STAT 528 - Advanced Regression Analysis II

GLMM and GEE

Daniel J. Eck
Department of Statistics
University of Illinois

Learning Objectives Today

- GLMM examples
- GEE theory
- GEE examples

We load in necessary packages.

```
library(faraway)
library(tidyverse)
library(ggplot2)
library(MASS)
library(lme4)
library(INLA)
library(glmm)
library(parallel)
```

In this example, we have data from a clinical trial of 59 epileptics.

For a baseline, patients were observed for 8 weeks and the number of seizures recorded. The patients were then randomized to treatment by the drug Progabide (31 patients) or to the placebo group (28 patients).

They were observed for four 2-week periods and the number of seizures recorded. We are interested in determining whether Progabide reduces the rate of seizures.

We first perform some data manipulations and then look at the first few observations:

```
data(epilepsy, package="faraway")
epilepsy$period <- rep(0:4, 59)
epilepsy$drug <- factor(c("placebo","treatment")[epilepsy$treat+1])
epilepsy$phase <- factor(c("baseline","experiment")[epilepsy$expind +1])
epilepsy %>% filter(id < 2.5) %>% head(3)
```

```
##   seizures id treat expind timeadj age period    drug      phase
## 1       11  1     0      0       8  31      0 placebo   baseline
## 2        5  1     0      1       2  31      1 placebo experiment
## 3        3  1     0      1       2  31      2 placebo experiment
```

The variables are:

- `expind` indicates the baseline phase by 0 and the treatment phase by 1.
- `timeadj` indicates the time phases.

Three new convenience variables are created:

- `period` denotes the separate 2- or 8- week periods
- `drug` records the type of treatment in nonnumeric form
- `phase` indicates the phase of the experiment

We now compute the mean number of seizures per week broken down by the treatment and baseline vs. experimental period.

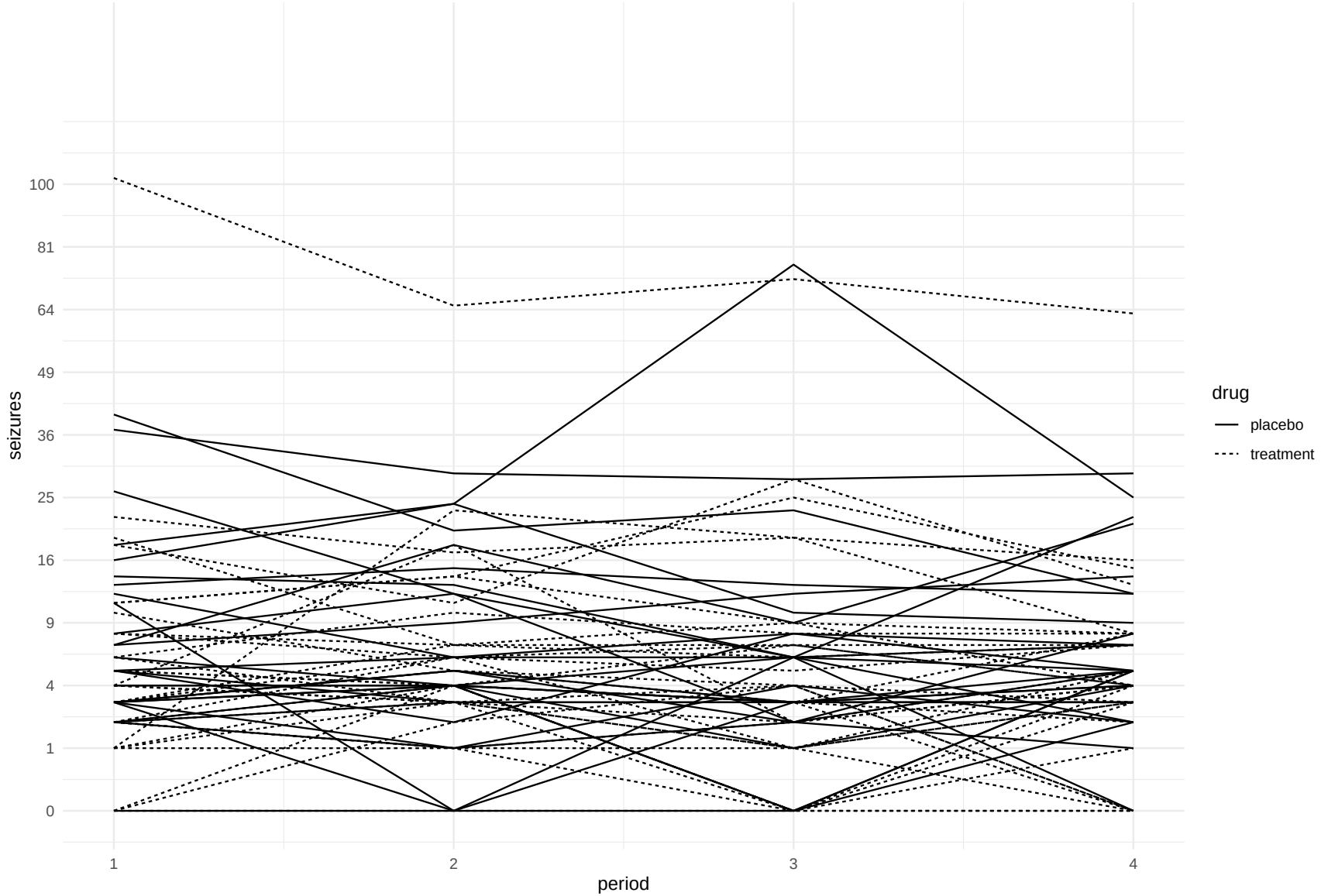
```
epilepsy %>%
  group_by(drug, phase) %>%
  summarise(rate=mean(seizures/timeadj)) %>%
xtabs(formula=rate ~ phase + drug)
```

```
##             drug
## phase         placebo treatment
##   baseline   3.848214  3.955645
##   experiment 4.303571  3.983871
```

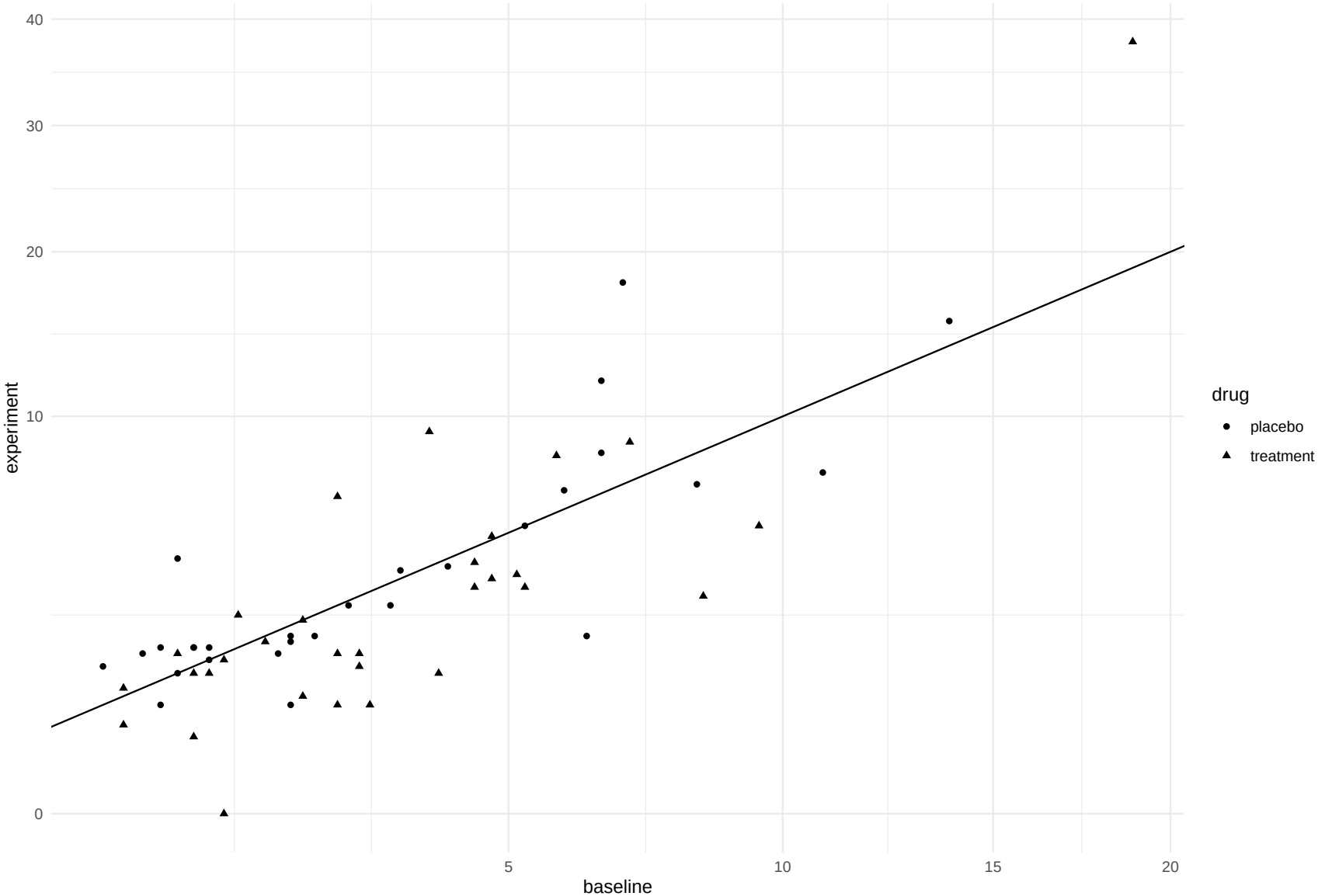
We see that the rate of seizures in the treatment group actually increases during the period in which the drug was taken. The rate of seizures increases even more in the placebo group.

Perhaps some other factor is causing the rate of seizures to increase during the treatment period and the drug is actually having a beneficial effect.

Now we make some plots to show the difference between the treatment and the control. The first plot shows the difference between the two groups during the experimental period only:


seizures
100
81
64
49
36
25
16
9
4
1
0
1
2
3
4
period
drug
placebo
treatment

We now compare the average seizure rate to the baseline for the two groups. The square-root transform is used to stabilize the variance; this is often used with count data.



A treatment effect, if one exists, is not readily apparent. Now we fit GLMM models. Patient #49 is unusual because of the high rate of seizures observed. We exclude it:

```
epilo <- filter(epilepsy, id != 49)
```

Excluding a case should not be taken lightly. For projects where the analyst works with producers of the data, it will be possible to discuss substantive reasons for excluding cases.

It is worth starting with a GLM even though the model is not correct due to the grouping of the observations. We must use an offset to allow for the difference in lengths in the baseline and treatment periods:

$$\log \frac{\mu_i}{t_i} = x_i^T \beta$$

```r
modglm <- glm(seizures ~offset(log(timeadj)) + expind + treat +
  I(expind*treat), family=poisson, data=epilo)
summary(modglm)
```

```
##
## Call:
## glm(formula = seizures ~ offset(log(timeadj)) + expind + treat +
##     I(expind * treat), family = poisson, data = epilo)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.4725  -2.3605  -1.0290   0.9001  14.0104
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        1.34761    0.03406  39.566  < 2e-16 ***
## expind             0.11184    0.04688   2.386    0.017 *
## treat             -0.10682    0.04863  -2.197    0.028 *
## I(expind * treat) -0.30238    0.06971  -4.338 1.44e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 2485.1  on 289  degrees of freedom
## Residual deviance: 2411.5  on 286  degrees of freedom
## AIC: 3449.7
##
## Number of Fisher Scoring iterations: 5
```

The interaction term is the primary parameter of interest. All the subjects were untreated in the baseline. This means that the main effect for treatment does not properly measure the response to treatment because it includes the baseline period.

As we have observed already, we suspect the response may have been different during the baseline time and the active period of the experiment. The interaction term represents the effect of the treatment during the baseline period after adjustment. In the output above we see that this interaction seems highly significant and negative (which is good since we want to reduce seizures).

But this inference is suspect because we have made no allowance for the correlated responses within individuals. The p-value is far smaller than it should be.

PQL methods

```
modpql <- glmmPQL(seizures ~offset(log(timeadj)) + expind + treat +
  I(expind*treat), random = ~1|id, family=poisson, data=epilo)
summary(modpql)
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: epilo
##   AIC BIC logLik
##    NA  NA     NA
##
## Random effects:
##  Formula: ~1 | id
##         (Intercept) Residual
## StdDev:   0.6820012 1.605385
##
## Variance function:
##  Structure: fixed weights
##  Formula: ~invwt
## Fixed effects:  seizures ~ offset(log(timeadj)) + expind + treat + I(expind *      treat)
##                       Value  Std.Error  DF   t-value p-value
## (Intercept)        1.0761832 0.09990514 230 10.772050  0.0000
## expind             0.1125119 0.07412152 230  1.517939  0.1304
## I(expind * treat) -0.3037615 0.10819095 230 -2.807642  0.0054
##  Correlation:
##                   (Intr) expind
## expind            -0.198
## I(expind * treat) -0.014 -0.656
##
## Standardized Within-Group Residuals:
##        Min         Q1        Med         Q3        Max
## -2.2934834 -0.5649468 -0.1492931  0.3224895  6.3123337
##
## Number of Observations: 290
## Number of Groups: 58
```

The parameter estimates from the PQL fit are comparable to the GLM fit. However, the standard errors are larger in the PQL fit as might be expected given that the correlated responses have been allowed for.

As with the binary response example, we still have some doubts about the accuracy of the inference. This is a particular concern when some count responses are small.

Numerical integration

Numerical quadrature can also be used. We use Gauss-Hermite in preference to Laplace as the model random effect structure is simple and so the computation is fast even though we have used the most expensive `nAGQ=25` setting.

```
modgh <- glmer(seizures ~offset(log(timeadj)) + expind + treat +
  I(expind*treat)+ (1|id), nAGQ=25, family=poisson, data=epilo)
```

```
summary(modgh)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 25) [glmerMod]
##  Family: poisson  ( log )
## Formula: seizures ~ offset(log(timeadj)) + expind + treat + I(expind *
##     treat) + (1 | id)
##    Data: epilo
##
##      AIC      BIC   logLik deviance df.resid
##    877.7    896.1   -433.9    867.7      285
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.8724 -0.8482 -0.1722  0.5697  9.8941
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  id     (Intercept) 0.515    0.7176
## Number of obs: 290, groups:  id, 58
##
## Fixed effects:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        1.035998   0.141256   7.334 2.23e-13 ***
## expind             0.111838   0.046877   2.386    0.017 *
## treat             -0.008152   0.196525  -0.041    0.967
## I(expind * treat) -0.302387   0.069714  -4.338 1.44e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) expind treat
## expind      -0.175
## treat       -0.718  0.126
## I(xpnd*trt)  0.118 -0.672 -0.173
```

We see that the interaction effect is significant. Notice that the estimate of this effect has been quite consistent over all the estimation methods so we draw some confidence from this. We have

```
exp(-0.302)
```

```
## [1] 0.7393381
```

So the drug is estimated to reduce the rate of seizures by about 26%. However, the subject SD is more than twice the drug effect of -0.3 at 0.718. This indicates that the expected improvement in the drug is substantially less than the variation between individuals.

Interpretation of the main effect terms is problematic in the presence of an interaction. For example, the treatment effect reported here represents the predicted difference in the response during the baseline period (i.e., `expind=0`).

Since none of the subjects are treated during the baseline period, we are reassured to see that this effect is not significant.

However, this does illustrate the danger in naively presuming that this is the treatment effect.

Bayesian methods

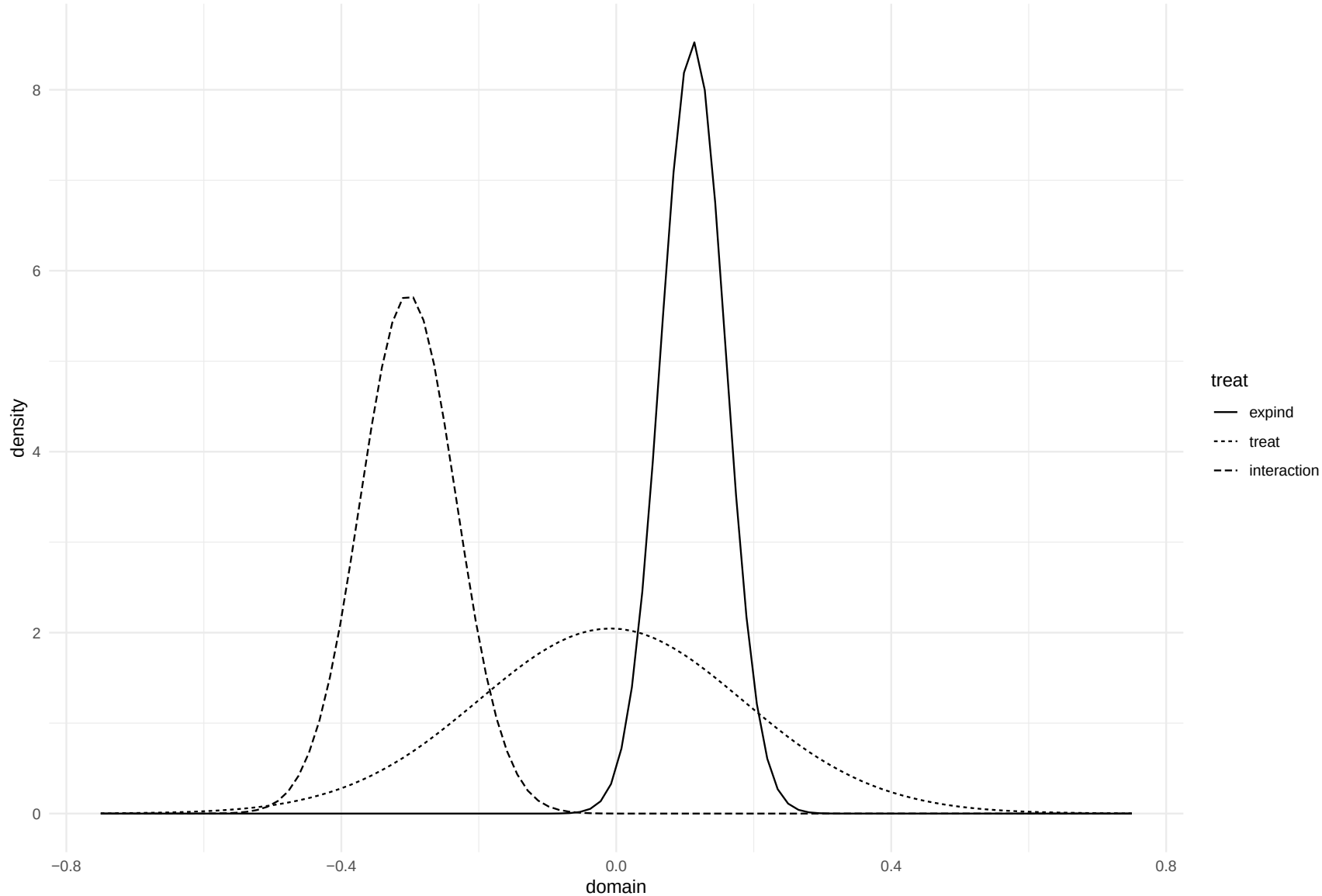
We can also take a Bayesian approach using `INLA`.

```
formula <- seizures ~ offset(log(timeadj)) + expind + treat +
  I(expind*treat) + f(id,model="iid")
result <- inla(formula, family="poisson", data = epilo)
```

We obtain a summary of the posteriors as:

```
sigmaalpha <- inla.tmarginal(function(x) 1/sqrt(x),
  result$marginals.hyperpar$"Precision for id")
restab <- sapply(result$marginals.fixed,
  function(x) inla.zmarginal(x, silent=TRUE))
restab <- cbind(restab,
  inla.zmarginal(sigmaalpha, silent=TRUE))
colnames(restab) = c("mu","expind","treat",
  "interaction","alpha")
data.frame(restab)
```

```
##                    mu     expind        treat interaction      alpha
## mean         1.036115  0.1118965  -0.00825552  -0.3025702  0.7254792
## sd          0.1421983 0.04685179    0.1978247  0.06967626 0.07183047
## quant0.025 0.7550146 0.01995068   -0.3978168  -0.4393086    0.59977
## quant0.25  0.9410767 0.08018646    -0.140758  -0.3497282  0.6746726
## quant0.5    1.036183  0.1117989 -0.008691486  -0.3027154   0.719905
## quant0.75   1.130938  0.1434113     0.123375  -0.2557025  0.7702153
## quant0.975  1.314419  0.2036471     0.380427  -0.1661221  0.8818008
```


8
6
4
2
0
density
−0.8
−0.4
0.0
0.4
0.8
domain
treat
expind
treat
interaction

Monte Carlo likelihood approximation

We can use the `glmm` package to implement the MCLA approach to fitting GLMM models with Poisson responses.

```
epilo$idF <- as.factor(epilo$id)
epilo$seizures <- as.integer(epilo$seizures)
set.seed(13)
clust <- makeCluster(8)
system.time(m1 <- glmm(seizures ~ offset(log(timeadj)) +
  expind + treat + I(expind*treat), random = list(~0+idF),
  family.glmm = poisson.glmm, m = 7e4,
  varcomps.names = c("idF"), cluster = clust, data=epilo))
```

```
##    user  system elapsed
##   6.095   0.653  66.986
```

We obtain summary information. However, the fit is buggy. The Monte Carlo standard error is not returned and the summary table estimates are not depicted.

```
## takes awhile to load
summary(m1)
```

```
##
## Call:
## glmm(fixed = seizures ~ offset(log(timeadj)) + expind + treat +
##     I(expind * treat), random = list(~0 + idF), varcomps.names = c("idF"),
##     data = epilo, family.glmm = poisson.glmm, m = 70000, cluster = clust)
##
##
## Link is: "log"
##
## Fixed Effects:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)               0          0  31.010  < 2e-16 ***
## expind                    0          0 -27.187  < 2e-16 ***
## treat                     0          0   0.018    0.986
## I(expind * treat)         0          0  -4.338 1.44e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Variance Components for Random Effects (P-values are one-tailed):
##     Estimate Std. Error z value Pr(>|z|)/2
## idF        0          0   5.098   1.72e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Monte Carlo standard errors
mcse_glmm <- mcse(m1)
mcse_glmm
```

```
##       (Intercept)             expind              treat I(expind * treat)
##               NaN                NaN                NaN                NaN
##               idF
##               NaN
```

That being said, we can obtain estimates of fixed effects and their standard errors from objects in the `glmm` object.

```
# standard errors
se_glmm <- se(m1)

# table for fixed effects
tab <- cbind(m1$beta, se_glmm[-5], m1$beta/se_glmm[-5])
colnames(tab) <- c("Estimate", "Std. Error", "z value")
round(tab, 3)
```

```
##                   Estimate Std. Error z value
## (Intercept)          3.106      0.100  31.010
## expind              -1.274      0.047 -27.187
## treat                0.003      0.185   0.018
## I(expind * treat)   -0.302      0.070  -4.338
```

We can also obtain estimates of random effect parameters and their standard errors from objects in the `glmm` object.

```
c(m1$nu, se_glmm[5])
```

```
##       idF       idF
## 0.5152511 0.1010674
```

Important parameter estimates are similar to the other fitting techniques which instills some confidence.

Discuss GEE

We will use the geepack package to fit GEEs. We will reanalyze the stability dataset using generalized estimating equations.

```
library(geepack)
data(ctsib)
ctsib$stable <- ifelse(ctsib$CTSIB==1,1,0)
ctsib <- ctsib %>%
  mutate(Age = scale(Age), Height = scale(Height), Weight = scale(Weight))
modgeep <- geeglm(stable ~ Sex + Age + Height + Weight + Surface + Vision,
                  id=Subject, corstr="exchangeable", scale.fix=TRUE,
                  data=ctsib, family=binomial)
```

We have specified the same fixed effects as in the corresponding GLMM earlier. Only simple groups are allowed while nested grouping variables cannot be accommodated easily in this function.

We are required to choose the correlation structure within each group. If we choose no correlation, then the problem reduces to a standard GLM. For this data, compound symmetry is selected as a covariance structure, since it seems reasonable that any pair of observations between subjects has the same correlation (ignoring a learning effect).

Note that compound symmetry is referred to as exchangeable correlation in the `corstr` argument of the `geeglm` fitting function.

Also note that we have chosen to fix ϕ at the default value of 1 to ensure that our analysis is comparable with the GLMM fit. Otherwise, there would not be a strong reason to fix this.

Here is the summary information:

```
summary(modgeep)
```

```
##
## Call:
## geeglm(formula = stable ~ Sex + Age + Height + Weight + Surface +
##     Vision, family = binomial, data = ctsib, id = Subject, corstr = "exchangeable",
##     scale.fix = TRUE)
##
##  Coefficients:
##             Estimate  Std.err   Wald Pr(>|W|)
## (Intercept) -6.16128  1.08020 32.534 1.17e-08 ***
## Sexmale      1.64487  0.90347  3.315   0.0687 .
## Age         -0.07659  0.30521  0.063   0.8019
## Height      -1.06930  0.44398  5.801   0.0160 *
## Weight       0.67199  0.52329  1.649   0.1991
## Surfacenorm  3.91631  0.56682 47.738 4.87e-12 ***
## Visiondome   0.35888  0.40403  0.789   0.3744
## Visionopen   3.17990  0.46063 47.657 5.08e-12 ***
## ---
## 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   0.2185 0.04467
## Number of clusters:   40  Maximum cluster size: 12
```

There is one clear difference with the GLMM output: the estimates for the GEE are about half the size of the GLMM β.

It is expected that the GEE estimates are smaller because GLMMs model the data at the subject or individual level. The correlation between the measurements on the individual is generated by the random effect.

Thus the βs for the GLMM represent the effect on an individual. A GEE models the data at the population level. Here is a good explanation of the difference. The βs for a GEE represent the effect of the predictors averaged across all individuals with the same predictor values. GEEs do not use random effects but model the correlation at the marginal level. This is a major distinction.

We can see that the estimated correlation between observations on the same subject is 0.22 with a standard error of 0.04. This suggests that there is correlation between responses within individuals.

The standard errors are constructed using a sandwich estimator mentioned above. Further motivation for sandwich estimation is described in Section 8.5 of Faraway (2016).

Note that sandwich estimation typically, but not always, leads to standard errors larger than those obtained directly from likelihood calculations.

The testing for vision is not entirely satisfactory since it has three levels meaning two tests—one being highly significant and the other not at all. If we want a single test for the significance of vision, we need to refit the model without vision and make the standard anova-type comparison:

```
modgeep2 <- geeglm(stable ~ Sex + Age + Height + Weight + Surface,
  id =Subject, corstr="exchangeable", scale.fix=TRUE, data=ctsib,
  family=binomial)
anova(modgeep2, modgeep)
```

```
## Analysis of 'Wald statistic' Table
##
## Model 1 stable ~ Sex + Age + Height + Weight + Surface + Vision
## Model 2 stable ~ Sex + Age + Height + Weight + Surface
##   Df   X2 P(>|Chi|)
## 1  2 58.4   2.1e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As expected, we see that vision is strongly significant.

We will now model the epilepsy data using GEEs.

We exclude the 49th case as before (all the same caveats apply).

An autoregressive AR(1) model for the correlation structure seems to be the most natural since consecutive measurements will be more correlated than measurements separated in time. Note that this does require that the clusters be sorted in time order (they are in this case).

```r
modgeep <- geeglm(seizures ~offset(log(timeadj)) + expind + treat +
  I(expind*treat), id=id, family=poisson, corstr="ar1", data=epilepsy,
  subset=(id!=49))
summary(modgeep)
```

```
##
## Call:
## geeglm(formula = seizures ~ offset(log(timeadj)) + expind + treat +
##     I(expind * treat), family = poisson, data = epilepsy, subset = (id !=
##     49), id = id, corstr = "ar1")
##
##  Coefficients:
##                   Estimate Std.err  Wald Pr(>|W|)
## (Intercept)         1.3138  0.1616 66.10  4.4e-16 ***
## expind              0.1509  0.1108  1.86    0.173
## treat              -0.0797  0.1983  0.16    0.688
## I(expind * treat)  -0.3987  0.1745  5.22    0.022 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = ar1
## Estimated Scale Parameters:
##
##             Estimate Std.err
## (Intercept)     10.6    2.35
##   Link = identity
##
## Estimated Correlation Parameters:
##       Estimate Std.err
## alpha    0.783  0.0519
## Number of clusters:   58  Maximum cluster size: 5
```

The drug effects, as measured by the interaction term, has a weakly significant effect.

The dispersion parameter is estimated as 10.6. This means that if we did not account for the overdispersion, the standard errors would be much larger.

The AR(1) correlation structure can be seen in the working correlation where adjacent measurements have 0.78 correlation.