STAT/BIOSTAT 571: Homework 10 — Key

1. ***[Conditional likelihoods, 3 points]*** *Consider the conditional likelihood approach to the progabide example. As an alternative to the model-based standard errors seen in class (slide 3.134) consider use of robust standard errors, applied to the conditional likelihood's score equations. Two different ways to do this are coded below;*

```
> library("sandwich")
> glm1 <- glm(cbind(Yi1, Ti-Yi1) ~ trtbas , family=binomial)
> round( cbind( est=coef(glm1), rob.se=sqrt(diag(vcovHC(glm1, "HC0"))) ) , 2)
              est rob.se
(Intercept) -0.11   0.12
trtbas       0.10   0.21
>
> Ylong   <- rep( rep(c(1,0),each=59), c(Yi1, Ti-Yi1) )
> trtbaslong <- rep( c(trtbas,trtbas),  c(Yi1, Ti-Yi1) )
> glm2 <- glm(Ylong ~ trtbaslong, family=binomial)
> round( cbind( est=coef(glm2), rob.se=sqrt(diag(vcovHC(glm2, "HC0"))) ) , 2)
              est rob.se
(Intercept) -0.11   0.05
trtbaslong   0.10   0.07
```

The point estimates are identical but these analyses have different standard error estimates. By thinking about the robustness that each provides, state why they are different, and which (if either) is appropriate for the use in the conditional likelihood method presented in class.

In the first version, we use the estimating equations from a Binomial regression of Y_{i1} on the product of the baseline and treatment indicators, and an intercept. Through use of the sandwich approaches, we gain robustness to the assumption that the variance of each Y_{i1} (conditional on total $\sum_j Y_{ij}$) depends on its mean (np) through the usual $np(1-p)$ relationship. The model-based approach does make this assumption. Furthermore, though it matters less, as this is a canonical link GLM we also gain robustness to the assumption that the mean model is correctly specified, i.e. that all clusters with `trtbas`=0 or 1 have the same proportion of 'successes' (p). In large-sample settings, the only active assumption is that the outcomes are independent across clusters. This is part of the conditional likelihood assumptions, so this use of robust standard errors is valid.

For the second version, while the point estimates are identical, the standard error estimates are using independence of each individual event counted in Y_{i1} (i.e. whether an event occurs at time 1 versus some other timepoint). This is effectively assuming that, within a cluster, whether each event happens at baseline versus some other timepoint is independent, given the total number of events. While we could motivate this step with conditionally-Poisson events and a mean model with a canonical link, if these assumptions are violated this independence does not hold. (Note that this is still a concern, despite our making valid assumptions about the mean-variance relationship for each individual event counted in Y_{i1}, i.e. variance $\mu(1-\mu)$ for mean μ.) So in general, this independence will not hold, and correspondingly this use of robust standard errors will not be valid. The fact that the two approaches here are so different is an indication that either the Poisson assumption or mean model **are** violated, in this case.

The discussion of independence alone is enough to gain full credit.

2. ***[Bayesian analysis of mixed models, 5 points]*** *The class site contains data from a non-randomized longitudinal study, in which subjects taking one of two treatments have their cholesterol measured at 5 time points. Some missing values are present, which you should assume are Missing At Random.*

(a) Fit the following model to the complete observations, and report posteriors for each of the parameters.

$$\begin{aligned}
\tau_b, \tau_Y &\overset{i.i.d.}{\sim} \Gamma(\text{shape} = 0.1, \text{rate} = 0.1) \\
\beta_0, \beta_1, \beta_2, \beta_3 &\overset{i.i.d}{\sim} N(0, 10000) \\
b_i|\tau_b &\overset{i.i.d}{\sim} N(0, 1/\tau_b) \\
Y_{it}|b_i, \text{trt}_i &\overset{indept}{\sim} N(\beta_0 + b_i + \beta_1 \text{trt}_i + \beta_2 t + \beta_3 \text{trt}_i t, 1/\tau_Y)
\end{aligned}$$

where $t = 1, 2, 3, 4, 5$ and trt_i is an indicator for receiving treatment 1 (versus treatment 0). Note: you are free to use any numerical method, and several options are available, but say what you did

Using INLA (which is perhaps most straightforward for this analysis) the following posterior quantiles are obtained;

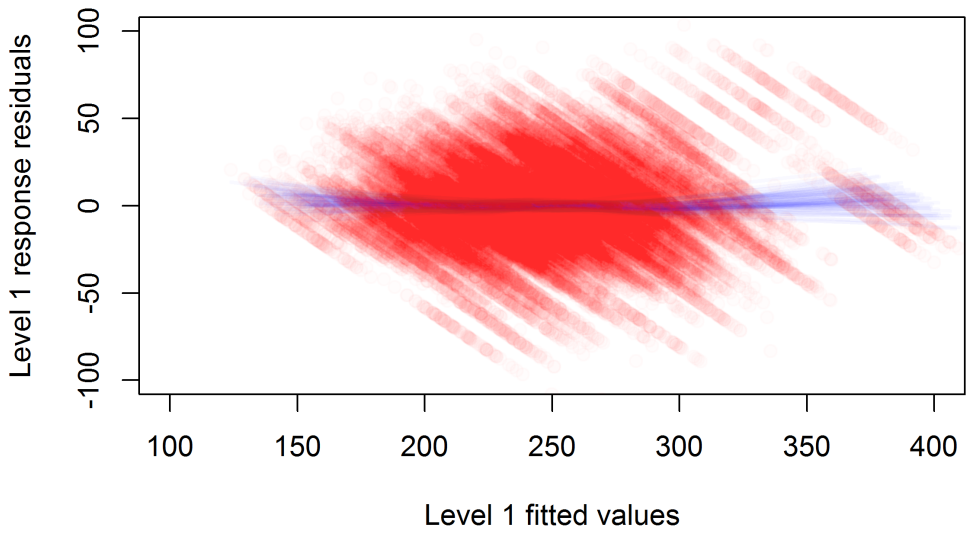
Effect	Parameter	Post'r Med'n	(2.5%, 97.5%)
Intercept	β_0	224	213,236.00
trt	β_1	6.0	-12.0,24.1
time	β_2	7.2	4.9,9.5
trt×time	β_3	-1.5	-5.0, 2.0
Error SD	$\sigma_Y^2 = \tau_Y^{-1/2}$	24.8	23.1,26.9
Rand Int'pt SD	$\sigma_b = \tau_b^{-1/2}$	37.5	32.4,45.9

Of course, presenting posteriors for τ_b, τ_Y also earns full credit. Do note, however, that the values of τ_b, τ_Y are small, and rounding every entry in the posterior summary down to e.g. 0.00 is a (serious) mistake. Proving very many decimal places in the summary is also a mistake, e.g. the 10th decimal place is of no practical use when the posterior has a range of many whole units. Use the statistical measures of uncertainty to tell you how much accuracy is available, and round to a corresponding number of decimal places or perhaps slightly more.

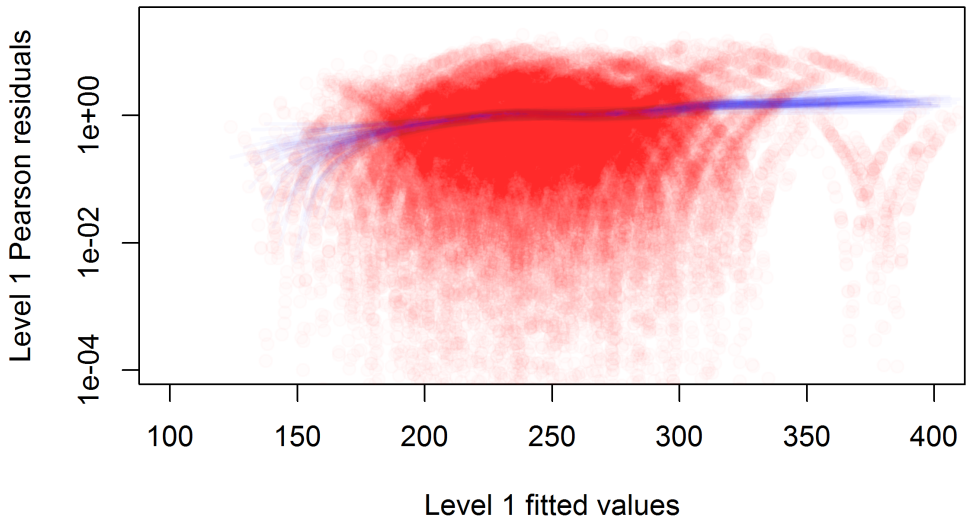
While not required for full credit, the posterior distributions of the (many) b_i can also be presented, in some form of graphical summary. Some students implemented the 'funnel plot' from class. Unfortunately in notes this was not implemented correctly, making its interpretation difficult. This has now been fixed; those who attempted this plot were not penalized for it.

(b) Give diagnostics plots for the model fitted in a), briefly describing what they show. Note that you are not being asked for MCMC diagnostics

Many possible choices are available. The plot below give the posterior distribution for level 1 residuals ($e_{ij} = Y_{ij} - \beta_0 - \beta_1 \text{trt}_i - \beta_2 t - \beta_3 \text{trt}_i t$) against fitted values ($\beta_0 + b_i + \beta_1 \text{trt}_i + \beta_2 t + \beta_3$) for 100 well-spaced draws from the posterior, with a smoother drawn through the points for each draw. This would be a standard option for checking that the mean model is not badly mis-specified. Given the limited number of covariates (2 treatment values and 5 times) it would also be feasible to give boxplots of the residuals at each of the 10 possible combinations, and giving a posterior distribution for the mean of each of these subsets of residuals.



For completeness, to assess the constant-variance assumption of the observations given the random intercepts, we similarly plot the Pearson residuals $e_{ij}^2 \tau_Y$, again averaged over the posterior;



Again, no major systematic problem is apparent; the mean of these squared residuals seems close to 1 across the great bulk of the data.

Leave-one-out diagnostics are also possible, but very slow to implement in any MCMC form of calculation. The attached code shows how to do this in INLA – but some care should be taken to ensure the posteriors for the precision parameters are calculated correctly. Several INLA users had problems with this.

(c) *The data contains some missing values. Suppose you had used MCMC in part a) but instead of using complete-observation data, had used Bayesian methods to impute the missing outcomes. Briefly, state how you could use a small number of steps from this chain in a multiple-*

imputation analysis. Note you are ***not*** *required to implement such an analysis*

A few draws of the missing Y_{it} from such a posterior, spaced sufficiently far apart so as to be essentially independent, can be used in mulitple imputation analysis, as seen in Chapter 2. Here, for each draw, the imputed-to-full data is analyzed using e.g. some mixed-model MLE analysis, and the various $\hat{\beta}$ and $\hat{\tau}$ combined using Rubin's rule. While not required for credit, note that the advantages of doing this over the fully Bayesian analysis (i.e. using the full chain) are typically not great; one would gain some robustness to the prior assumptions on the β and τ parameters. If GEE was used on each imputed dataset, some robustness to the modeling assumptions is achieved, but only by making explicit modeling assumptions about the missing data. This is not 'wrong' but it is hard to justify in practice.

3. ***[Marginal and conditional: 6 points]*** *Read the paper by Ritz and Spiegelman (Statistical Methods in Medical Research, 2004) that is available on the course website. Using full sentences and paragraphs throughout, write a brief summary of all the main ideas and results (1–2 double spaced typed pages, with 12 point font and 1 inch margins). You may omit §2.5, which deals with survival analysis. While it may be helpful for your summary's structure to be similar to the original paper, do not plagiarize the paper or other sources; write using your own words and do not copy complete sentences or phrases from elsewhere.*

   Obviously there is no uniquely 'correct' answer for this question. Some key points are below. Compressing this to two pages, double-spaced, is a challenge; it is reasonable to omit the data-based examples and the special cases for which general results are available.

   - Marginal parameters and conditional (a.k.a. subject-specific) parameters are not typically the same, and have different meanings
   - When they do coincide, methods (like GEE) that are designed to estimate marginal parameters can be usefully employed to estimate conditional parameters
   - **§2.1:** For additive random effects, and in particular linear mixed models with random intercepts (but not necessarily assuming Normality) then marginal and conditional variance coincide. The marginal variance of the outcomes can also be obtained easily, in this case
   - **§2.1.1:** For example: conditionally Poisson outcomes with gamma-distributed random intercepts the marginal distribution is negative binomial
   - **§2.1.2:** The authors give a short example, showing broad agreement in estimates and confidence intervals using LMM-based and GEE-based examples of a linear regression analysis using a Poisson family. For the 'naïve' GEE analysis to be valid in this case would require that the (working) covariance matrix is actually correct, which is not consistent with assumptions of the model-based version
   - **§2.2** With a log link and i.i.d. random intercepts, the marginal and conditional means are equivalent except for their intercepts. As in §2.1.1, Poisson-gamma mixtures give closed form examples, in addition to the Normal examples. For the multivariate extension, also assuming Normality of the random intercepts, we see the results from slide 3.113.
   - **§2.2.1** Using much the same example as before, the authors illustrate the similar point estimates, but notable difference in standard error estimates; the robust GEE ones are larger, which may suggest mis-specification somewhere in the modeling approach
   - **§2.3** Using a probit link (i.e. the inverse CDF of a $N(0,1)$ distribution) then with Normal random intercepts, the slope parameters (β) in the marginal model are attenuated relative to those in the conditional model. The attenuation factor, $1/\sqrt{1+\mathrm{Var}[\,b_i\,]}$ is larger for more diffuse random effects
   - **§2.4** Using a logit link and with binary outcomes, no closed-form exists for the attenuation factor, but it can be approximated. With exponential or double-exponential random intercepts, closed form marginal mean models are available, but the connection of parameters

that might describe them are not easily connected to the conditional parameters. (Note this paper pre-dates widespread knowledge of the 'Bridge' distribution discussed in class)

- **§2.4.1** The example describes the challenge in fitting some of these models. No connection between the estimated conditional parameters, attenuation factor and marginal parameters is made
- **§3** The difference between marginal and conditional can be viewed as one of omitting the variables whose effect is captured by the random effects – although this is not the only way to view them. The various formula developed to describe the marginal/conditional differences fit into the literature on omitted-variable biases

Also note this is an active research area; see for example this pre-print of a just-published paper, showing that random effects distributions with the 'bridging' property are essentially unique, i.e. there is only one per family of mixed model.

4. ***[Missing data in GEE and mixed models, 5 points]*** *Suppose you are working with a statistician, on linear regression-based analysis of data that contains missing values. To understand the behavior of GEE and LMMs under MAR, your co-author has simulated data very similarly to the setting of HW6, and 3.204–3.206;*

$$\begin{aligned} b_i &\overset{i.i.d}{\sim} N(0, \sigma_b^2) \\ X_i &\overset{i.i.d}{\sim} Bern(0.5) \\ Y_{it}|b_i, X_i = x &\overset{indept}{\sim} N(\beta_0 + b_i + \beta_1 t + \beta_2 x, \sigma_Y^2), for\ t = 1, 2, 3, \end{aligned}$$

where $n = 1000$, $\sigma_b = 1, \sigma_Y = 0.5, \beta_0 = 1, \beta_1 = -1, \beta_2 = 0.5$, *and if any* $Y_{it} < -1$ *then all subsequent* $Y_{it'}$ *are missing. (Note the original version used* $Y_{it} < 0$*; this version is less extreme)*

Despite the robustness of GEE, and correct mean-model and variance (i.e. second-moment) assumptions, your coauthor is surprised to see that 95% CIs from GEE with an exchangeable working matrix do not provide as good coverage of β_1 *and* β_2 *as those from LMM analysis. They feel the GEE approach should provide the same robustness to MAR missing outcome as LMM, as it relies on assumptions about the first two moments, just like LMM.*

(a) *Provide simulation results confirming your co-authors' findings; that GEE linear regression here does not give good coverage, but LMM analysis does. These simulations need not be large-scale, but should be clear enough to confirm this finding unambiguously. As always, you should report what you did*

Given the focus on 95% CIs in the question, it makes sense to summarize simulation results using some measure of coverage. Based on 1000 replications, the empirical coverage of the nominal 95% CIs was

Method	$\beta_0 = 1$	$\beta_1 = -1$	$\beta_2 = 0.5$
LMM	0.943	0.940	0.948
GEE	0.829	0.009	0.926

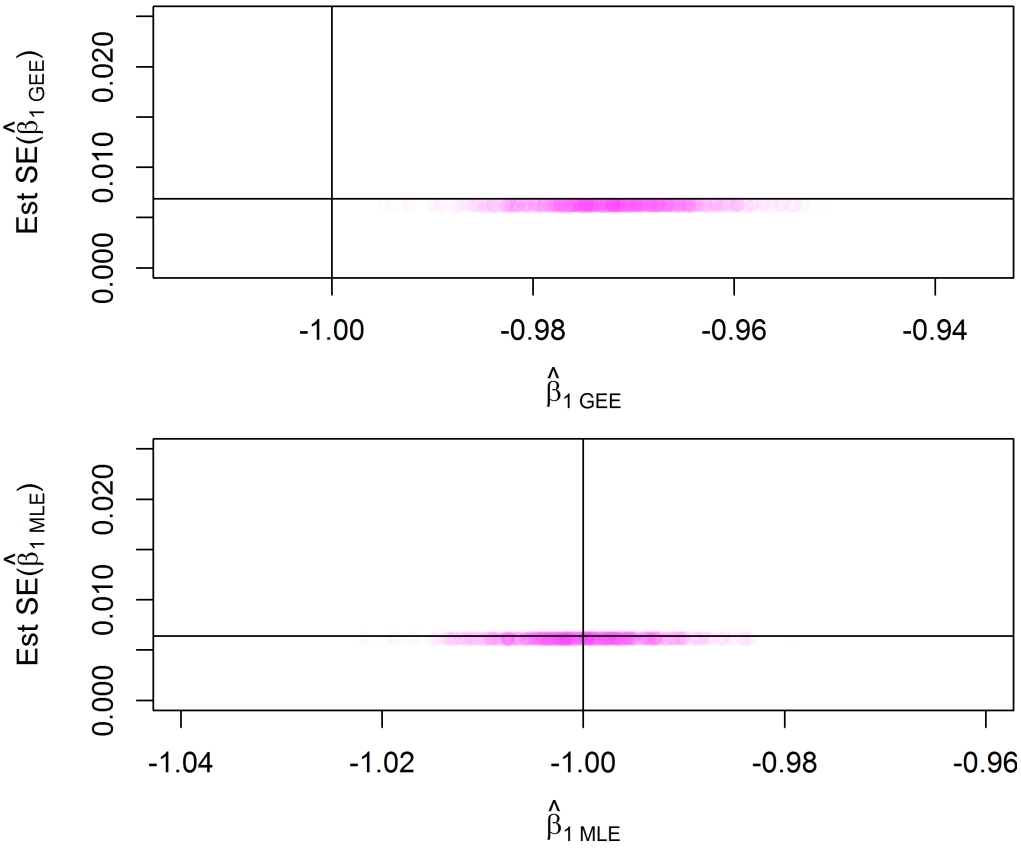
Note that with 1000 replications, the margin of error here is $\approx \pm 0.01$, so the differences between the methods are not plausibly due to chance.

(b) *After carefully considering the differences between the two anlayses, write a short explanation for your co-author, of why the GEE approach does not work as well as LMM*

The statement about "first two moments" in the question is important; it is not sufficient to quote results about GEE being generally invalid except under MCAR; these results do not rule out situations where special cases of GEE analysis can be valid under MAR. Given the close connections between the two approaches here (see e.g. 3.14, or the estimating equations

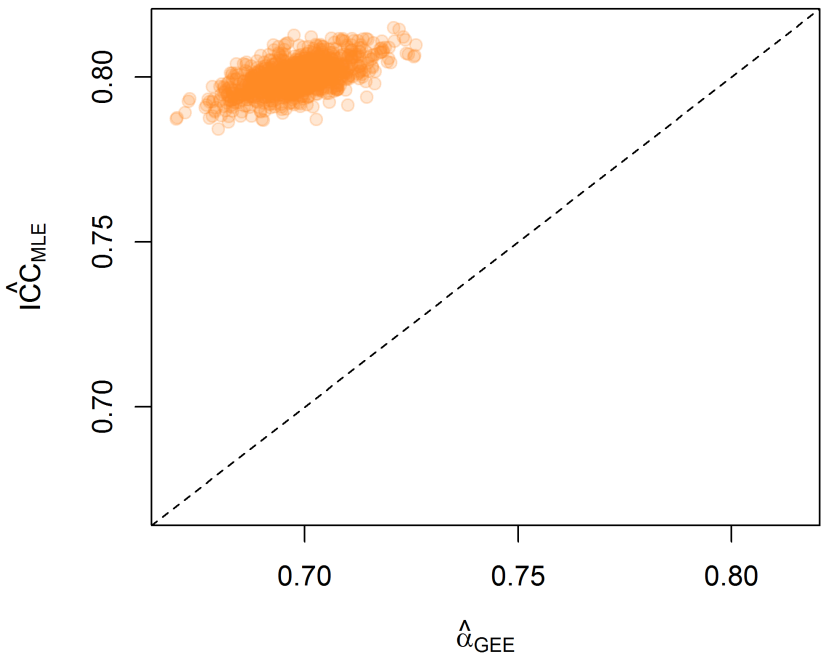
for $\boldsymbol{\beta}$ under GEE vs LMM linear regression) it's reasonable to expect that GEE might provide some robustness to MAR missingness.

To investigate what's happening, a more detailed look at the simulation results for β_1 is given below;



For each plot, the horizontal line indicates the empirical standard deviation of the point estimates over all the simulations, and the vertical line indicates the true parameter value. In both cases, the uncertainty in the standard error estimates is negligible, compared to the uncertainty in $\hat{\beta}$. We see that, as we expect from theory, the MLE approach is basically valid. The modest undercoverage appears to be due to standard error estimates that are biased small; using REML here would help a little. GEE is estimating a different parameter – it estimates some component of a linear trend summarizing the population, as discussed on e.g. slide 2.37. However, this is not the same parameter as the model-based approach is estimating, and in this situation it's reasonable to call GEE biased. Moreover, GEE is slightly underestimating the standard error of its estimate – because the mean model used is not correct; the mean of the complete observations is not linear in x and t. However, the dramatically bad coverage can not be explained by slightly-wrong standard error estimates; the problem is due overwhelmingly to the bias.

The underlying reason for the bias lies in the estimation of α in GEE, versus the variance components σ_b^2 and σ_Y^2 in the LMM, and in particular the estimated $ICC = \sigma_b^2/(\sigma_b^2 + \sigma_Y^2)$. As seen below, these estimates are both consistent for some value, but not the same one – the diagonal line indicates equality;



This in turn affects the weighting of different components of the estimating equations for $\boldsymbol{\beta}$, leading to the problems above.

If we had taken care to ensure that GEE's α was estimated in the same way as the ICC in the LME analysis, the two procedures would have essentially identical point estimates, both would give valid standard error estimates. This approach would essentially just add robust standard errors around the LMM MLE estimates. It is not available in off-the-shelf R functions, but the `robust` option in Stata's `xtmixed` command provides it.

5. ***[(G)LMMs and assumptions, 8 points]*** *Suppose we have a data-generating mechanism where*

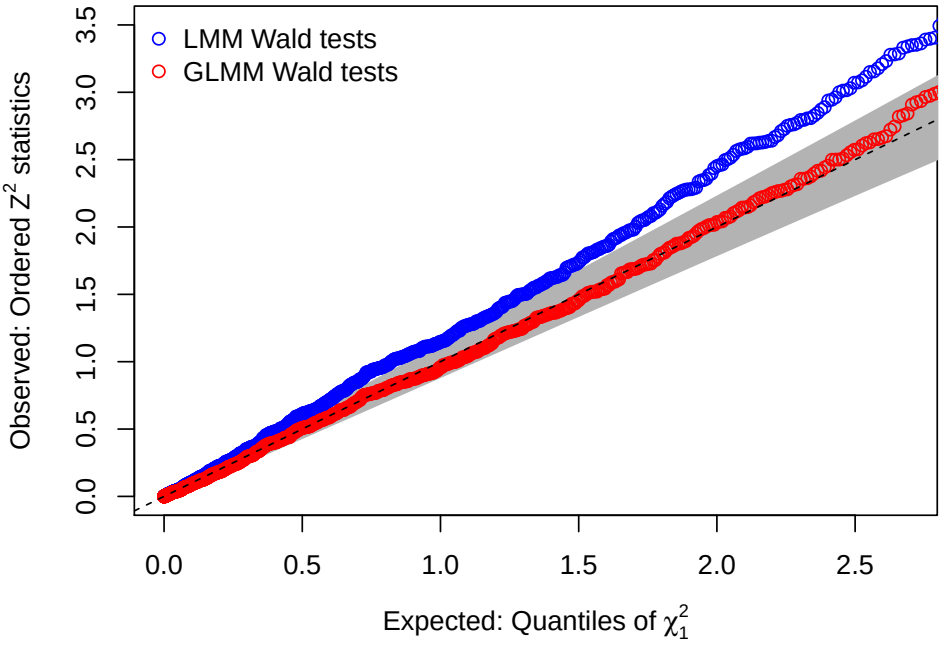
$$\begin{aligned} Z_i &\overset{i.i.d.}{\sim} Bern(p) \\ X_{ij}|Z_i = z &\overset{indept}{\sim} Bin(2, \text{expit}(\theta_0 + \theta_1 z)) \\ b_i|\boldsymbol{X}_i, Z_i &\overset{i.i.d.}{\sim} N(0, \sigma_b^2) \\ Y_{ij}|\boldsymbol{X}_i, Z_i, b_i &\overset{indept}{\sim} Bern(\text{expit}(\beta_0 + b_i + \beta_1 X_{ij} + \beta_2 Z_i)), \end{aligned}$$

and the inference of interest is a test of the null hypothesis, that keeping cluster and confounding variable Z fixed, there is no association between Y and X. Throughout, our simulations will use $\beta_1 = 0$, i.e. generating data under the null hypothesis.

(a) Using a simulation study for $n_i = 3$ and $n = 1000$, with $p = 1/4, \theta_0 = -2, \theta_1 = 1.5, \beta_0 = -2.5, \beta_2 = 1.5, \sigma_b = 0.1$, summarize the distribution of p-values produced by fitting LMM regression of Y on X and Z, with random intercepts for each cluster, and performing a Wald test using the X coefficient. (The choice of how to summarize is up to you, but at minimum you should indicate the nominal and actual Type I error rates of the test at $\alpha = 0.05$, indicating the extent of any Monte Carlo error in your answers)

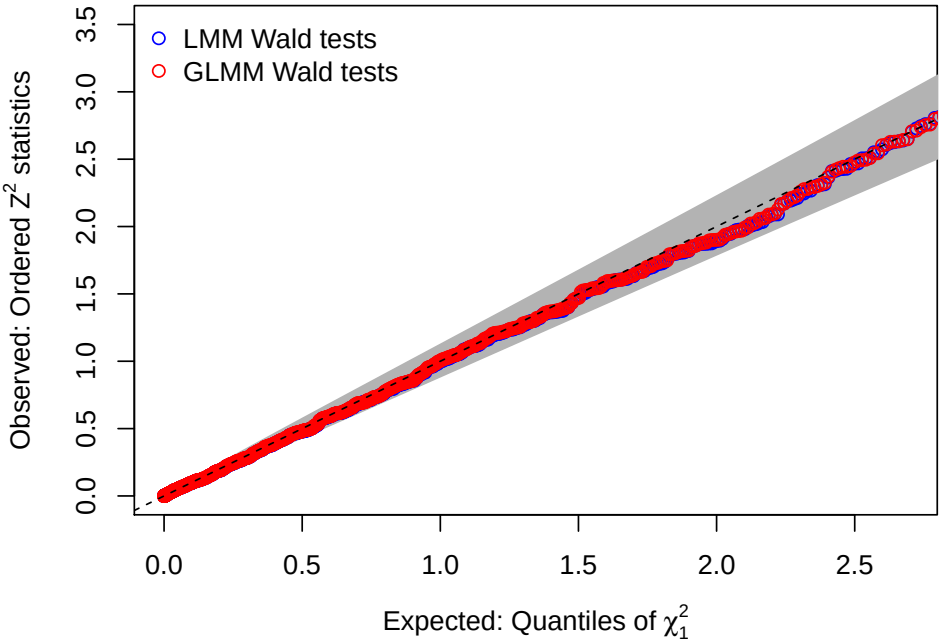
The QQ plot below is of the Z^2 statistics for the tests from 1000 replicate experiments. These are compared to the χ^2_1 distribution, i.e. the usual reference distribution for these tests. The plot also shows GLMM results, used in part b). On the plot, the Monte Carlo error has been described by plotting 95% prediction intervals for the various order statistics of a sample of χ^2_1 distributions. Also, the calculations were from 1000 replicates (excluding a small number for which `glmer()` did not converge) but the plot focuses on the smaller ranked Z^2 statistics,

which are less prone to noise. We see that the LME results are systematically 'inflated', i.e. they overstate the true significance of the results. This is a problem, as it means the Type I error rate would not be controlled below the nominal α, at any corresponding significance threshold. For example, around 14% of the simulations have $Z^2 > 2.7$ using the LMM approach, but the reference distribution has 10% of its support beyond this value.
(Note: instead of the Z^2, χ^2 'scale', these plots are often shown for $-\log_{10}$ p-values, i.e. the number of zeroes in the p-value.)



(b) *Contrast your summary in a) with the same summary of Wald test p-values from fitting GLMM logistic regression, of Y on X adjusting for Z with random intercepts for each cluster. Hint: use the same datasets as in part a)*
See the plot above for results. The GLMM analysis has no such problems; its observed quantiles line up almost exactly with those of the χ^2_1 reference distribution.

(c) Repeat a) and b) for the same data-generating mechanism, but with $\beta_0 = -0.75$
The QQ plot of results is given below. Here, both methods give well-calibrated p-values



Note: for the LMM analyses, you should face no difficulties with calculations; the estimates of σ_b^2 may go to zero, but this does not invalidate or otherwise complicate estimation of the standard error of $\hat{\beta}_1$. For GLMM, `glmer()`'s default number of quadrature points will be sufficient to illustrate the main points; for modestly-greater accuracy in the Type I error rates for the GLMM analysis, a greater number of points should be used. The computational burden is non-trivial, but e.g. running overnight should (easily) give enough simulations to see the main points.

(d) *Using what you know about the importance of assumptions of LMMs and GLMM (correct mean model, correct within-cluster variance, mean-variance relationship, Normality of outcomes etc) carefully explain any differences or lack of difference in the results between the two simulation setups*

In both situations, Z acts as a confounder for the relationship of Y on X, and so we expect to have to adjust the regression in order to obtain valid inference, i.e. p-values that are approximately $U(0,1)$. Modulo small-sample concerns, the GLMM approach should provide valid inference, as it fits a model which is fully correctly specified. As discussed in class, the LMM analysis is valid in large samples when its first two moments are well-specified. We do not have this in a) and b), as the variance of the observations varies with their mean. As the observations are binary, if they have mean μ their variance must be $\mu(1-\mu) = 1/4-(\mu-1/2)^2$, and the range of values of μ present is large enough that this non-constant variance invalidates the LMM analysis. In c) we do not see this; the values of μ differ (they are approximately 0.32 and 0.68 for $z = 0, 1$ respectively) but these values are symmetric around $\mu = 0.5$ and so have the same $\mu(1-\mu)$, meaning that homoskedasticity is retained.

(To see that the other assumptions are met, note that mis-specification of the mean model under the null cannot occur, as only a single binary covariate Z is present. Both models also assume exchangeability of the observations (i.e. the correlation matrix within each cluster have off-diagonal entries all zero) but this follows from stated model.)

A final note: this question reflects a recent research problem – no important simplification has been made. For binary outcomes (Y) clustered in families, where ancestry (Z) confounds the association between outcome and a genetic variant (X), it has long been standard to use LMM analysis in the way described here in a) – not least because this is fast. The question shows that ignoring non-constant variance in this way leads to invalid inference, when the heteroskedasticity is strong enough. Some users have been reluctant to accept this, as in practice the confounding tend to give a mix of over-stated and under-stated Z^2 statistics, depending on the genetic variant being assessed, making the QQ plots in look acceptable. The question shows this acceptance is formally wrong – though it takes large but not implausible levels of confounding to illustrate it clearly. The work will appear shortly in the American Journal of Human Genetics.

6. ***[Review, 3 points]*** *Using the most up-to-date version of the class slides, find 6 remaining typos, and state what a correct version would be. Note: do not start this question until noon on Monday, March 14th, or until notified by the instructor.*

Thank you for all your contributions! Any collection of 6 typos, suspected typos, or suggested re-wording received full credit.

R code

See the original TEXfile for code with better indentation.

```
##
## Q2
##

## a)

dd <- read.csv("hw10q2.csv")
dd2 <- dd[complete.cases(dd),]

library("INLA")
hyperprior <- list(theta=list(prior="loggamma",param=c(0.1, 0.1)))
fixedprior <- list( mean.intercept=0, prec.intercept=.0001,
   mean=0, prec=.0001 )
formula1 <- y ~ trt*time + f(id,model="iid",hyper=hyperprior)
familyprior <- list(hyper = list(prec = list(prior = "loggamma", param = c(0.1, 0.1))))

inla1 <- inla(formula1, family="gaussian", data=dd2,
control.fixed= fixedprior, control.family=familyprior)

signif( rbind( inla1$summary.fixed[,c(4,3,5)],
1/sqrt(inla1$summary.hyperpar)[,c(5,3,4)] ) , 3)

# sanity check
library("nlme")
lme1 <- lme( y ~ trt*time, ~1|id, data=dd2)
intervals(lme1)

## b) - diagnostic plots using WinBUGS output

dd.list <- list(id=dd2$id, time=dd2$time, y=dd2$y, trt=dd2$trt, N=nrow(dd2),
n=length(unique(dd2$id)) )
library("R2WinBUGS")
library("coda")
bugs1 <- bugs( data=dd.list, inits=NULL,
parameters.to.save=c("beta0","beta1","beta2","beta3","sigb","sigY","b"), DIC=FALSE,
model.file="hw10q2winbugscode.txt", bugs.seed=4,
n.chains=1, n.iter=40000, n.burnin=6000, n.thin=1,
working.dir=getwd(),
bugs.dir="C:/Program Files/winbugs14/WinBUGS14",
codaPkg=TRUE)
chain1 <- read.coda(output.file=bugs1, index.file=paste(getwd(),"/codaIndex.txt", sep=""))
chain1.short <- chain1[seq(1,34000,339),]
rm(chain1) # to save memory

# construct posteriors of the fitted values
X <- model.matrix(y ~ trt*time, data=dd2)
fixed.mu  <- X %*% t(chain1.short[,104:107]) # posterior for X*beta terms
```

```
random.mu <- t(chain1.short[,1:103])[dd2$id,] # posterior for b[i] terms
mu <- fixed.mu + random.mu

# make the diagnostic plots
png("hw10q2plot1.png", w=6*400, h=4*400, res=400)
plot( y=residuals(lme1, level=1), x=fitted(lme1, level=1), type="n", ylim=c(-100,100),
xlim=c(100,400), xlab="Level 1 fitted values", ylab="Level 1 response residuals")
for(j in 1:101){
points( x=mu[,j], y=dd2$y-mu[,j], col="#FF000003", pch=19)
lines(lowess( x=mu[,j], y=dd2$y-mu[,j], iter=0), col="#0000FF06", lwd=2)
}
dev.off()

png("hw10q2plot2.png", w=6*400, h=4*400, res=400)
plot( y=residuals(lme1, level=1, type="pearson")^2, x=fitted(lme1, level=1), type="n", ylim=c(1E-4
xlim=c(100,400), log="y", xlab="Level 1 fitted values", ylab="Level 1 Pearson residuals")
for(j in 1:101){
points( x=mu[,j], y=(dd2$y-mu[,j])^2/chain1.short[j,"sigY"]^2, col="#FF000003", pch=19)
lines(lowess( x=mu[,j], y=(dd2$y-mu[,j])^2/chain1.short[j,"sigY"]^2, iter=0), col="#0000FF06", lwd
}
dev.off()

## b) - leave-one-out, with some INLA issues

hyperprior <- list(theta=list(prior="loggamma",param=c(0.1, 0.1), initial=0.0007))
fixedprior <- list( mean.intercept=0, prec.intercept=.0001,
   mean=0, prec=.0001 )
formula1 <- y ~ trt*time + f(id,model="iid",hyper=hyperprior)
familyprior <- list(hyper = list(prec = list(prior = "loggamma", param = c(0.1, 0.1), initial=0.00

inla.many <- sapply(c(4,14,17), function(i){
inlai <- inla(formula1, family="gaussian", data=subset(dd2,id!=i),
control.fixed= fixedprior, control.family=familyprior)
inlai.h <- inla.hyperpar(inlai, h=0.001)
list(inlai.h$summary.fixed, inlai.h$summary.hyperpar)
}, simplify=FALSE)

alarm()
save.image("hw10work.Rdata")
#load("hw10work.Rdata")

loo.ests <- matrix(NA, 103, 6)
for(i in 1:103){
loo.ests[i,1:4] <- inla.many[[i]][[1]][,4]
loo.ests[i,5:6] <- 1/sqrt(inla.many[[i]][[2]][,4])
}
loo.ests <- as.data.frame(loo.ests)
names(loo.ests) <- c("beta0","beta1","beta2","beta3","sigY","sigb")

library("MASS")
parcoord(loo.ests, var.label=TRUE)
```

```
# a few INLA problems
# - MCMC more reliable for this, albeit slower
useme <- loo.ests[,6]<40 & loo.ests[,5]<40

parcoord(loo.ests[useme,], var.label=TRUE)
int.out <- signif( rbind( inla1$summary.fixed[,c(4,3,5)],
1/sqrt(inla1$summary.hyperpar)[,c(4,3,5)] ) , 3)
mtext( at=1:6, side=1, line=2, int.out[,3])
mtext( at=1:6, side=1, line=3, int.out[,2])

##
## Q4
##

rm(list=ls())
library("lme4")
library("gee")

n <- 5000
sigB <- 1
sigY <- 0.5
beta0 <- 1
beta1 <- -1
beta2 <- 0.5

gen.one.miss <- function(){
bi <- rnorm(n, 0, sigB)
x <- rbinom(n, 1, 0.5)
dd <- data.frame(
id=1:n,
x=x,
y1=rnorm(n, beta0 + bi + beta1*1 + beta2*x, sigY),
y2=rnorm(n, beta0 + bi + beta1*2 + beta2*x, sigY),
y3=rnorm(n, beta0 + bi + beta1*3 + beta2*x, sigY)
)
dd$y2[dd$y1< -1] <- NA
dd$y3[dd$y1< -1] <- NA
dd$y3[dd$y2< -1] <- NA
dd.l <- reshape(dd, idvar="id", varying=3:5, sep="", timevar="t",direction="long")
dd.l <- dd.l[order(dd.l$id, dd.l$t),]
dd.l
}

set.seed(5)
bigB <- 1000
do.many <- replicate(bigB, {
ddx <- gen.one.miss()
lmerx <- lmer(y~t+x +(1|id), data=ddx, REML=FALSE)
geexE <- gee(y~t+x, id=id, data=ddx, corstr="exchangeable")
vv <- as.data.frame(VarCorr(lmerx))
alpha <- c(geexE$working.cor[1,2],vv[1,4]/(vv[1,4]+ vv[2,4]))
```

```
c(fixef(lmerx), coef(geexE),
  sqrt(diag(vcov(lmerx))), sqrt(diag(geexE$robust.variance)),
  alpha)
})
alarm()

rowMeans( ((do.many[c(1,4),] - beta0)/do.many[c(7,10),])^2 < qchisq(0.95, df=1) )
rowMeans( ((do.many[c(1,4)+1,] - beta1)/do.many[c(7,10)+1,])^2 < qchisq(0.95, df=1) )
rowMeans( ((do.many[c(1,4)+2,] - beta2)/do.many[c(7,10)+2,])^2 < qchisq(0.95, df=1) )

matplot(do.many, type="l")

# 1:3 MLEs
# 4:6 GEE ests
# 7:9 MLE SEs
#10:12 GEE SEs
# 13 GEE alpha
# 14 LMM ICC

png("hw10q4plot1.png", w=6*400, h=5*400, res=400)
par(mfrow=c(2,1))
par(mar=c(4,5,0.5,0)+0.1)
plot( x=do.many[5,], y=do.many[11,], xlim=c(-1,-0.95), ylim=c(0,0.025), asp=1,
xlab=expression(hat(beta)["1 GEE"]), ylab=expression("Est SE("*hat(beta)["1 GEE"]*")"),
pch=19, col="#FF00FF03")
abline( v=-1 )
abline(h= sd(do.many[5,]))

plot( x=do.many[2,], y=do.many[8,], xlim=c(-1.025,-0.975), ylim=c(0,0.025), asp=1,
xlab=expression(hat(beta)["1 MLE"]), ylab=expression("Est SE("*hat(beta)["1 MLE"]*")"),
pch=19, col="#FF00FF03")
abline( v=-1 )
abline(h= sd(do.many[2,]))
dev.off()

png("hw10q4plot2.png", w=5*400, h=4*400, res=400)
par(mar=c(4,5,0.5,0)+0.1)
plot( x=do.many[13,], y=do.many[14,],
xlab=expression(hat(alpha)["GEE"]), ylab=expression(hat("ICC")["MLE"]),
pch=19, col="#FF882233", xlim=range(do.many[13:14,]), ylim=range(do.many[13:14,]))
abline(0,1, lty=2)
dev.off()

##
## Q5
##

rm(list=ls())
library("nlme")
library("lme4")
expit <- function(x){exp(x)/(1+exp(x))}
```

```
do.one <- function(n=1000, ni=3, p, theta0, theta1, beta0, beta1=0, beta2, sigb=0.5){
zi   <- rbinom(n, 1, p)
xi   <- rbinom(n, 2, expit(theta0 + theta1*zi))
bi   <- rnorm(n, 0, sigb)
dd   <- data.frame(id=rep(1:n, each=ni))
dd$z <- zi[dd$id]
dd$x <- xi[dd$id]
dd$d <- bi[dd$id]
dd$y <- with(dd, rbinom(n*ni, 1, expit(beta0 + beta1*x + beta2*z)))

lme1 <- lme(y~x+z, random=~1|id, data=dd)
glmer1 <- glmer( y~x+z + (1|id), data=dd, family=binomial)
c( fixef(lme1)[2], sqrt(vcov(lme1)[2,2]), fixef(glmer1)[2], sqrt(vcov(glmer1)[2,2]) )
}

expit(-2.5+c(0,1.5))
expit(-2 + c(0, 1.5))
expit(-0.75 + c(0, 1.5))

## a) and b)

set.seed(4)
do.many1 <- replicate(1000,
tryCatch( do.one(1000, 3, p=1/4, theta0=-2, theta1=1.5,
                      beta0=-2.5, beta1=0, beta2=1.5, sigb=0.1),
error=function(e){ c(NA,NA,NA,NA) } )
)
alarm()
save.image("hw10q5.Rdata")

z2l <- (do.many1[1,]/do.many1[2,])^2
z2g <- (do.many1[3,]/do.many1[4,])^2

z2l <- z2l[complete.cases(z2l)]
z2g <- z2g[complete.cases(z2g)]
bigB <- length(z2l)

pdf("hw10q5fig1.pdf", w=6, h=4)
par(mar=c(4,5,0,0)+0.2)
plot(  y=sort(z2l), x=qchisq(ppoints(bigB), df=1), col=1,
xlab=expression("Expected: Quantiles of "*chi[1]^2),
ylab=expression("Observed: Ordered Z"^2*" statistics"), type="n",
xlim=c(0,qchisq(ppoints(bigB)[bigB-100], df=1)), ylim=c(0,3.5) )
polygon( x=c(
qchisq(ppoints(bigB), df=1),
rev(qchisq(ppoints(bigB), df=1))
),
y=c(
qchisq(qbeta(0.025, 1:bigB, bigB:1), df=1),
qchisq(qbeta(0.975, bigB:1, 1:bigB), df=1) ),
```

```
density=NA, col="gray70" )
points(y=sort(z2l), x=qchisq(ppoints(bigB), df=1), col=4)
points(y=sort(z2g), x=qchisq(ppoints(bigB), df=1), col=2)
abline(0,1,lty=2)
legend("topleft", bty="n", pch=1, col=c(4,2), c("LMM Wald tests","GLMM Wald tests"))
dev.off()

## c) - rinse and repeat for the other settings

set.seed(5)
do.many2 <- replicate(1000,
tryCatch( do.one(1000, 3, p=1/4, theta0=-2, theta1=1.5,
                      beta0=-0.75, beta1=0, beta2=1.5, sigb=0.1),
error=function(e){ c(NA,NA,NA,NA) } )
)
alarm()
save.image("hw10q5b.Rdata")

z2lb <- (do.many2[1,]/do.many2[2,])^2
z2gb <- (do.many2[3,]/do.many2[4,])^2

z2lb <- z2lb[complete.cases(z2lb)]
z2gb <- z2gb[complete.cases(z2gb)]
bigB <- length(z2lb)

pdf("hw10q5fig2.pdf", w=6, h=4)
par(mar=c(4,5,0,0)+0.2)
plot(  y=sort(z2lb), x=qchisq(ppoints(bigB), df=1), col=1,
xlab=expression("Expected: Quantiles of "*chi[1]^2),
ylab=expression("Observed: Ordered Z"^2*" statistics"), type="n",
xlim=c(0,qchisq(ppoints(bigB)[bigB-100], df=1)), ylim=c(0,3.5) )
polygon( x=c(
qchisq(ppoints(bigB), df=1),
rev(qchisq(ppoints(bigB), df=1))
),
y=c(
qchisq(qbeta(0.025, 1:bigB, bigB:1), df=1),
qchisq(qbeta(0.975, bigB:1, 1:bigB), df=1) ),
density=NA, col="gray70" )
points(y=sort(z2lb), x=qchisq(ppoints(bigB), df=1), col=4)
points(y=sort(z2gb), x=qchisq(ppoints(bigB), df=1), col=2)
abline(0,1,lty=2)
legend("topleft", bty="n", pch=1, col=c(4,2), c("LMM Wald tests","GLMM Wald tests"))
dev.off()
```

WinBUGS code for Q2

```
model{
for(i in 1:N){
y[i] ~ dnorm(mu[i], tauY)
mu[i] <- beta0 + b[id[i]] + beta1*trt[i] + beta2*time[i] + beta3*trt[i]*time[i]
}
for(j in 1:n){
   b[j]~dnorm(0, taub)
}
   beta0~dnorm(0,.0001)
   beta1~dnorm(0,.0001)
   beta2~dnorm(0,.0001)
   beta3~dnorm(0,.0001)
   taub~dgamma(0.1, 0.1)
   tauY~dgamma(0.1, 0.1)

   sigb <- 1/sqrt(taub)
   sigY <- 1/sqrt(tauY)
}
```