Notes from: Random Effects: Generalized Linear Mixed Models

Chapter 10 Agresti (2007)
Saúl Sotomayor Leytón
April 2012

Setup

```
> library(MASS)
> library(glmmML)
> library(lme4)
> library(vcd)
> options(width=70)
```

1 Introduction

In the previous chapter it was shown how, sometimes, observations occur in clusters, that is groups based on any type of association among observations; for example in longitudinal studies where a subject's response is recorded at various times the observations for a particular subject form a cluster. We expect observations within a cluster to be more alike than those from another cluster, and because of that we need models that consider that association, otherwise, as noted by Agresti the standard error of the estimates would be biased. Chapter 9 described one way to handle correlated, clustered data that focused on marginal distributions and estimated factor effects for all the population; in contrast the models presented in this chapter estimate factor effects conditional on the subjects. This is reflected by the inclusion of a separate term for each cluster, terms that are supposed to vary across clusters. As it is discussed throughout the chapter the cluster–specific term may reflect unmeasured predictors that translate in heterogeneity among observations.

2 Random effects modeling of clustered categorical data

The author starts this section by distinguishing *fixed* and *random* effects; the former apply to all categories of interest, like gender or treatment, while the latter apply to a sample or a specific cluster.

To understand how these two types of effects are included in the model, the author first reviews how generalized linear models (GLMs) extend ordinary regression to allow non–normal responses and a link function of the mean. As a next step, generalized linear mixed models GLMMs extend GLMs to allow the inclusion of random effects as well as fixed effects in the linear predictor. For cluster i a random effect is denoted as μ_i. It should be noted that, in practice, μ_i is unknown, however it's treated as random variable that comes from a normal distribution with mean α and variance of σ^2 $(\mu, N(\alpha, \sigma^2))$. The random effect can be included in the model as an intercept, as a covariate or both. In this section the author then describes the common case where μ_i is included as a intercept, models that receive the name of *random intercept models* and it's represented as follows,

$$g(\mu_{it}) = \mu_i + \beta x_{it}$$

Note that this model has as many intercepts as clusters, however if we include the expected value of μ_i into the model and regard it as a random normal variable with a mean of 0 and a variance of σ we have

a model,

$$g(\mu_{it}) = \mu_i + \alpha + \beta x_{it}$$

that takes the value of α when $x_{it} = 0$, but more importantly it reduces the number of additional parameters to be estimated to σ^2. This value represents the variability among clusters, which in some studies may be interpreted as heterogeneity among clusters caused by not including certain explanatory variables. In other words it reflects terms that would be in the fixed effects if those explanatory variables had been included.

Next, the author describes GLMM models for binomial responses, that is,

$$logit[P(y_{i1} = 1)] = \mu_i + \alpha + \beta; \quad logit[P(y_{i2} = 1)] = \mu_i + \alpha$$

model that is called *logistic–normal model*, normal because the intercept term is assumed to come from a normal distribution (see above). As for any GLMM this model assumes that the observations from the same cluster are more alike than observations from different clusters. For example if we apply this model to the questions about helping the environment discussed in chapter 8, this would mean that a subject who answers "yes" to question 1 is more likely to answer "yes" to the second question, and the same for a "no" answer; something that is reflected in the contingency table. where the "yes/yes"

Table 1: Opinions related to environment

Pay Higher taxes	Cut living standards	
	Yes	No
Yes	227	132
No	107	678

and "no/no" cells represent the higher proportion of the sample, which implies that the odds ratio is positive just as the log-odds ratio. This is explained in Agresti as, "when a higher proportion of cases have outcomes $(y_{i1} = 1, y_{i2} = 1)$ or $(y_{i1} = 0, y_{i2} = 0)$ the association between repeated responses is positive and greater association results from greater heterogeneity, larger σ.

2.1 Sacrifices for the environment

This subsection revises the example of chapter 8 about sacrifices that people are willing to make in order to help the environment. The data is presented in table 1.
To fit these models Thompson uses the function `glmmPQL` from the `MASS` package. This function needs un-grouped data so first we construct the data frame:

```
> tb10.1 <- data.frame(question = c(0, 1), response = c(rep(c(1,
+     1), 227), rep(c(1, 0), 132), rep(c(0, 1), 107), rep(c(0,
+     0), 678)))
> tb10.1$question <- ifelse(tb10.1$question == 0, "high.tax", "cut.liv")
> tb10.1$question <- factor(tb10.1$question, levels = c("cut.liv",
+     "high.tax"))
> tb10.1$case <- rep(1:(nrow(tb10.1)/2), each = 2)
```

Note that we reversed the levels of the `question` vector and how the index vector (`case`) was constructed.
With the data correctly constructed we can fit the model

```
> ## library(MASS)
> glmmmPQL10.1 <- glmmPQL(response ~ question, random = ~1 | case,
+     family = binomial, data = tb10.1)
> summary(glmmmPQL10.1)
```

```
Linear mixed-effects model fit by maximum likelihood
  Data: tb10.1
  AIC BIC logLik
   NA  NA     NA

Random effects:
 Formula: ~1 | case
        (Intercept)  Residual
StdDev:    2.363765 0.6431358

Variance function:
 Structure: fixed weights
 Formula: ~invwt
Fixed effects:  response ~ question
                     Value  Std.Error   DF    t-value p-value
(Intercept)      -1.4266156 0.09522882 1143 -14.980923  0.0000
questionhigh.tax  0.2097344 0.08341640 1143   2.514306  0.0121
 Correlation:
                  (Intr)
questionhigh.tax -0.452

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-1.5041498 -0.3973594 -0.3577616  0.5485945  1.7850774

Number of Observations: 2288
Number of Groups: 1144
```

Note that the β estimate is the same as that reported in Agresti, however its standard error is lower, as well as the estimate of the standard deviation (σ), 2.36 versus 2.85.
A different β estimate but with closer standard error and standard deviation estimates are found with the `glmmML` function from the package with the same name.

```
> ## library(glmmML)
> summary(glmmML(response ~ question, cluster = tb10.1$case, family = binomial,
+     data = tb10.1))
```

```
Call:  glmmML(formula = response ~ question, family = binomial, data = tb10.1,      cluster = tb
```

```
                    coef se(coef)       z Pr(>|z|)
(Intercept)      -1.9705   0.1693 -11.637    0.000
questionhigh.tax  0.1982   0.1262   1.571    0.116

Scale parameter in mixing distribution:  2.697 gaussian
Std. Error:                              0.1745

        LR p-value for H_0: sigma = 0:  5.278e-53

Residual deviance: 2572 on 2285 degrees of freedom         AIC: 2578
```

Agresti mentions that the β estimate of 0.210 is the same as that calculated with conditional maximum likelihood as it was done in chapter 8. According to the author this is common when the sample log odds ratio is positive.

2.2 Differing effects in conditional models and marginal models

This subsection describes the differences among marginal and conditional models about theirs interpretations; basically one can say that the inferences in marginal models apply to the population, that is, they are population–averaged, while the interpretations in the conditional models are cluster–specific. The author exemplifies this by comparing the estimates obtained in the environmental example, for the marginal model ($exp(0.104) = 1.11$) and the conditional model ($exp(0.210) = 1.23$). He mentions that when the link function is non-linear, like the logit, the population averaged effects of marginal models are typically smaller in magnitude than the cluster–specific effects as it's illustrated in the following figure: This is because the marginal effects are the average of the cluster–specific effects ; the difference

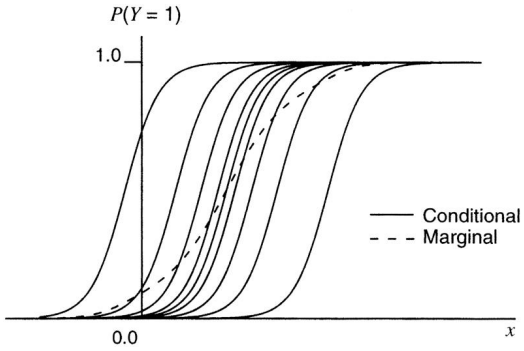

Figure 1: Logistic random-intercept model, showing the conditional (subject-specific) curves and the marginal (population-averaged) curve averaging over these

between the two effects is greater as the cluster–specific curves are more spread out, in other words as the spread of the random effect (σ) is greater.

3 Examples of random effects models for binary data

3.1 Small–area estimation of binomial probabilities

Although the name and the description suggest that this approach is specific for estimation of parameters in geographical studies with few observation, one can generalize to say that this approach is used to

estimate parameter for categorical predictors (factors) when the sample size for each level (or combination of levels, if we are working with multiple explanatory variables) is small. In the area example, if we have i areas, where $i = 1, \ldots, n$, the fixed effect logit model would be, $logit(\pi_i = 1) = \beta_i$ which is a saturated model having i parameters. Under this model, let T_i be the number of observations for area i of which y_i are the successes, then when we treat y_i as an independent binomial variate, the maximum likelihood estimate for π_i is the sample proportion $\pi = y_i/T_i$. Now, because we have, in some cities/levels, few observations, the sample proportions may be a poorly estimate of π_i, among other things, because these proportions would have a large standard error.
On the other hand, random effects model that treat each area as a cluster, express the model as, $logit(\pi_i = 1) = \mu_i + \alpha$, where μ_i is a normally (but unknown) variable with a mean of 0 and standard deviation of σ, thus the model has only two parameters, α and the variance of μ_i, σ instead of the n parameters in the previous model. As the author mentions, the assumptions that the logits of probabilities ($logit(\pi_i)$) vary according to a normal distribution, the estimation of a particular probability uses all the information available, thus the estimate for a given area is a *weighted average* of the sample proportion for the area of interest and the overall proportions for all the areas. This estimate, $\hat{\pi}_i$ gives more weight to sample proportions as T_i grows.

Agresti illustrates these models with the data of free-throws showed in table 10.2.
In R we fit this data either with the `glmmPQL` or the `glmmML` function. Note that as the previous cases the results obtained with these functions do not match exactly those presented in Agresti (2007, p. 303). First we fit the models, with the two functions mentioned,

```
> ## library(MASS)
>  load("supp_data/tb10-2.rda")
> summary(glmmPQL(y/n ~ 1, random = ~1 | player, weights = n, family = binomial,
+     data = tb10.2))
```

```
Linear mixed-effects model fit by maximum likelihood
  Data: tb10.2
  AIC BIC logLik
   NA  NA     NA

Random effects:
 Formula: ~1 | player
         (Intercept) Residual
StdDev: 3.799351e-05  1.20199

Variance function:
 Structure: fixed weights
 Formula: ~invwt
Fixed effects:  y/n ~ 1
                Value Std.Error DF  t-value p-value
(Intercept) 0.8774509 0.2284363 15 3.841118  0.0016

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-2.4144553 -0.5237744 -0.1773034  1.0367917  1.1189059

Number of Observations: 15
```

```
Number of Groups: 15
```

```
> ## library(glmmML)
> summary(glmmML(y/n ~ 1, cluster = tb10.2$player, weights = tb10.2$n,
+                family = binomial, data = tb10.2))
```

```
Call:  glmmML(formula = y/n ~ 1, family = binomial, data = tb10.2, cluster = tb10.2$player,      w


             coef se(coef)     z Pr(>|z|)
(Intercept) 0.9058   0.2214 4.091  4.3e-05

Scale parameter in mixing distribution:  0.4015 gaussian
Std. Error:                              0.3858

        LR p-value for H_0: sigma = 0:  0.2664

Residual deviance: 22.72 on 13 degrees of freedom         AIC: 26.72
```

Note that the first function gives very different results, particularly for the standard deviation estimate, than those mentioned in Agresti (α=0.908, $\hat{\sigma}$=0.422). On the other hand, the results of the second function are closer to those mentioned.

Update Another function that can fit GLMMs is `lmer` from the `lme4` package. Among the advantages of this functions is that it provides values for the BIC or AIC information criteria, but also it allows the calculation of fitted values, thing that, at least, is difficult with the other functions.

```
> ## library(lme4)
> summary(lmer.tb10.2 <- glmer(y/n ~ 1 | player, weights = n, family = binomial,
+     data = tb10.2)) ## previously lmer
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: y/n ~ 1 | player
   Data: tb10.2
Weights: n

     AIC      BIC   logLik deviance df.resid
    60.3     61.8    -28.2     56.3       13

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.2615 -0.4952 -0.2034  1.0048  1.1712

Random effects:
 Groups Name        Variance Std.Dev.
```

```
 player (Intercept) 0.1611   0.4014
Number of obs: 15, groups:  player, 15

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.9057     0.2211   4.097 4.18e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

With this variable we can obtain the fitted values and compare them with the sample proportions as it is shown on table 10.2.

```
> cbind(tb10.2[, 1:2], Obs = round(tb10.2[, 3]/tb10.2[, 2], 3),
+     Fit = round(exp(fitted(lmer.tb10.2))/(1 + exp(fitted(lmer.tb10.2))),
+         3))
```

```
       player  n   Obs   Fit
1         Yao 13 0.769 0.675
2        Frye 10 0.900 0.681
3       Camby 15 0.667 0.668
4        Okur 14 0.643 0.666
5      Blount  6 0.667 0.669
6        Mihm 10 0.900 0.681
7   Ilgauskas 10 0.600 0.665
8       Brown  4 1.000 0.678
9       Curry 11 0.545 0.661
10     Miller 10 0.900 0.681
11    Haywood  8 0.500 0.661
12 Olowokandi  9 0.889 0.679
13   Mourning  9 0.778 0.674
14    Wallace  8 0.625 0.667
15   Ostertag  6 0.167 0.649
```

Agresti highlights the much narrower range of the estimated probabilities (0.61–0.76) than the sample proportions (0.17–1.00). Note that those calculated with the `lmer` function give an even narrower range (0.65–0.68).
Now, whether the fitted model is appropriate is left for the discussion in section 10.5.2.

3.2 Teratology example

This section analyzes the data from a teratology study, first introduced in section 9.2.4. Because we already constructed the data (both in grouped and ungrouped form) we can simply import into the present R session.

```
> #grouped data
> tb9.4 <- read.table("supp_data/tb9-4")
> colnames(tb9.4) <- c("group", "litter.size", "num.dead")
> tb9.4$case <- 1:nrow(tb9.4)
> tb9.4$group <- factor(tb9.4$group, levels = 1:4)
> ## ungrouped data
```

```
> tb9.4long <- read.table("supp_data/tb9-4long", header = TRUE)
> tb9.4long$group <- factor(tb9.4long$group, levels = 1:4)
```

With `glmmML` or `lmer` functions we can use either of them, just remember that if we use the grouped data we need to express the response variable as a proportion of positive outcomes and include a `weights` argument in the formula. Note also that in case we didn't have a data frame it would be easier to construct and work a grouped data than an ungropued one, an advantage of these functions compared with those that were used for GEEs.
Following are the commands used to fit the model with both mentioned functions as well as with both data frames, however, only the results of two out of the four commands is shown.

```
> #With grouped data
> ## library(glmmML)
> glmmML(num.dead/litter.size ~ group, cluster = case, weights = tb9.4$litter.size,
+     family = binomial, data = tb9.4)
```

```
Call:  glmmML(formula = num.dead/litter.size ~ group, family = binomial,      data = tb9.4, cluste


             coef se(coef)      z Pr(>|z|)
(Intercept)  1.809   0.3605  5.019 5.19e-07
group2      -4.540   0.7326 -6.196 5.77e-10
group3      -5.884   1.1743 -5.010 5.43e-07
group4      -5.607   0.9050 -6.195 5.82e-10

Scale parameter in mixing distribution:  1.511 gaussian
Std. Error:                              0.2759

        LR p-value for H_0: sigma = 0:  3.223e-15

Residual deviance: 112.7 on 53 degrees of freedom         AIC: 122.7
```

```
> ## library(lme4)
> glmer(num.dead/litter.size ~ group + (1 | case), family = binomial,
+     weights = litter.size, data = tb9.4)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: num.dead/litter.size ~ group + (1 | case)
   Data: tb9.4
Weights: litter.size
     AIC      BIC   logLik deviance df.resid
194.1626 204.4648 -92.0813 184.1626       53
Random effects:
 Groups Name        Std.Dev.
 case   (Intercept) 1.511
Number of obs: 58, groups:  case, 58
```

```
Fixed Effects:
(Intercept)       group2       group3       group4
      1.809       -4.540       -5.883       -5.606
```

```
> # With ungrouped data
> glmer(outcome ~ group + (1 | case), family = binomial, data = tb9.4long)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: outcome ~ group + (1 | case)
   Data: tb9.4long
      AIC       BIC    logLik  deviance  df.resid
 445.9399  467.9825 -217.9699  435.9399       602
Random effects:
 Groups Name        Std.Dev.
 case   (Intercept) 1.511
Number of obs: 607, groups:  case, 58
Fixed Effects:
(Intercept)       group2       group3       group4
      1.809       -4.540       -5.883       -5.606
```

```
> glmmML(outcome ~ group, cluster = tb9.4long$case, family = binomial,
+     data = tb9.4long)
```

```
Call:  glmmML(formula = outcome ~ group, family = binomial, data = tb9.4long,      cluster = tb9


             coef se(coef)      z Pr(>|z|)
(Intercept)  1.809   0.3605  5.019 5.19e-07
group2      -4.540   0.7326 -6.197 5.77e-10
group3      -5.884   1.1743 -5.010 5.43e-07
group4      -5.607   0.9050 -6.195 5.82e-10

Scale parameter in mixing distribution:  1.511 gaussian
Std. Error:                              0.2759

        LR p-value for H_0: sigma = 0:  3.223e-15

Residual deviance: 435.9 on 602 degrees of freedom         AIC: 445.9
```

Note how the random intercept is specified in the `lmer` function

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: num.dead/litter.size ~ group + (1 | case)
   Data: tb9.4
Weights: litter.size
```

```
     AIC      BIC   logLik deviance df.resid
194.1626 204.4648 -92.0813 184.1626       53
Random effects:
 Groups Name        Std.Dev.
 case   (Intercept) 1.511
Number of obs: 58, groups:  case, 58
Fixed Effects:
(Intercept)       group2       group3       group4
      1.809       -4.540       -5.883       -5.606
```

```
Call:  glmmML(formula = num.dead/litter.size ~ group, family = binomial,      data = tb9.4, cluste


             coef se(coef)      z Pr(>|z|)
(Intercept)  1.809   0.3605  5.019 5.19e-07
group2      -4.540   0.7326 -6.196 5.77e-10
group3      -5.884   1.1743 -5.010 5.43e-07
group4      -5.607   0.9050 -6.195 5.82e-10

Scale parameter in mixing distribution:  1.511 gaussian
Std. Error:                              0.2759

        LR p-value for H_0: sigma = 0:  3.223e-15

Residual deviance: 112.7 on 53 degrees of freedom         AIC: 122.7
```

Table 2: Comparison of estimates from Agresti (2007) and those calculated in R with two functions

Estimate	Agresti-2007	`glmmML`	`lmer`
(Intercept)	1.802 (0.362)	1.809 (0.3605)	1.8095 (0.3286)
group2	-4.515 (0.736)	-4.540 (0.7326)	-4.5398 (0.6779)
group3	-5.855 (1.190)	-5.884 (1.1743)	-5.8838 (1.1764)
group4	-5.594 (0.919)	-5.607 (0.9050)	-5.6067 (0.8619)
σ	1.53	1.511	1.5113

Note that the results calculated in R vary from those in Agresti, however the difference isn't big. In table 10.2 it's shown the results from the GEE model as well as those from the binomial maximum likelihood. Interesting the results are similar, however the assumptions that underlie those models are different. First the logistic model, $logit(\pi_{it}) = \alpha + \beta_2 z_{i2} + \beta_3 z_{i3} + \beta_4 z_{i4}$, assumes that the group effect on the outcome are the same across the different litters and more important that the outcomes are independent of the others, an assumption that isn't realistic because fetuses in a litter are likely to be related in someway. This relationship is indicated in table 2.

3.3 Repeated responses on similar survey items

This section shows data of a survey on whether a person supports or not abortions under 3 different scenarios (variable treated as random), the responses distinguished between males and females (fixed

variable). The model for responses in favor of abortion ($y_{it} = 1$) is,

$$logit[P(y_{it} = 1)] = \mu_i + \beta_t + \gamma x_i \tag{1}$$

where x_i is an indicator variable that equals 1 for males and 0 for females, β_t represents the parameters for all the 3 different scenarios under which abortion is considered. Agresti notes that because there is no intercept term (no α) there are no constraints for β_t (*i.e.* no category is set to zero).
For fitting this model in R, Thompson describes a different way of setting up the data, ungrouped data. First we construct the grouped data from table 10.4. Note that in table 10.4 the question that varies fastest is question 3, then question 1 and finally question 2, this order is reflected in the constriction. Next, we "collapse" the columns for the questions into one, this with the `reshape` function.

```
> tb10.4 <- data.frame(expand.grid(Q3 = c(1, 0), Q1 = c(1, 0),
+     Q2 = c(1, 0), gender = c("M", "F")), count = c(342, 26, 6,
+     21, 11, 32, 19, 356, 440, 25, 14, 18, 14, 47, 22, 457))
> tb10.4long <- reshape(tb10.4, varying = c("Q1", "Q2", "Q3"),
+     direction = "long", v.names = "response", timevar = "question")
> tb10.4long$question <- factor(tb10.4long$question, levels = 3:1)
```

A couple of notes about the last command. First note that the `varying` argument is for those columns that will be collapsed, then the `direction` argument must be either `"long"` or `"wide"`, depending what are we trying to do; in this case we are converting a "wide" table into a "long" one, thus the value is the latter. By omitting some of the above arguments it seems that the `v.names` argument represents the name of the collapsed column and it is mandatory for the function to work; its value can be anything we want, of course it should be something that makes sense, in this case `response`. On the other hand the `timevar` argument is optional, if we omit it a column with the name `time` will be created, as with the previous argument it's better to give it a useful name, in this case `question`.
Once we have created the long table we need to make it "ungrouped" by repeating each row `count` times, a step that was done before with other data frames, however note that in the index command some columns are dropped. Later those dropped columns are re-created with the correct values,

```
> tb10.4long <- tb10.4long[rep(1:nrow(tb10.4long), tb10.4long$count),
+     c("response", "gender", "question")]
> tb10.4long$id <- factor(rep(1:1850, 3))
```

Now we can fit the model with the functions `glmmPQL` and `lmer`

```
> ## library(MASS)
> (glmmPQL(response ~ gender + question, random = ~1 | id, family = binomial,
+     data = tb10.4long))
```

```
Linear mixed-effects model fit by maximum likelihood
  Data: tb10.4long
  Log-likelihood: NA
  Fixed: response ~ gender + question
 (Intercept)      genderF    question2    question1
-0.510887814  0.006461863  0.306730804  0.867734859

Random effects:
 Formula: ~1 | id
```

```
        (Intercept)  Residual
StdDev:    4.308419 0.4507951

Variance function:
 Structure: fixed weights
 Formula: ~invwt
Number of Observations: 5550
Number of Groups: 1850
```

```
> ## library(lme4)
> (glmer(response ~ gender + question + (1 | id), family = binomial,
+     data = tb10.4long))
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: response ~ gender + question + (1 | id)
   Data: tb10.4long
      AIC       BIC    logLik  deviance  df.resid
 4942.684  4975.792 -2466.342  4932.684      5545
Random effects:
 Groups Name        Std.Dev.
 id     (Intercept) 5.59
Number of obs: 5550, groups:  id, 1850
Fixed Effects:
(Intercept)      genderF    question2    question1
   -0.46922      0.01073      0.20949      0.59851
```

However, as noted above, Agresti (2007, p. 306) fits a model with no intercept, $logit[P(Y_{it})] = \mu_i + \beta_t + \gamma x_i$. In R to indicate that the model has no intercept we need to include in the formula the term `-1`, also we need to put first the variable whose estimates won't be restricted, in this case `question`,

```
> (glmmPQL(response ~ -1 + question + gender, random = ~1 | id,
+     family = binomial, data = tb10.4long))
```

```
Linear mixed-effects model fit by maximum likelihood
  Data: tb10.4long
  Log-likelihood: NA
  Fixed: response ~ -1 + question + gender
   question3    question2    question1      genderF
-0.510887809 -0.204157009  0.356847041  0.006461863

Random effects:
 Formula: ~1 | id
        (Intercept)  Residual
StdDev:    4.308419 0.4507951

Variance function:
 Structure: fixed weights
```

```
 Formula: ~invwt
Number of Observations: 5550
Number of Groups: 1850
```

```
> (glmer(response ~ -1 + question + gender + (1 | id), family = binomial,
+     data = tb10.4long))
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: response ~ -1 + question + gender + (1 | id)
   Data: tb10.4long
      AIC       BIC    logLik  deviance  df.resid
 4942.684  4975.792 -2466.342  4932.684      5545
Random effects:
 Groups Name        Std.Dev.
 id     (Intercept) 5.59
Number of obs: 5550, groups:  id, 1850
Fixed Effects:
question3  question2  question1    genderF
 -0.46921   -0.25973    0.12930    0.01072
```

Note that the estimates obtained with `glmmPQL` are closer to those reported in Agresti, however the estimate of the standard error is half the value reported.

Because all the question's parameters were estimated the odds ratio of being in favor of abortion under one scenario instead of another is computed by subtracting the estimates, for instance the odds ob being in favor of abortion under option 1 instead of option 3 is: $exp(\beta_1 - \beta_3) = exp(0.356847045 - (-0.510887814)) = exp(0.8677349) = 2.4$ times the odds for option 3. Although the standard error reported in `R` is smaller than that reported in Agresti, 4.3 versus 8.6, it still represents a high degree of heterogeneity and also suggest a strong association among the responses, note that responses 1,1,1 and 0,0,0 represent the great majority of the responses. The author mentions that in the US people tend to be either uniformly opposed or in favor about abortion.

Agresti indicates that including an interaction term does not improve the fitting, in fact if one looks at the p-values it seems sensible to drop the gender term. We can test this by fitting a model without gender and compare with the previous model.

```
> ## library(lme4)
> lmer.tb10.4 <- glmer(response ~ gender + question + (1 | id),
+     family = binomial, data = tb10.4long)
> lmer2.tb10.4 <- glmer(response ~ question + (1 | id), family = binomial,
+     data = tb10.4long)
> anova(lmer2.tb10.4, lmer.tb10.4, test = "Chisq")
```

```
Data: tb10.4long
Models:
lmer2.tb10.4: response ~ question + (1 | id)
lmer.tb10.4: response ~ gender + question + (1 | id)
```

```
             npar    AIC    BIC  logLik deviance Chisq Df Pr(>Chisq)
lmer2.tb10.4    4 4940.2 4966.7 -2466.1   4932.2
lmer.tb10.4     5 4942.7 4975.8 -2466.3   4932.7     0  1          1
```

This provides strong evidence in favor of dropping the gender effect. Note that the comparison was made with `lmer` models; for some reason the `anova` function doesn't seem to work with `glmmPQL` objects.

3.4 Depression study revisited

This subsection re-evaluates the example of section 9.1.2 about the effects of a new drug on mental depression which was evaluated at 3 times in patients with two initial conditions. The model fitted in chapter 9 allowed a treatment–time interaction.

$$logit[P(Y_t = 1)] = \alpha + \beta_1 s + \beta_2 d + \beta_3 t + \beta_4 (d \times t)$$

Now, the subject specific model would be,

$$logit[P(Y_{it} = 1)] = \mu_i + \alpha + \beta_1 s + \beta_2 d + \beta_3 t + \beta_4 (d \times t)$$

This model can be fitted in R either with `glmmPQL`, `glmmML` or `lmer`, with the same estimates and basically the same standard errors, which are very close to those reported in Agresti which in turn are similar to those found with GEE.

```
> tb9.1 <- read.table('supp_data/tb9-1', header=TRUE)
> glmer(outcome ~ diagnose + time + treat + time * treat + (1 |
+     case), family = binomial, data = tb9.1)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula:
outcome ~ diagnose + time + treat + time * treat + (1 | case)
   Data: tb9.1
     AIC      BIC   logLik deviance df.resid
1173.940 1203.505 -580.970 1161.940     1014
Random effects:
 Groups Name        Std.Dev.
 case   (Intercept) 0.05684
Number of obs: 1020, groups:  case, 340
Fixed Effects:
(Intercept)     diagnose         time        treat   time:treat
   -0.02797     -1.31488      0.48274     -0.05967      1.01817
```

The author notes that the small standard deviation implies little heterogeneity and therefore the estimates are not only similar to those of GEE but also to those found by ordinary logistic regression.

```
> glm(outcome ~ diagnose + time + treat + time * treat, family = binomial,
+     data = tb9.1)
```

```
Call:  glm(formula = outcome ~ diagnose + time + treat + time * treat,
    family = binomial, data = tb9.1)

Coefficients:
(Intercept)     diagnose         time        treat   time:treat
   -0.02799     -1.31391      0.48241     -0.05960      1.01744

Degrees of Freedom: 1019 Total (i.e. Null);  1015 Residual
Null Deviance:            1412
Residual Deviance: 1162         AIC: 1172
```

3.5 Choosing marginal or conditional models

This subsection compares situation where GEEs are better than GLMMs and vice versa.

GEEs are better when the main focus is on comparing groups that are independent samples or effects of interest are between clusters rather than within cluster.

GLMMs are better when we want to model the joint distribution or when we want to estimate cluster–specific effects or its variability; also when we want to specify a mechanism that could generate a positive correlation among clustered observations.
Agresti (2007, p. 309) mentions that with some extra calculation, one can recover information about marginal distributions from a conditional model; that is, "a conditional model implies a marginal model but not vice versa, which means that a conditional model has more information."
Despite the choice of model the inferential conclusions do not vary considerably.

3.6 Conditional models: random effects versus conditional ML

This subsection compares random effects models with conditional maximum likelihood models, the latter discussed in section 8.2.4. The author notes that because conditional ML models eliminate the random effect, one can not estimate its variability nor its effects on the probability, thus the inferences are limited to the fixed effects. Also if the number of observation is large, the computation of the conditional ML can be intensive.

4 Extensions to multinomial responses of multiple random effect terms

As the name suggest, this section extends the models to handle multiple responses but more interesting to include more than one random effect in the form of a random slope.
It should be noted that when working with ordered responses the structure used is that for proportional odds.

4.1 Insomnia study revisited

This example was analyzed in section 9.3.2 thus we can use the data to fit the logistic–normal model, $logit[P(Y_{it} = 1)] = \mu_i + \alpha_j + \beta_i t + \beta_2 x + \beta_3 (t \times x)$.

As with other examples the results of including a random intercept μ_i doesn't change the estimate's effects, however with the random effects model we can obtain a value (σ) that tell us that there is a relatively large heterogeneity among the clusters.

Note on R fitting In Thompson (2007, p. 243) a method is described for fitting multinomial random effects model, however is very complicated. Looking on the internet I've found the package `mixcat`[1] which supposedly can fit these models, however it doesn't report an estimate of the standard deviation, and the estimates are the same as those from a GEE. From an initial test the package `lme4` is able to fit the model, however the estimates have a different sign (and value) than those from Agresti, also the estimate of the standard deviation has a different value.

```
> ## library(lme4)
> tb9.6 <- read.table('supp_data/tb9-6', header = TRUE)
> lmer(outcome ~ treat * occasion + (1 | case), data = tb9.6)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: outcome ~ treat * occasion + (1 | case)
   Data: tb9.6
REML criterion at convergence: 1323.964
Random effects:
 Groups   Name        Std.Dev.
 case     (Intercept) 0.6770
 Residual             0.7537
Number of obs: 478, groups:  case, 239
Fixed Effects:
   (Intercept)           treat        occasion  treat:occasion
     3.0250000       0.0002101      -0.5750000      -0.4418067
```

4.2 Bivariate random effects and association heterogeneity

This section evaluates a study about a comparison between two surgical procedures, a standard and a new one, about whether they provoke an adverse effect on the patient. Data were collected from 41 different studies. Agresti indicates that when the strata (in this case the studies) are themselves a sample, a random effects approach is natural. Thus we move from the model,

$$logit[P(Y_{i1} = 1)] = \mu_i + \alpha + \beta \qquad logit[P(Y_{i2} = 1)] = \mu_i + \alpha$$

which is a model that assumes the same treatment effect for all the studies, to the model,

$$logit[P(Y_{i1} = 1)] = \mu_i + \alpha + (\beta + \upsilon_i) \qquad logit[P(Y_{i2} = 1)] = \mu_i + \alpha$$

which is a model that allows the treatment effect to vary between studies. The standard deviation of υ_i describes the variability in the odds ratio for all the 41 studies, something that seems reasonable when we look at some of the data (table 10.8).
To fit the model in R we need to import the data from Agresti's website and then use the `lmer` function.

```
> tb10.8 <- read.table("supp_data/tb10-8", header = TRUE)
> # Model with random intercept only
> glmer(y/n ~ treat + (1 | study), weights = n, family = binomial,
+     data = tb10.8)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
```

[1] In order to install this package the linux package, **libgls0-dev** must be installed first

```
 Family: binomial  ( logit )
Formula: y/n ~ treat + (1 | study)
   Data: tb10.8
Weights: n
      AIC       BIC    logLik  deviance  df.resid
 523.9590  531.1792 -258.9795  517.9590        79
Random effects:
 Groups Name        Std.Dev.
 study  (Intercept) 0.8161
Number of obs: 82, groups:  study, 41
Fixed Effects:
(Intercept)        treat
    -0.3211      -1.1727
```

```
> # Model with a random intercept and random slope
> fit.lmer.tb10.8 <- glmer(y/n ~ treat + (treat | study), weights = n, family = binomial,
+     data = tb10.8)
>
```

The estimates and their standard errors are very close to those reported in Agresti, namely β =-1.299 with SE =0.277 and σ_v =1.52 versus β =-1.297 with SE =0.269 and σ_v =1.50. This suggest that there is considerable variation among odds ratio from different studies.

Comparing with the model with only a random intercept, Agresti indicates that the one with a random slope has a smaller p-value for the null hypothesis of $\beta = 0$ (-4.819 versus -10.122). He explains that as $\hat{\sigma}_v$ increases so does the standard error of the treatment estimate ($\hat{\beta}$) tends to increase, in other words, the more the treatment effect varies among studies, the more difficult is to estimate precisely the mean of that effect. When $\sigma_v = 0$ both models have the same $\hat{\beta}$ estimate.

At the end of the section Agresti notes that, as with other examples, the model odds ratios are comprised in a range much narrower than those calculated from the observed values.
To do this in R, first, we need to calculate the fitted values, both positive and negative outcomes, then rearrange these values so that they are in one column. This should be done for the observed values too.

```
> # Sample negative outcomes
> tb10.8$no <- tb10.8$n - tb10.8$y
> # Fitted positive and negative outcomes
> tb10.8$fit.y <- tb10.8$n * fitted(fit.lmer.tb10.8)
> tb10.8$fit.no <- tb10.8$n * (1 - fitted(fit.lmer.tb10.8))
> # Re-arrangement of the data frame
> tb10.8b <- tb10.8
> tb10.8b$response <- rep(1, nrow(tb10.8b))
> tb10.8c <- tb10.8
> tb10.8c$response <- rep(0, nrow(tb10.8b))
> tb10.8group <- rbind(tb10.8b, tb10.8c)
> tb10.8group$count <- c(tb10.8b$y, tb10.8$no)
> tb10.8group$fit <- c(tb10.8b$fit.y, tb10.8$fit.no)
> # Delete unused columns and variables
> tb10.8group <- tb10.8group[, -c(3, 4, 5, 6, 7)]
> rm(tb10.8b, tb10.8c)
```

To check that the data frame was correctly constructed we extract the same observations reported in table 10.8

```
> xtabs(count ~ treat + response + study, data = tb10.8group)[,
+     , c(1, 5, 6)]
```

```
, , study = 1

     response
treat  0  1
    0  2 11
    1  8  7

, , study = 5

     response
treat  0  1
    0 12  0
    1  9  3

, , study = 6

     response
treat  0  1
    0  0  4
    1  3  4
```

Now we can calculate the odds ratio with the function `oddsratio` from the `vcd` package; first we compute for those studies showed in table 10.8

```
> ## library(vcd)
> # Sample odds ratio
> oddsratio(xtabs(count ~ treat + response + study, data = tb10.8group)[,
+     , c(1, 5, 6)], log = FALSE)
```

```
 odds ratios for treat and response by study

        1         5         6
0.1918159 9.2105263 0.1428571
```

```
> # Fitted odds ratio
> oddsratio(xtabs(fit ~ treat + response + study, data = tb10.8group)[,
+     , c(1, 5, 6)], log = FALSE)
```

```
 odds ratios for treat and response by study

        1         5         6
0.1465000 2.5393926 0.1252579
```

Then, with the `min` and `max` functions we can extract the boundaries of the range of odds ratios from all the studies.

```
> ## Sample odds ratio (min and max values)
> odds_sample <- oddsratio(xtabs(count ~ treat + response + study, data = tb10.8group),
+     log = FALSE)
> min(exp(odds_sample$coefficients))
```

```
[1] 0.0002100399
```

```
> max(exp(odds_sample$coefficients))
```

```
[1] 9.210526
```

```
> ## Fitted odds ratio (min and max values)
> odds_fit <- oddsratio(xtabs(fit ~ treat + response + study, data = tb10.8group),
+     log = FALSE)
> min(exp(odds_fit$coefficients))
```

```
[1] 0.004142094
```

```
> max(exp(odds_fit$coefficients))
```

```
[1] 2.539393
```

Note that for study 5 the sample odds ratio isn't infinite (due to a division by zero) because the `oddsratio` function sums 0.5 to each cell before computing in order avoid ending up with undefined estimates (see Agresti 2007, p. 31).

5 Multilevel (hierarchical) models

Hierarchical models refers to observations that have a nested nature, for example a study that measures the performance of students may classify them according to their gender or socio-economic status, but at the same time one may wish to classify students based on the school they attend. Fort this, other level, own may be interested in measure characteristics like the amount spend by student, salary of the teachers, etc.

An advantage of using random effects for nested models, as pointed out in Agresti, p. 314, is that the number of parameters (an therefore the degrees of freedom) get reduced. However as noted latter, we can not, always consider a particular variable as random, we need to analyze if the number and the nature of the sample suggest so.
For the student/school example, the model for the student level may measure whether the student passes grade or not (response variable) in function of gender, race or nay previous failure. This model would be,

$$logit[P(Y_{it} = 1)] = \alpha_i + \beta_1 x_{it1} + \beta_2 x_{it2} + \ldots + \beta_k x_{itk}$$

where i indicates school and t student.
The school level, then, could be related through the intercept value,

$$\alpha_i = \mu_i + \alpha + \gamma_1 w_{i1} + \ldots + \gamma_l w_{il}$$

where $w_{i1}, \ldots, w_{il}$ are the explanatory variables that vary only at the school level (i); for example expenditure per student at school i.
Substituting the value of α_i in the first equation we have,

$$logit[P(Y_{it} = 1)] = \mu_i + \alpha + \gamma_1 w_{i1} + \ldots + \gamma_l w_{il} + \beta_1 x_{it1} + \beta_2 x_{it2} + \ldots + \beta_k x_{itk}$$

Note that the random effect (μ_i) enters only at the level 2, however more general model might include random terms in both levels, or include a random intercept.

Section 10.4.2 shows an example of this type of model from **Raudenbush-2002**. This study measured, for the student level, the socio–economic status, gender, whether he/she spoke central thai dialect, if he/she had breakfast and if he/she had some pre–primary experience. This is represented in the model,

$$logit[P(Y_{it} = 1)] = \alpha_i + \beta_1 ses_{it} + \beta_2 gender_{it} + \beta_3 DI_{it} + \beta_4 BR_{it} + \beta_5 PRE_{it}$$

The second level (school) related to the first through the intercept α_i and meassures the school mean SES, size of the school enrollment and the availability of the texts.

$$\alpha_i = \mu_i + \alpha + \gamma_1 meanses_i + \gamma_2 size_i + \gamma_3 texts_i$$

For comparison, the authors, first, fit a model that only considers the differences for the sampled schools ($n = 356$)

$$logit[P(Y_{it} = 1)] = \mu_i + \alpha$$

This model has estimates, $\hat{\alpha} = -2.22$ and $\hat{\sigma} = 1.30$, so the estimate of at least one retention is $exp(-2.22)/(1 + exp(-2.22)) = 0.10$. For this, the 95% CI is (0.01, 0.58) which is a wide, non very informative, interval[2].

6 Model fitting and inference for GLMMs

The final section briefly discuss the steps and problems involved in the fitting of Generalized Linear Mixed Models (GLMMs). The author mentions that the likelihood function for a GLMM refers to the fixed effects parameters $\alpha, \beta_1, \beta_2, \ldots, \beta_k$ and the parameter σ from a normal distribution with mean of zero and standard deviation of σ. Now, to obtain this function, software eliminates μ_i **(1)** by forming the likelihood function as if the μ_i values were known, and then **(2)** averaging that function with respect to the normal distribution of μ_i ($N(0, \sigma)$). The last step is the difficult one because the calculus base integral used to average with respect to the normal distribution of the random effects doesn't have a closed form.

Section 10.5.2 describes the inference about the fixed effects. As it is usually done for other models the significance of a particular term is evaluated comparing the deviances for a model with an another

[2]It is important to note that the data set isn't available in the book nor in the author's website. Also the estimates for the nested model aren't provided in the book

without that term. This difference has an approximate chi-squared distribution. In R this is done with the `anova` function. Now, inference about the random effects like testing the null hypothesis that the variance is zero, $H_0 : \sigma^2 = 0$ is more complicated, because the variance can not be negative and therefore the null distribution of $\hat{\sigma}$ is not even approximate normal. An alternative for models containing a single random effect is to test the null hypothesis $H_0 : \sigma = 0$ versus $H_a : \sigma > 0$. The null distribution has probability $1/2$ at 0 and $1/2$ following the shape of a chi-squared distribution with df=1. The test statistic value of 0 occurs when $\sigma = 0$, in which case the maximum of the likelihood function is identical under H_0 and H_a. When $\sigma > 0$ and the observed test statistic equals t (*i.e.* the $\hat{\sigma}$ value), the P -value for this test is half the right-tail probability above t for a chi-squared distribution with df = 1.

References

Agresti, Alan (2007). *An Introduction to Categorical Data Analysis.* 2nd ed. Hoboken NJ: Wiley-Interscience. ISBN: 978-0-471-22618-5.

Thompson, Laura A. (2007). *S-plus (and R) Manual to Accompany Agresti's "Categorical Data Analysis" (2002).*