y77_m) %>% left_join(y78_m) %>%
  left_join(y79_m) %>% left_join(y80_m) %>%
  dplyr::select(cat, count76, count77, count78, count79, count80)
```

Plot for male.

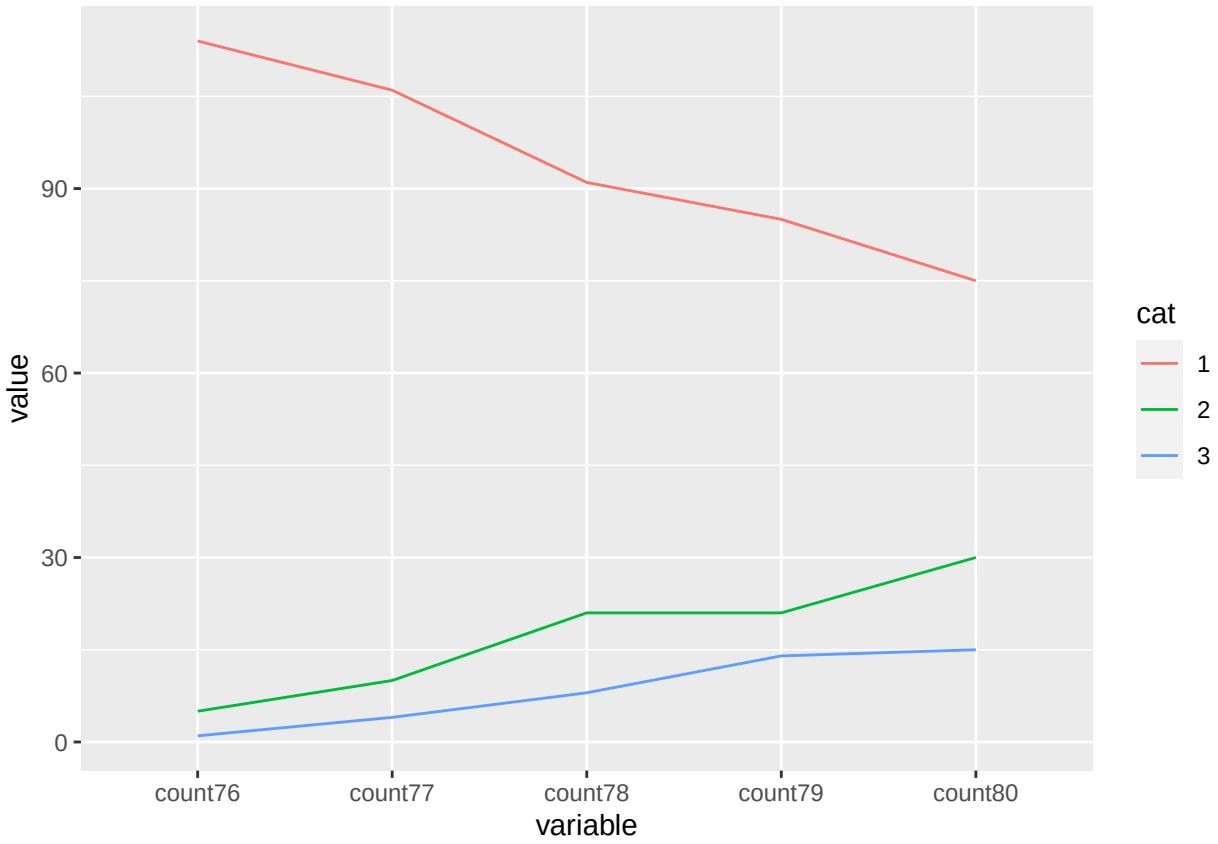
```
df_male <- melt(data.frame(count_male))[-c(1,2,3),]
df_male$cat <- 1:3
df_male$cat <- as.factor(df_male$cat)
ggplot(df_male, aes(variable, value, group=cat)) + geom_line(aes(color=cat))
```




```
y76_f <- potuse %>% filter(sex == 2) %>% group_by(year.76) %>% summarise(count76 = sum(count)) %>% rena
y77_f <- potuse %>% filter(sex == 2)%>% group_by(year.77) %>% summarise(count77 = sum(count))%>% mutate
y78_f <- potuse %>% filter(sex == 2)%>% group_by(year.78) %>% summarise(count78 = sum(count))%>% mutate
y79_f <- potuse %>% filter(sex == 2)%>% group_by(year.79) %>% summarise(count79 = sum(count))%>% mutate
y80_f <- potuse %>% filter(sex == 2)%>% group_by(year.80) %>% summarise(count80 = sum(count))%>% mutate
count_female <- y76_f %>% left_join(y77_f) %>% left_join(y78_f) %>% left_join(y79_f) %>% left_join(y80_
  dplyr::select(cat, count76, count77, count78, count79, count80)
```

Plot for female.

```
df_female <- melt(data.frame(count_female))[-c(1,2,3),]
df_female$cat <- 1:3
df_female$cat <- as.factor(df_female$cat)
ggplot(df_female, aes(variable, value, group=cat)) + geom_line(aes(color=cat))
```



b)

Condense responses, and transform data from wide format to long format, scale predictors.

```
potuse_long <- potuse %>% filter(count > 0) %>% mutate(resp76 = ifelse(year.76 == 1, 0, 1),
                                 resp77 = ifelse(year.77 == 1, 0, 1),
                                 resp78 = ifelse(year.78 == 1, 0, 1),
                                 resp79 = ifelse(year.79 == 1, 0, 1),
                                 resp80 = ifelse(year.80 == 1, 0, 1)) %>% uncount(count) %>%
                          dplyr::select(sex, resp76, resp77,resp78, resp79, resp80)
potuse_long$subject <- 1:236
potuse_long <- melt(potuse_long, id.vars=c('subject', 'sex'))
potuse_long$year <- as.numeric(gsub('resp','', potuse_long$variable))

potuse_long$sex <- as.factor(potuse_long$sex)
potuse_long$year_num <- (potuse_long$year - mean(potuse_long$year)) / sd(potuse_long$year)
potuse_long$year <- as.factor(potuse_long$year)
head(potuse_long)
```

```
##   subject sex variable value year  year_num
## 1       1   1   resp76     0   76 -1.413614
## 2       2   1   resp76     0   76 -1.413614
## 3       3   1   resp76     0   76 -1.413614
## 4       4   1   resp76     0   76 -1.413614
## 5       5   1   resp76     0   76 -1.413614
## 6       6   1   resp76     0   76 -1.413614
```

```
system.time(mod_large <- glmer(value ~ sex + year_num + sex:year_num + (1|subject),
                               nAGQ=25, family=binomial, data=potuse_long,
                               control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5))))
```

```
##    user  system elapsed
##   0.228   0.001   0.230
```

```
summary(mod_large)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 25) [glmerMod]
##  Family: binomial  ( logit )
## Formula: value ~ sex + year_num + sex:year_num + (1 | subject)
##    Data: potuse_long
## Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05))
##
##      AIC      BIC   logLik deviance df.resid
##   1004.4   1029.7   -497.2    994.4     1175
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.5035 -0.3333 -0.1202  0.1739  5.8374
##
## Random effects:
##  Groups  Name        Variance Std.Dev.
##  subject (Intercept) 8.181    2.86
## Number of obs: 1180, groups:  subject, 236
##
## Fixed effects:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.8018     0.3338  -5.398 6.73e-08 ***
## sex2           -1.1273     0.4609  -2.446   0.0145 *
## year_num        1.2231     0.1594   7.671 1.70e-14 ***
## sex2:year_num   0.2198     0.2296   0.957   0.3385
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) sex2   yer_nm
## sex2        -0.593
## year_num    -0.229  0.032
## sex2:yer_nm  0.061 -0.199 -0.595
```

sex2 has negetive estimated coefficients, meaning that being female has negative effect on the probability of using pot. So the model says that females are less likely to use pot, compared with males.

c)

```
system.time(mod_mid <- glmer(value ~ year_num + sex:year_num + (1|subject),
                               nAGQ=25, family=binomial, data=potuse_long,
                               control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5))))
```

```
##    user  system elapsed
##   0.190   0.001   0.192
```

```
system.time(mod_small <- glmer(value ~ year_num  + (1|subject),
                               nAGQ=25, family=binomial, data=potuse_long,
                               control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5))))
```

```
##    user  system elapsed
##   0.127   0.001   0.128
```

```
anova(mod_large, mod_mid, mod_small)
```

```
## Data: potuse_long
## Models:
## mod_small: value ~ year_num + (1 | subject)
## mod_mid: value ~ year_num + sex:year_num + (1 | subject)
## mod_large: value ~ sex + year_num + sex:year_num + (1 | subject)
##           npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## mod_small    3 1006.7 1021.9 -500.35  1000.71
## mod_mid      4 1008.4 1028.7 -500.22  1000.44 0.2659  1    0.60610
## mod_large    5 1004.4 1029.7 -497.18   994.35 6.0917  1    0.01358 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Turns out the effect of sex is significant at significance level of 0.05.

d)

```
system.time(mod_fac <- glmer(value ~ sex + year + sex:year + (1|subject),
                               nAGQ=25, family=binomial, data=potuse_long,
                               control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5))))
```

```
##    user  system elapsed
##   0.943   0.007   0.949
```

```
summary(mod_fac)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 25) [glmerMod]
##  Family: binomial  ( logit )
## Formula: value ~ sex + year + sex:year + (1 | subject)
##    Data: potuse_long
## Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05))
##
##      AIC      BIC   logLik deviance df.resid
##   1003.2   1059.1   -490.6    981.2     1169
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.1175 -0.3016 -0.1564  0.1508  8.5825
##
## Random effects:
##  Groups  Name        Variance Std.Dev.
##  subject (Intercept) 8.511    2.917
## Number of obs: 1180, groups:  subject, 236
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.2898     0.5632  -7.616 2.61e-14 ***
```

```
## sex2         -1.3285     0.8270  -1.606 0.108182
## year77        1.8687     0.5240   3.566 0.000362 ***
## year78        2.9702     0.5345   5.557 2.75e-08 ***
## year79        3.3590     0.5419   6.198 5.70e-10 ***
## year80        3.9635     0.5569   7.118 1.10e-12 ***
## sex2:year77  -0.3330     0.8387  -0.397 0.691339
## sex2:year78   0.2357     0.8259   0.285 0.775326
## sex2:year79   0.3695     0.8284   0.446 0.655590
## sex2:year80   0.5552     0.8357   0.664 0.506457
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) sex2   year77 year78 year79 year80 sx2:77 sx2:78 sx2:79
## sex2        -0.539
## year77      -0.660  0.393
## year78      -0.715  0.399  0.683
## year79      -0.730  0.398  0.683  0.730
## year80      -0.747  0.393  0.677  0.733  0.746
## sex2:year77  0.381 -0.683 -0.612 -0.407 -0.405 -0.397
## sex2:year78  0.396 -0.745 -0.415 -0.606 -0.426 -0.420  0.695
## sex2:year79  0.399 -0.757 -0.415 -0.429 -0.599 -0.424  0.696  0.757
## sex2:year80  0.400 -0.771 -0.412 -0.428 -0.429 -0.587  0.693  0.761  0.773
```

AIC prefers the model with year as factor, but BIC disagrees. In my humble opinion, treating year as a categorical variable should be preferred. When year is included as numeric variable, since its value monotonically increases, the effect of year in the model is also monotone, and varying at a constant rate. To be specific, assume estimated coefficient for year is $\hat{\beta} > 0$, then the effect of `year=77` is greater than `yead=76` by $\hat{\beta}$, the effect of `year=78` is greater than `yead=77` by $\hat{\beta}$, and so on(this not exactly the case in the above models since year is scaled, but close enough), which not necessarily makes sense. But when year is included as categorical variable, the model can estimate different effect for each year, making the model more expressive.

Now when we look at the summary table of model with year as factor, we notice that with year76 as reference level, each of the following years has stronger positive effect on pot usage than the previous year. But we can also note that the increasing rate of effect is slowing down. The increase in estimated coefficient from year 78 to 79 is greater than that from 79 to 80.

e)

We choose the model with year as a categorical variable.

PQL

```
mod_fac_PQL <- glmmPQL(value ~ sex + year + sex:year, random = ~ 1|subject,
                                family=binomial, data=potuse_long)
```

```
## iteration 1
## iteration 2
## iteration 3
## iteration 4
## iteration 5
```

```
summary(mod_fac_PQL)
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: potuse_long
##   AIC BIC logLik
##    NA  NA     NA
##
## Random effects:
##  Formula: ~1 | subject
##         (Intercept)  Residual
## StdDev:    2.371001 0.7538634
##
## Variance function:
##  Structure: fixed weights
##  Formula: ~invwt
## Fixed effects:  value ~ sex + year + sex:year
##                 Value Std.Error  DF   t-value p-value
## (Intercept) -3.684649 0.3844182 936 -9.585002  0.0000
## sex2        -1.098355 0.6088835 234 -1.803883  0.0725
## year77       1.667289 0.3703422 936  4.502022  0.0000
## year78       2.679829 0.3727999 936  7.188385  0.0000
## year79       3.039851 0.3751940 936  8.102079  0.0000
## year80       3.595952 0.3798049 936  9.467894  0.0000
## sex2:year77 -0.288316 0.6013066 936 -0.479482  0.6317
## sex2:year78  0.231650 0.5951434 936  0.389235  0.6972
## sex2:year79  0.355766 0.5974479 936  0.595477  0.5517
## sex2:year80  0.523788 0.6026208 936  0.869184  0.3850
##  Correlation:
##             (Intr) sex2   year77 year78 year79 year80 sx2:77 sx2:78 sx2:79
## sex2        -0.631
## year77      -0.629  0.397
## year78      -0.666  0.420  0.665
## year79      -0.676  0.427  0.665  0.703
## year80      -0.689  0.435  0.661  0.706  0.717
## sex2:year77  0.388 -0.664 -0.616 -0.409 -0.409 -0.407
## sex2:year78  0.417 -0.724 -0.416 -0.626 -0.441 -0.442  0.689
## sex2:year79  0.424 -0.737 -0.418 -0.442 -0.628 -0.450  0.690  0.752
## sex2:year80  0.434 -0.754 -0.417 -0.445 -0.452 -0.630  0.687  0.756  0.770
##
## Standardized Within-Group Residuals:
##        Min         Q1        Med         Q3        Max
## -5.3047520 -0.4130275 -0.2258331  0.1950520  9.5835955
##
## Number of Observations: 1180
## Number of Groups: 236
```

INLA for Bayesian method.

```
library(INLA)
```

```
## Loading required package: sp
```

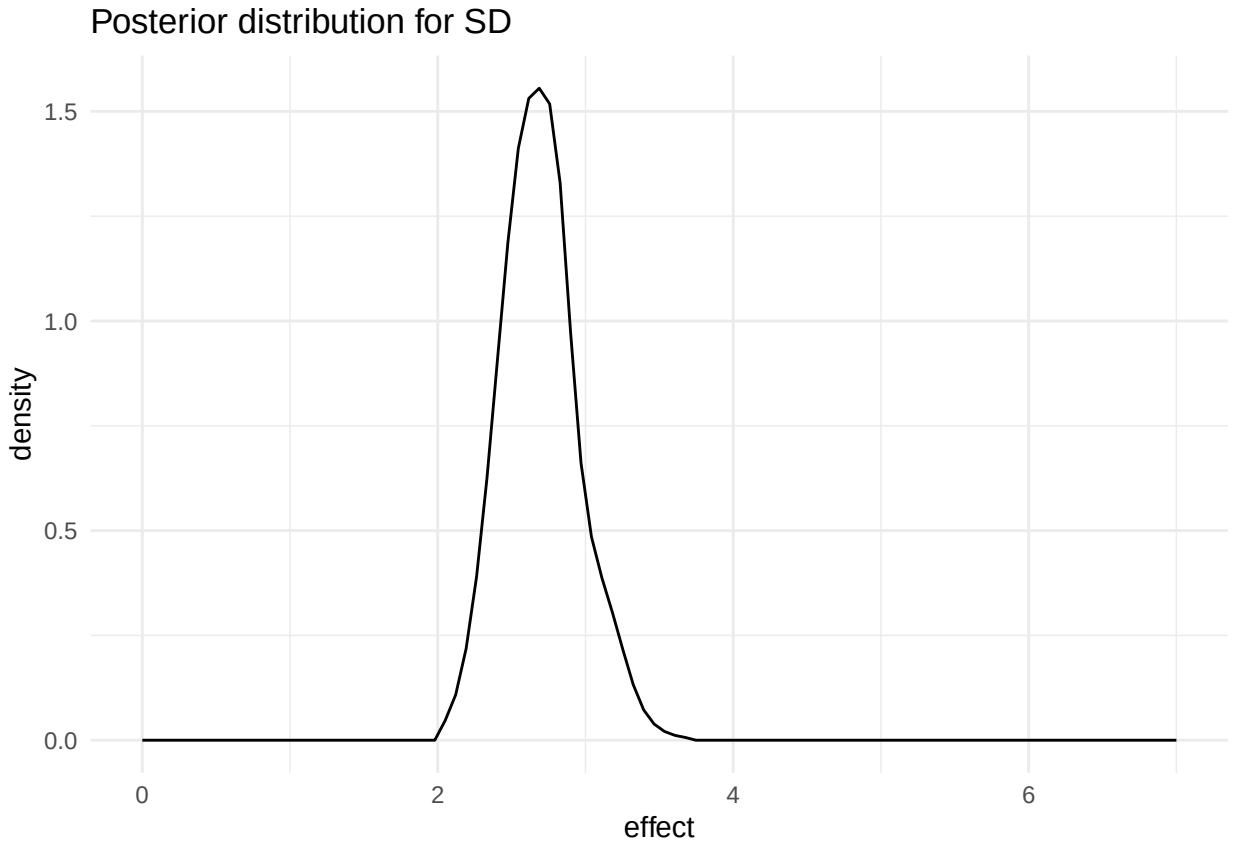
```
## This is INLA_22.12.16 built 2022-12-23 13:43:44 UTC.
##  - See www.r-inla.org/contact-us for how to get help.
##  - To enable PARDISO sparse library; see inla.pardiso()
```

```
# formula <- use ~  year + f(id, model = "iid")
# result <- inla(formula, family = "binomial", data = potuse3)
```

```
formula <- value ~ sex + year + sex:year + f(subject, model="iid")
result <- inla(formula, family="binomial", data=potuse_long)
```

```
sigmaalpha <- inla.tmarginal(function(x) 1/sqrt(x), result$marginals.hyperpar$"Precision for subject")
```

```
x <- seq(0,7,length.out = 100)
sdf <- data.frame(effect = x, density=inla.dmarginal(x, sigmaalpha))
ggplot(sdf,aes(x=effect,y=density)) +
  ggtitle("Posterior distribution for SD") +
  geom_line() +
  theme_minimal()
```




```
restab <- sapply(result$marginals.fixed,
  function(x) inla.zmarginal(x, silent=TRUE))
restab <- cbind(restab, alpha=inla.zmarginal(sigmaalpha,silent=TRUE))
#colnames(restab) = c("mu","norm","dome","open","alpha")
restab
```

```
##            (Intercept) sex2       year77    year78    year79    year80
## mean       -4.161088   -1.313602  1.814879  2.89476   3.282207  3.894626
## sd         0.5141166   0.7479847  0.4764481 0.4861902 0.4934075 0.5090432
## quant0.025 -5.20134    -2.791772  0.8845452 1.949536  2.324765  2.9104
## quant0.25  -4.501674   -1.816834  1.491695  2.564267  2.946505  3.547713
## quant0.5   -4.151662   -1.312021  1.812164  2.89051   3.277245  3.888226
## quant0.75  -3.811131   -0.8100491 2.134191  3.219688  3.611554  4.233591
## quant0.975 -3.183193   0.1433134  2.75301   3.856355  4.259957  4.907103
##            sex2:year77 sex2:year78 sex2:year79 sex2:year80 alpha
```

```
## mean       -0.3044377  0.2513454   0.3865613   0.5836255   2.693169
## sd         0.7652073   0.7506316   0.751718    0.7560237   0.26209
## quant0.025 -1.807341   -1.221722   -1.088097   -0.8982552  2.217509
## quant0.25  -0.8218916  -0.2563631  -0.1219449  0.07203385  2.511476
## quant0.5   -0.3057117  0.2496159   0.3846208   0.5812175   2.680079
## quant0.75  0.2101729   0.7557375   0.8915172   1.091144    2.850749
## quant0.975 1.193416    1.72217     1.860154    2.067028    3.262101
```

MCLA

```
library(glmm)
```

```
## Loading required package: trust
```

```
## Loading required package: mvtnorm
```

```
set.seed(13)
clust <- makeCluster(detectCores()-2)
# subject needs to be a factor
potuse_long$subjectF <- as.factor(potuse_long$subject)
system.time(m_MCLA <- glmm(value ~ sex + year + sex:year, random = list(~0+subjectF),
family.glmm=bernoulli.glmm, m = 1e3,
varcomps.names = c("subjectF"), cluster = clust,
data=potuse_long))
```

```
##    user  system elapsed
## 267.434   3.301 308.037
```

```
m_MCLA$beta
```

```
## (Intercept)        sex2      year77      year78      year79      year80
##  -3.5725090  -1.0851557   1.6354978   2.5919303   2.9254103   3.4393445
## sex2:year77 sex2:year78 sex2:year79 sex2:year80
##  -0.2781660   0.1768498   0.2714756   0.3961934
```

```
se(m_MCLA)
```

```
## (Intercept)        sex2      year77      year78      year79      year80
##   0.4606245   0.8331778   0.4783824   0.4791197   0.4818802   0.4870940
## sex2:year77 sex2:year78 sex2:year79 sex2:year80    subjectF
##   0.7891274   0.7812601   0.7836443   0.7879967   0.4718590
```

There are of course some differences on the exact values of estimates. A notable one is that estimated coeffcients of interaction terms given by MCLA is slightly off, compared with other approaches(probably because insufficient sample size. I believe it is safe to say that all above approaches give similar results.

f)

```
library(geepack)
modgee <- geeglm(value ~ sex + year + sex:year,
                 id=subject, corstr="exchangeable", scale.fix=TRUE,
                 data=potuse_long, family=binomial)
summary(modgee)
```

```
##
## Call:
## geeglm(formula = value ~ sex + year + sex:year, family = binomial,
##     data = potuse_long, id = subject, corstr = "exchangeable",
```

```
##     scale.fix = TRUE)
##
##  Coefficients:
##             Estimate  Std.err   Wald Pr(>|W|)
## (Intercept) -2.15948  0.30487 50.171 1.41e-12 ***
## sex2        -0.78495  0.51806  2.296  0.12973
## year77       1.01435  0.37420  7.348  0.00671 **
## year78       1.55557  0.36149 18.518 1.68e-05 ***
## year79       1.73963  0.35913 23.465 1.27e-06 ***
## year80       2.02133  0.35721 32.021 1.52e-08 ***
## sex2:year77 -0.09429  0.62955  0.022  0.88094
## sex2:year78  0.24531  0.59294  0.171  0.67909
## sex2:year79  0.31751  0.58715  0.292  0.58868
## sex2:year80  0.41228  0.58189  0.502  0.47862
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = exchangeable
## Scale is fixed.
##
##   Link = identity
##
## Estimated Correlation Parameters:
##       Estimate Std.err
## alpha        0       0
## Number of clusters:   1180  Maximum cluster size: 1
```

Note that most of the estimated coefficients of GEE is smaller in magnitude than that of GLMM by numerical integration. The reason why lies in the underlying modelling assumptions.

In GLMM, it is assumed that random effects are normally distributed, and the link function correctly captures the relation between conditional expectation and linear combination of predictors. So given the logit link for this GLMM, the estimated coefficient should be interpreted as the change in log odds ratio for unit change in the corresponding predictor, given the subject and other predictors fixed.

In GEE, there's no distributional assumptions. We only assume a link function and a variance-covariance structure. In other words, there are weaker assumptions in GEE. And given logit link for GEE, the estimated coefficient should be interpreted as the average change in log odds ratio for unit change in the corresponding predictor, across all subjects, given other predictors fixed. So, GEE is modelling population-average effect, rather than subject-specific effect that is modeled by GLMM. Hence it makes sense that GEE gives estimates with smaller magnitude.

Problem 4: The `nitrofen` data in `boot` package come from an experiment to measure the reproductive toxicity of the pesticide `nitrofen` on a species of zooplankton called *Ceriodaphnia dubia*. Each animal produced three broods in which the number of live offspring was recorded. Fifty animals in total were used and divided into five batches. Each batch was treated in a solution with a different concentration of the pesticide. Do the following:

a) Plot the total number of live offspring as they vary with concentration and comment. Now plot the numbers for each brood, taking care to distinguish the different broods. Is the trend within each brood the same?

b) Fit a GLMM for the number of live offspring within each brood that varies with concentration and brood number (including an interaction). The model should take account of the dependence between observations from the same animal. Describe what the model says about how the number of live offspring change with concentration for the different broods.

c) Fit an equivalent GEE model and compare it to the GLMM result.

Soln:

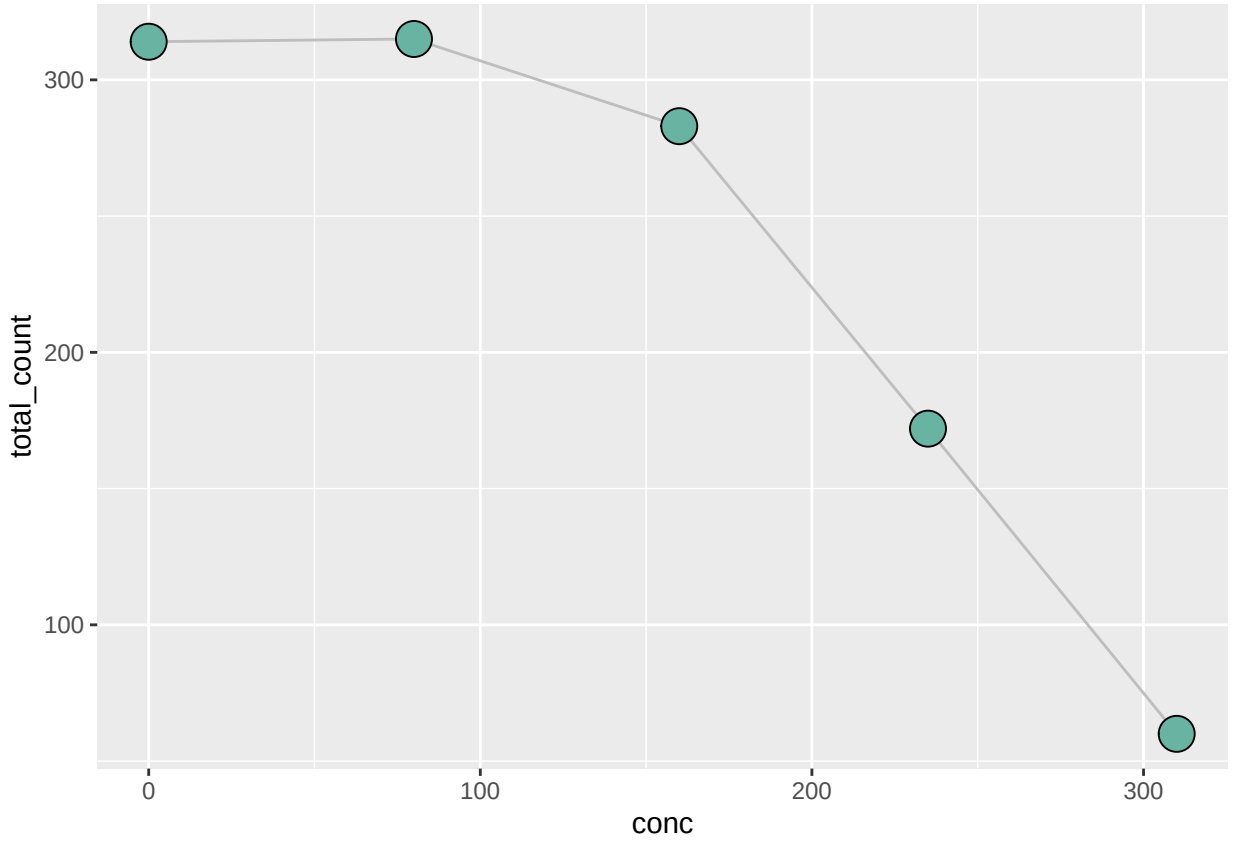
a)

```
library(boot)
library(hrbrthemes)
```

```
head(nitrofen)
```

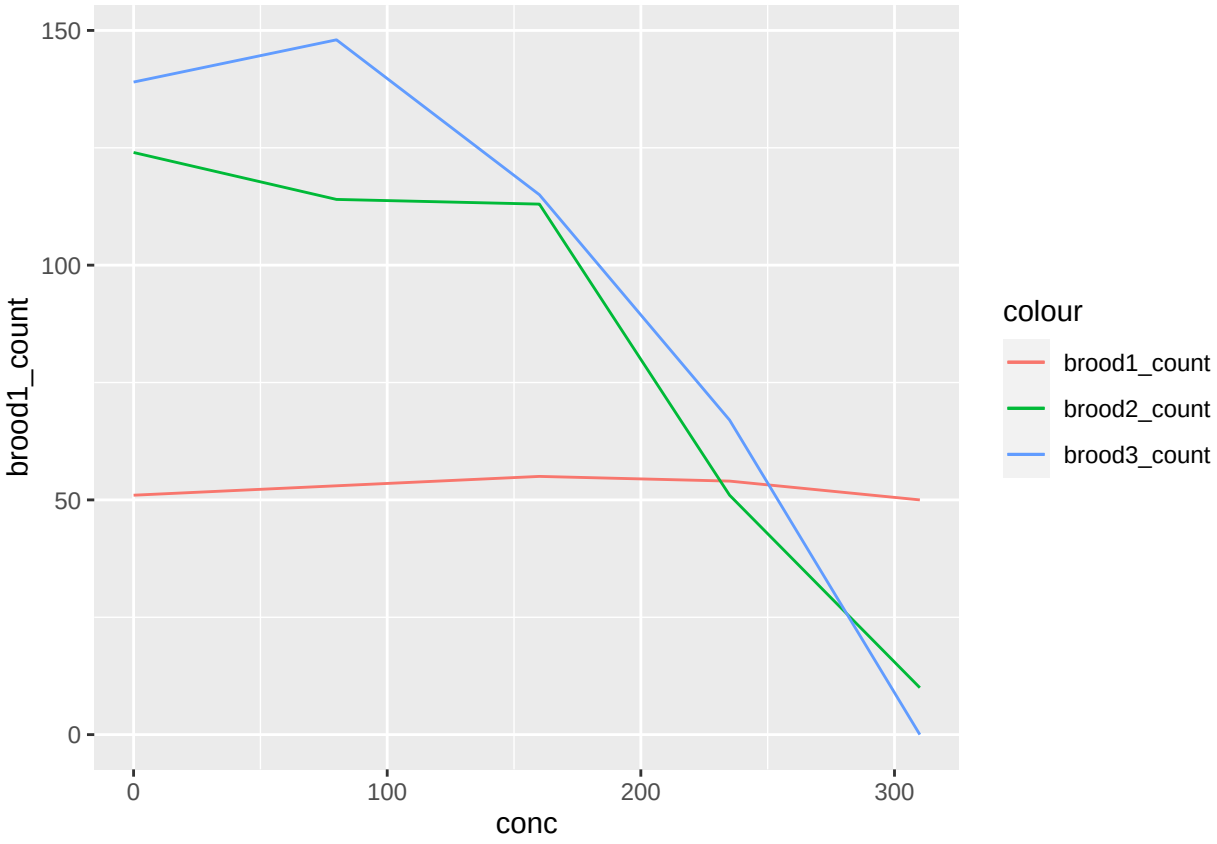
```
##   conc brood1 brood2 brood3 total
## 1    0      3     14     10    27
## 2    0      5     12     15    32
## 3    0      6     11     17    34
## 4    0      6     12     15    33
## 5    0      6     15     15    36
## 6    0      5     14     15    34
```

```
nitrofen %>% group_by(conc) %>% summarise(total_count = sum(total)) %>%
  ggplot(aes(x=conc, y=total_count)) + geom_line( color="grey") +
    geom_point(shape=21, color="black", fill="#69b3a2", size=6)
```



According to plot, concentration 80 does not make much difference from 0. But when greater than 80, concentration clearly has negetive effect on number of live offspring.

```
nitrofen %>% group_by(conc) %>%
  summarise(brood1_count = sum(brood1), brood2_count = sum(brood2), brood3_count = sum(brood3)) %>%
  ggplot(aes(conc)) + geom_line(aes(y=brood1_count, colour='brood1_count')) +
  geom_line(aes(y=brood2_count, colour='brood2_count')) +
  geom_line(aes(y=brood3_count, colour='brood3_count'))
```



Similar trend of decreasing as total count is observed for brood2 and brood3. But seems that brood1 does not vary much with different concentration values.

b)

```
nitrofen_long <- nitrofen %>% dplyr::select(conc, brood1, brood2, brood3)
nitrofen_long$subject <- 1:50
nitrofen_long <- melt(nitrofen_long, id.vars=c('subject', 'conc'))
nitrofen_long$brood_num <- gsub('brood','', nitrofen_long$variable)

nitrofen_long$conc_s <- (nitrofen_long$conc - mean(nitrofen_long$conc)) / sd(nitrofen_long$conc)
head(nitrofen_long)
```

```
##   subject conc variable value brood_num conc_s
## 1       1    0   brood1     3         1 -1.427
## 2       2    0   brood1     5         1 -1.427
## 3       3    0   brood1     6         1 -1.427
## 4       4    0   brood1     6         1 -1.427
## 5       5    0   brood1     6         1 -1.427
```

```
## 6       6    0   brood1     5         1 -1.427
```

```
system.time(mod_nitro <- glmer(value ~ conc_s + brood_num + conc_s:brood_num + (1|subject),
                               nAGQ=25, family=poisson, data=nitrofen_long))
```

```
##    user  system elapsed
##   0.102   0.000   0.103
```

```
summary(mod_nitro)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 25) [glmerMod]
##  Family: poisson  ( log )
## Formula: value ~ conc_s + brood_num + conc_s:brood_num + (1 | subject)
##    Data: nitrofen_long
##
##      AIC      BIC   logLik deviance df.resid
##    313.9    335.0   -150.0    299.9      143
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -2.208 -0.606 -0.008  0.618  3.565
##
## Random effects:
##  Groups  Name        Variance Std.Dev.
##  subject (Intercept) 0.0911   0.302
## Number of obs: 150, groups:  subject, 50
##
## Fixed effects:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)         1.6157     0.0762   21.22  < 2e-16 ***
## conc_s             -0.0160     0.0804   -0.20  0.84216
## brood_num2          0.2932     0.0841    3.49  0.00049 ***
## brood_num3          0.3929     0.0827    4.75  2.0e-06 ***
## conc_s:brood_num2  -0.6133     0.0912   -6.73  1.7e-11 ***
## conc_s:brood_num3  -0.6714     0.0898   -7.47  7.9e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) conc_s brd_n2 brd_n3 cn_:_2
## conc_s       0.016
## brood_num2  -0.585  0.002
## brood_num3  -0.593  0.002  0.551
## cnc_s:brd_2  0.032 -0.609  0.230  0.018
## cnc_s:brd_3  0.036 -0.617  0.017  0.237  0.613
```

Note that both brood number and the interaction between brood number and concentration chooses `brood1` as the reference level. With this in mind, we note that the estimate of `conc_s` effect, which actually stands for the effect of concentration in brood1, is really close to 0 and not significant, meaning number of live offspring is not significantly related to concentration level for brood1. On the other hand, both `conc_s:brood_num2` and `conc_s:brood_num3` have significant negative effect, meaning that compared with that of brood1, number of live offspring in brood2 and brood3 are much more significantly negatively correlated with concentration level. What's more, the magnitude of estimated effect for `conc_s:brood_num3` is greater than that of `conc_s:brood_num2`. All above conclusions agree with what we observed from the plot.

c)

```
mod_nitro_gee <- geeglm(value ~ conc_s + brood_num + conc_s:brood_num,
                  id=subject, corstr="exchangeable", scale.fix=TRUE,
                  data=nitrofen_long, family=poisson)
summary(mod_nitro_gee)
```

```
##
## Call:
## geeglm(formula = value ~ conc_s + brood_num + conc_s:brood_num,
##     family = poisson, data = nitrofen_long, id = subject, corstr = "exchangeable",
##     scale.fix = TRUE)
##
##  Coefficients:
##                    Estimate  Std.err    Wald Pr(>|W|)
## (Intercept)         1.66013  0.03947 1769.30  < 2e-16 ***
## conc_s             -0.00212  0.04126    0.00   0.9590
## brood_num2          0.31693  0.11082    8.18   0.0042 **
## brood_num3          0.42105  0.08797   22.91  1.7e-06 ***
## conc_s:brood_num2 -0.51884  0.10176   25.99  3.4e-07 ***
## conc_s:brood_num3 -0.56826  0.08577   43.90  3.5e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = exchangeable
## Scale is fixed.
##
##   Link = identity
##
## Estimated Correlation Parameters:
##       Estimate Std.err
## alpha        0       0
## Number of clusters:   150  Maximum cluster size: 1
```

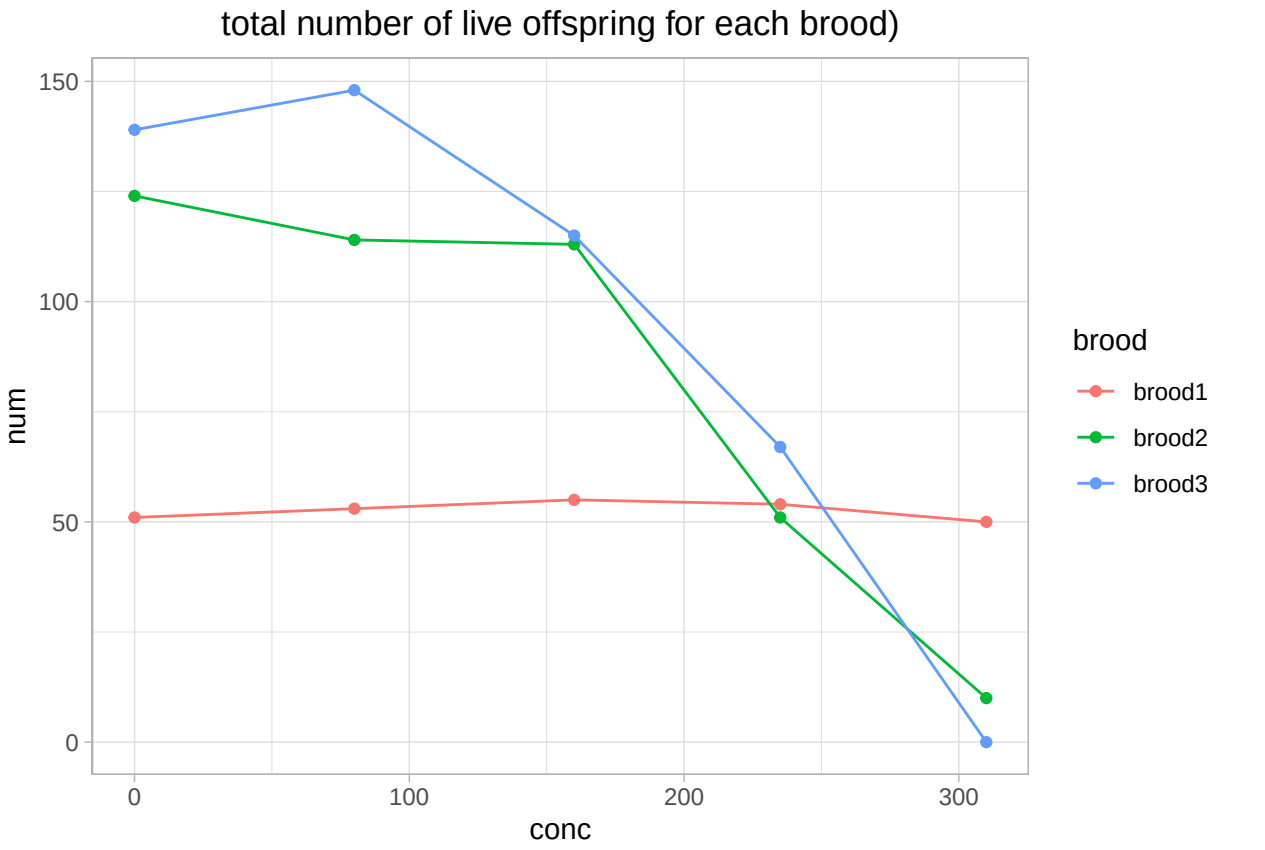
Interestingly, this time, we can no longer say that estimates given GEE are notably smaller in magnitude than that of GLMM. This is because the reason explained in Problem 3 is critical only for logit link. When modeling count response, which requires log link, the difference of interpretations of estimated coefficients of GEE and GLMM are not that crucial. Also, we can note that this time GEE estimates do have larger standar errors.

We can observe from the plots that when the concentration of the solution is higher than 80 mug/litre, the higher the concentration, the less the number of live offspring.

Then we can continue to observe the trend for each brood.

```
nitrofen2  <- nitrofen %>%
      dplyr::select(-total) %>%
      pivot_longer(
        cols = brood1:brood3,
        names_to = "brood",
        values_to = "num"
      ) %>%
  group_by(conc,brood) %>%
  summarise(num = sum(num), .groups = "drop")
ggplot(data = nitrofen2, mapping = aes(x = conc, y = num, col = brood)) +
  geom_point() +
```

```
  geom_line() +
  theme_light() +
  labs(title = "total number of live offspring for each brood)")+
  theme(plot.title = element_text(hjust = 0.5))
```



We can see that brood2 and brood3 has the similar trend with the last plot, while the brood1 is quite different. It seems that in brood3, the total number of live offspring isn't affected by the nitrofen concentration in the solution.

###(b)

First, do data wrangling as following:

```
nitrofen3 <- nitrofen %>%
  dplyr::select(-total) %>%
  pivot_longer(
    cols = brood1:brood3,
    names_to = "brood",
    values_to = "num") %>%
  mutate(brood = substring(brood, 6,6)) %>%
  mutate(brood = as.factor(brood))
nitrofen3 <- cbind(nitrofen3, id = rep(c(1:50), each = 3)) %>%
  mutate(conc = scale(conc)) %>%
  dplyr::select(c(id,brood,conc,num))

head(nitrofen3)
```

```
##   id brood  conc num
## 1  1     1 -1.43   3
```

```
## 2  1     2 -1.43  14
## 3  1     3 -1.43  10
## 4  2     1 -1.43   5
## 5  2     2 -1.43  12
## 6  2     3 -1.43  15
```

Then, we use Numerical integration for GLMM

```
mod2glmm <- glmer(num ~ conc + brood + conc*brood + (1|id),nAGQ=25,
                  family=poisson, data=nitrofen3)
summary(mod2glmm)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 25) [glmerMod]
##  Family: poisson  ( log )
## Formula: num ~ conc + brood + conc * brood + (1 | id)
##    Data: nitrofen3
##
##      AIC      BIC   logLik deviance df.resid
##      314      335     -150      300      143
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -2.208 -0.606 -0.008  0.618  3.565
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  id     (Intercept) 0.0911   0.302
## Number of obs: 150, groups:  id, 50
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.6157     0.0762   21.22  < 2e-16 ***
## conc         -0.0160     0.0804   -0.20  0.84216
## brood2        0.2932     0.0841    3.49  0.00049 ***
## brood3        0.3929     0.0827    4.75  2.0e-06 ***
## conc:brood2  -0.6133     0.0912   -6.73  1.7e-11 ***
## conc:brood3  -0.6714     0.0898   -7.47  7.9e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) conc   brood2 brood3 cnc:b2
## conc         0.016
## brood2      -0.585  0.002
## brood3      -0.593  0.002  0.551
## conc:brood2  0.032 -0.609  0.230  0.018
## conc:brood3  0.036 -0.617  0.017  0.237  0.613
```

we can observe that the random effect id has variance 0.0911 with standard deviance 0.302. For fixed effects, only `conc` is not significant. This agrees with the observation in (a).

From the estimated coefficients of `conc:brood2` and `coc:brood3` are all negative, we can learn that when the nitrofen concentration in the solution increase in brood2 and brood3, the number of live offspring will decrease. This means that the nitrofen concentration has negative effect on the number of live offspring.

Also, the estimated coefficients of `brood2` and `brood3` are positive. This indicates that brood2 and brood3

tend to have more offspring than brood1.

###(c)

For GEE model:

```
mod2geep <- geeglm(num ~  conc + brood + conc*brood,
                   id=id, corstr="ar1",
                   data = nitrofen3, family=poisson)
summary(mod2geep)
```

```
##
## Call:
## geeglm(formula = num ~ conc + brood + conc * brood, family = poisson,
##     data = nitrofen3, id = id, corstr = "ar1")
##
##  Coefficients:
##             Estimate Std.err    Wald Pr(>|W|)
## (Intercept)   1.6642  0.0393 1797.11  < 2e-16 ***
## conc          0.0307  0.0414    0.55   0.4585
## brood2        0.3168  0.1071    8.75   0.0031 **
## brood3        0.4169  0.0780   28.57  9.0e-08 ***
## conc:brood2  -0.5420  0.1006   29.01  7.2e-08 ***
## conc:brood3  -0.6006  0.0760   62.46  2.8e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = ar1
## Estimated Scale Parameters:
##
##             Estimate Std.err
## (Intercept)     1.65   0.184
##   Link = identity
##
## Estimated Correlation Parameters:
##       Estimate Std.err
## alpha    0.329   0.106
## Number of clusters:   50  Maximum cluster size: 3
```

We can see that the estimated correlation between observations on the same subject is 0.329 with a standard error of 0.106, which indicates the correlation between responses within individuals.

Only `conc` is not significant and all the other estimated coefficients are very closed for both GEE and GLMM methods. Thus, we can get the similar results with what we have got in GLMM. Besides, the estimated βs are also very close, even though they have different meanings.