glmmEP: fast and accurate binary response mixed model analysis via expectation propagation

M.P. Wand and J.C.F. Yu

25th May, 2018

1 Introduction

The glmmEP package supports binary response mixed model analysis based on an *expectation propagation* approximation to the log-likelihood. Full details of the methodology, as well as theoretical back-up, are in the article by Hall, Johnstone, Ormerod, Wand & Yu (2018).

The package's central function is `glmmEP()`. In this vignette we explain how to set up the input data matrices for `glmmEP()` and then obtain inferential summaries of the fit. We first use a simulated data set, corresponding to a simulation study described in Hall *et al.* (2018). We then conduct some analyses involving a dataset that arose for a contraception use study.

2 Nature of the Computations and a Caveat

Exact likelihood-based inference for binary response mixed models involves numerical integration with dimension matching that of the random effects vectors. Typically this dimension is a low number between 1 and 5, with 1 (corresponding to random intercepts models) and 2 (corresponding to adding a random slopes effect) being the most common. The essence of the expectation propagation approach is to replace each of the multivariate integrals required for a log-likelihood evaluation by a fixed-point iterative algorithm which, in certain cases, has closed formed updates. This has the attraction of circumventing the numerical integration requirement. In Hall *et al.* (2018) we derive theoretically justifiable starting values, which are used in glmmEP. In our extensive simulation testing (see e.g. Section 4.1 of Hall *et al.*, 2018) we have witnessed excellent convergence of expectation propagation. The approximate likelihood evaluations are then used to search for the maximum value, and approximate the corresponding Hessian, over a multivariate parameter space using the Broyden-Fletcher-Goldfarb-Shanno quasi-Newton iterative algorithm. Starting values for this algorithm are obtained by a preliminary Nelder-Mead iterative algorithm search. However, iterative algorithms are susceptible to breakdown and it is difficult to guard against problems when glmmEP is used to fit an arbitrary data set. We welcome feedback on experiences that users have with glmmEP applied to their data. Our e-mail addresses are `james.yu@student.uts.edu.au` and `matt.wand@uts.edu.au`

3 Illustration for Simulated Data

In an R session the glmmEP package is loaded via the command:

```
library(glmmEP)
```

3.1 Generation and Format of the Simulated Data

The next chunk of code obtains simulated data corresponding to the simulation study in Section 4.1.2. of Hall *et al.* (2018):

```
dataObj <- glmmSimData(seed=54321) ; y <- dataObj$y ; idNum <- dataObj$idNum
Xfixed <- dataObj$Xfixed ; Xrandom <- dataObj$Xrandom
```

The dimension values for these data are:

$$
\begin{aligned}
m &= \text{number of groups} = 2,500,\\
n_i &= \text{number of measurements in the } i\text{th group which is}\\
&\qquad \text{a randomly generated integer between 20 and 30,}\\
\sum_{i=1}^{m} n_i &= \text{total number of measurements} = 6,229,\\
d^{\mathrm{F}} &= \text{fixed effects dimension} = 6\\
\text{and}\quad d^{\mathrm{R}} &= \text{random effects dimension} = 2.
\end{aligned}
$$

The command:

```
print(y[1:100])
```

leads to the output:

```
  [1] 1 0 1 1 1 0 0 1 0 0 0 0 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1 1 1 0 0 1 1 0 1 1 0
 [38] 0 0 1 1 0 0 1 1 1 1 0 1 1 1 0 1 0 1 1 0 0 1 1 1 1 0 0 1 0 1 1 0 0 1 0 1 1
 [75] 1 1 0 1 1 1 1 1 0 0 1 0 1 1 0 0 1 0 1 0 1 1 0 0 0 1
```

which are the first 100 entries of the response vector `y`, which must be numerical with all entries either 0 or 1. The length of `y` is 6,229. The command:

```
print(idNum[1:100])
```

gives

```
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [38] 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4
 [75] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5
```

which are the first 100 entries of the identification number vector `idNum`. The length of `idNum` is 6,229. It is apparent that the sample sizes in the first four groups are $n_1 = 24$, $n_2 = 25$, $n_3 = 21$ and $n_4 = 23$. Next, if one issues:

```
print(Xfixed[1:10,])
```

then the resultant output is:

```
                x1          x2         x3        x4        x5
 [1,] 1 0.36289082 0.002376411 0.09104755 0.1273508 0.3222632
 [2,] 1 0.04930613 0.190524008 0.26741325 0.3577301 0.3285285
 [3,] 1 0.57960823 0.032975174 0.80433496 0.3783690 0.5370373
 [4,] 1 0.81654150 0.092146002 0.94197020 0.4696319 0.6734873
 [5,] 1 0.45523877 0.016108288 0.25312640 0.3708289 0.3986340
 [6,] 1 0.45992203 0.436338940 0.96050362 0.7808369 0.7156205
 [7,] 1 0.23505112 0.710328994 0.57337595 0.8998197 0.7668252
 [8,] 1 0.75835507 0.991308090 0.34212789 0.1302605 0.5837071
 [9,] 1 0.32415353 0.080589397 0.33139670 0.4897236 0.3431149
[10,] 1 0.33242435 0.559411192 0.01339585 0.2474313 0.6049820
```

which displays the first 10 rows of the $6,229 \times 6$ fixed effects design matrix `Xfixed`. An important aspect of `glmmEP()` is that it insists on the first column having all entries equal to 1, corresponding to the fixed effects intercept. Lastly, issuing

```
print(Xrandom[1:10,])
```

gives

```
      x1
 [1,] 1 0.36289082
 [2,] 1 0.04930613
 [3,] 1 0.57960823
 [4,] 1 0.81654150
 [5,] 1 0.45523877
 [6,] 1 0.45992203
 [7,] 1 0.23505112
 [8,] 1 0.75835507
 [9,] 1 0.32415353
[10,] 1 0.33242435
```

Note that `Xrandom` coincides with the first 2 columns of `Xfixed`. This means that, in the upcoming call to `glmmEP()`, there will be random intercepts, and random slopes corresponding to the first predictor `x1`.

3.2 Probit Mixed Model Analysis of the Data

The appropriate fitting command is:

```
fitSimulRanIntAndSlp <- glmmEP(y,Xfixed,Xrandom,idNum)
```

and takes about 20–30 seconds to fit on typical 2018 computers.

An inferential summary of the fit is obtained via:

```
summary(fitSimulRanIntAndSlp)
```

and leads to

```
          95% C.I.low    estimate  95% C.I. upp
intercept  0.13272044  0.30246606   0.4722118
x1         0.69789090  0.88043814   1.0629854
x2        -0.54163761 -0.41337685  -0.2851161
x3        -0.03726938  0.09050807   0.2182855
x4        -1.45392531 -1.31994887  -1.1859724
x5         1.04426639  1.17685492   1.3094436
sigma1     0.60352982  0.70879985   0.8324305
sigma2     0.76176324  0.94008393   1.1601476
rho12     -0.60952776 -0.44659755  -0.2474686
```

The first six rows of this summary table are estimates and 95% confidence intervals for the fixed effects parameters, corresponding to the intercept and the predictors $x_1, \ldots, x_5$. The last three rows are estimates and 95% confidence intervals for the parameters σ_1, σ_2 and ρ_{12} within the random effects covariance matrix:

$$\begin{bmatrix} u_{i0} \\ u_{i1} \end{bmatrix} \overset{\text{ind.}}{\sim} N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 \\ \rho_{12}\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix} \right).$$

For example, the expectation propagation-approximate estimate of the fixed effect associated with x_1 is

$$\underset{\sim}{\widehat{\beta}}_1 = 0.8804 \text{ with corresponding 95\% confidence interval } (0.6979, 1.0630).$$

The estimate of the standard deviation of the random slope is

$$\underset{\sim}{\widehat{\sigma}}_2 = 0.9401 \text{ with corresponding 95\% confidence interval } (0.7618, 1.160).$$

The code

```
uHat <-  fitSimulRanIntAndSlp$randomEffects
plot(uHat[,1],uHat[,2],col="dodgerblue",xlab="random intercepts predicted values",
     ylab="random slopes predicted values",bty="l",lwd=2,cex.lab=1.5,cex.axis=1.5)
abline(v=0,col="slateblue",lwd=2) ; abline(h=0,col="slateblue",lwd=2)
```

leads to the plot shown in Figure 1 concerning the expectation propagation-approximate best predictions of the random effects.

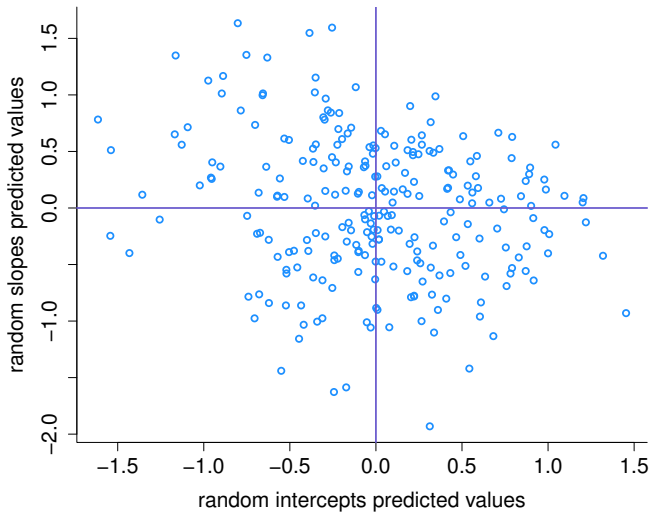

Figure 1: *Scatterplot of the expectation propagation-approximate best predictions of the random slopes and corresponding random intercepts, which are part of the* `fitSimulRanIntAndSlp` *fit object.*

3.3 Controlling the Convergence Parameters

The convergence parameters can be controlled using the function `glmmEP.control()` inside the call to `glmmEP()`. For example, the default number of Nelder-Mead iterations is 100. To increase this to 150 in the creation of `fitSimulRanIntAndSlp` the call to `glmmEP()` should be changed to

```
fitSimultRanIntAndSlp <- glmmEP(y,Xfixed,Xrandom,idNum,
                                control=glmmEP.control(NMmaxit=150))
```

The command

```
help(glmmEP.control)
```

can be used to find the names and default values of each of the other convergence variables.

3.4 Other Control Parameters

There are three other control parameters apart from those concerning convergence criteria. These are

`confLev` which controls the level of the confidence intervals (defaulted to 0.95),

`quiet` which controls whether or not a running commentary of the search for the maximum of the expectation propagation-approximate log-likelihood is printed to the screen (defaulted to `TRUE`),

`preTransfData` which controls whether or not each of the predictor data vectors (i.e. the columns of `Xfixed` and `Xrandom` apart from the intercepts) are transformed to the unit interval for the purposes of finding the expectation propagation-approximate log-likelihood maximum (defaulted to `FALSE`),

Each of these parameters can be controlled via a call to `glmmEP.control()` within the call to `glmmEP()`.

4 Analysis of Data from a Contraception Use Study

Data from the 1988 Bangladesh Fertility Survey are stored in the data frame `Contraception` within the R package mlmRev (Bates, Maechler and Bolker, 2014). Steele, Diamond and Amin (1996) contains details of the study and some multilevel analyses. Variables in the `Contraception` data frame include:

`use` a two-level factor variable indicating whether a woman is a user of contraception at the time of the survey, with levels `Y` for use and `N` for non-use.

`age` age of the woman in years at the time of the survey, centred about the average age of all women in the study.

`district` a multi-level factor variable that codes the district, out of 60 districts in total, in which the woman lives,

`urban` a two-level factor variable indicating whether or not the district in which the woman lives is urban, with levels `Y` for urban dwelling and `N` for rural dwelling.

`livch` a four-level factor variable that indicates the number of living children of the woman, with levels `0` for no children, `1` for one child, `2` for two children and `3+` for three or more children.

The following code leads to the visualisation of the data shown in Figure 2:

```
library(mlmRev) ; data(Contraception) ; library(lattice)
colourVec <- c("forestgreen","sienna")
ContraceptionHiLivCh <- Contraception[Contraception$livch=="3+",]
figRaw <- xyplot(jitter((as.numeric(use)-1),factor=0.5)
          ~ age|district,groups=district,data=ContraceptionHiLivCh,
       layout=c(6,10),
       xlab=list(label="age (years) centred about average",cex=1.35),
       ylab=list(label="indicator of contraception use (jittered)",cex=1.35),
       scales=list(cex=1.25),strip=FALSE,as.table=TRUE,
       key=list(title="subset of data for women with three or more living children",
       columns=2, points=list(pch=rep(1,2),col=colourVec[1:2]),
       text=list(c("rural district","urban district"),cex=1.55)),
       panel=function(x,y,subscripts,groups)
       {
          panel.grid()
          colourInd <- 3 - as.numeric(ContraceptionHiLivCh$urban[subscripts[1]])
          panel.superpose(x,y,subscripts,groups,col=colourVec[colourInd],
                          pch=1,cex=0.5)
       })
print(figRaw)
```

In Figure 2 each panel corresponds to a different district and each point is the `age`/`use` pair for a woman in that district. The `use` data are re-coded as 0 if the woman is a non-user of contraception and 1 if the woman uses contraception. Jittering has been added to these data to aid visualisation. Lastly, Figure 2 is restricted to the subset for which `livch`=`3+`, namely women with three or more living children.

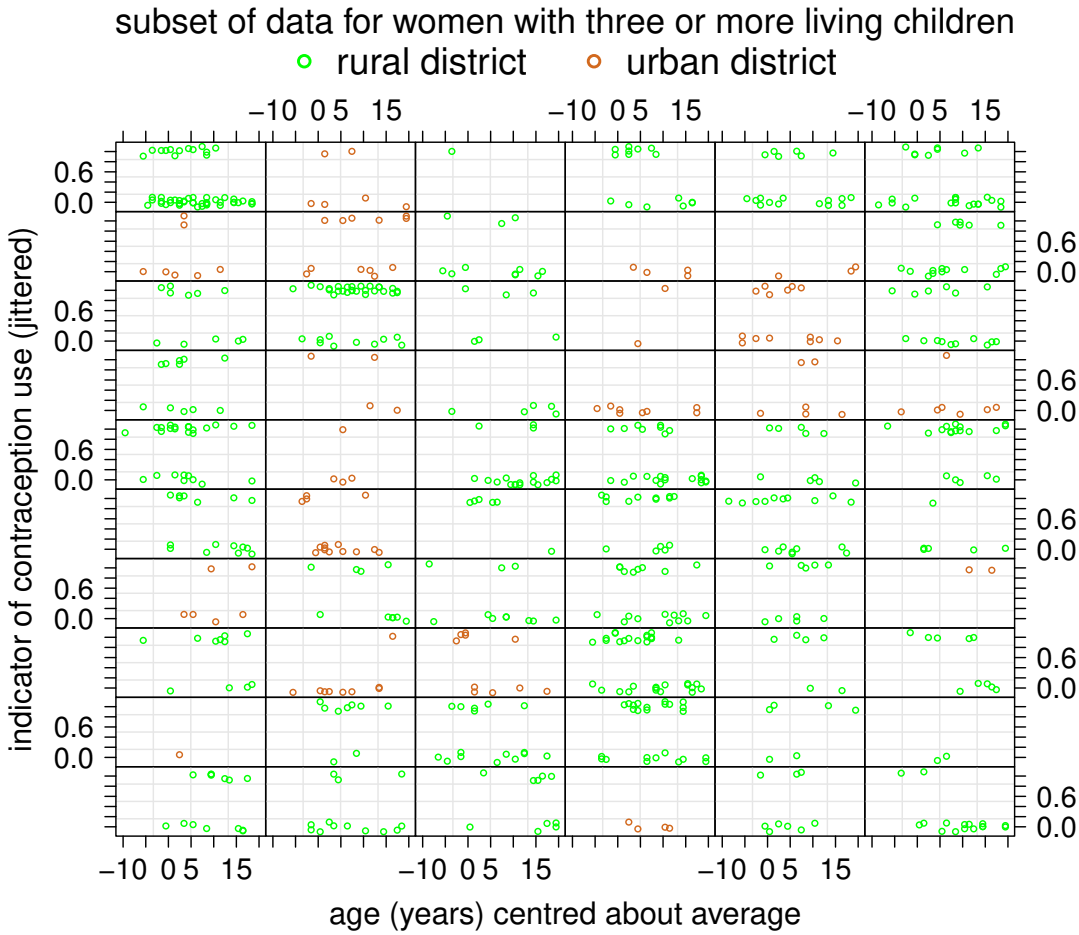

Figure 2: *Visualisation of the* `Contraception` *data frame for the* `livch=3+` *subset, corresponding to women with three or more living children. Each panel is for a different district with colour-coding of the points according to whether the district is rural or urban. The indicators of contraception use values have been jittered to aid visualisation.*

4.1 Random Intercepts Model

Our first analysis of these data using `glmmEP()` involves a probit mixed model with the response variable being the indicator of contraception use and predictors urban versus rural status, age and number of living children. The district is the grouping variable. The following code sets up the input data for fitting via `glmmEP()` (since the current release of `glmmEP()` does not support factor-type variables we are required to use indicator variable coding for categorical variables):

```
y <- as.numeric(as.character(Contraception$use)=="Y")
age <- Contraception$age
isUrban <- as.numeric(as.character(Contraception$urban)=="Y")
livchFactor <- Contraception$livch
livChEq1 <- as.numeric(as.character(Contraception$livch)=="1")
livChEq2 <- as.numeric(as.character(Contraception$livch)=="2")
livChGe3 <- as.numeric(as.character(Contraception$livch)=="3+")
Xfixed <- cbind(1,isUrban,age,livChEq1,livChEq2,livChGe3)
colnames(Xfixed) <- c("intercept","isUrban","age",
                      "livChEq1","livChEq2","livChGe3")
idNum <- as.numeric(as.character(Contraception$district))
```

```
Xrandom <- as.matrix(rep(1,length(y)))
colnames(Xrandom) <- "intercept"
```

Fitting is then achieved via:

```
fitContracRanInt <- glmmEP(y,Xfixed,Xrandom,idNum)
```

and the inferential summary of the model parameters is produced from the command:

```
summary(fitContracRanInt)
```

resulting in the output:

```
          95% C.I. low  estimate  95% C.I. upp
intercept -1.18917625 -1.02853837 -0.867900420
isUrban    0.30660335  0.44911632  0.591629316
age       -0.02568918 -0.01628592 -0.006882654
livChEq1   0.48452054  0.67017817  0.855835821
livChEq2   0.62921680  0.83480529  1.040393799
livChGe3   0.60436497  0.81479526  1.025225571
sigma      0.20312843  0.28250518  0.392900097
```

We see from this output that each of the fixed parameters are statistically significant. For example, the coefficient of the indicator of the district being urban has an estimate of 0.4491163 and a corresponding 95% confidence interval of $(0.3066, 0.5916)$. The random intercept standard deviation corresponds to the row labelled `sigma` and its estimate is 0.2825 with a 95% confidence interval of $(0.2031, 0.3929)$, indicating a significant amount of within-district correlation.

The following code:

```
uHat <- as.numeric(fitContracRanInt$randomEffects)
hist(uHat,xlab="random intercepts predicted values",probability=TRUE,
     col="dodgerblue",breaks=15,main="",cex.lab=1.5)
abline(v=0,col="slateblue",lwd=2)
```

leads to the histogram shown in Figure 3. This is a visual summary of the expectation propagation-approximate best predictions of the random intercepts.

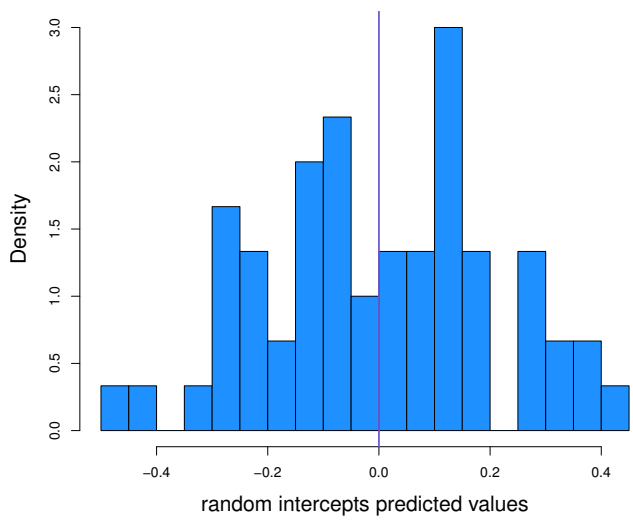

Figure 3: *Histogram of the expectation propagation-approximate best predictions of the random intercepts, which are part of the* `fitContracRanInt` *fit object.*

Lastly, we embellish Figure 2 by computing the estimated probability of contraception use curves. The code for this is:

```
betaHat <- fitContracRanInt$parameters[,2]
ng <- 101 ; ageg <- seq(min(age),max(age),length=ng)
probRanIntg <- vector("list",60)
idNumOrig <- idNum
idNum <- match(idNumOrig,unique(idNumOrig))
distSttInds <- c(1,(1:length(y))[diff(idNum)==1] + 1)
for (i in 1:60)
   probRanIntg[[i]] <- pnorm(betaHat[1]+uHat[i]+betaHat[2]*ageg
                      + betaHat[3]*isUrban[distSttInds[i]]+betaHat[6])
figFitRanInt <- xyplot(jitter((as.numeric(use)-1),factor=0.5)
                 ~ age|district,groups=district,data=ContraceptionHiLivCh,
             layout=c(6,10),
             xlab=list(label="age (years) centred about average",cex=1.35),
             ylab=list(label="indicator of contraception use (jittered)",cex=1.35),
             scales=list(cex=1.25),strip=FALSE,as.table=TRUE,
             key=list(title="subset of data for women with three or more living children",
                      columns=2,
                      points=list(pch=rep(1,2),col=colourVec[1:2],lwd=rep(2,2)),
                      text=list(c("rural district","urban district"),cex=1.55)),
             panel=function(x,y,subscripts,groups)
             {
                iDistrict <- panel.number()  ;  panel.grid()
                colourInd <- 3 - as.numeric(ContraceptionHiLivCh$urban[subscripts[1]])
                panel.superpose(x,y,subscripts,groups,
                                col=colourVec[colourInd],pch=1,cex=0.5)
                panel.xyplot(ageg,probRanIntg[[iDistrict]],col="blue",lwd=2,type="l")
             })
```

Note that the calculation of the ordinate vectors in `probRanIntg` is simplified by the fact that Figure 2 is restricted to the subset of women with three or more living children. The plot that results from this code is shown in Figure 4. It shows that the estimated probability of contraception use increases steeply with age about 5 years either side of the average age. Differences between the districts and those with rural and urban status is difficult to discern visually. However the confidence intervals given earlier in this subsection show that there are, indeed, significant differences.

4.2 Random Intercepts and Slopes Model

We now extend the model to allow the slope of the urban district indicator to be random. The only change in the design matrix set-up is that `Xrandom` is now:

```
Xrandom <- cbind(1,isUrban)
colnames(Xrandom) <- c("intercept","isUrban")
```

With this new version of `Xrandom` we call `glmmEP()` as before using:

```
fitContracRanIntAndSlp <- glmmEP(y,Xfixed,Xrandom,idNum)
```

The inferential summary from the command:

```
summary(fitContracRanIntAndSlp)
```

resulting in the output:

```
          95% C.I. low    estimate 95% C.I. upp
intercept  -1.2184525 -1.04179990 -0.865142851
isUrban     0.2956345  0.50025501  0.704877047
age        -0.0258905 -0.01634954 -0.006808459
livChEq1    0.4932917  0.68153267  0.869774289
```

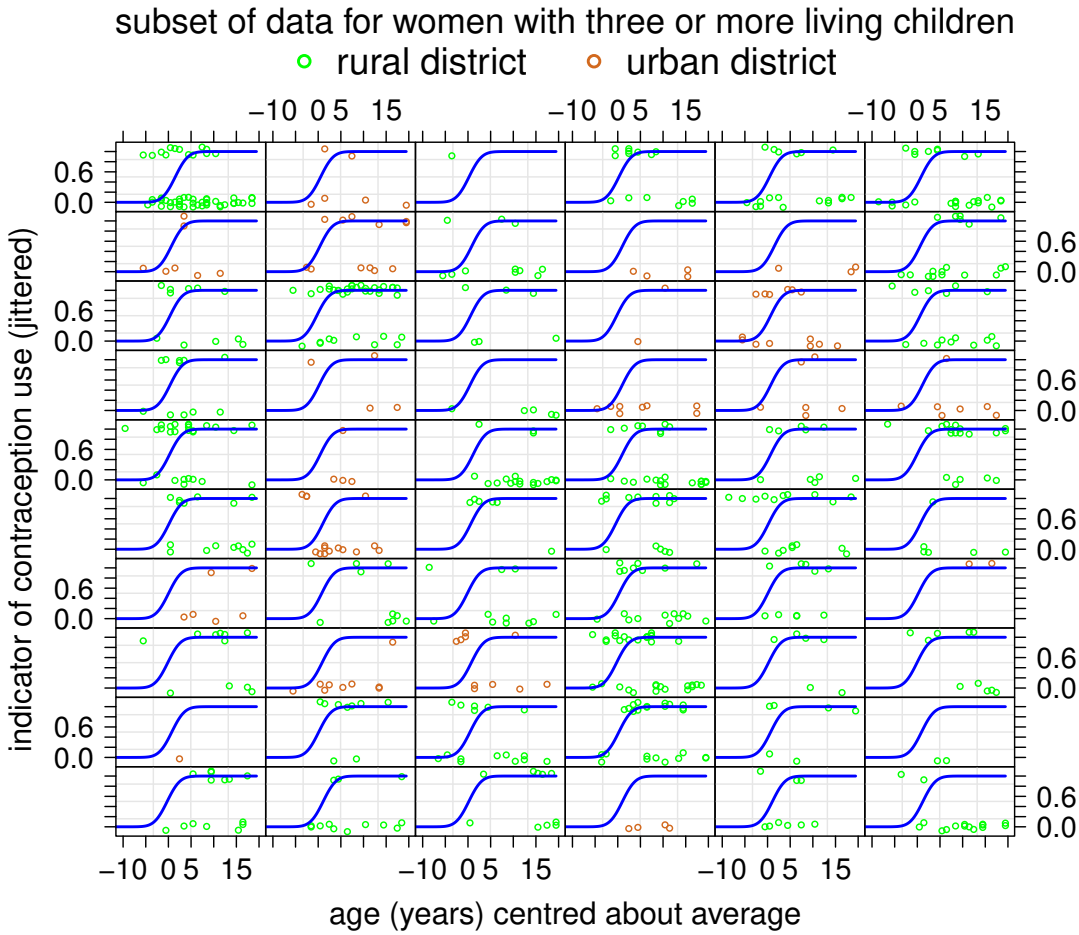

Figure 4: *The data from Figure 2 with the addition of the estimated probability of contraception use curves, according to the expectation propagation random intercepts model fit within the* `fitContracRanInt` *fit object.*

```
livChEq2     0.6222459  0.83057389  1.038903736
livChGe3     0.6101980  0.82444298  1.038693956
sigma1       0.2748112  0.37854225  0.521427402
sigma2       0.3095888  0.49648685  0.796214721
rho12       -0.9367186 -0.79843366 -0.444619228
```

Note that random slope coefficient has estimate

$$\widehat{\sigma}_2 = 0.4965 \text{ with corresponding } 95\% \text{ confidence interval } (0.3108, 0.7931).$$

The tight confidence interval well away from zero verifies significant variability in the random slopes associated with the indicator of a district being urban.

The scatterplot shown in Figure 5 is a visual summary of the expectation propagation-approximate best predictions of the bivariate random intercepts and slopes, and is produced from the following code:

```
uHat <-  fitContracRanIntAndSlp$randomEffects
plot(uHat[,1],uHat[,2],col="dodgerblue",lwd=2,xlab="random intercepts predicted values",
     ylab="random slopes predicted values",bty="l",cex.lab=1.5,cex.axis=1.5)
abline(v=0,col="slateblue",lwd=2) ; abline(h=0,col="slateblue",lwd=2)
```

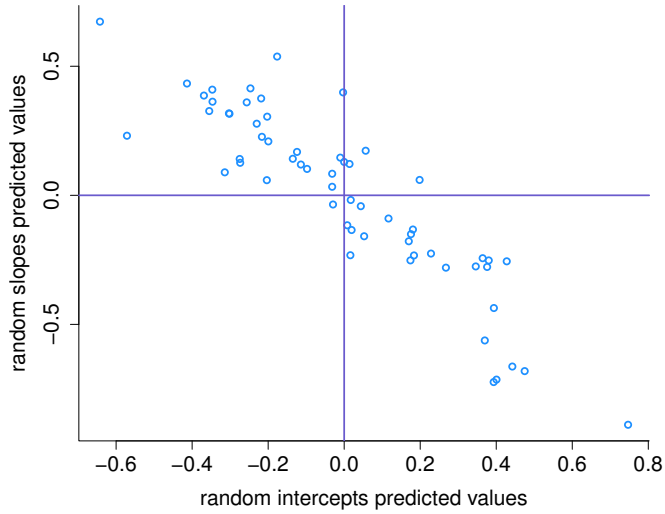

Figure 5: *Scatterplot of the expectation propagation-approximate best predictions of the random slopes and corresponding random intercepts, which are part of the* `fitContracRanIntAndSlp` *fit object.*

Our final plot is the analogue of Figure 4 for the random intercepts and slopes model, which is shown in Figure 6.

Figure 6 is produced using:

```
betaHat <- fitContracRanIntAndSlp$parameters[,2]
probRanIntAndSlpg <- vector("list",60)
for (i in 1:60)
   probRanIntAndSlpg[[i]] <- pnorm(betaHat[1]+uHat[i,1]+betaHat[2]*ageg
                             +(betaHat[3]+uHat[i,2])*isUrban[distSttInds[i]]
                             +betaHat[6])
figFitRanIntAndSlp <- xyplot(jitter((as.numeric(use)-1),factor=0.5)
                  ~ age|district,groups=district,data=ContraceptionHiLivCh,
                  layout=c(6,10),
                  xlab=list(label="age (years) centred about average",cex=1.35),
                  ylab=list(label="indicator of contraception use (jittered)",cex=1.35),
                  scales=list(cex=1.25),strip=FALSE,as.table=TRUE,
                  key=list(title="subset of data for women with three or more living children",
                            columns=2,
                            points=list(pch=rep(1,2),col=colourVec[1:2],lwd=rep(2,2)),
                            text=list(c("rural district","urban district"),cex=1.55)),
                  panel=function(x,y,subscripts,groups)
                  {
                     iDistrict <- panel.number() ; panel.grid()
                     colourInd <- 3 - as.numeric(ContraceptionHiLivCh$urban[subscripts[1]])
                     panel.superpose(x,y,subscripts,groups,
                                     col=colourVec[colourInd],pch=1,cex=0.5)
                     panel.xyplot(ageg,probRanIntAndSlpg[[iDistrict]],col="blue",lwd=2,type="l")
                  })
```

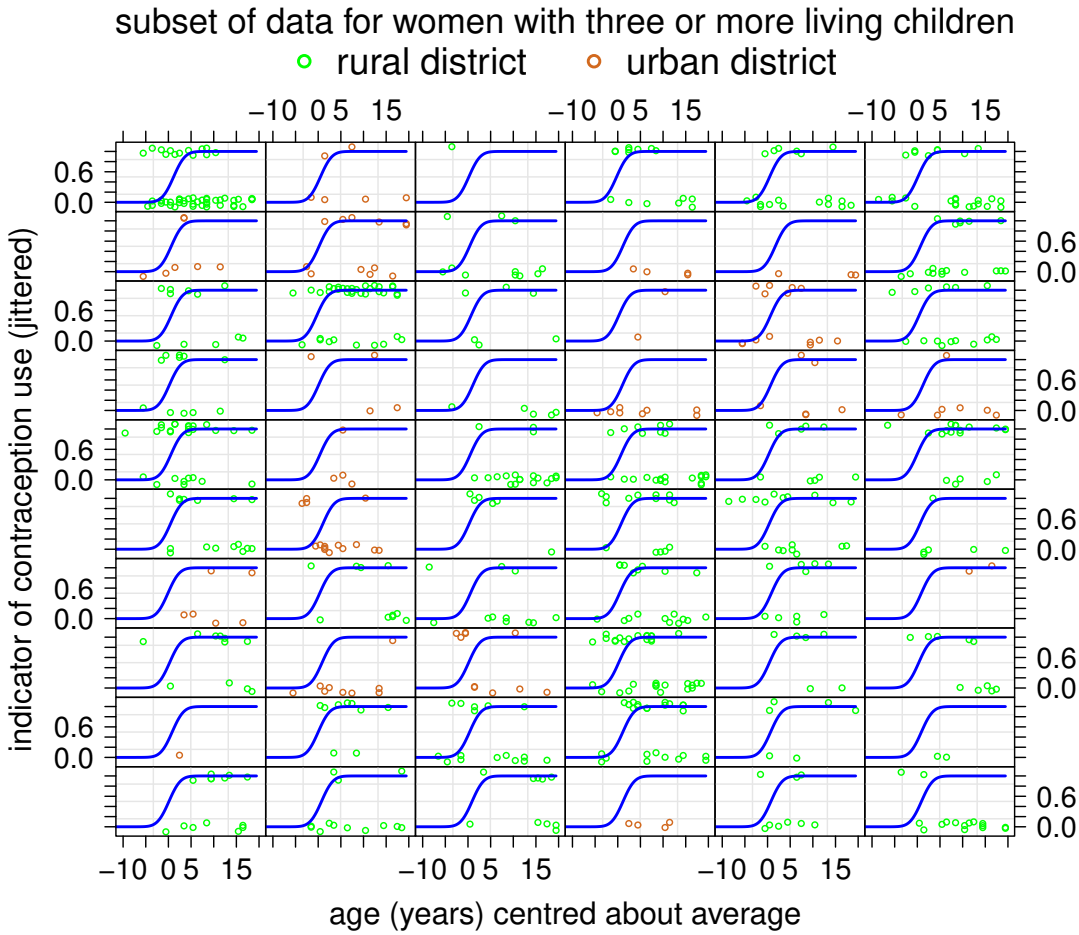

Figure 6: *The data from Figure 2 with the addition of the estimated probability of contraception use curves, according to the expectation propagation random intercepts and slopes model fit within the* `fitContracRanIntAndSlp` *fit object.*

References

Bates, D., Maechler, M. and Bolker, B. (2014). mlmRev: Examples from multilevel modelling software review. R package version 1.0. `http://cran.r-project.org`.

Steele, F., Diamond, I. and Amin, S. (1996). Immunization uptake in rural Bangladesh: a multilevel analysis. *Journal of the Royal Statistical Society, Series A*, **159**, 289–299.

Hall, P., Johnstone, I.M., Ormerod, J.T., Wand, M.P. and Yu, J.C.F. (2018). Fast and accurate binary response mixed model analysis via expectation propagation. Submitted for publication. `http://matt-wand.utsacademics.info/statsPapers.html`