

Generalized Linear Mixed Models

Outline

- Subject-specific Models
- Conditional Inference
- Hierarchical Generalized Linear Model
 - Beta-Binomial Model
 - Poisson-Gamma Model
- Generalized Linear Mixed Model

Subject-Specific Models

Assumptions

- Given the subject-specific effects \mathbf{b}_i (a q -vector), the responses Y_{ij} ($i = 1, \dots, m, j = 1, \dots, n_i$) are *independent* and follow a distribution from the exponential family

$$Y_{ij} | \mathbf{b}_i \sim f(y_{ij} | \mathbf{b}_i, \theta).$$

Let $E(Y_{ij} | \mathbf{b}_i) = \mu_{ij}$ then

$$g(\mu_{ij}) = \eta_{ij} = \mathbf{X}_{ij}^T \boldsymbol{\beta} + \mathbf{Z}_{ij}^T \mathbf{b}_i,$$

where η_{ij} is the linear predictor and g is the link function. \mathbf{X}_{ij} and \mathbf{Z}_{ij} are p - and q -vector of covariates, with \mathbf{Z} often being a subset of \mathbf{X} .

Three Ways to Handle Subject-specific Parameters

- Treated as fixed unknown parameters. Neyman and Scott (1948) showed that the ML estimates may be inconsistent due to the fact that the number of unknown parameters increases with m .
- Conditional inference
 - Subject-specific effects $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_m$ are treated as nuisance **parameters**;
 - Estimate $\boldsymbol{\beta}$ using the conditional likelihood given the *sufficient* statistics for \mathbf{b}_i .

- Random-effect approach
 - Treat \mathbf{b}_i as unobserved random variables and integrate them out to get the marginal likelihood of the parameters $\boldsymbol{\beta}$.
 - The random effects \mathbf{b}_i are independent and identically distributed with mean $\mathbf{0}$ and variance $D(\boldsymbol{\alpha})$. Its distribution G is completely specified with parameters $\boldsymbol{\alpha}$. That is, G does not depend on any covariates.
 - Estimation of inference for $\boldsymbol{\beta}$ is obtained from ML estimation, based on the marginal density for \mathbf{Y}_i .
 - Examples include linear mixed model, hierarchical generalized linear model (beta-binomial model, poisson-gamma model) and generalized linear mixed model.

Conditional Inference

Sufficiency

Suppose a random vector \mathbf{Y} has density indexed by parameter θ , and $s = s(\mathbf{y})$ is a statistic. s is said to be **sufficient** for θ if

$$f(\mathbf{y}; \theta) \propto g(s; \theta)h(\mathbf{y} | s).$$

The inference for θ can be based on the marginal density of s and no information is lost. The conditional density $h(\mathbf{y} | s)$ is useful for model checking but not in inference for θ .

Ancillary

If

$$f(\mathbf{y}; \theta) \propto g(\mathbf{y} | t; \theta)h(t),$$

then $t = t(\mathbf{y})$ is said to be **ancillary** for θ . In this case the conditional density $g(\mathbf{y} | t, \theta)$ is used for inference about θ .

When there is a nuisance parameter λ , an extension of the factorization is

$$f(\mathbf{y}; \theta, \lambda) = g(\mathbf{y} \mid s; \theta)h(s; \lambda).$$

- s is sufficient for λ and is ancillary for θ .

We can use the conditional density $g(\mathbf{y} \mid s; \theta)$ for inference about θ and the marginal density $h(s; \lambda)$ for inference about λ .

Oftentimes, we only have

$$f(\mathbf{y}; \theta, \lambda) = g(\mathbf{y} \mid s; \theta)h(s; \theta, \lambda).$$

We can still use $g(\mathbf{y} \mid s; \theta)$ for inference about θ but there is potential information in $h(s; \theta, \lambda)$ about θ .

Conditional Likelihood

- We will consider the binary (Bernoulli) and count (Poisson) data.
- Assume $a(\phi) = 1$ (no overdispersion) to simplify the discussion.
- Restrict to the canonical link, thus

$$\theta_{ij} = \eta_{ij} = g(\mu_{ij}) = \mathbf{x}_{ij}^T \boldsymbol{\beta} + \mathbf{z}_{ij}^T \mathbf{b}_i.$$

The likelihood function for $\boldsymbol{\beta}$ for individual i is

$$\prod_{j=1}^{n_i} f(y_{ij} | \boldsymbol{\beta}, \mathbf{b}_i) \propto \prod_{j=1}^{n_i} \exp \{ \theta_{ij} y_{ij} - \psi(\theta_{ij}) \} = \exp \left\{ \boldsymbol{\beta}^T \sum_{j=1}^{n_i} \mathbf{x}_{ij} y_{ij} + \mathbf{b}_i^T \sum_{j=1}^{n_i} \mathbf{z}_{ij} y_{ij} - \sum_{j=1}^{n_i} \psi(\theta_{ij}) \right\}. \quad (1)$$

Hence the **sufficient statistic** for \mathbf{b}_i is

$$\mathbf{T}_i = \sum_{j=1}^{n_i} \mathbf{z}_{ij} Y_{ij}.$$

Let

$$\mathbf{S}_i = \sum_{j=1}^{n_i} \mathbf{x}_{ij} Y_{ij},$$

which is sufficient for $\boldsymbol{\beta}$.

The **conditional distribution** of $\mathbf{y}_i \mid \mathbf{T}_i = \mathbf{t}_i$ is

$$\begin{aligned} \Pr(\mathbf{y}_i \mid \mathbf{T}_i = \mathbf{t}_i) &= \frac{\Pr(\mathbf{y}_i, \mathbf{T}_i = \mathbf{t}_i)}{\Pr(\mathbf{T}_i = \mathbf{t}_i)} \\ &= \frac{\exp(\boldsymbol{\beta}^T \mathbf{s}_i + \mathbf{b}_i^T \mathbf{t}_i)}{\sum_{\mathbf{y}_i^* \in \mathcal{R}_{t,i}} \exp(\boldsymbol{\beta}^T \mathbf{s}_i^* + \mathbf{b}_i^T \mathbf{t}_i)} \\ &= \frac{\exp(\boldsymbol{\beta}^T \mathbf{s}_i)}{\sum_{\mathbf{y}_i^* \in \mathcal{R}_{t,i}} \exp(\boldsymbol{\beta}^T \mathbf{s}_i^*)} \end{aligned}$$

where

$$\mathcal{R}_{t,i} = \{(y_{i1}, \dots, y_{in_i}) : \mathbf{T}_i = \mathbf{t}_i\},$$

that is, the set of outcomes for which the statistic \mathbf{T}_i takes the value \mathbf{t}_i .

For all the data, the conditional likelihood is proportional to

$$\prod_{i=1}^m \frac{\exp(\boldsymbol{\beta}^T \mathbf{s}_i)}{\sum_{\mathbf{y}_i^* \in \mathcal{R}_{t,i}} \exp(\boldsymbol{\beta}^T \mathbf{s}_i^*)}.$$

- The conditional likelihood uses part of the data that does not contain information about $(\mathbf{b}_1, \dots, \mathbf{b}_m)$ to estimate $\boldsymbol{\beta}$.
- It is not necessary to specify the distribution of \mathbf{b}_i .
- For simple cases such as the random intercept model, the conditional likelihood is relatively easy to maximize. Next, we will consider random intercept model for binary and count data in more detail.

Random Intercept Model

In the random intercept model, the linear predictor is

$$\eta_{ij} = \gamma_i + \mathbf{x}_{ij}^T \boldsymbol{\beta} \quad [+\text{offsets if necessary}],$$

where $\gamma_i = \beta_0 + b_i$ and \mathbf{x}_{ij} does not include an intercept term. The sufficient statistic for γ_i is

$$T_i = \sum_{j=1}^{n_i} Y_{ij} = Y_{i+}.$$

For binary data, the joint likelihood function for $\boldsymbol{\beta}$ and the γ_i is proportional to

$$\prod_{i=1}^m \exp \left[\gamma_i \sum_{j=1}^{n_i} y_{ij} + \left(\sum_{j=1}^{n_i} y_{ij} \mathbf{x}_{ij}^T \right) \boldsymbol{\beta} - \sum_{j=1}^{n_i} \log \{ 1 + \exp(\gamma_i + \mathbf{x}_{ij}^T \boldsymbol{\beta}) \} \right].$$

The conditional likelihood for $\boldsymbol{\beta}$ is proportional to

$$\prod_i^m \frac{\exp(\sum_{j=1}^{n_i} y_{ij} \mathbf{x}_{ij}^T \boldsymbol{\beta})}{\sum_{\{L \in R_i\}} \exp(\sum_{\{l \in L\}} \mathbf{x}_{il}^T \boldsymbol{\beta})}. \quad (2)$$

For example, if for subject i , there were $y_{i+} = 2$ (out of $n_i = 10$) defected pregnancies were observed, then R_i contains all ways of choosing 2 y_{ij} 's from the 10 outcomes $\{y_{ij}; j = 1, \dots, 10\}$. There are 10-choose-2 (=45) possible ways in total. $\sum_{\{L \in R_i\}}$ is the sum across all 45 ways.

Conditional Logistic Regression for Binary Responses

Consider conditional MLE of random intercept logistic model for binary data.

2 × 2 Crossover Trial Data

$$n_i = 2, i = 1, \dots, m$$

Compare two treatments

- A: active drug.
- B: placebo.

Responses:

- 1 for a normal electrocardiogram reading.
- 0 for a abnormal reading.

	$y_1 = 1$	$y_1 = 1$	$y_1 = 0$	$y_1 = 0$
Group	$y_2 = 1$	$y_2 = 0$	$y_2 = 1$	$y_2 = 0$
AB	22	6	0	6
BA	18	2	4	9
AB	a_1	b_1	c_1	d_1
BA	a_2	b_2	c_2	d_2

If $y_{i+} = 2$ or 0 (2 successes or no successes in two trials), R_i has a single element

$$(y_{i1}, y_{i2}) = (1, 1) \text{ or } (0, 0)$$

and the contribution to (2) is 1. Therefore, and we do not need to consider responses (1,1) or (0,0) in calculating the conditional likelihood.

Let

$$x_1 = \begin{cases} 1 & \text{if A (active)} \\ 0 & \text{if B (placebo)} \end{cases}$$
$$x_2 = \begin{cases} 1 & \text{if period 2} \\ 0 & \text{if period 1} \end{cases}$$
$$x_3 = x_1 \times x_2$$

The conditional likelihood for $\boldsymbol{\beta}$ is

$$\mathcal{L}(\boldsymbol{\beta}) = \prod_{i: y_{i1}=1, y_{i2}=0} \frac{\exp(\mathbf{x}_{i1}^T \boldsymbol{\beta})}{\exp(\mathbf{x}_{i1}^T \boldsymbol{\beta}) + \exp(\mathbf{x}_{i2}^T \boldsymbol{\beta})} \times \prod_{i: y_{i2}=1, y_{i1}=0} \frac{\exp(\mathbf{x}_{i2}^T \boldsymbol{\beta})}{\exp(\mathbf{x}_{i1}^T \boldsymbol{\beta}) + \exp(\mathbf{x}_{i2}^T \boldsymbol{\beta})}.$$

Group AB				Group BA			
period 1 (A)				period 1 (B)			
<div><div></div><div>10</div></div>				<div><div></div><div>10</div></div>			
period 2	1	a_1	c_1	period 2	1	a_2	c_2
(B)	0	b_1	d_1	(A)	0	b_2	d_2

a_1, d_1, a_2 and d_2 do not contribute to the conditional likelihood.

Values of $\sum_{j=1}^2 y_{ij} \mathbf{x}_{ij}^T \boldsymbol{\beta}$

- We can ignore calculating these for $(y_{i1}, y_{i2}) = (1,1)$ or $(0,0)$
- For $i \in \text{group AB}$

$$\mathbf{x}_i^T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

So,

$$\mathbf{x}_{i1}^T \boldsymbol{\beta} = \beta_1$$

$$\mathbf{x}_{i2}^T \boldsymbol{\beta} = \beta_2$$

and there are b_1 and c_1 of those terms, respectively, in the conditional likelihood.

- For $i \in \text{group BA}$

$$\mathbf{x}_i^T = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

So,

$$\mathbf{x}_{i1}^T \boldsymbol{\beta} = 0$$

$$\mathbf{x}_{i2}^T \boldsymbol{\beta} = \beta_1 + \beta_2 + \beta_3$$

and there are b_2 and c_2 of those terms, respectively, in the conditional likelihood.

- The conditional likelihood is proportional to

$$\underbrace{\left(\frac{e^{\beta_1}}{e^{\beta_1} + e^{\beta_2}}\right)^{b_1} \left(\frac{e^{\beta_2}}{e^{\beta_1} + e^{\beta_2}}\right)^{c_1}}_{\text{(AB group)}} \times \underbrace{\left(\frac{1}{1 + e^{\beta_1 + \beta_2 + \beta_3}}\right)^{b_2} \left(\frac{e^{\beta_1 + \beta_2 + \beta_3}}{1 + e^{\beta_1 + \beta_2 + \beta_3}}\right)^{c_2}}_{\text{(BA group)}}$$

- Special case 1: When $\beta_2 = \beta_3 = 0$, i.e. only consider the treatment effect, the conditional likelihood reduces to

$$\left(\frac{e^{\beta_1}}{1 + e^{\beta_1}}\right)^{b_1} \left(\frac{1}{1 + e^{\beta_1}}\right)^{c_1} \left(\frac{1}{1 + e^{\beta_1}}\right)^{b_2} \left(\frac{e^{\beta_1}}{1 + e^{\beta_1}}\right)^{c_2} = \left(\frac{e^{\beta_1}}{1 + e^{\beta_1}}\right)^{b_1 + c_2} \left(\frac{1}{1 + e^{\beta_1}}\right)^{c_1 + b_2}$$

Therefore

$$\hat{\beta}_1 = \log \left(\frac{b_1 + c_2}{c_1 + b_2} \right) = \log \left(\frac{6 + 4}{0 + 2} \right) = 1.61$$

$$\widehat{\text{s.e.}}(\hat{\beta}_1) = \sqrt{\frac{1}{b_1 + c_2} + \frac{1}{c_1 + b_2}} = 0.77$$

- Special case 2: When $\beta_3 = 0$, the conditional likelihood is

$$\left(\frac{e^{\beta_1}}{e^{\beta_1} + e^{\beta_2}}\right)^{b_1} \left(\frac{e^{\beta_2}}{e^{\beta_1} + e^{\beta_2}}\right)^{c_1} \left(\frac{1}{1 + e^{\beta_1 + \beta_2}}\right)^{b_2} \left(\frac{e^{\beta_1 + \beta_2}}{1 + e^{\beta_1 + \beta_2}}\right)^{c_2} = p_1^{b_1} (1 - p_1)^{c_1} (1 - p_2)^{b_2} p_2^{c_2}$$

where

$$\text{logit}(p_1) = \beta_1 - \beta_2, \quad \text{logit}(p_2) = \beta_1 + \beta_2.$$

Then

$$\begin{aligned} \hat{p}_1 &= \frac{b_1}{b_1 + c_1}, & \hat{p}_2 &= \frac{c_2}{b_2 + c_2} \\ \hat{\beta}_1 &= \frac{1}{2}(\text{logit } \hat{p}_1 + \text{logit } \hat{p}_2) = \frac{1}{2} \log \left(\frac{b_1 c_2}{c_1 b_2} \right) \\ \widehat{\text{s.e.}}(\hat{\beta}_1) &= \frac{1}{2} \sqrt{b_1^{-1} + c_1^{-1} + b_2^{-1} + c_2^{-1}}. \end{aligned}$$

Since $c_1 = 0$, use *ad hoc* convention of adding 0.5 to give

$$\hat{\beta}_1 = \frac{1}{2} \log \left(\frac{6 \cdot 4}{0.5 \cdot 2} \right) = 1.59$$
$$\widehat{\text{s.e.}}(\hat{\beta}_1) = 0.85$$

- Note:

1. In both models, $\hat{\beta}_1 \approx 1.6$ is marginally significant at a (2-sided) 5% level.
2. $\exp(1.6) \approx 5$ indicates the odds of a normal result for a treated patient is about 5 times of the odds for a non-treated patient.
3. In conditional logistic regression we can only estimate effects for within-cluster covariates and any interactions between a within-cluster covariate and a cluster-level covariate.

Conditional Logistic Regression Results

The `clogit` function in package `survival` can fit conditional logistic regression models.

```
> xover <- read.table ("../data/xover1.data",
+                      col.names = c("id", "class", "y", "intercept",
+                      "trt", "period", "xover", "BA"))
> xover$trtA <- 1-xover$trt
> with (xover, ftable (BA, trtA, y))
      y  0  1
BA trtA
0  0      12 22
   1       6 28
1  0      13 20
   1      11 22

> library (survival)
Loading required package: splines

Attaching package: 'survival'

> xover.cl <- clogit (y ~ trtA + strata (id), data = xover)
> summary (xover.cl)
Call:
coxph(formula = Surv(rep(1, 134L), y) ~ trtA + strata(id), data = xover,
      method = "exact")
```

```
n= 134
      coef exp(coef) se(coef)      z      p
trtA 1.61          5    0.775 2.08 0.038

      exp(coef) exp(-coef) lower .95 upper .95
trtA          5          0.2      1.10      22.8

Rsquare= 0.043   (max possible= 0.117 )
Likelihood ratio test= 5.82  on 1 df,   p=0.0158
Wald test          = 4.32  on 1 df,   p=0.0377
Score (logrank) test = 5.33  on 1 df,   p=0.0209
```

GEE Results

```
> library (gee)
> summary (gee (y ~ trtA, data = xover, cor = "exchangeable",
+             id = id, family = binomial, scale.fix = TRUE))
Beginning Cgee S-function, @(#) geeformula.q 4.13 98/01/27
running glm to get initial regression estimate
(Intercept)      trtA
  0.5187938    0.5600159
```

```
GEE:  GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
gee S-function, version 4.13 modified 98/01/27 (1998)
```

Model:

```
Link:                      Logit
Variance to Mean Relation: Binomial
Correlation Structure:     Exchangeable
```

Call:

```
gee(formula = y ~ trtA, id = id, data = xover, family = binomial,
    corstr = "exchangeable", scale.fix = TRUE)
```

Summary of Residuals:

	Min	1Q	Median	3Q	Max
	-0.7462687	-0.6268657	0.2537313	0.3731343	0.3731343

Coefficients:

	Estimate	Naive S.E.	Naive z	Robust S.E.	Robust z
(Intercept)	0.5187938	0.2526055	2.053771	0.2526055	2.053771
trtA	0.5600159	0.2328349	2.405205	0.2356943	2.376026

Estimated Scale Parameter: 1

Number of Iterations: 1

Working Correlation

	[,1]	[,2]
[1,]	1.0000000	0.6233823
[2,]	0.6233823	1.0000000

Note

- The marginal coefficients are smaller in absolute value.

$$|0.56| \leq |1.61|$$

(consistent with the theoretical inequality in Neuhaus Theorem).

- The marginal and conditional coefficients have different interpretations.
- The Z statistics are similar for the two regressions.

Poisson Regression for Count Responses

Consider conditional MLE of random intercept log-linear model for count data.

- $(y_{i1}, \dots, y_{in_i})$ are independent Poisson r.v.s with

$$\log E(y_{ij} \mid \gamma_i, \beta) = \gamma_i + \mathbf{x}_{ij}^T \boldsymbol{\beta} + \log(t_{ij})$$

where $\gamma_i = \beta_0 + b_i$ and \mathbf{x}_{ij} does not include the intercept term.

- The likelihood contributed by the i -th individual is proportional to

$$\exp \left\{ \gamma_i \sum_j y_{ij} + \boldsymbol{\beta}^T \sum_j \mathbf{x}_{ij} y_{ij} + \sum_j y_{ij} \log(t_{ij}) - \sum_j t_{ij} \exp(\gamma_i + \mathbf{x}_{ij}^T \boldsymbol{\beta}) \right\} \frac{1}{\prod_j y_{ij}!}$$

Then $y_{i+} = \sum_j y_{ij}$ is a sufficient statistic for γ_i .

- The distribution of y_{i+} is Poisson with mean

$$\sum_j e^{\gamma_i + \mathbf{x}_{ij}^T \boldsymbol{\beta}} = \lambda_i.$$

- Let $\mathbf{s}_i = \sum_j \mathbf{x}_{ij} y_{ij}$. The conditional distribution of \mathbf{y}_i given y_{i+}

$$\begin{aligned} \Pr(\mathbf{y}_i | y_{i+}) &= \frac{\exp \left\{ \gamma_i y_{i+} + \boldsymbol{\beta}^T \mathbf{s}_i + \sum_j y_{ij} \log(t_{ij}) - \sum_j t_{ij} \exp(\gamma_i + \mathbf{x}_{ij}^T \boldsymbol{\beta}) \right\}}{\lambda_i^{y_{i+}} \exp(-\lambda_i) / y_{i+}!} \\ &= \binom{y_{i+}}{y_{i1}, \dots, y_{in_i}} \frac{\exp \left\{ \boldsymbol{\beta}^T \mathbf{s}_i + \sum_j y_{ij} \log t_{ij} \right\}}{\left(\sum_j \exp \{ \mathbf{x}_{ij}^T \boldsymbol{\beta} + \log t_{ij} \} \right)^{y_{i+}}} \end{aligned}$$

- The conditional likelihood for all data is proportional to:

$$\prod_{i=1}^m \frac{\exp \left\{ \boldsymbol{\beta}^T \mathbf{s}_i + \sum_j y_{ij} \log t_{ij} \right\}}{\left(\sum_j \exp \{ \mathbf{x}_{ij}^T \boldsymbol{\beta} + \log t_{ij} \} \right)^{y_{i+}}}$$

Seizure Data

- $x_{ij1} = 1$ or 0 for progabide or placebo, respectively.
- $x_{ij2} = 1$ if $j = 1, 2, 3, 4$; 0 if $j = 0$.
- $x_{ij3} = x_{ij1}x_{ij2}$ (interaction term). Note that β_3 is the parameter of interest.

The model:

$$\log E(y_{ij} \mid \gamma_i, \boldsymbol{\beta}) = \gamma_i + \beta_1 x_{ij1} + \beta_2 x_{ij2} + \beta_3 x_{ij3} + \log t_{ij}$$

Note e^{γ_i} is the expected baseline count for individual i .

- For the placebo group,

$$\mathbf{x}_{ij}^T = \begin{cases} (0, 0, 0) & \text{if } j = 0 \\ (0, 1, 0) & \text{if } j = 1, 2, 3, 4 \end{cases}$$

- For the treatment group,

$$\mathbf{x}_{ij}^T = \begin{cases} (1, 0, 0) & \text{if } j = 0 \\ (1, 1, 1) & \text{if } j = 1, 2, 3, 4 \end{cases}$$

Thus

$$\begin{aligned} \mathbf{s}_i &= \sum_{j=0}^4 \mathbf{x}_{ij} y_{ij} \\ &= \mathbf{x}_{i0} y_{i0} + \mathbf{x}_{i1} \sum_{j=1}^4 y_{ij} \\ &= \mathbf{x}_{i0} y_{i0} + \mathbf{x}_{i1} (y_{i+} - y_{i0}) \end{aligned}$$

If $i \in$ placebo group,

$$\mathbf{s}_i^T \boldsymbol{\beta} = \beta_2 (y_{i+} - y_{i0}),$$

otherwise it equals

$$y_{i0}\beta_1 + (y_{i+} - y_{i0})(\beta_1 + \beta_2 + \beta_3) = (y_{i+} - y_{i0})(\beta_2 + \beta_3) + y_{i+}\beta_1.$$

The conditional likelihood is proportional to

$$\prod_i \frac{\exp(\mathbf{s}_i^T \boldsymbol{\beta}) \exp(\sum_j y_{ij} \log(t_{ij}))}{\left(\sum_{j=0}^4 \exp\{\mathbf{x}_{ij} \boldsymbol{\beta} + \log t_{ij}\}\right)^{y_{i+}}}.$$

Here we have $t_{i0} = 8$ and $t_{i1} = t_{i2} = t_{i3} = t_{i4} = 2$.

- For $i \in$ placebo group, the likelihood is proportional to

$$\frac{\exp\{\beta_2(y_{i+} - y_{i0})\} \prod_{j=0}^4 t_{ij}^{y_{ij}}}{(t_{i0} + \exp\{\beta_2\} \sum_{j=1}^4 t_{ij})^{y_{i+}}} = \frac{e^{\beta_2(y_{i+} - y_{i0})}}{(1 + e^{\beta_2})^{y_{i+}}} \times \frac{\prod_{j=0}^4 t_{ij}^{y_{ij}}}{8^{y_{i+}}} \propto (1 - \pi_1)^{y_{i+} - y_{i0}} \pi_1^{y_{i0}}$$

which is a Binomial (y_{i+}, π) , with

$$\pi_1 = \frac{1}{1 + e^{\beta_2}}.$$

- For $i \in$ progabide group, the likelihood is proportional to

$$\frac{e^{y_{i+}\beta_1} e^{(\beta_2+\beta_3)(y_{i+}-y_{i0})}}{e^{y_{i+}\beta_1} (1 + e^{\beta_2+\beta_3})^{y_{i+}}} = \pi_2^{y_{i0}} (1 - \pi_2)^{y_{i+}-y_{i0}}$$

where

$$\pi_2 = \frac{1}{1 + e^{\beta_2+\beta_3}}$$

- In summary, the conditional likelihood is proportional to

$$\prod_{i=1}^{28} \pi_1^{y_{i0}} (1 - \pi_1)^{y_{i+}-y_{i0}} \prod_{i=29}^{59} \pi_2^{y_{i0}} (1 - \pi_2)^{y_{i+}-y_{i0}}$$

- π_1 and π_2 are the probabilities that an individual's seizure occurs before rather than after the randomization, for placebo and progabide groups, respectively.
- If progabide is helpful in reducing seizures we would observe $\pi_1 < \pi_2$, or equivalently, $1 + e^{\beta_2} > 1 + e^{\beta_2+\beta_3} \Leftrightarrow \beta_3 < 0$.

- Results (patient 207 deleted)

	GEE	Cond. Like.
β_2	0.11 (0.12)	0.11 (0.047)
β_3	-0.30 (0.17)	-0.30 (0.07)

Notes:

1. Conditional likelihood inference leads to conclusion that progabide's effect is highly significant.
2. But the fitted model is inadequate based on a Pearson χ^2 statistic (see DHLZ 9.4.1).
3. Because of 2, the estimated s.e.'s in conditional likelihood approach may be too small in this example.
4. The homogeneity assumption that everyone's response to the treatment is the same is inadequate, that is, a random slope for the treatment is needed. However, the conditional likelihood method would no longer be appropriate since all relevant information about β_3 will be conditioned away. We must instead use the random effects approach.

Hierarchical Generalized Linear Model

Overdispersion

- For independent count (Poisson or Binomial) data, there is often overdispersion. One possible cause is heterogeneity, for example, due to an unobserved important covariate.
- When data are correlated/clustered, ignoring the correlation can result in overdispersion which, when ignored, leads to underestimate of the standard errors of the regression parameters.
- Example: In teratology experiments, pregnant rats are randomized to receive a teratogenic or a control agent, then the total number of animals in a litter and the number of birth defects are recorded. Because in a litter, all births have the same mother and hence their outcomes (having birth defect or not) are correlated.
- Example: Hospitals are randomized to use a new treatment program for alcoholics or stay with the current program. The response variable is the number of hospitalization in the year following enrollment in the program. One might expect a “hospital” effect such that the hospitalization events at the same hospital are correlated.

- Previously we used quasi-likelihood methods to take into account of the overdispersion by using a scale parameter ϕ .
- For likelihood-based methods, it is natural to model the “litter” or “hospital” effects as random effects. Early examples are beta-binomial and Poisson-Gamma (negative binomial) models.

Beta-Binomial Model

- For the teratology experiment, let $Y_{ijk} = 1$ if animal k from litter j in treatment group i has a birth defect, and 0 otherwise. Then we can assume:

$$\begin{aligned} Y_{ijk} \mid \pi_{ij} &\overset{iid}{\sim} \text{Bernoulli}(\pi_{ij}) \\ \pi_{ij} &\overset{iid}{\sim} \text{Beta}(\alpha_i, \beta_i) \end{aligned}$$

- The reason to choose Beta distribution for π_{ij} is that it is the **conjugate distribution** for Binomial distribution and hence is computational convenient (important in early years!)

- Under this model, Y_{ijk} has a marginal Bernoulli distribution with mean

$$\mu_i = E(Y_{ijk}) = \frac{\alpha_i}{\alpha_i + \beta_i}.$$

- Note that $Y_{ij+} = \sum_k Y_{ijk}$ does not have a Binomial distribution, but a Beta-Binomial distribution with density:

$$\Pr(Y_{ij+} = y; \alpha_i, \beta_i) = \binom{n_{ij}}{y} \frac{B(y + \alpha_i, n_{ij} - y + \beta_i)}{B(\alpha_i, \beta_i)},$$

where B is the Beta function.

- The mean and variances are:

$$E(Y_{ij+}) = n_{ij}\mu_i$$

$$\text{Var}(Y_{ij+}) = n_{ij}\mu_i(1 - \mu_i) \{1 + \rho_i(n_{ij} - 1)\}, \text{ where } \rho_i = \frac{1}{\alpha_i + \beta_i + 1}.$$

- Note that $\rho_i = \text{Corr}(Y_{ijk}, Y_{ijk'})$. (Check this.)
- The Beta-Binomial model is **not** in the exponential family. Even though it is adequate for simple experiments but is more difficult to extend, for example, to model covariates or to allow more complicated correlation structure.
- In contrast, in a **quasi-likelihood model** we may use

$$\text{Var}(Y_{ij+}) = \phi n_{ij}\mu_i(1 - \mu_i), \tag{3}$$

or use a variance function motivated by Beta-Binomial distribution with fixed ρ ,

$$\text{Var}(Y_{ij+}) = n_{ij}\mu_i(1 - \mu_i)\{1 + (n_{ij} - 1)\rho\}. \tag{4}$$

There is no clear conclusion which model is better (Liang and McCullagh, 1993).

Moore's Teratology Data

This data example is from the website of Agresti (2002). Female rats were put on iron-deficient diets and then randomized to receive placebo (group 1) and three different iron supplements (groups 2, 3, and 4). They are sacrificed 3 weeks after pregnant.

n is the total number of fetuses in a litter and y is the number of dead fetuses.

```
> tera <- read.table ("../data/tera.dat", col.names = c("litter", "group", "n", "y"))
>
> tera$group <- factor (tera$group)
> tera[1:5,]
  litter group  n  y
1      1     1 10  1
2      2     1 11  4
3      3     1 12  9
4      4     1  4  4
5      5     1 10 10
```

First we fit a naive Binomial model. From the residual plot, there is clearly overdispersion that depends on the litter size.

```
> tera.glm <- glm (cbind(y, n - y) ~ group, data = tera,
+                 family = binomial)
> summary (tera.glm)
```

Call:

```
glm(formula = cbind(y, n - y) ~ group, family = binomial, data = tera)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-4.42952	-0.97500	-0.02846	1.40242	2.78260

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	1.1440	0.1292	8.855	< 2e-16 ***
group2	-3.3225	0.3308	-10.043	< 2e-16 ***
group3	-4.4762	0.7311	-6.122	9.22e-10 ***
group4	-4.1297	0.4762	-8.672	< 2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

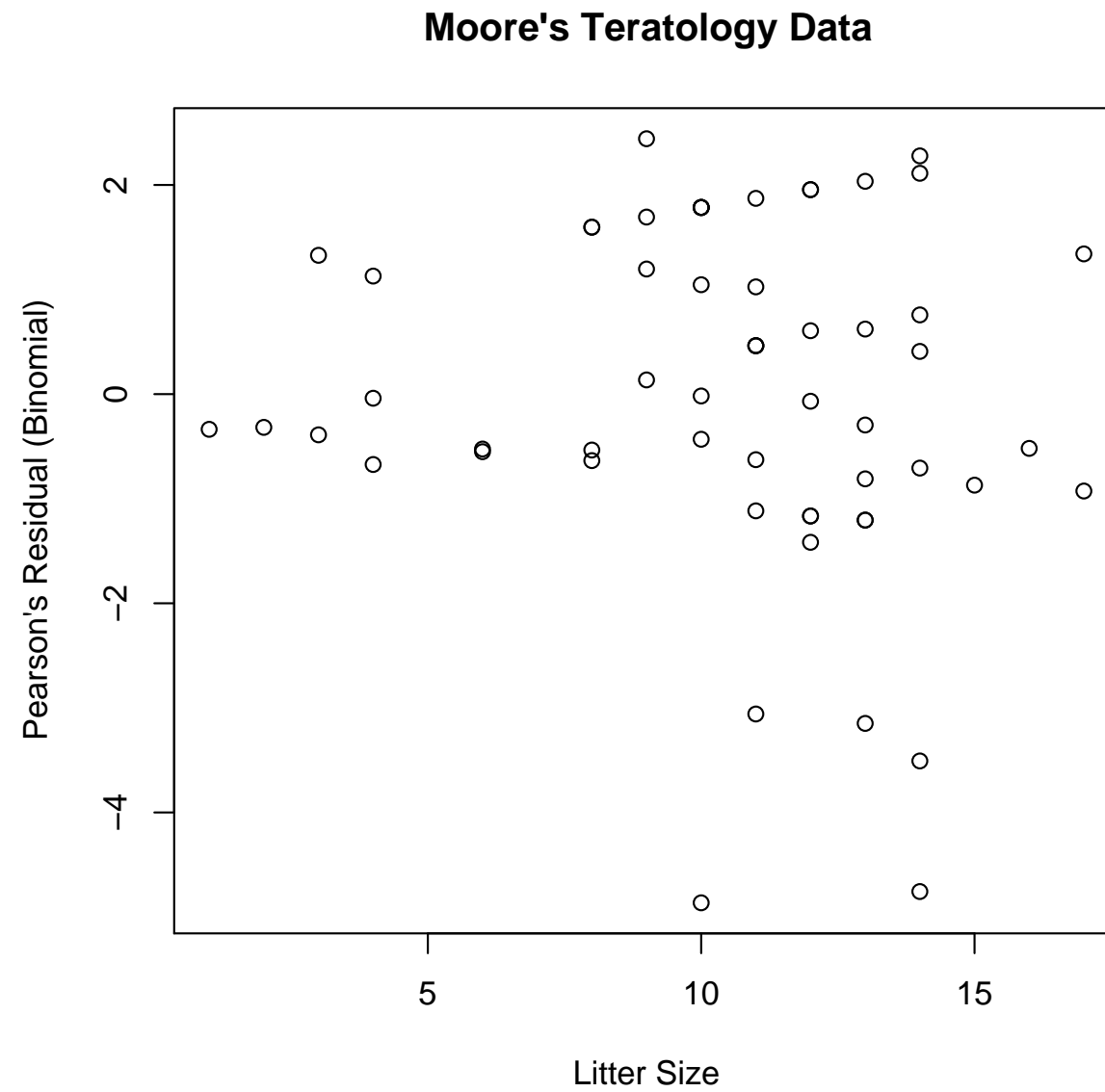
Null deviance: 509.43 on 57 degrees of freedom

Residual deviance: 173.45 on 54 degrees of freedom

[AIC: 252.92](#)

Number of Fisher Scoring iterations: 5

```
> plot (tera$n, resid (tera.glm, type = "p"), xlab = "Litter Size",
+       ylab = "Pearson's Residual (Binomial)",
+       main = "Moore's Teratology Data")
```



Quasi-likelihood Model 1

Next we fit a quasi-likelihood model with variance model in (3).

```
> tera.quasi <- glm (cbind(y, n - y) ~ group, data = tera,
+                    family = quasibinomial)
> summary (tera.quasi)
```

Call:

```
glm(formula = cbind(y, n - y) ~ group, family = quasibinomial,
    data = tera)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-4.42952	-0.97500	-0.02846	1.40242	2.78260

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	1.1440	0.2187	5.231	2.81e-06	***
group2	-3.3225	0.5600	-5.933	2.18e-07	***
group3	-4.4762	1.2375	-3.617	0.000656	***
group4	-4.1297	0.8061	-5.123	4.14e-06	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasibinomial family taken to be 2.864945)

Null deviance: 509.43 on 57 degrees of freedom

Residual deviance: 173.45 on 54 degrees of freedom

AIC: NA

Number of Fisher Scoring iterations: 5

Note the point estimates are the same. The scale parameter does not affect the point estimates.

Quasi-likelihood Model 2

This R function by Patrick Heagerty at UW fits quasi-likelihood model with the variance model in (4), that is,

$$\text{Var}(Y_{ij}) = n_{ij}\mu_i(1 - \mu_i)(1 + \rho(n_{ij} - 1)).$$

Note that the point estimates are different now.

```
> source ("bod_regn.R")

> tera$g2 <- as.numeric (tera$group == 2)
> tera$g3 <- as.numeric (tera$group == 3)
> tera$g4 <- as.numeric (tera$group == 4)
> attach (tera)

> x <- cbind (1, tera$g2, tera$g3, tera$g4)
> tera.bod <- bod.regn (y = tera$y, n = tera$n, x = x,
+                       dispersion = "correlation",
+                       beta = c(1, -3, -4, -4), alpha = 2)

Z is NULL. Common alpha assumed.
> tera.bod
```

```
Regression for Binomial Overdispersed Data
dispersion = correlation
```


number of observations = 58

Mean parameters:

	estimate	mod. s.e.	emp. s.e.	Z	emp. p	emp.
beta1	1.212	0.222	0.270	4.496		0
beta2	-3.369	0.559	0.430	-7.827		0
beta3	-4.584	1.294	0.624	-7.350		0
beta4	-4.249	0.843	0.605	-7.021		0

Dispersion parameters:

	estimate	emp. s.e.
alpha	-1.669719	0.3082800

In this model $\hat{\rho} = \exp(-1.67) = 0.188$.

GEE and Empirical Standard Errors

The GEE model is the same as the Quasi-likelihood Model 1 above but empirical (robust) standard errors are calculated.

```
> tera.gee <- gee (cbind(y, n - y) ~ group, data = tera, id = litter,
+                 family = binomial)
[1] "Beginning Cgee S-function, @(#) geeformula.q 4.13 98/01/27"
[1] "running glm to get initial regression estimate"
[1] 1.143981 -3.322513 -4.476185 -4.129663
> summary (tera.gee)
```

```
GEE:  GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
gee S-function, version 4.13 modified 98/01/27 (1998)
```

Model:

```
Link:                               Logit
Variance to Mean Relation: Binomial
Correlation Structure:              Independent
```

Call:

```
gee(formula = cbind(y, n - y) ~ group, id = litter, data = tera,
    family = binomial)
```

Summary of Residuals:

Min	1Q	Median	3Q	Max
-----	----	--------	----	-----

-0.10169492 0.03453548 2.74159021 8.24159021 13.24159021

Coefficients:

	Estimate	Naive S.E.	Naive z	Robust S.E.	Robust z
(Intercept)	1.143981	0.2186717	5.231499	0.2758667	4.146861
group2	-3.322513	0.5599915	-5.933149	0.4400582	-7.550168
group3	-4.476185	1.2375175	-3.617068	0.6104577	-7.332507
group4	-4.129663	0.8060677	-5.123221	0.5763810	-7.164814

Estimated Scale Parameter: 2.864945

Beta-Binomial Model: Maximum Likelihood

The estimation of the “Beta-Binomial” regression models above (Quasi-likelihood Model 2) are actually done using estimating equation (i.e. quasi-score) approach. The maximum likelihood estimation based on the Beta-Binomial model is not trivial. To use ML, we use the reparametrization: $(\alpha_i, \beta_i) \rightarrow (\mu_i, \theta_i)$, where $\mu_i = \frac{\alpha_i}{\alpha_i + \beta_i}$, $\theta_i = \alpha_i + \beta_i$, α_i and β_i are Beta distribution parameters. In the next model, we assume:

$$\text{location: } \text{logit}(\mu_i) = b_1 + b_2(\text{group}_2) + b_3(\text{group}_3) + b_4(\text{group}_4)$$

$$\text{shape (constant): } \log(\theta_i) = \text{constant}$$

```
> library (rmutil) # for 'finterp' function (formula interpreter)
> library (gnlm)   # for 'gnlr' function
> library (boot)   # for 'inv.logit' function
> tera.gnlr <- gnlr (cbind (y, n - y), distribution = "beta binomial",
+                   mu = function (beta)
+                     inv.logit (beta[1] + beta[2] * g2 +
+                               beta[3] * g3 + beta[4] * g4)
+                   , pmu = c(1, -3, -4, -4),
+                   pshape = 2)
> tera.gnlr
```

Call:

```
gnlr(cbind(y, n - y), distribution = "beta binomial", mu = function(beta)
  inv.logit(beta[1] + beta[2] * g2 + beta[3] * g3 + beta[4] *
    g4)
, pmu = c(1, -3, -4, -4), pshape = 2)
```

beta binomial distribution

Response: cbind(y, n - y)

Log likelihood function:

```
m <- mu1(p)
s <- exp(sh1(p))
t <- s * m
u <- s * (1 - m)
-sum(wt * (lbeta(y[, 1] + t, y[, 2] + u) - lbeta(t, u)))
```

Location function:

```
structure(
  inv.logit(beta[1] + beta[2] * g2 + beta[3] * g3 + beta[4] *
    g4)
), srcfile = <environment>
```

Log shape function:

```
structure(p[1] * rep(1, n)), srcfile = <environment>
```

-Log likelihood 93.45675

Degrees of freedom 53

AIC 98.45675

Iterations 7

Location parameters:

	estimate	se
beta[1]	1.346	0.2482
beta[2]	-3.114	0.5018
beta[3]	-3.868	0.8081
beta[4]	-3.923	0.6675

Shape parameters:

	estimate	se
p[1]	1.146	0.3299

Correlations:

	1	2	3	4	5
1	1.0000	-0.5338	-0.3383	-0.4025	0.1720
2	-0.5338	1.0000	0.2211	0.2548	-0.3143
3	-0.3383	0.2211	1.0000	0.1679	-0.2350
4	-0.4025	0.2548	0.1679	1.0000	-0.2436
5	0.1720	-0.3143	-0.2350	-0.2436	1.0000

Here $\hat{\rho} = 1/(\exp(1.146) + 1) = 0.242$.

In the next model we assume,

$$\log(\theta_i) = \gamma_0 + \gamma_1 n_{ij}.$$

```
> tera.gnlr2 <- gnlr (cbind (y, n - y), distribution = "beta binomial",
+                     mu = function (beta)
+                         inv.logit (beta[1] + beta[2] * g2 + beta[3] * g3 +
+                                     beta[4] * g4)
+                     , pmu = c(1, -3, -4, -4),
+                     shape = finterp (~ n),
+                     pshape = c(2, 0.5))
> tera.gnlr2
```

Call:

```
gnlr(cbind(y, n - y), distribution = "beta binomial", mu = function(beta)
  inv.logit(beta[1] + beta[2] * g2 + beta[3] * g3 + beta[4] *
    g4)
, pmu = c(1, -3, -4, -4), shape = finterp(~n), pshape = c(2,
  0.5))
```

beta binomial distribution

Response: cbind(y, n - y)

Log likelihood function:

```

m <- mu1(p)
s <- exp(sh1(p))
t <- s * m
u <- s * (1 - m)
-sum(wt * (lbeta(y[, 1] + t, y[, 2] + u) - lbeta(t, u)))

```

Location function:

```

structure(
  inv.logit(beta[1] + beta[2] * g2 + beta[3] * g3 + beta[4] *
    g4)
), srcfile = <environment>

```

Log shape function:

$\sim n$

```

-Log likelihood      93.44569
Degrees of freedom  52
AIC                  99.44569
Iterations           22

```

Location parameters:

	estimate	se
beta[1]	1.335	0.2587
beta[2]	-3.105	0.5061
beta[3]	-3.859	0.8103


```
beta[4]    -3.921  0.6673
```

Shape parameters:

	estimate	se
(Intercept)	0.93694	1.4375
n	0.01776	0.1192

Correlations:

	1	2	3	4	5	6
1	1.0000	-0.5448	-0.34467	-0.39068	0.31051	-0.27987
2	-0.5448	1.0000	0.22836	0.25494	-0.19686	0.12883
3	-0.3447	0.2284	1.00000	0.16874	-0.12470	0.07299
4	-0.3907	0.2549	0.16874	1.00000	-0.06879	0.01326
5	0.3105	-0.1969	-0.12470	-0.06879	1.00000	-0.97347
6	-0.2799	0.1288	0.07299	0.01326	-0.97347	1.00000

From AIC the simpler Beta-Binomial is preferable and fit the data much better than the naive Binomial model (AIC = 98 vs 253).

Poisson-Gamma Model

For clustered count data, we can assume

$$Y_{ij} \mid \mu_{ij} \stackrel{\text{iid}}{\sim} \text{Poisson}(\mu_{ij})$$
$$\mu_{ij} \stackrel{\text{iid}}{\sim} \text{Gamma}(\lambda_i, \theta_i/\lambda_i)$$

The marginal distribution of Y_{ij} is *negative-Binomial* with

$$E(Y_{ij}) = E\{E(Y_{ij} \mid \mu_{ij})\} = \theta_i$$

$$\text{Var}(Y_{ij}) = \theta_i + \theta_i^2/\lambda_i = \theta_i(1 + \theta_i/\lambda_i)$$

Note that we use a somewhat unusual parameterization for Gamma distribution such that $E(\mu_{ij}) = \theta_i$ and $\text{Var}(\mu_{ij}) = \theta_i^2/\lambda_i$.

We can then model the marginal mean response as

$$\log(E(Y_{ij})) = \log(\theta_i) = \mathbf{X}_i^T \boldsymbol{\beta}.$$

For fixed λ_i , the negative binomial is in the exponential family. Therefore to find the MLE for the negative binomial model, we can alternate the two iterative steps until convergence.

- For fixed λ , fit the GLM to solve for $\boldsymbol{\beta}$ in a regression model.
- For fixed μ , estimate λ using Newton-Raphson method.

The `glm.nb` function in R **MASS** library implements this iterative procedure while `gnlr` can estimate all parameters simultaneously.

Seizure Data

```
> library(MASS)
> s.nb <- glm.nb (seizure ~ progabide * post + offset (log (time)),
+               data = seize.full)
> summary (s.nb)
```

Call:

```
glm.nb(formula = seizure ~ progabide * post + offset(log(time)),
       data = seize.full, init.theta = 1.048836922, link = log)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.1575	-0.8444	-0.4752	0.1339	4.2488

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	1.34761	0.18765	7.182	6.89e-13 ***
progabide	0.02651	0.25882	0.102	0.918
post	0.11080	0.21157	0.524	0.600
progabide:post	-0.10368	0.29196	-0.355	0.723

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

(Dispersion parameter for Negative Binomial(1.0488) family taken to be 1)

Null deviance: 335.32 on 294 degrees of freedom
 Residual deviance: 334.84 on 291 degrees of freedom

AIC: 2031.5

Number of Fisher Scoring iterations: 1

Theta: 1.0488
Std. Err.: 0.0932

2 x log-likelihood: -2021.5290

- Negative-Binomial model result is similar to GEE model's.
- Both methods' s.e. estimates are larger than the naive Poisson regression model (underestimated s.e.).

Conjugated Mixture Models

- The Beta-Binomial and Poisson-Gamma models are examples of **conjugated mixture models** where the marginal distribution has closed form.
- These models are mainly used to account for overdispersion or simple clustered data and are less suitable to study longitudinal data where the correlation structure may be more complicated.
- In Lee and Nelder (1996), they extended this class of model to what they called **hierarchical generalized linear models** which are like *generalized linear mixed models* but the distribution of the random effects need not be normal.

Full Likelihood-Based Inference

Assumptions

- Given the random effects \mathbf{b}_i (a q -vector), the responses Y_{ij} ($i = 1, \dots, m, j = 1, \dots, n_i$) are *independent* and follow a distribution from the exponential family.

$$Y_{ij} | \mathbf{b}_i \sim f(y_{ij} | \mathbf{b}_i, \theta).$$

Let $E(Y_{ij} | \mathbf{b}_i) = \mu_{ij}$ then

$$g(\mu_{ij}) = \eta_{ij} = \mathbf{X}_{ij}^T \boldsymbol{\beta} + \mathbf{Z}_{ij}^T \mathbf{b}_i,$$

where η_i is the linear predictor and g is the link function. \mathbf{X}_{ij} and \mathbf{Z}_{ij} are p - and q -vector of covariates, with \mathbf{Z} often being a subset of \mathbf{X} .

- The random effects \mathbf{b}_i are independent and identically distributed with mean $\mathbf{0}$ and variance $D(\boldsymbol{\alpha})$. Its distribution F is completely specified with parameters $\boldsymbol{\alpha}$, so that $f(\mathbf{b}_i, \boldsymbol{\alpha})$. That is, F does not depend on any covariates.

Likelihood

The likelihood function for $(\boldsymbol{\beta}, \boldsymbol{\alpha})$ is the marginal distribution of \mathbf{Y} ,

$$\mathcal{L}(\boldsymbol{\beta}, \boldsymbol{\alpha}) = \prod_{i=1}^m \int \prod_{j=1}^{n_i} f(y_{ij} | \boldsymbol{\beta}, \mathbf{b}_i) f(\mathbf{b}_i; \boldsymbol{\alpha}) d\mathbf{b}_i.$$

Maximum Likelihood for GLMM

We need specify a distribution for \mathbf{b}_i to use maximum likelihood estimation. For *generalized linear mixed models*, we will assume

$$\mathbf{b}_i \stackrel{iid}{\sim} \mathcal{N}(\mathbf{0}, D(\boldsymbol{\alpha})),$$

where $\boldsymbol{\alpha}$ is the variance parameter for the random effects.

The joint likelihood for $(\boldsymbol{\beta}, \boldsymbol{\alpha})$ is the marginal distribution of \mathbf{Y} ,

$$\mathcal{L}(\boldsymbol{\beta}, \boldsymbol{\alpha}) = \prod_{i=1}^m \int \prod_{j=1}^{n_i} f(y_{ij} | \boldsymbol{\beta}, \mathbf{b}_i) f(\mathbf{b}_i; \boldsymbol{\alpha}) d\mathbf{b}_i.$$

where $f(\mathbf{b}_i; \boldsymbol{\alpha})$ is the multivariate normal density function. Note that there are q levels of integral where q is the dimension of \mathbf{b}_i .

In general, the marginal likelihood has no closed form and the integration is quite difficult.

Estimation Methods for GLMM

- Numerical maximization of the likelihood
 - Expectation-maximization (EM) algorithm
 - Monte Carlo EM
 - Monte Carlo Newton-Raphson
- Numerical evaluation of the likelihood
 - Gaussian-Hermite quadrature
 - Adaptive quadrature
 - Importance sampling and Monte Carlo integration
- Approximate likelihood
 - Penalized quasi-likelihood (PQL), Bias-corrected PQL
 - Linearization
- Generalized estimating equations (GEE)
- Bayesian Markov chain Monte Carlo (Zeger and Karim, 1991; Clayton, 1996)
 - Gibbs sampling
 - Metropolis-Hastings algorithm

Numerical Maximization of the Likelihood

Expectation-Maximization Algorithm for GLMM

- Treat the random effects as missing data, the complete data is (\mathbf{y}, \mathbf{b}) . The complete data log-likelihood is given by

$$\ell(\boldsymbol{\beta}, D; \mathbf{y}, \mathbf{b}) = \sum_i (\log f(\mathbf{y}_i | \boldsymbol{\beta}, \mathbf{b}_i) + \log f(\mathbf{b}_i; D)).$$

- E-step: evaluate the conditional expectation of the complete data log-likelihood

$$E[\ell | \mathbf{Y}; \boldsymbol{\beta}^{(l)}, D^{(l)}]$$

where the expectation is taken with respect to the conditional distribution $f(\mathbf{b} | \mathbf{y})$ using current estimates of $\boldsymbol{\beta}^{(l)}$ and $D^{(l)}$.

- M-step: maximize the expected complete data log-likelihood. Since the parameters $(\boldsymbol{\beta}, D)$ are separated into two terms, the maximization can be done separately

$$\boldsymbol{\beta}^{(l+1)} = \arg \max_{\boldsymbol{\beta}} \sum_i \mathbb{E} \left\{ \log f(\mathbf{y}_i | \boldsymbol{\beta}, \mathbf{b}_i) | \mathbf{Y}; \boldsymbol{\beta}^{(l)}, D^{(l)} \right\}$$

$$D^{(l+1)} = \arg \max_D \sum_i \mathbb{E} \left\{ \log f(\mathbf{b}_i; D) | \mathbf{Y}; \boldsymbol{\beta}^{(l)}, D^{(l)} \right\}$$

- For GLMM, the EM algorithm involves solving estimating equations iteratively:

$$0 = \sum_{i=1}^m \sum_{j=1}^{n_i} \mathbf{x}_{ij} \left\{ y_{ij} - \mathbb{E}_{\mathbf{b}_i | \mathbf{y}_i} [\mu_{ij}(\mathbf{b}_i) | \mathbf{y}_i] \right\},$$

$$0 = \frac{1}{2} D^{-1} \left\{ \sum_{i=1}^m \mathbb{E}_{\mathbf{b}_i | \mathbf{y}_i} (\mathbf{b}_i \mathbf{b}_i^T | \mathbf{y}_i) \right\} D^{-1} - \frac{m}{2} D^{-1},$$

where $\mu_{ij}(\mathbf{b}_i) = g^{-1}(\mathbf{x}_{ij}^T \boldsymbol{\beta} + \mathbf{z}_{ij}^T \mathbf{b}_i)$.

- The dimension of the integration involved in the conditional expectation $\mathbb{E}_{\mathbf{b}_i | \mathbf{y}_i}$ is q . When $q = 1$ or 2 , numerical integration can be implemented; for higher q , Monte Carlo integration can be used.

Monte Carlo EM

- The conditional expectation required in the EM algorithm typically does not have a closed form since the conditional distribution is

$$f(\mathbf{b} | \mathbf{y}) = \frac{f(\mathbf{y}, \mathbf{b})}{f(\mathbf{y})} = \frac{f(\mathbf{y}, \mathbf{b})}{\int f(\mathbf{y}, \mathbf{b}) d\mathbf{b}}$$

and the integration in the denominator is what we want to avoid.

- We can draw dependent samples from $f(\mathbf{b} | \mathbf{y})$ using Metropolis algorithm without calculating $f(\mathbf{y})$ and use Monte Carlo method to calculate the conditional expectation (McCulloch, 1997).

Monte Carlo Newton-Raphson

$$\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} + E_{\mathbf{b}|\mathbf{y}} \left[\mathbf{X}^T \mathbf{W}(\boldsymbol{\beta}^{(k)}, \mathbf{b}) \mathbf{X} \right]^{-1} \mathbf{X}^T E_{\mathbf{b}|\mathbf{y}} \left[\mathbf{W}(\boldsymbol{\beta}^{(k)}, \mathbf{b}) \frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\mu}} \bigg|_{\boldsymbol{\beta}^{(k)}} \{ \mathbf{y} - \boldsymbol{\mu}(\boldsymbol{\beta}^{(k)}, \mathbf{b}) \} \right]$$

where

$$\mathbf{W}(\boldsymbol{\beta}^{(k)}, \mathbf{b}) = \left(\frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\eta}} \right)^T \mathbf{V}^{-1} \left(\frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\eta}} \right) \bigg|_{\boldsymbol{\beta}^{(k)}}.$$

- The conditional expectation can be calculated using the Monte Carlo integration with the Metropolis algorithm described above.
- When the likelihood is not unimodal, EM may converge to the wrong place and Newton-Raphson may not converge.
- For accurate estimates, many iterations are required. It might be desirable to combine difference approaches.

Numerical Evaluation of the Likelihood

Gaussian-Hermite Quadrature

The Gaussian-Hermite quadrature uses a fixed set of Q nodes (quadrature points) and weights (z_q, w_q) to approximate an integral:

$$\int_{-\infty}^{\infty} f(z)\Phi(z)dz \approx \sum_{q=1}^Q w_q f(z_q).$$

Consider a single scalar random effect $b_i \sim \mathcal{N}(0, \tau^2)$, the GLMM likelihood for one subject i is given by

$$\begin{aligned}\mathcal{L}(\boldsymbol{\beta}, \tau; \mathbf{y}_i) &= \int \left\{ \prod_j^{n_i} f(y_{ij} | \boldsymbol{\beta}, b_i) \right\} f(b_i; \tau) db_i \\ &= \int \left[\exp \left\{ \sum_j \log f(y_{ij} | \boldsymbol{\beta}, b_i) \right\} \right] \frac{1}{\tau} \Phi \left(\frac{b_i}{\tau} \right) db_i \\ &\approx \sum_{q=1}^Q w_q \left[\exp \left\{ \sum_j \log f(y_{ij} | \boldsymbol{\beta}, b_i = z_q \tau) \right\} \right]\end{aligned}$$

- The higher Q , the more accurate the approximation will be.
- Gaussian-Hermite quadrature is less accurate when τ increases.
- The z_q 's are chosen based on $\phi(z)$ rather than $f(z)$. Depending on the support of $f(z)$, z_q may or may not lie in the region of interest. The quadrature only performs well when $f(z)\phi(z)$ behaves like $\phi(z)$.

Adaptive Gaussian Quadrature

The adaptive Gaussian quadrature centers the Gaussian approximation at the posterior mode of the random effects and scaled. For subject i ,

$$\begin{aligned}
 \mathcal{L}(\boldsymbol{\beta}, \tau; \mathbf{y}_i) &= \int \left[\exp \left\{ \sum_j \log f(y_{ij} | \boldsymbol{\beta}, b_i) \right\} \right] \frac{1}{\tau} \Phi \left(\frac{b_i}{\tau} \right) db_i \\
 &= \int \left[\exp \left\{ \sum_j \log f(y_{ij} | \boldsymbol{\beta}, b_i) \right\} \frac{1}{\tau} \frac{\Phi(b_i/\tau)}{\Phi((b_i - c)/s)} \right] \times \Phi \left(\frac{b_i - c}{s} \right) db_i \\
 &= \int \exp \left\{ \sum_j \log f(y_{ij} | \boldsymbol{\beta}, b_i = c + sz) \right\} \frac{1}{\tau} \times \frac{s\Phi((c + sz)/\tau)}{\Phi(z)} \Phi(z) dz \\
 &\approx \sum_{q=1}^Q w_q \cdot \exp \left\{ \sum_j \log f(y_{ij} | \boldsymbol{\beta}, b_i = c + sz_q) \right\} \frac{1}{\tau} \times \frac{s\Phi((c + sz_q)/\tau)}{\Phi(z_q)}
 \end{aligned}$$

where c is the posterior mode of $\log f(z)\Phi(z)$ and s is the posterior curvature, $\left[\frac{\partial^2}{\partial z^2} \log[f(z)\Phi(z)] \right]^{-1/2}$.

- In the adaptive quadrature method, different values c_i and s_i are used for each subject.
- Adaptive quadrature requires less points to achieve the same accuracy as fixed-point quadrature.
- The quadrature methods do not perform very well for higher-dimension integrations.

Monte Carlo Methods for Integration

Suppose we want to calculate this integral:

$$E(h(X)) = \int h(x)f(x)dx$$

where X can be a vector and f is a density function.

Naive Monte Carlo

If we can draw a random iid sample $x^{(1)}, x^{(2)}, \dots, x^{(M)}$ from f , then

$$\int h(x)f(x)dx \approx \frac{1}{M} \sum_j^M h(x^{(j)}).$$

It also works when *dependent* samples are drawn from a Markov chain with stationary distribution $f(x)$ (Markov chain Monte Carlo).

Importance Sampling

It is often difficult or impossible to draw independent samples from an arbitrary distribution f (which may not even have a closed form). We can find another density function g whose support includes that of f . Since

$$\int h(x)f(x)dx = \int \frac{h(x)f(x)}{g(x)}g(x)d(x),$$

we can draw a random iid sample $x^{(1)}, x^{(2)}, \dots, x^{(M)}$ from g , then

$$\int h(x)f(x)dx \approx \frac{1}{M} \sum_j^M w_j h(x^{(j)}) \quad \text{where} \quad w_j = \frac{f(x^{(j)})}{g(x^{(j)})}.$$

While the algorithm works for any g , the efficiency depends on the choice of g (the variance of w_j).

Simulated Maximum Likelihood for GLMM

Geyer and Thompson (1992) and Gelfand and Carlin (1993) suggested simulation to estimate the GLMM likelihood which can then be numerically maximized.

$$\begin{aligned}\mathcal{L}(\boldsymbol{\beta}, D | \mathbf{y}) &= \int f(\mathbf{y} | \boldsymbol{\beta}, \mathbf{b}) f(\mathbf{b} | D) d\mathbf{b} \\ &= \int \frac{f(\mathbf{y} | \boldsymbol{\beta}, \mathbf{b}) f(\mathbf{b} | D)}{g(\mathbf{b})} g(\mathbf{b}) d\mathbf{b} \\ &\approx \frac{1}{M} \sum_{k=1}^M \frac{f(\mathbf{y} | \boldsymbol{\beta}, \mathbf{b}^{(k)}) f(\mathbf{b}^{(k)} | D)}{g(\mathbf{b}^{(k)})}\end{aligned}$$

where $\mathbf{b}^{(k)}$ is drawn from the importance sampling distribution $g(\mathbf{b})$.

- To maximize the likelihood, we need evaluate it at different values of $(\boldsymbol{\beta}, D)$. The advantage of IS method is that only one sample from g is needed.
- To improve Monte Carlo efficiency, multiple g 's that depend on the value of $(\boldsymbol{\beta}, D)$ can be used.
- When g itself is difficult to draw independent samples from, Markov chain Monte Carlo (MCMC) can be used to draw dependent samples from a Markov chain with g as stationary distribution.

Approximate Likelihood

Penalized Quasi-Likelihood (PQL)

- Use conditional modes rather than conditional means in the score function, and thus avoid integrations.
 - This is equivalent to approximating the conditional distribution of \mathbf{b}_i given \mathbf{y}_i by a Gaussian distribution with the same mode and curvature (incorporated into the M-step).
 - In other words, we plug in the posterior mode or BLUP $\hat{\mathbf{b}}$ for \mathbf{b} .
- From another perspective, consider the random effects \mathbf{b} as fixed parameters, we can maximize the joint likelihood with respect to $(\boldsymbol{\beta}, \mathbf{b})$

$$\log f(\mathbf{y} | \boldsymbol{\beta}, \mathbf{b}) - \frac{1}{2} \mathbf{b}^T \mathbf{D}^{-1} \mathbf{b}.$$

This *penalized likelihood* idea is similar to REML, and this approach is called *penalized quasi-likelihood* (Breslow and Clayton, 1993).

- Direct differentiation of the penalized likelihood leads to the score equations for $(\boldsymbol{\beta}, \mathbf{b})$ (assuming canonical exponential family)

$$\begin{bmatrix} X^T(\mathbf{y} - \boldsymbol{\mu}) \\ Z^T(\mathbf{y} - \boldsymbol{\mu}) - D^{-1}\mathbf{b} \end{bmatrix} = \mathbf{0},$$

where $g(\boldsymbol{\mu}) = X\boldsymbol{\beta} + Z\mathbf{b}$. Alternating solving these two sets of score equations gives the previous algorithm which can be implemented easily in standard software.

- In PQL, only the mean and variance need to be specified in the conditional mean model of $\mathbf{y} \mid \mathbf{b}$.

PQL and Laplace Approximation

More formally, PQL can be derived via Laplace approximation to the GLMM likelihood,

$$\mathcal{L}(\boldsymbol{\beta}, D) \propto |D|^{-1/2} \int_{\mathbb{R}^q} \exp(h(\mathbf{b})) d\mathbf{b},$$

where

$$h(\mathbf{b}) = \mathbf{y}^T (X\boldsymbol{\beta} + Z\mathbf{b}) - \mathbf{1}^T b(X\boldsymbol{\beta} + Z\mathbf{b}) - \frac{1}{2} \mathbf{b}^T D^{-1} \mathbf{b}.$$

The Laplace approximation of $\mathcal{L}(\boldsymbol{\beta}, D)$ starts with the Taylor series approximation of $h(\mathbf{b})$

$$h(\mathbf{b}) \approx h(\mathbf{b}_0) + \dot{h}(\mathbf{b}_0)(\mathbf{b} - \mathbf{b}_0) + \frac{1}{2}(\mathbf{b} - \mathbf{b}_0)^T \ddot{h}(\mathbf{b}_0)(\mathbf{b} - \mathbf{b}_0).$$

We can (for example) choose \mathbf{b}_0 to solve

$$\dot{h}(\mathbf{b}_0) = \mathbf{0}$$

that is, the mode.

We then have

$$\begin{aligned}
 h(\mathbf{b}) &\approx h(\mathbf{b}_0) + \frac{1}{2}(\mathbf{b} - \mathbf{b}_0)^T \ddot{h}(\mathbf{b}_0)(\mathbf{b} - \mathbf{b}_0), \\
 \exp(h(\mathbf{b})) &\approx \\
 \int \exp(h(\mathbf{b})) d\mathbf{b} &\approx
 \end{aligned}$$

Note that

$$\int_{\mathbb{R}^q} \frac{1}{(2\pi)^{q/2}} |\Sigma|^{-1/2} \exp \left\{ -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}) \right\} = 1.$$

We have (up to a constant)

$$\begin{aligned}
 \mathcal{L}(\boldsymbol{\beta}, D) &\approx |D|^{-1/2} |-\ddot{h}(\mathbf{b}_0)|^{-1/2} \exp(h(\mathbf{b}_0)) \\
 \log \mathcal{L}(\boldsymbol{\beta}, D) &\approx
 \end{aligned}$$

Here

$$\dot{h}(\mathbf{b}) = \left\{ \mathbf{y} - \dot{b}(X\boldsymbol{\beta} + Z\mathbf{b}) \right\}^T Z - \mathbf{b}^T D^{-1}$$

$$\ddot{h}(\mathbf{b}) = -Z^T \text{diag}(\ddot{b}(X\boldsymbol{\beta} + Z\mathbf{b}))Z - D^{-1}$$

$$| -\ddot{h}(\mathbf{b}_0) | =$$

$$\log | -\ddot{h}(\mathbf{b}_0) | =$$

The resulting approximation (up to a constant) to the log-likelihood is

$$\ell(\boldsymbol{\beta}, D) \approx \mathbf{y}^T (X\boldsymbol{\beta} + Z\mathbf{b}_0) - \mathbf{1}^T b(X\boldsymbol{\beta} + Z\mathbf{b}_0) - \frac{1}{2} \mathbf{b}_0^T D^{-1} \mathbf{b}_0 - \frac{1}{2} \log | I + DZ^T \text{diag}(\ddot{b}(X\boldsymbol{\beta} + Z\mathbf{b}_0))Z |.$$

- PQL uses an additional approximation by assuming $\ddot{b}(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b}_0)$ to be relatively constant with respect to $\boldsymbol{\beta}$ hence can be dropped and we are left with

$$\ell(\boldsymbol{\beta}, D) \approx h(\mathbf{b}_0) = \log f(\mathbf{y} \mid \boldsymbol{\beta}, \mathbf{b}_0).$$

- Expand h at the true mean of \mathbf{b} leads to another approximation of the likelihood.
- The approximation adopted by PQL induces bias in the estimates of $\boldsymbol{\beta}$. Better approximating can be achieved by considering high-order expansions. Breslow and Lin (1995) and Lin and Breslow (1996) derived the bias-corrected estimates.

Linearization

Another approximation approach is based the idea of linearization:

- The data Y is decomposed into the mean $\boldsymbol{\mu}$ which is a non-linear function of the linear predictor and an appropriate error term.

$$E(\mathbf{Y} \mid \mathbf{b}) = \boldsymbol{\mu} = g^{-1}(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b}) = g^{-1}(\boldsymbol{\eta})$$

$$\text{Var}(\mathbf{Y} \mid \mathbf{b}) = \mathbf{C}^{1/2} \mathbf{R} \mathbf{C}^{1/2}$$

$$\mathbf{b} \sim \mathcal{N}(\mathbf{0}, \mathbf{G})$$

- Taylor series expansion of $\boldsymbol{\mu}$ about $\tilde{\boldsymbol{\beta}}$ and $\tilde{\mathbf{b}}$:

$$g^{-1}(\boldsymbol{\eta}) \approx g^{-1}(\tilde{\boldsymbol{\eta}}) + \tilde{\Delta}\mathbf{X}(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}) + \tilde{\Delta}\mathbf{Z}(\mathbf{b} - \tilde{\mathbf{b}})$$

where

$$\tilde{\Delta} = \left. \frac{\partial}{\partial \boldsymbol{\eta}} g^{-1}(\boldsymbol{\eta}) \right|_{\tilde{\boldsymbol{\beta}}, \tilde{\boldsymbol{b}}}$$

- Rearrange:

$$\tilde{\Delta}^{-1}(\boldsymbol{\mu} - g^{-1}(\tilde{\boldsymbol{\eta}})) + (X\tilde{\boldsymbol{\beta}} + Z\tilde{\boldsymbol{b}}) \approx X\boldsymbol{\beta} + Z\boldsymbol{b}$$

- Define pseudo-data \mathbf{Y}^* :

$$\mathbf{Y}^* \equiv \tilde{\Delta}^{-1}(\mathbf{Y} - g^{-1}(\tilde{\boldsymbol{\eta}})) + (X\tilde{\boldsymbol{\beta}} + Z\tilde{\boldsymbol{b}})$$

we have

$$\mathbb{E}(\mathbf{Y}^* | \mathbf{b}) = X\boldsymbol{\beta} + Z\mathbf{b}$$

$$\text{Var}(\mathbf{Y}^* | \mathbf{b}) = \tilde{\Delta}^{-1} C^{1/2} R C^{1/2} \tilde{\Delta}^{-1}$$

- Thus we can consider the linear mixed model:

$$\mathbf{Y}^* = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b} + \boldsymbol{\epsilon}$$

where $\boldsymbol{\epsilon}$ has a Gaussian distribution.

- The model on pseudo-data can be fitted using ML or REML.
- Double-iterative algorithm:
 - Start with $\tilde{\boldsymbol{\beta}}, \tilde{\mathbf{b}}$, compute the pseudo data and fit the linear mixed model, which may itself require an iterative algorithm.
 - Update $\tilde{\boldsymbol{\beta}}, \tilde{\mathbf{b}}$.
- Choices of $\tilde{\boldsymbol{\beta}}, \tilde{\mathbf{b}}$:
 - Penalized quasi-likelihood, subject-specific expansion: BLUP of the random effects.

$$\tilde{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}} \qquad \tilde{\mathbf{b}} = \hat{\mathbf{b}}$$

– Marginal quasi-likelihood, population-average expansion:

$$\tilde{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}} \quad \tilde{\boldsymbol{b}} = \mathbf{0}$$

- The full likelihood of \mathbf{Y} is not required (hence 'quasi-likelihood').
- Accuracy depends on the distribution assumption of the pseudo-data. There can be substantial bias for binary data with relatively small number of repeated observations (not “sufficiently continuous”).

Fitting GLMM: Software

- R
 - `glmmPQL` (**MASS**): penalized quasi-likelihood, allows the use of an additional correlation structure.
 - `glmmML` (**glmmML**): maximum likelihood using adaptive Gaussian quadrature or Laplace (default) methods; random intercept only model.
 - `glmm` (**repeated**): Gaussian-Hermite quadrature, models with random intercept only.
 - `gnlmix` (**repeated**): non-linear regression with mixed random effects for the location parameters. Non-Gaussian mixing distributions are allowed.
 - `glmm` (**GLMMGibbs**): Gibbs sampling.
- SAS:
 - `PROC NLMIXED` uses adaptive Gaussian quadrature method by default.
 - `PROC GLIMMIX` uses pseudo-likelihood estimation based on linearization to fit models with Gaussian random effects.
- BUGS (OpenBUGS/WinBUGS)

2 × 2 Crossover Trial

Gaussian-Hermite quadrature using R function `glmm(repeated)`

```
> library(repeated)
> xover.glmm <- glmm(y ~ trtA, family=binomial, nest=id, data=xover)
There were 43 warnings (use warnings() to see them)
> summary(xover.glmm)
Call:
glmm(y ~ trtA, family = binomial, nest = id, data = xover)
```

Deviance Residuals:

	Min	1Q	Median	3Q	Max
	-1.576e+00	-1.027e-05	1.922e-06	3.595e-03	1.392e+00

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	2.2029	0.9065	2.430	0.01510 *
trtA	1.4845	0.6392	2.322	0.02021 *
sd	5.6464	1.7462	3.234	0.00122 **

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 166.65 on 133 degrees of freedom

Residual deviance: 138.38 on 131 degrees of freedom

AIC: 144.38

Number of Fisher Scoring iterations: 20

Normal mixing variance: 31.88128

Adaptive Gaussian quadrature using SAS PROC NLMIXED

```
data xover;
infile "C:/xover1.data";
input id class y intercept trt period xover BA;
trtA=1-trt;
proc nlmixed data = xover qpoints = 20;
  parms beta0 = 1 beta1 = 2 tau = 4;
  eta = beta0 + b1 + beta1 * trtA ;
  p = exp (eta) / (1 + exp (eta));
  model y ~ binary (p);
  random b1 ~ normal (0, tau ** 2) subject = id;
run;
```

Fit Statistics

-2 Log Likelihood	138.3
AIC (smaller is better)	144.3
AICC (smaller is better)	144.4
BIC (smaller is better)	150.9

Parameter Estimates

Parameter	Estimate	Standard Error	DF	t Value	Pr > t	Alpha	Lower	Upper	Gradient
beta0	1.5242	0.8594	66	1.77	0.0808	0.05	-0.1917	3.2400	4.953E-7
beta1	1.6029	0.7692	66	2.08	0.0410	0.05	0.06721	3.1385	5.455E-7
tau	4.3681	1.4909	66	2.93	0.0047	0.05	1.3914	7.3449	-8.16E-7

Comparison of results

	GEE	CLR	R glmm	SAS nlmixed
Intercept	0.52 (0.25)		2.2 (0.91)	1.52 (0.86)
Treatment	0.56 (0.24)	1.61 (0.78)	1.48 (0.64)	1.60 (0.77)
$D^{1/2}$			5.65 (1.75)	4.37 (1.49)
ρ	0.62 (0.1)			

- There is substantial variation in baseline risk, $D^{1/2} = 4 \sim 6$.
- There is strong within-subject association, $\rho = 0.6$.

3 × 3 Crossover Trial

Penalized quasi-likelihood method using R function `glmmPQL(MASS)`

```
> library (MASS)
> xover.glmm <- glmmPQL (fixed = relief ~ tx2 + tx3 + p2 + p3
>                        + ptx2 + ptx3,
+                        random = ~ 1 | id, family = "binomial",
+                        data = xover3)
```

iteration 1

iteration 2

```
> summary (xover.glmm)
```

Linear mixed-effects model fit by maximum likelihood

Data: xover3

	AIC	BIC	logLik
	1194.117	1226.094	-588.0585

Random effects:

x Formula: ~1 | id

(Intercept) Residual

StdDev: 0.04334977 1.002844

Variance function:

Structure: fixed weights

Formula: ~invwt

Fixed effects: relief ~ tx2 + tx3 + p2 + p3 + ptx2 + ptx3

	Value	Std.Error	DF	t-value	p-value
(Intercept)	-1.0865137	0.3337615	166	-3.255359	0.0014
tx2	2.1056712	0.4084809	166	5.154882	0.0000
tx3	2.0683181	0.3895146	166	5.309989	0.0000
p2	0.4123220	0.4685381	166	0.880018	0.3801
p3	0.5865394	0.4830292	166	1.214294	0.2264
ptx2	-0.1282313	0.5101216	166	-0.251374	0.8018
ptx3	-0.9285876	0.4954788	166	-1.874122	0.0627

Correlation:

	(Intr)	tx2	tx3	p2	p3	ptx2
tx2	-0.539					
tx3	-0.599	0.472				
p2	-0.232	-0.232	-0.198			
p3	-0.215	-0.237	-0.205	0.699		
ptx2	-0.259	0.425	0.269	-0.574	-0.593	
ptx3	-0.174	0.059	0.347	-0.532	-0.550	0.511

Standardized Within-Group Residuals:

	Min	Q1	Med	Q3	Max
	-2.2261231	-0.5793593	0.4702024	0.5989729	2.2217509

Number of Observations: 258

Number of Groups: 86

Maximum likelihood method using adaptive Gaussian quadrature method for random intercept only model, using R function `glmmML(glmmML)`

```
> library (glmmML)
> xover.glmmML1 <- glmmML (relief ~ tx2 + tx3 + p2 + p3 +
+                          ptx2 + ptx3,
+                          cluster = xover3$id, family = binomial, data = xover3,
+                          n.points = 20, method="ghq")
> xover.glmmML1
Call:  glmmML(formula = relief ~ tx2 + tx3 + p2 + p3 + ptx2 + ptx3,
family = binomial, data = xover3, cluster = xover3$id, method = "ghq", n.points = 20)
```

	coef	se(coef)	z	Pr(> z)
(Intercept)	-1.0866	0.3281	-3.3119	9.27e-04
tx2	1.9494	0.3888	5.0135	5.35e-07
tx3	2.2223	0.3946	5.6318	1.78e-08
p2	0.4141	0.4610	0.8984	3.69e-01
p3	0.5885	0.4753	1.2383	2.16e-01
ptx2	-0.1922	0.5070	-0.3791	7.05e-01
ptx3	-0.8309	0.4818	-1.7244	8.46e-02

Scale parameter in mixing distribution: 4.615e-05 gaussian

Std. Error: 0.5071

LR p-value for H₀: sigma = 0: 0.5

Residual deviance: 283.2 on 250 degrees of freedom AIC: 299.2

- The two methods give consistent point estimate and slightly different s.e. estimates.

Indonesian Children Health Study (ICHS)

Random intercept and random slope (for time) model using glmmPQL(MASS)

```
> library(MASS)
> ICHS <- read.table("../data/ICHS.dat", header = TRUE)
> ichs.glmm <- glmmPQL (RESPONSE ~ VITA + AGE + I(AGE^2) +
+                       GENDER + TIME,
+                       random = ~ TIME | ID,
+                       data = ICHS, family = "binomial")
> ICHS$AGE <- ICHS$AGE * 12
> summary (ichs.glmm)
```

Linear mixed-effects model fit by maximum likelihood

Data: ICHS

	AIC	BIC	logLik
	7918.793	7971.926	-3949.397

Random effects:

Formula: ~TIME | ID

Structure: General positive-definite, Log-Cholesky parametrization

	StdDev	Corr
(Intercept)	2.8959344	(Intr)
TIME	0.1635745	-0.482
Residual	0.6537811	

Variance function:

Structure: fixed weights

Formula: ~invwt

Fixed effects: RESPONSE ~ VITA + AGE + I(AGE^2) + GENDER + TIME

	Value	Std.Error	DF	t-value	p-value
(Intercept)	-2.0547139	0.7325637	1249	-2.804826	0.0051
VITA	0.4156394	0.3749237	245	1.108597	0.2687
AGE	0.6095973	0.4359468	245	1.398330	0.1633
I(AGE^2)	-0.0974293	0.0564336	245	-1.726441	0.0855
GENDER	-0.9629119	0.3613544	245	-2.664730	0.0082
TIME	0.0396433	0.0169233	1249	2.342525	0.0193

Correlation:

	(Intr)	VITA	AGE	I(AGE^	GENDER
VITA	-0.085				
AGE	-0.850	-0.130			
I(AGE^2)	0.770	0.128	-0.976		
GENDER	-0.189	0.028	-0.073	0.066	
TIME	-0.182	0.005	-0.009	0.008	-0.003

Standardized Within-Group Residuals:

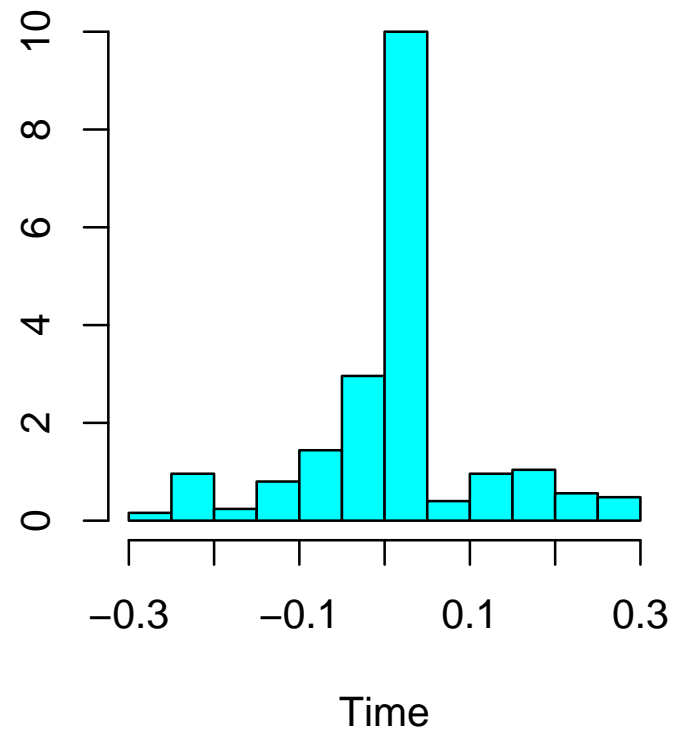
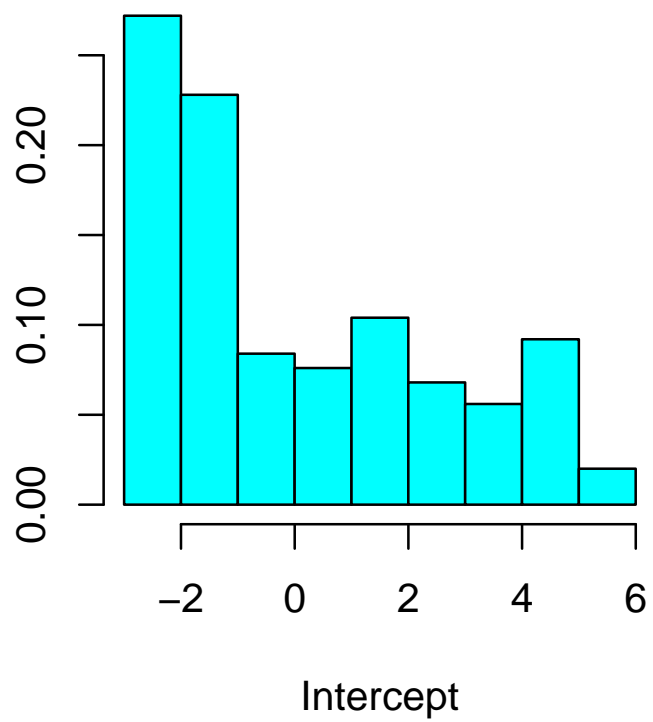
Min	Q1	Med	Q3	Max
-3.0848406	-0.3567323	-0.2277855	0.3358386	3.4221396

Number of Observations: 1500

Number of Groups: 250

```
> truehist (ranef (ichs.glmm)[,1], xlab = "Intercept")
```

```
> truehist (ranef (ichs.glmm)[,2], xlab = "Time")
```



Software differences (using the random intercept only model as an example)

For GLMM, difference algorithms and software may give quite different answers.

```
> library (MASS)
> summary (glmmPQL (RESPONSE ~ VITA + AGE + I(AGE^2) +
+                 GENDER + TIME,
+                 random = ~ 1 | ID,
+                 data = ICHS, family = "binomial"))
```

Linear mixed-effects model fit by maximum likelihood

Data: ICHS

	AIC	BIC	logLik
	7709.842	7752.348	-3846.921

Random effects:

Formula: ~1 | ID

(Intercept) Residual

StdDev: 2.330407 0.729902

Variance function:

Structure: fixed weights

Formula: ~invwt

Fixed effects: RESPONSE ~ VITA + AGE + I(AGE^2) + GENDER + TIME

	Value	Std.Error	DF	t-value	p-value
(Intercept)	-1.8854716	0.6745516	1249	-2.795148	0.0053
VITA	0.3995431	0.3479949	245	1.148129	0.2520
AGE	0.5629374	0.4046444	245	1.391190	0.1654

I(AGE^2)	-0.0890962	0.0523895	245	-1.700650	0.0903
GENDER	-0.8961807	0.3353563	245	-2.672324	0.0080
TIME	0.0327428	0.0114319	1249	2.864150	0.0043

Correlation:

	(Intr)	VITA	AGE	I(AGE^	GENDER
VITA	-0.084				
AGE	-0.859	-0.131			
I(AGE^2)	0.778	0.128	-0.976		
GENDER	-0.191	0.027	-0.071	0.064	
TIME	-0.132	0.002	0.003	-0.003	-0.006

Standardized Within-Group Residuals:

	Min	Q1	Med	Q3	Max
	-3.1088570	-0.5485138	-0.2382666	0.3486534	3.6635217

Number of Observations: 1500

Number of Groups: 250

```
> library(glmml)
> glmml (RESPONSE ~ VITA + AGE + I(AGE^2) + GENDER + TIME,
+       cluster = ID,
+       data = ICHS, family = binomial,
+       n.points = 20,
+       method="ghq")
```

```
Call: glmml(formula = RESPONSE ~ VITA + AGE + I(AGE^2) + GENDER + TIME, family = binomial, data =
```

	coef	se(coef)	z	Pr(> z)
(Intercept)	-2.30652	0.86359	-2.671	0.00757
VITA	0.50403	0.43939	1.147	0.25100
AGE	0.72195	0.51338	1.406	0.16000
I(AGE^2)	-0.11440	0.06662	-1.717	0.08590
GENDER	-1.14038	0.42613	-2.676	0.00745
TIME	0.03422	0.01602	2.136	0.03260

```
Scale parameter in mixing distribution: 2.819 gaussian
Std. Error: 0.2565
```

```
Residual deviance: 1330 on 1493 degrees of freedom      AIC: 1344
```

```
> library (repeated)
> ichs.glmm2 <- glmm (RESPONSE ~ VITA + AGE + I(AGE^2) +
+                     GENDER + TIME, nest = ID,
+                     data = ICHS, family = "binomial",
+                     points = 20)
There were 17 warnings (use warnings() to see them)
> summary (ichs.glmm2)
Call:
glmm(RESPONSE ~ VITA + AGE + I(AGE^2) + GENDER + TIME, nest = ID,
     data = ICHS, family = "binomial", points = 20)
```

Deviance Residuals:

	Min	1Q	Median	3Q	Max
	-1.732e+00	-1.233e-04	-4.726e-11	1.846e-13	1.842e+00

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-2.32668	0.37305	-6.237	4.46e-10	***
VITA	0.60359	0.17127	3.524	0.000425	***
AGE	0.79657	0.20543	3.878	0.000105	***
I(AGE^2)	-0.12761	0.02686	-4.751	2.03e-06	***
GENDER	-1.21289	0.16934	-7.163	7.91e-13	***
TIME	0.03423	0.01594	2.147	0.031802	*
sd	2.84078	0.15638	18.166	< 2e-16	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1822.3 on 1499 degrees of freedom

Residual deviance: 1328.7 on 1493 degrees of freedom

AIC: 1342.7

Number of Fisher Scoring iterations: 17

Normal mixing variance: 8.070035

SAS PROC GLIMMIX

```

data ichs;
    infile 'ICHS.dat' firstobs = 2;
    input id response time gender vita age;
run;

proc glimmix data = ichs method = RSPL;
    class id;
    model response (event = '1') = vita age age*age
        gender time / dist = binary solutions;
    nloptions maxit = 100;
    random intercept / subject = id;
run;

```

Solutions for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > t
Intercept	-1.5583	0.5826	245	-2.67	0.0080
vita	0.3305	0.2978	1249	1.11	0.2673
age	0.4674	0.3474	1249	1.35	0.1787
age*age	-0.07484	0.04507	1249	-1.66	0.0971
gender	-0.7758	0.2869	1249	-2.70	0.0070
time	0.02685	0.01414	1249	1.90	0.0577

SAS PROC NLMIXED for model with random intercept only

```

proc nlmixed data = ichs qpoints = 20;
  parms beta0 = -1.5 beta1 = 0.3 beta2 = 0.5 beta3 = 0
    beta4 = -0.8 beta5 = 0 tau = 3;
  eta = beta0 + b1 + beta1 * vita + beta2 * age +
    beta3 * age * age + beta4 * gender +
    beta5 * time;
  p = exp (eta) / (1 + exp (eta));
  model response ~ binary (p);
  random b1 ~ normal (0, tau ** 2) subject = id;
run;

```

Parameter Estimates

Parameter	Standard								
	Estimate	Error	DF	t Value	Pr > t	Alpha	Lower	Upper	Gradient
beta0	-2.3064	0.8635	249	-2.67	0.0081	0.05	-4.0071	-0.6056	-0.00182
beta1	0.5037	0.4394	249	1.15	0.2527	0.05	-0.3616	1.3691	-0.00228
beta2	0.7219	0.5134	249	1.41	0.1609	0.05	-0.2892	1.7330	-0.00893
beta3	-0.1144	0.06662	249	-1.72	0.0872	0.05	-0.2456	0.01681	-0.04717
beta4	-1.1402	0.4261	249	-2.68	0.0079	0.05	-1.9794	-0.3010	-0.00007
beta5	0.03422	0.01602	249	2.14	0.0337	0.05	0.002667	0.06577	-0.03968
tau	2.8189	0.2566	249	10.99	<.0001	0.05	2.3135	3.3243	-0.00252

SAS PROC NLMIXED for model with random intercept and random slope

```
proc nlmixed data = ichs qpoints = 20;
  parms beta0 = -1.5 beta1 = 0.3 beta2 = 0.5 beta3 = 0
    beta4 = -0.8 beta5 = 0 s1 = 3 c1 = -0.5 s2 = 0.2;
  eta = beta0 + b1 + beta1 * vita + beta2 * age +
    beta3 * age * age + beta4 * gender +
    beta5 * time + b2 * time;
  p = exp (eta) / (1 + exp (eta));
  model response ~ binary (p);
  random b1 b2 ~ normal ([0, 0], [s1, c1, s2])
    subject = id;
run;
```

Parameter Estimates

Parameter	Estimate	Standard Error	DF	t Value	Pr > t	Alpha	Lower	Upper	Gradient
beta0	-2.4632	0.9023	248	-2.73	0.0068	0.05	-4.2403	-0.6861	0.00001
beta1	0.4867	0.4512	248	1.08	0.2818	0.05	-0.4020	1.3755	3.461E-6
beta2	0.7655	0.5252	248	1.46	0.1462	0.05	-0.2689	1.7998	0.000079
beta3	-0.1221	0.06840	248	-1.79	0.0755	0.05	-0.2568	0.01261	0.000515
beta4	-1.1531	0.4369	248	-2.64	0.0088	0.05	-2.0136	-0.2926	8.488E-6
beta5	0.04592	0.02416	248	1.90	0.0585	0.05	-0.00166	0.09351	0.000171
s1	9.8096	2.7618	248	3.55	0.0005	0.05	4.3701	15.2492	-2.05E-7
c1	-0.1111	0.1395	248	-0.80	0.4266	0.05	-0.3858	0.1636	2.188E-6
s2	0.004986	0.006383	248	0.78	0.4354	0.05	-0.00759	0.01756	0.00005

SAS PROC GENMOD for marginal model

```
proc genmod data=ichs DESCENDING;
  class id;
  model response = vita age age*age  gender time  / dist = bin;
  repeated subject = id / type=cs;
run;
```

Analysis Of GEE Parameter Estimates

Empirical Standard Error Estimates

		Standard	95% Confidence			
Parameter	Estimate	Error	Limits		Z	Pr > Z
Intercept	-1.0904	0.4477	-1.9679	-0.2130	-2.44	0.0149
vita	0.2211	0.2232	-0.2164	0.6586	0.99	0.3220
age	0.3462	0.2527	-0.1490	0.8414	1.37	0.1706
age*age	-0.0565	0.0319	-0.1190	0.0059	-1.77	0.0761
gender	-0.5780	0.2168	-1.0029	-0.1531	-2.67	0.0077
time	0.0170	0.0083	0.0007	0.0333	2.04	0.0409

	GLMM with Random Intercept					GEE
	glmmPQL	glmmML	glmm	SAS GLIMMIX	SAS NL MIXED	SAS GENMOD
Intercept	-1.89 (0.67)	-2.31 (0.86)	-2.33 (0.37)	-1.56 (0.58)	-2.31 (0.86)	-1.09 (0.45)
vita	0.40 (0.35)	0.50 (0.44)	0.60 (0.17)	0.33 (0.30)	0.50 (0.44)	0.22 (0.22)
age	0.56 (0.40)	0.72 (0.51)	0.80 (0.21)	0.47 (0.35)	0.72 (0.51)	0.35 (0.25)
age ²	-0.09 (0.05)	-0.11 (0.07)	-0.13 (0.03)	-0.07 (0.05)	-0.11 (0.07)	-0.06 (0.03)
gender	-0.90 (0.34)	-1.14 (0.43)	-1.21 (0.17)	-0.78 (0.29)	-1.14 (0.43)	-0.58 (0.22)
time	0.03 (0.01)	0.03 (0.02)	0.03 (0.02)	0.03 (0.01)	0.03 (0.02)	0.02 (0.01)
τ	2.33	2.82 (0.26)	2.84 (0.16)	3.48 (0.47)	2.82 (0.26)	

- `glmm(repeated)` uses Gaussian-Hermite quadrature. `glmmML(glmmML)` in R and `PROC NL MIXED` in SAS, both using adaptive Gaussian quadrature methods, give the same results.
- Fitting model with random intercept and random slope using SAS `PROC NL MIXED` is quite time consuming.

Seizure Data

Read and rearrange data

```
> seize <- read.table ("../data/seize.data", col.names = c("id", "seizure", "week",  
+ "progabide","baseline8", "age"))  
> seize.base <- data.frame (id = seize$id, seizure = seize$baseline8, week = seize$week,  
+ "progabide" = seize$progabide, age = seize$age)  
> seize.base <- seize.base[seize.base$week == 1,]  
> seize.base$week <- 0  
> seize.full <- rbind (seize[,-5], seize.base)  
> seize.full <- seize.full[order(seize.full$id, seize.full$week),]  
> seize.full$time <- ifelse (seize.full$week == 0, 8, 2)  
> seize.full$post <- as.numeric (seize.full$week != 0)  
> seizure.r <- seize.full[seize.full$id != 207,]
```

1. R gee function

```
> library(gee)
> gee.seize <- gee (seizure ~ post+progabide + post:progabide +offset (log(time)),
+                   data = seizure.r, id = id, family = "poisson",cor = "exchangeable")

> summary (gee.seize)
```

Coefficients:

	Estimate	Naive S.E.	Naive z	Robust S.E.	Robust z
(Intercept)	1.3476092	0.1105906	12.1855676	0.1573571	8.5640166
post	0.1107981	0.1232531	0.8989482	0.1160997	0.9543358
progabide	-0.1080280	0.1579475	-0.6839487	0.1936732	-0.5577850
post:progabide	-0.3015995	0.1936050	-1.5578080	0.1712004	-1.7616751

Estimated Scale Parameter: 10.54250

Number of Iterations: 1

Working Correlation

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	1.0000000	0.5941486	0.5941486	0.5941486	0.5941486
[2,]	0.5941486	1.0000000	0.5941486	0.5941486	0.5941486
[3,]	0.5941486	0.5941486	1.0000000	0.5941486	0.5941486
[4,]	0.5941486	0.5941486	0.5941486	1.0000000	0.5941486
[5,]	0.5941486	0.5941486	0.5941486	0.5941486	1.0000000

2. R glmmPQL function for model with random intercept only

```
> library (MASS)
> summary (glmmPQL (seizure ~ post+progabide+progabide * post + offset (log (time)),
+                  random = ~ 1 | id, family = poisson, data = seizure.r))
```

Random effects:

Formula: ~1 | id
(Intercept) Residual

StdDev: 0.6829581 1.604582

Variance function:

Structure: fixed weights

Formula: ~invwt

Fixed effects: seizure ~ post + progabide + progabide * post + offset(log(time))

	Value	Std.Error	DF	t-value	p-value
(Intercept)	1.0805705	0.14386103	230	7.511211	0.0000
post	0.1107981	0.07574634	230	1.462752	0.1449
progabide	-0.0108566	0.20048439	56	-0.054152	0.9570
post:progabide	-0.3015995	0.11268482	230	-2.676487	0.0080

3. R glmmPQL function for model with random intercept and random slope

```
> summary (glmmPQL (seizure ~ post+progabide+progabide * post + offset (log (time)),
+                  random = ~ post | id,family = poisson, data = seizure.r))
```

Random effects:

```
Formula: ~post | id
Structure: General positive-definite, Log-Cholesky parametrization
              StdDev   Corr
(Intercept) 0.6349095 (Intr)
post         0.3831382 0.17
Residual    1.3905660
```

Variance function:

```
Structure: fixed weights
Formula: ~invwt
```

Fixed effects: seizure ~ post + progabide + progabide * post + offset(log(time))

	Value	Std.Error	DF	t-value	p-value
(Intercept)	1.1214970	0.1326771	230	8.452830	0.0000
post	-0.0049557	0.1049458	230	-0.047222	0.9624
progabide	-0.0243675	0.1845746	56	-0.132020	0.8954
post:progabide	-0.3006763	0.1492665	230	-2.014359	0.0451

4. SAS GENMOD

```
PROC IMPORT OUT= WORK.seiz
            DATAFILE= "seizeurel.csv"
            DBMS=CSV REPLACE; GETNAMES=YES;DATAROW=2;
proc genmod data = seiz;
  class id;
  model seizure = post | progabide / dist = poisson offset = logtime
    scale = Pearson;
  repeated subject = id / type = exch covb;
run;
```

Exchangeable Working

Correlation

Correlation 0.5941485833

Analysis Of GEE Parameter Estimates

Empirical Standard Error Estimates

		Standard	95% Confidence			
Parameter	Estimate	Error	Limits		Z	Pr > Z
Intercept	1.3476	0.1574	1.0392	1.6560	8.56	<.0001
post	0.1108	0.1161	-0.1168	0.3383	0.95	0.3399
progabide	-0.1080	0.1937	-0.4876	0.2716	-0.56	0.5770
post*progabide	-0.3016	0.1712	-0.6371	0.0339	-1.76	0.0781

5. SAS GLIMMIX for model with random intercept only

```
proc glimmix data = seiz method = RSPL;
  class id;
  model seizure = post | progabide / dist = poi offset = logtime solutions;
  random intercept / subject = id;
run;
```

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error
Intercept	id	0.5249	0.1043

Solutions for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > t
Intercept	1.0477	0.1424	56	7.36	<.0001
post	0.1108	0.04689	230	2.36	0.0190
progabide	-0.00910	0.1982	230	-0.05	0.9634
post*progabide	-0.3016	0.06975	230	-4.32	<.0001

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
post	1	230	5.58	0.0190
progabide	1	230	0.00	0.9634
post*progabide	1	230	18.70	<.0001

6. SAS NLMIXED for model with random intercept only

```
proc nlmixed data = seiz;
  parms b0 = 1 b1 = 0 b2 = 0 b3 = 0 sig1 = 0.1;
  lambda = b0 + b1*post + b2*progabide + b3*post*progabide + u0 + log(time);
  mu = exp(lambda);
  model seizure ~ Poisson(mu);
  random u0 ~ Normal(0, sig1) subject=id;
run;
```

Fit Statistics

-2 Log Likelihood	1897.2
AIC (smaller is better)	1907.2
AICC (smaller is better)	1907.4
BIC (smaller is better)	1917.5

Parameter Estimates

Parameter	Estimate	Standard Error	DF	t Value	Pr > t	Alpha	Lower	Upper	Gradient
b0	1.0360	0.1414	57	7.33	<.0001	0.05	0.7528	1.3191	0.000323
b1	0.1108	0.04689	57	2.36	0.0216	0.05	0.01691	0.2047	0.000155
b2	-0.01049	0.1967	57	-0.05	0.9577	0.05	-0.4044	0.3835	0.000213
b3	-0.3016	0.06975	57	-4.32	<.0001	0.05	-0.4413	-0.1619	0.00003
sig1	0.5161	0.1012	57	5.10	<.0001	0.05	0.3135	0.7187	0.000054

7. SAS GLIMMIX for model with random intercept and random slope

```
proc glimmix data = seiz method = RSPL;
  class id;
  model seizure = post | progabide / dist = poi offset = logtime solutions;
  random intercept post/ subject = id type=un;run;
```

Covariance Parameter Estimates

Cov			Standard
Parm	Subject	Estimate	Error
UN(1,1)	id	0.4578	0.09544
UN(2,1)	id	0.01434	0.05369
UN(2,2)	id	0.2194	0.05931

Solutions for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > t
Intercept	1.0872	0.1344	56	8.09	<.0001
post	0.009620	0.1058	56	0.09	0.9278
progabide	-0.00912	0.1868	174	-0.05	0.9611
post*progabide	-0.3425	0.1490	174	-2.30	0.0227

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
post	1	56	0.01	0.9278
progabide	1	174	0.00	0.9611
post*progabide	1	174	5.28	0.0227

8. SAS NLMIXED for model with random intercept and slope

```
proc nlmixed data = seiz;
  parms b0 = 1 b1 = 1 b2 = 0 b3 = 0 s11 = 0.1 s12 = 0.05 s22 = 0.1;
  lambda = b0 + b1*post + b2*progabide + b3*post*progabide + u0 + u1*post + log(time);
  mu = exp(lambda);
  model seizure~ Poisson(mu);
  random u0 u1 ~ Normal([0,0],[s11,s12,s22]) subject = id;
run;
```

Fit Statistics

-2 Log Likelihood	1786.5
AIC (smaller is better)	1800.5
AICC (smaller is better)	1800.9
BIC (smaller is better)	1814.9

Parameter Estimates

Parameter	Estimate	Standard Error	DF	t Value	Pr > t	Alpha	Lower	Upper	Gradient
b0	1.0698	0.1342	56	7.97	<.0001	0.05	0.8009	1.3386	0.000173
b1	0.005823	0.1068	56	0.05	0.9567	0.05	-0.2081	0.2197	-0.00029
b2	-0.00976	0.1858	56	-0.05	0.9583	0.05	-0.3820	0.3625	0.000062
b3	-0.3469	0.1485	56	-2.34	0.0231	0.05	-0.6445	-0.04935	-0.00019
s11	0.4517	0.09327	56	4.84	<.0001	0.05	0.2649	0.6386	0.000065
s12	0.01753	0.05264	56	0.33	0.7403	0.05	-0.08791	0.1230	0.000245
s22	0.2147	0.05821	56	3.69	0.0005	0.05	0.09810	0.3313	-0.00056

Comparison of Results

Also see Table 8.11 (p165), Table 9.6 (p186), Table 9.7 (p189), DHLZ.

	GEE	GLMM: Intercept			CL DHLZ
		glmmPQL	GLIMMIX	NLMIXED	
Intercept	1.35 (0.16)	1.08 (0.14)	1.05 (0.14)	1.04 (0.14)	0.11 (0.047)
post	0.11 (0.12)	0.11 (0.08)	0.11 (0.05)	0.11 (0.05)	
progabide	-0.11 (0.19)	-0.01 (0.20)	-0.01 (0.20)	-0.01 (0.20)	
post*progabide	-0.30 (0.17)	-0.30 (0.11)	-0.30 (0.07)	-0.30 (0.07)	-0.30 (0.070)
D_{11}		0.47	0.52 (0.10)	0.51 (0.10)	
ρ	0.59				
ϕ	10.5				

	GLMM: Intercept and slope		
	glmmPQL	GLIMMIX	NLMIXED
Intercept	1.12 (0.13)	1.09 (0.13)	1.07 (0.13)
post	-0.003 (0.10)	0.01 (0.11)	0.008 (0.11)
progabide	-0.023 (0.18)	-0.009 (0.19)	-0.008 (0.19)
post*progabide	-0.30 (0.15)	-0.34 (0.15)	-0.35 (0.15)
D_{11}	0.40	0.46 (0.10)	0.45 (0.09)
D_{22}	0.15	0.22 (0.06)	0.22 (0.06)
D_{12}	0.041	0.014 (0.06)	0.015 (0.05)

- The PQL method is much faster than the quadrature method.
- The interaction term's point estimate is similar across all models.
- From AIC, random slope model is better (1801 vs 1907).

Further Reading

- Chapter 9 of DHLZ.
- Chapter 13 of Molenberghs and Verbeke (2005).

References

- Agresti A (2002) Categorical data analysis. John Wiley & Sons, 2nd edn.
- Breslow NE and Clayton DG (1993) Approximate inference in generalized linear mixed models. *Journal of American Statistician Association* 88:9-25.
- Breslow NE and Lin X (1995) Bias correction in generalised linear mixed models with a single component of dispersion. *Biometrika* 82:81-91.
- Clayton DG (1996) Generalized linear mixed models. In *Markov Chain Monte Carlo in Practice* (edited by WR Gilks, S Richardson, and DJ Spiegelhalter), Chapman & Hall, London, pp. 259-273.
- Gelfand AE and Carlin BP (1993) Maximum likelihood estimation for constrained- or missing-data problems. *Canadian Journal of Statistics* 21:303-311.
- Geyer CJ and Thompson EA (1992) Constrained Monte Carlo maximum likelihood for dependent data. *Journal of the Royal Statistical Society, Series B* 54:657-699.
- Lee Y and Nelder JA (1996) Hierarchical Generalized Linear Models. *JRSSB*, 58:619-678.
- Liang KY and McCullagh P (1993) Case studies in binary dispersion. *Biometrics* 49:623-630.
- Lin X and Breslow NE (1996) Bias correction in generalized linear mixed models with multiple component of dispersion. *Journal of American Statistician Association* 91:1007-1016.
- McCulloch CE (1997) Maximum likelihood algorithms for generalized linear mixed models. *Journal of American Statistician Association* 92:162-170.

- McCulloch CE (2003) Generalized linear mixed models, vol. 7 of NSF-CBMS Regional Conference Series in Probability and Statistics. Institute of Mathematical Statistics.
- Zeger SL and Karim MR (1991) Generalized linear models with random effects: a Gibbs sampling approach. Journal of American Statistician Association 86:79-86.
- Zeger SL, Liang KY, and Albert PS (1988) Models for longitudinal data: a generalized estimating equation approach. Biometrics 44:1049-1060.