The code below provides another way to visualize the difference between marginal and conditional logistic regression models for clustered data . The basic idea is that both models for a binary outcome are valid, but they provide estimates for different quantities.

Code Chunks - When marginal and conditional logistic model estimates diverge

Say we have an intervention that is assigned at a group or cluster level but the outcome is measured at an individual level (e.g. students in different schools, eyes on different individuals). And, say this outcome is binary; that is, something happens, or it doesn’t. (This is important, because none of this is true if the outcome is continuous and close to normally distributed.) If we want to measure the *effect* of the intervention - perhaps the risk difference, risk ratio, or odds ratio - it can really matter if we are interested in the *marginal* effect or the *conditional* effect, because they likely won’t be the same.

My aim is to show this through a couple of data simulations that allow us to see this visually.

**First example**

In the first scenario, I am going to use a *causal inference* framework that uses the idea that everyone has a potential outcome under one exposure (such as an intervention of some sort), and another potential outcome under a different exposure (such as treatment as usual or control). (I may discuss potential outcomes and causal inference in more detail in the future.) The potential outcome can be written with a superscript, lie Y^0*Y*0 or Y^1*Y*1.

To generate the data, I will use this simple model for each potential outcome:log\left[\frac{P(Y^0\_{ij})}{1-P(Y^0\_{ij})}\right] = \gamma + \alpha\_i*log*[1−*P*(*Yij*0​)*P*(*Yij*0​)​]=*γ*+*αi*​

andlog\left[\frac{P(Y^1\_{ij})}{1-P(Y^1\_{ij})}\right] = \gamma + \alpha\_i + \delta.*log*[1−*P*(*Yij*1​)*P*(*Yij*1​)​]=*γ*+*αi*​+*δ*.\delta*δ* is the treatment effect and is constant across the clusters, on the log-odds scale. \alpha\_i*αi*​ is the cluster specific effect for cluster i*i*. Y^a\_{ij}*Yija*​ is the potential outcome for individual j*j* under exposure a*a*.

Now let’s generate some data and look at it:

# Define data

def1 <- defData(varname = "clustEff", formula = 0, variance = 2,

id = "cID")

def1 <- defData(def1, varname = "nInd", formula = 10000,

dist = "noZeroPoisson")

def2 <- defDataAdd(varname = "Y0", formula = "-1 + clustEff",

dist = "binary", link = "logit")

def2 <- defDataAdd(def2, varname = "Y1",

formula = "-1 + clustEff + 2",

dist = "binary", link = "logit")

options(width = 80)

def1

## varname formula variance dist link

## 1: clustEff 0 2 normal identity

## 2: nInd 10000 0 noZeroPoisson identity

def2

## varname formula variance dist link

## 1: Y0 -1 + clustEff 0 binary logit

## 2: Y1 -1 + clustEff + 2 0 binary logit

# Generate cluster level data

set.seed(123)

dtC <- genData(n = 100, def1)

# Generate individual level data

dt <- genCluster(dtClust = dtC, cLevelVar = "cID", numIndsVar = "nInd",

level1ID = "id")

dt <- addColumns(def2, dt)

Since we have repeated measurements for each cluster (the two potential outcomes), we can transform this into a “longitudinal” data set, though the periods are not time but different exposures.

dtLong <- addPeriods(dtName = dt, idvars = c("id","cID"),

nPeriods = 2,timevars = c("Y0","Y1"),

timevarName = "Y"

)

When we look at the data visually, we get a hint that the marginal (or average) effect might not be the same as the conditional (cluster-specific) effects.

# Calculate average potential outcomes by exposure (which is called period)

dtMean <- dtLong[, .(Y = mean(Y)), keyby = .(period, cID)] # conditional mean

dtMMean <- dtLong[, .(Y = mean(Y)), keyby = .(period)] # marginal mean

dtMMean[, cID := 999]

ggplot(data = dtMean, aes(x=factor(period), y = Y, group= cID)) +

# geom\_jitter(width= .25, color = "grey75") +

geom\_line(color = "grey75", position=position\_jitter(w=0.02, h=0.02)) +

geom\_point(data=dtMMean) +

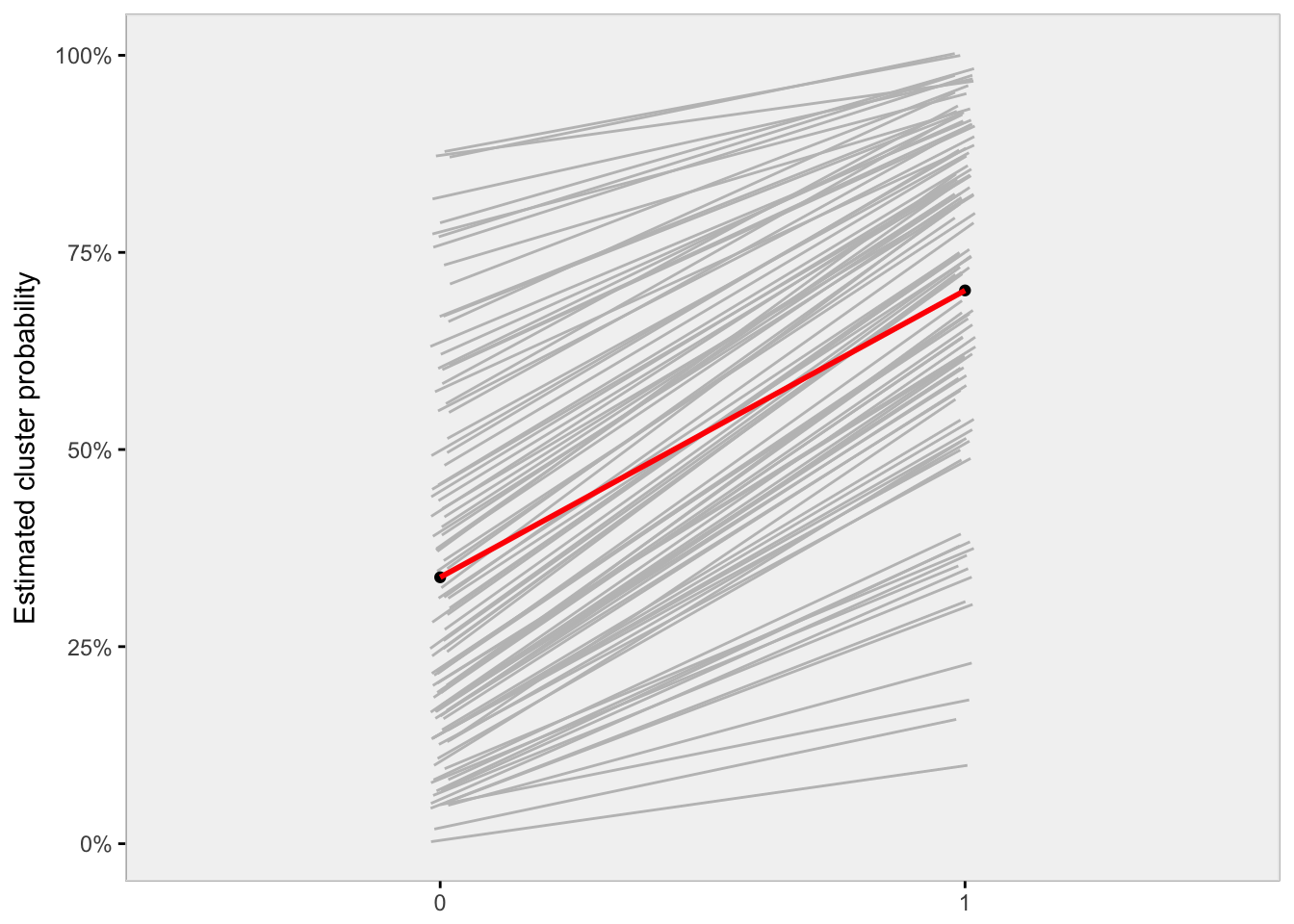
geom\_line(data=dtMMean, size = 1, color = "red") +

ylab("Estimated cluster probability") +

scale\_y\_continuous(labels = scales::percent) +

theme(axis.title.x = element\_blank()) +

my\_theme()



Looking at the plot, we see that the slopes of the grey lines - each representing the change in probability as a result of the exposure for each cluster - vary quite a bit. When the probability without exposure (Y^0*Y*0) is particularly low or high, the absolute effect of the intervention is small (the slope is minimal). The slope or absolute effect increases when the starting probability is closer to 50%. The red line represents the averages of Y^0*Y*0 and Y^1*Y*1 across all individuals in all clusters. There is no reason to believe that the average slope of the grey lines is the same as the slope of the red line, which is slope of the averages. We will see that more clearly with the next data generation scenario.

Finally, if we look at cluster-specific effects of exposure, we see that on the risk difference scale (difference in probabilities), there is much variation, but on the log-odds ratio scale there is almost no variation. This is as it should be, because on the log-odds scale (which is how we generated the data), the difference between exposure and non-exposure is additive. On the probability scale, the difference is multiplicative. Here are some estimated differences for a sample of clusters:

dtChange <- dt[, .(Y0 = mean(Y0), Y1 = mean(Y1)), keyby = cID]

dtChange[, riskdiff := round(Y1 - Y0, 2)]

dtChange[, loratio := round( log( (Y1 / (1-Y1)) / (Y0 / (1-Y0) )), 2)]

dtChange[sample(1:100, 10, replace = F),

.(Y0 = round(Y0,2), Y1 = round(Y1,2), riskdiff, loratio),

keyby=cID]

## cID Y0 Y1 riskdiff loratio

## 1: 18 0.02 0.14 0.12 1.95

## 2: 19 0.50 0.89 0.39 2.06

## 3: 22 0.22 0.67 0.45 1.97

## 4: 24 0.12 0.49 0.37 1.94

## 5: 30 0.69 0.94 0.26 2.00

## 6: 31 0.40 0.83 0.42 1.95

## 7: 34 0.56 0.90 0.35 1.99

## 8: 38 0.26 0.72 0.46 2.01

## 9: 72 0.01 0.09 0.08 1.97

## 10: 99 0.22 0.66 0.44 1.93

**Second example**

This time around, we will add an additional individual level covariate that will help us visualize the difference a bit more clearly. Let us say that *age* is positively associated with increased probability in the outcome. (In this case, we measured age and then normalized it so that the mean age in the sample is 0.) And this time around, we are not going to use potential outcomes, but will randomly assign clusters to an intervention or treatment group.

This is the data generating model and the code:

log\left[\frac{P(Y\_{ij})}{1-P(Y\_{ij})}\right] = \gamma + \alpha\_j + \beta\_1\*Trt\_j + \beta\_2\*Age\_i*log*[1−*P*(*Yij*​)*P*(*Yij*​)​]=*γ*+*αj*​+*β*1​∗*Trtj*​+*β*2​∗*Agei*​

def1 <- defData(varname = "clustEff", formula = 0, variance = 2, id = "cID")

def1 <- defData(def1, varname = "nInd", formula = 100, dist = "noZeroPoisson")

# Each individual now has a measured age

def2 <- defDataAdd(varname = "age", formula = 0, variance = 2)

def2 <- defDataAdd(def2, varname = "Y",

formula = "-4 + clustEff + 2\*trt + 2\*age",

dist = "binary", link = "logit")

# Generate cluster level data

dtC <- genData(200, def1)

dtC <- trtAssign(dtC, grpName = "trt") #

# Generate individual level data

dt <- genCluster(dtClust = dtC, cLevelVar = "cID", numIndsVar = "nInd",

level1ID = "id")

dt <- addColumns(def2, dt)

By fitting a conditional model (generalized linear mixed effects model) and a marginal model (we should fit a generalized estimating equation model to get the proper standard error estimates, but will estimate a generalized linear model, because the GEE model does not have a “predict” option in R; the point estimates for both marginal models should be quite close), we can see that indeed the conditional and marginal averages can be quite different.

glmerFit1 <- glmer(Y ~ trt + age + (1 | cID), data = dt, family="binomial")

glmFit1 <- glm(Y ~ trt + age, family = binomial, data = dt)

## Intercept Trt Age

## conditional model -3.82 1.99 2.01

## marginal model -2.97 1.60 1.54

Now, we’d like to visualize how the conditional and marginal treatment effects diverge. We can use the model estimates from the conditional model to predict probabilities for each cluster, age, and treatment group. (These will appear as grey lines in the plots below). We can also predict marginal probabilities from the marginal model based on age and treatment group while ignoring cluster. (These marginal estimates appear as red lines.) Finally, we can predict probability of outcomes for the conditional model also based on age and treatment group alone, but fixed at a mythical cluster whose random effect is 0. (These “average” conditional estimates appear as black lines.)

newCond <- expand.grid(cID = unique(dt$cID), age=seq(-4, 4, by =.1))

newCond0 <- data.table(trt = 0, newCond)

newCond1 <- data.table(trt = 1, newCond)

newMarg0 <- data.table(trt = 0, age = seq(-4, 4, by = .1))

newMarg1 <- data.table(trt = 1, age = seq(-4, 4, by = .1))

newCond0[, pCond0 := predict(glmerFit1, newCond0, type = "response")]

newCond1[, pCond1 := predict(glmerFit1, newCond1, type = "response")]

newMarg0[, pMarg0 := predict(glmFit1, newMarg0, type = "response")]

newMarg0[, pCAvg0 := predict(glmerFit1, newMarg0[,c(1,2)],

re.form = NA, type="response")]

newMarg1[, pMarg1 := predict(glmFit1, newMarg1, type = "response")]

newMarg1[, pCAvg1 := predict(glmerFit1, newMarg1[,c(1,2)],

re.form = NA, type="response")]

dtAvg <- data.table(age = newMarg1$age,

avgMarg = newMarg1$pMarg1 - newMarg0$pMarg0,

avgCond = newMarg1$pCAvg1 - newMarg0$pCAvg0

)

p1 <- ggplot(aes(x = age, y = pCond1), data=newCond1) +

geom\_line(color="grey", aes(group = cID)) +

geom\_line(data=newMarg1, aes(x = age, y = pMarg1), color = "red", size = 1) +

geom\_line(data=newMarg1, aes(x = age, y = pCAvg1), color = "black", size = 1) +

ggtitle("Treatment group") +

xlab("Age") +

ylab("Probability") +

my\_theme()

p0 <- ggplot(aes(x = age, y = pCond0), data=newCond0) +

geom\_line(color="grey", aes(group = cID)) +

geom\_line(data=newMarg0, aes(x = age, y = pMarg0), color = "red", size = 1) +

geom\_line(data=newMarg0, aes(x = age, y = pCAvg0), color = "black", size = 1) +

ggtitle("Control group") +

xlab("Age") +

ylab("Probability") +

my\_theme()

pdiff <- ggplot(data = dtAvg) +

geom\_line(aes(x = age, y = avgMarg), color = "red", size = 1) +

geom\_line(aes(x = age, y = avgCond), color = "black", size = 1) +

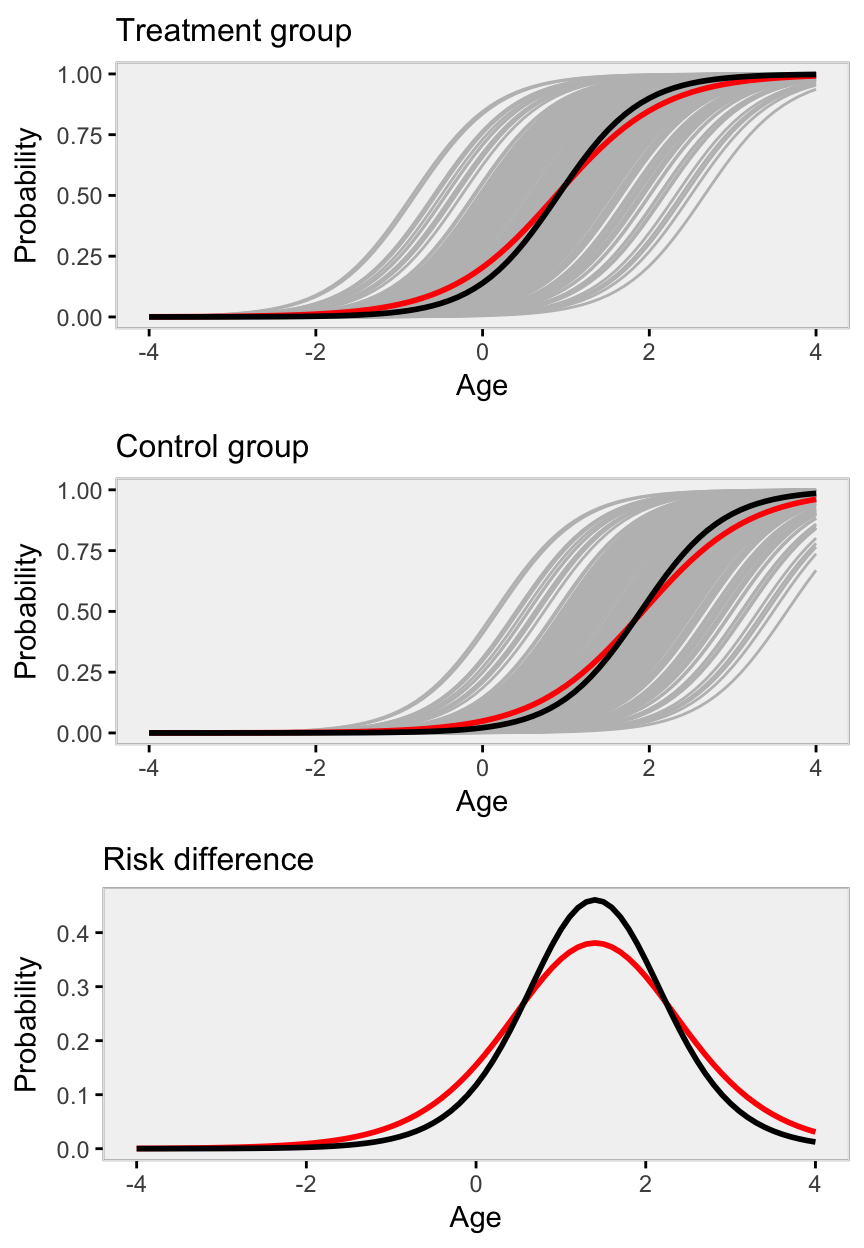
ggtitle("Risk difference") +

xlab("Age") +

ylab("Probability") +

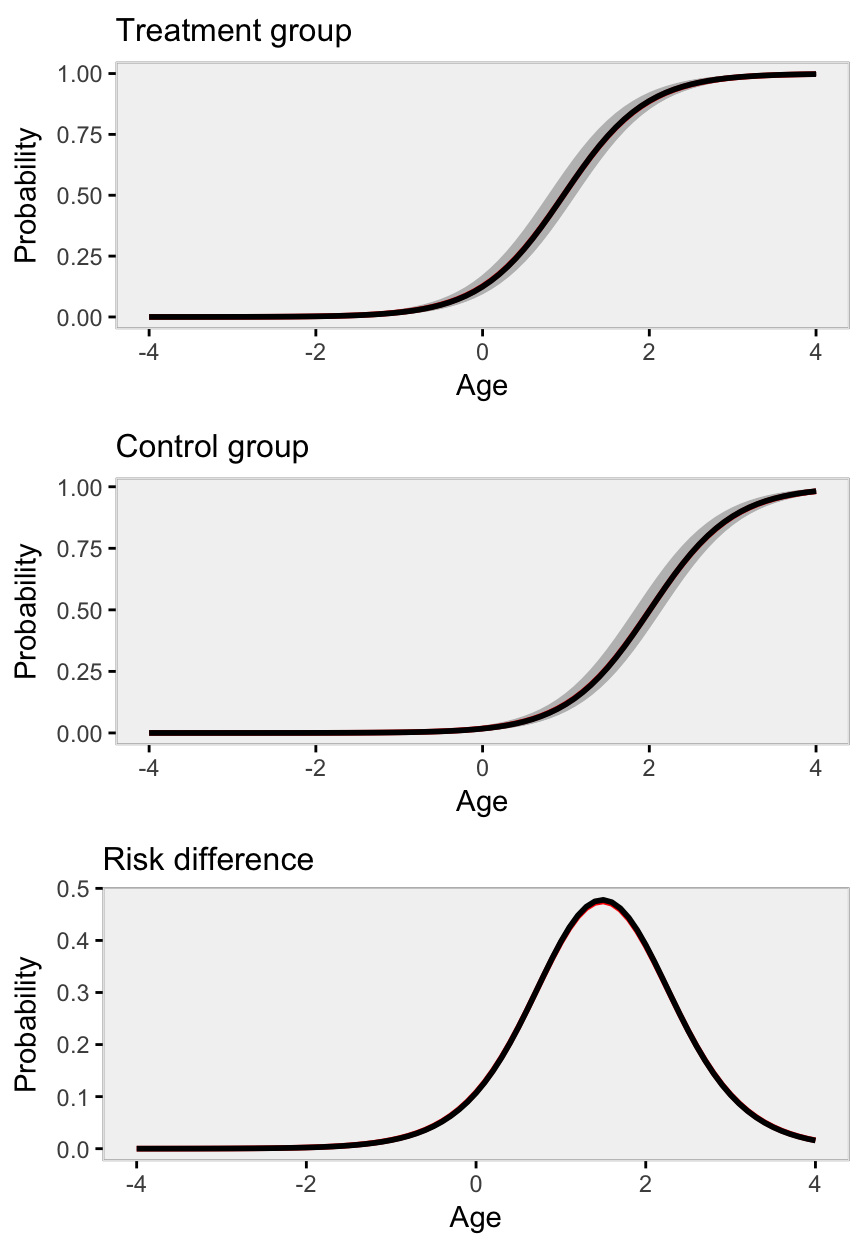
my\_theme()

grid.arrange(p1, p0, pdiff)



We see pretty clearly across all ages that the marginal and conditional estimates of average treatment differences differ quite dramatically.

Below are point estimates and plots for data generated with very little variance across clusters, that is var(\alpha\_i)*var*(*αi*​) is close to 0. (We change this in the simulation by setting def1 <- defData(varname = "clustEff", formula = 0, variance = 0.05, id = "cID").)



The black lines obscure the red - the marginal model estimate is not much different from the conditional model estimate - because the variance across clusters is negligible.

The marginal model is estimated using a generalized estimating equation (GEE) model (here using function geeglm in package geepack). If the intervention is binary, the intervention effect (log-odds ratio) is interpreted as the average effect across all individuals regardless of the group or cluster they might belong to. (This estimate is sensitive to the relative sizes of the clusters.)

The conditional model is estimated using a random mixed effect generalized linear model (using function glmer in package lme4), and provides the log-odds ratio conditional on the cluster. (The estimate is not as sensitive to the relative sizes of the clusters since it is essentially providing a within-cluster effect.)

As the variation across clusters increases, so does the discrepancy between the conditional and marginal models. Using a generalized linear model that ignores clustering altogether will provide the correct (marginal) point estimate, but will underestimate the underlying variance (and standard errors) as long as there is between cluster variation. If there is no between cluster variation, the GLM model should be fine.

**Simulation**

To start, here is a function that uses simstudy to define and generate a data set of individuals that are clustered in groups. A key argument passed to this function is the across cluster variation.

library(lme4)

library(geepack)

library(broom)

genFunc <- function(nClusters, effVar) {

# define the cluster

def1 <- defData(varname = "clustEff", formula = 0,

variance = effVar, id = "cID")

def1 <- defData(def1, varname = "nInd", formula = 40,

dist = "noZeroPoisson")

# define individual level data

def2 <- defDataAdd(varname = "Y", formula = "-2 + 2\*grp + clustEff",

dist = "binary", link = "logit")

# generate cluster level data

dtC <- genData(nClusters, def1)

dtC <- trtAssign(dtC, grpName = "grp")

# generate individual level data

dt <- genCluster(dtClust = dtC, cLevelVar = "cID", numIndsVar = "nInd",

level1ID = "id")

dt <- addColumns(def2, dt)

return(dt)

}

A plot of the average site level outcome from data generated with across site variance of 1 (on the log-odds scale) shows the treatment effect:

set.seed(123)

dt <- genFunc(100, 1)

dt

## cID grp clustEff nInd id Y

## 1: 1 0 -0.5604756 35 1 1

## 2: 1 0 -0.5604756 35 2 0

## 3: 1 0 -0.5604756 35 3 0

## 4: 1 0 -0.5604756 35 4 0

## 5: 1 0 -0.5604756 35 5 0

## ---

## 3968: 100 1 -1.0264209 45 3968 0

## 3969: 100 1 -1.0264209 45 3969 0

## 3970: 100 1 -1.0264209 45 3970 1

## 3971: 100 1 -1.0264209 45 3971 0

## 3972: 100 1 -1.0264209 45 3972 0

dplot <- dt[, mean(Y), keyby = .(grp, cID)]

davg <- dt[, mean(Y)]

ggplot(data = dplot, aes(x=factor(grp), y = V1)) +

geom\_jitter(aes(color=factor(grp)), width = .10) +

theme\_ksg("grey95") +

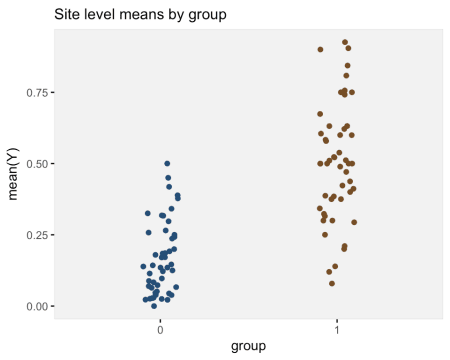
xlab("group") +

ylab("mean(Y)") +

theme(legend.position = "none") +

ggtitle("Site level means by group") +

scale\_color\_manual(values = c("#264e76", "#764e26"))



**Model fits**

First, the conditional model estimates a log-odds ratio of 1.89, close to the actual log-odds ratio of 2.0.

glmerFit <- glmer(Y ~ grp + (1 | cID), data = dt, family="binomial")

tidy(glmerFit)

## term estimate std.error statistic p.value group

## 1 (Intercept) -1.8764913 0.1468104 -12.781729 2.074076e-37 fixed

## 2 grp 1.8936999 0.2010359 9.419711 4.523292e-21 fixed

## 3 sd\_(Intercept).cID 0.9038166 NA NA NA cID

The marginal model that takes into account clustering yields an estimate of 1.63. This model is not wrong, just estimating a different quantity:

geeFit <- geeglm(Y ~ grp, family = binomial, data = dt,

corstr = "exchangeable", id = dt$cID)

tidy(geeFit)

## term estimate std.error statistic p.value

## 1 (Intercept) -1.620073 0.1303681 154.42809 0

## 2 grp 1.628075 0.1740666 87.48182 0

The marginal model that ignores clustering also estimates a log-odds ratio, 1.67, but the standard error estimate is much smaller than in the previous model (0.076 vs. 0.174). We could say that this model is not appropriate given the clustering of individuals:

glmFit <- glm(Y ~ grp, data = dt, family="binomial")

tidy(glmFit)

## term estimate std.error statistic p.value

## 1 (Intercept) -1.639743 0.0606130 -27.05267 3.553136e-161

## 2 grp 1.668143 0.0755165 22.08978 3.963373e-108

**Multiple replications**

With multiple replications (in this case 100), we can see how each model performs under different across cluster variance assumptions. I have written two functions (that are shown at the end in the appendix) to generate multiple datasets and create a plot. The plot shows (1) the average point estimate across all the replications in black, (2) the true standard deviation of all the point estimates across all replications in blue, (3) the average estimate of the standard errors in orange.

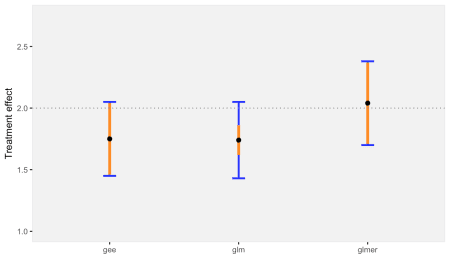
In the first case, the variability across sites is highest. The discrepancy between the marginal and conditional models is relatively large, but both the GEE and mixed effects models estimate the standard errors correctly (the orange line overlaps perfectly with blue line). The generalized linear model, however, provides a biased estimate of the standard error – the orange line does not cover the blue line:

set.seed(235)

res1.00 <- iterFunc(40, 1.00, 100)

s1 <- sumFunc(res1.00)

s1$p

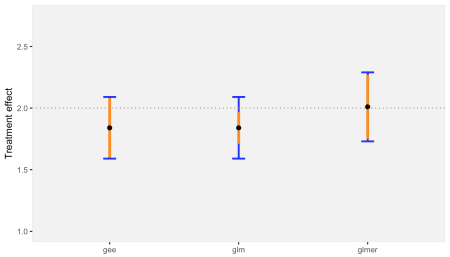


When the across cluster variation is reduced, the discrepancy between the marginal and conditional models is reduced, as is the bias of standard error estimate for the GLM model:

res0.50 <- iterFunc(40, 0.50, 100)

s2 <- sumFunc(res0.50)

s2$p

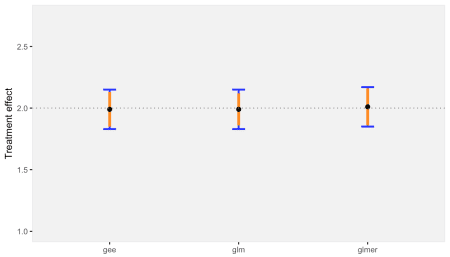


Finally, when there is negligible variation across sites, the conditional and marginal models are pretty much one and the same. And even the GLM model that ignores clustering is unbiased (which makes sense, since there really is no clustering):

res0.05 <- iterFunc(40, 0.05, 100)

s3 <- sumFunc(res0.05)

s3$p



**Appendix**

Here are the two functions that generated the the replications and created the plots shown above.

iterFunc <- function(nClusters, effVar, iters = 250) {

results <- list()

for (i in 1:iters) {

dt <- genFunc(nClusters, effVar)

glmerFit <- glmer(Y ~ grp + (1 | cID), data = dt, family="binomial")

glmFit <- glm(Y ~ grp, data = dt, family="binomial")

geeFit <- geeglm(Y ~ grp, family = binomial, data = dt,

corstr = "exchangeable", id = dt$cID)

res <- unlist(c(coef(summary(glmerFit))[2,1:2],

coef(summary(glmFit))[2,1:2],

as.vector(coef(summary(geeFit))[2,1:2])))

results[[i]] <- data.table(t(res))

}

return(rbindlist(results))

}

sumFunc <- function(dtRes, precision = 2) {

setnames(dtRes, c("estGlmer", "sdGlmer",

"estGlm","sdGlm",

"estGEE", "sdGEE"))

meanEst <- round(apply(dtRes[, c(1, 3, 5)], 2, mean), precision)

estSd <- round(sqrt(apply(dtRes[, c(2, 4, 5)]^2, 2, mean)), precision)

sdEst <- round(apply(dtRes[, c(1, 3, 5)], 2, sd), precision)

x <- data.table(rbind(c(meanEst[1], estSd[1], sdEst[1]),

c(meanEst[2], estSd[2], sdEst[2]),

c(meanEst[3], estSd[3], sdEst[3])

))

setnames(x, c("estMean","estSD","sd"))

x[, method := c("glmer","glm","gee")]

p <- ggplot(data = x, aes(x = method, y = estMean)) +

geom\_errorbar(aes(ymin = estMean - sd, ymax = estMean + sd),

width = 0.1, color = "#2329fe", size = 1) +

geom\_errorbar(aes(ymin = estMean - estSD, ymax = estMean + estSD),

width = 0.0, color = "#fe8b23", size = 1.5) +

geom\_point(size = 2) +

ylim(1,2.75) +

theme\_ksg("grey95") +

geom\_hline(yintercept = 2, lty = 3, color = "grey50") +

theme(axis.title.x = element\_blank()) +

ylab("Treatment effect")

return(list(mean=meanEst, sd=sdEst, p=p))

}