Bayesian inference for linear mixed models

Peter Hoff
Duke STA 610

Multilevel data

```
length(y)

## [1] 2742

length(groups)

## [1] 2742

groups[1:50]

##  [1] 1 1 1 1 1 1 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
## [39] 2 2 2 2 2 2 2 3 3 3 3 3

dim(X)

## [1] 2742    3

X[1:5,]

##         hwh   ses
## [1,] 1    2 -0.23
## [2,] 1    0  0.69
## [3,] 1    4 -0.68
## [4,] 1    5 -0.89
## [5,] 1    3 -1.28

m<-max(groups)
p<-ncol(X)
```

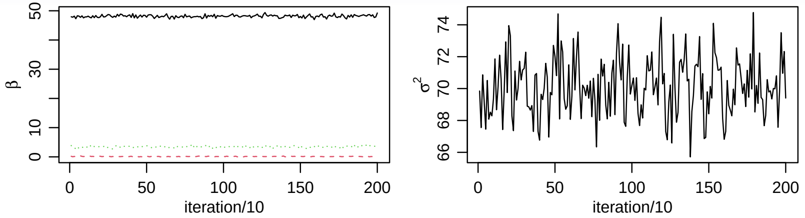
Notice there are no macro predictors in this X-matrix.

Unit information hyperparameters

```
B<-NULL

SSE<-df<-0

for(j in 1:m){

  yj<-y[groups==j]
  Xj<-X[groups==j,,drop=FALSE]
  bj<-c(solve( t(Xj)%*%Xj + diag(p) )%*%(t(Xj)%*%yj))

  B<-rbind(B,bj)
  SSE<-SSE + sum( (yj-Xj%*%bj)^2 ) ; df<-df+max(0,length(yj)-p)

}

s20<-SSE/df ; nu0<-2

beta0<-apply(B,2,mean) ; V0<-diag(p)*s20 ; iV0<-solve(V0)

Psi0<-cov(B) ; iPsi0<-solve(Psi0) ; eta0<-p+1
```

```
## starting values
s2<-s20
beta<-beta0
iPsi<-iPsi0
```

```
BSIM<-array(dim=c(m,p,200))
BETA<-S2<-PSI<-NULL
for(s in 1:2000){

  ## update within-groups parameters
  SSE<-0
  for(j in 1:m){
    yj<-y[groups==j]
    Xj<-X[groups==j,,drop=FALSE]

    Vbj<-solve( iPsi + t(Xj)%*%Xj/s2 )
    Ebj<-Vbj%*%( iPsi%*%beta + t(Xj)%*%yj/s2 )

    bj<-Ebj + t(chol(Vbj))%*%rnorm(p)
    B[j,]<-bj

    SSE<-SSE + sum( (yj-Xj%*%bj)^2 )
  }
  s2<-1/rgamma(1,(nu0+length(y))/2, (nu0*s20 + SSE)/2 )

  ## update across-group parameters
  Vbeta<-solve( iV0 + m*iPsi )
  Ebeta<-Vbeta%*%( iV0%*%beta0 + m*iPsi%*%apply(B,2,mean) )
  beta<-c(Ebeta + t(chol(Vbeta))%*%rnorm(p) )

  SSB<-crossprod( sweep(B,2,beta,"-") )
  iPsi<-rWishart(1,eta0+m,solve( Psi0 + SSB ) )[,,1]

  if(s%%10==0){
    S2<-c(S2,s2)
    BETA<-rbind(BETA,beta)
    PSI<-rbind(PSI,c(solve(iPsi)))
```




```
apply(BETA,2,mean)

## [1] 48.1465488  0.1270704  3.4540289

apply(BETA,2,sd)

## [1] 0.43895252 0.06605531 0.30678029

matrix( apply(PSI,2,mean),p,p)

##             [,1]        [,2]      [,3]
## [1,] 16.60149082 -0.09685625 2.5402417
## [2,] -0.09685625  0.18229805 0.1098322
## [3,]  2.54024167  0.10983218 3.3823442
```

```
library(lme4)
X0<-X[,-1]
fit<-lmer( y ~ X0 + (X0|groups) )

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
Model failed to converge with max|grad| = 0.00735232 (tol = 0.002, component 1)

summary(fit)

## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ X0 + (X0 | groups)
##
## REML criterion at convergence: 19742.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.1139 -0.6458  0.0073  0.6411  4.4791
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  groups   (Intercept) 16.03462 4.0043
##           X0hwh        0.02486 0.1577   0.40
##           X0ses        2.26061 1.5035   0.34 0.57
##  Residual             70.70688 8.4087
## Number of obs: 2742, groups:  groups, 146
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) 48.18445    0.40472 119.055
## X0hwh        0.10105    0.05252   1.924
## X0ses        3.45490    0.27773  12.440
##
```

Gibbs sampler

Given $\beta, \Psi, \sigma^2, b_1, \ldots, b_m$:

1. update $b_1, \ldots, b_m$ given $y_1, \ldots, y_m, \beta, \Psi, \sigma^2$;
2. update σ^2 given $y_1, \ldots, y_m, b_1, \ldots, b_m$;
3. update β given $b_1, \ldots, b_m, \Psi$;
4. update Ψ given $b_1, \ldots, b_m, \beta$.

Updating $b_1, \ldots, b_m$

$$b_j | y_j, \beta, \Psi, \sigma^2 \sim N_p(E_j, V_j)$$

$$V_j = (\Psi^{-1} + X_j^\top X_j/\sigma^2)^{-1}$$

$$E_j = (\Psi^{-1} + X_j^\top X_j/\sigma^2)^{-1}(\Psi^{-1}\beta + X_j^\top y_j/\sigma^2)$$

$$b_j \stackrel{d}{=} E_j + V_j^{1/2} z, \quad z \sim N_p(0, I).$$

```
SSE<-0
for(j in 1:m){
  yj<-y[groups==j]
  Xj<-X[groups==j,,drop=FALSE]

  Vbj<-solve( iPsi + t(Xj)%*%Xj/s2 )
  Ebj<-Vbj%*%( iPsi%*%beta + t(Xj)%*%yj/s2 )

  bj<-Ebj + t(chol(Vbj))%*%rnorm(p)
  B[j,]<-bj

  SSE<-SSE + sum( (yj-Xj%*%bj)^2 )
}
```

Simulating the multivariate normal distribution

Goal: Simulate $b \sim N_p(E, V)$.

Method:

1. simulate $z \sim N_p(0, I)$;
2. set $b = E + V^{1/2} z$, where $V^{1/2} V^{\top/2} = V$.

Check

$$\mathrm{E}[b] = \mathrm{E}[E + V^{1/2} z] = E + 0 = E$$
$$\mathrm{Var}[b] = \mathrm{Var}[E + V^{1/2} z] = \mathrm{Var}[V^{1/2} z] = V^{1/2} V^{\top/2} = V.$$

Updating σ^2

$$1/\sigma^2 | y_1, \ldots, y_m, b_1, \ldots, b_m \sim \text{gamma}((\nu_0 + N)/2, (\nu_0 \sigma_0^2 + SSE(b))/2)$$

$$SSE(b) = \sum_j \sum_i (y_{i,j} - x_{i,j}^\top b_j)^2 = \sum_j ||y_j - X_j b_j||^2$$

```
s2<-1/rgamma(1,(nu0+length(y))/2, (nu0*s20 + SSE)/2 )
```

Updating β

$$\beta | b_1, \ldots, b_m, \Psi \sim N_p(E, V)$$
$$V = (m\Psi^{-1} + V_0^{-1})^{-1}$$
$$E = (m\Psi^{-1} + V_0^{-1})^{-1}(m\Psi_0^{-1}\bar{b} + V_0^{-1}\beta_0)$$

$$\beta \stackrel{d}{=} E + V^{1/2}z, \; z \sim N_p(0, I).$$

```
Vbeta<-solve( iV0 + m*iPsi )
Ebeta<-Vbeta%*%( iV0%*%beta0 + m*iPsi%*%apply(B,2,mean) )
beta<-c(Ebeta + t(chol(Vbeta))%*%rnorm(p) )
```

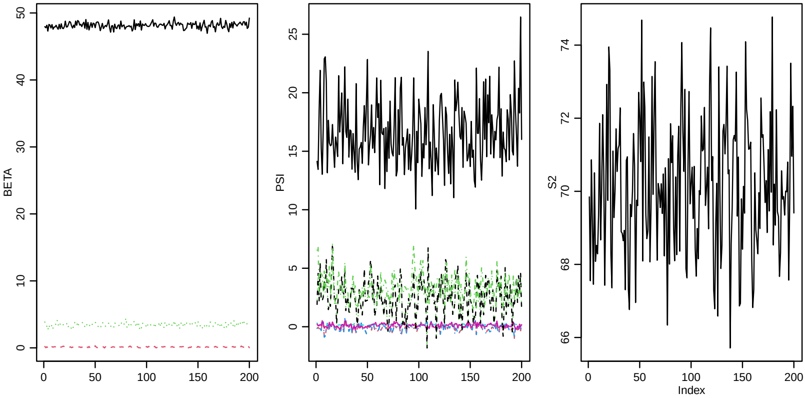
Updating Ψ

$$\Psi^{-1}|b_1,\ldots,b_m,\beta \sim \text{Wishart}(\eta_0 + m, [\Psi_0 + SSB(\beta)]^{-1})$$
$$SSB(\beta) = \sum_j (b_j - \beta)(b_j - \beta)^\top$$

```
SSB<-crossprod( sweep(B,2,beta,"-") )
iPsi<-rWishart(1,eta0+m,solve( Psi0 + SSB ) )[,,1]
```

Notice that the sampler depends on Ψ^{-1} - inversion of Ψ is not necessary.

Posterior diagnostics and summaries




```r
apply(BETA,2,mean)

## [1] 48.1465488  0.1270704  3.4540289

apply(BETA,2,sd)

## [1] 0.43895252 0.06605531 0.30678029
```

Macro predictors

```
WX[1:20,]

##          hflp hwh   ses
##  [1,] 1    1   2 -0.23
##  [2,] 1    1   0  0.69
##  [3,] 1    1   4 -0.68
##  [4,] 1    1   5 -0.89
##  [5,] 1    1   3 -1.28
##  [6,] 1    1   5 -0.93
##  [7,] 1    1   1  0.36
##  [8,] 1    1   4 -0.24
##  [9,] 1    1   8 -1.07
## [10,] 1    1   2 -0.10
## [11,] 1    1   1  0.16
## [12,] 1    1   1 -0.74
## [13,] 1    1   3 -0.58
## [14,] 1    1   0  0.88
## [15,] 1    1   1  0.24
## [16,] 1    1   2  0.08
## [17,] 1    1   1 -1.36
## [18,] 1    1   0 -0.73
## [19,] 1    1   1  1.29
## [20,] 1    1   0 -0.49
```

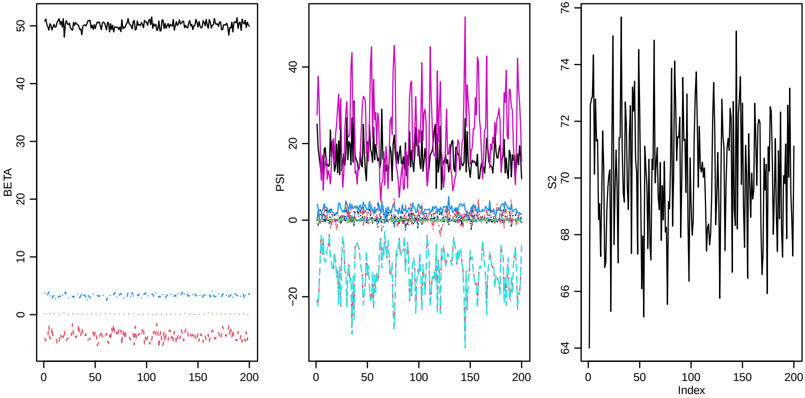
Unit information hyperparameters

```
p<-ncol(WX)

B<-NULL

SSE<-df<-0

for(j in 1:m){

  yj<-y[groups==j]
  Xj<-WX[groups==j,,drop=FALSE]
  bj<-c(solve( t(Xj)%*%Xj + diag(p) )%*%(t(Xj)%*%yj))

  B<-rbind(B,bj)
  SSE<-SSE + sum( (yj-Xj%*%bj)^2 ) ; df<-df+max(0,length(yj)-p)

}

s20<-SSE/df ; nu0<-2

beta0<-apply(B,2,mean) ; V0<-diag(p)*s20 ; iV0<-solve(V0)

Psi0<-cov(B) ; iPsi0<-solve(Psi0) ; eta0<-p+1
```

```
## starting values
s2<-s20
beta<-beta0
iPsi<-iPsi0
```

```
BSIM<-array(dim=c(m,p,200))
BETA<-S2<-PSI<-NULL
for(s in 1:2000){

  ## update within-groups parameters
  SSE<-0
  for(j in 1:m){
    yj<-y[groups==j]
    Xj<-WX[groups==j,,drop=FALSE]

    Vbj<-solve( iPsi + t(Xj)%*%Xj/s2 )
    Ebj<-Vbj%*%( iPsi%*%beta + t(Xj)%*%yj/s2 )

    bj<-Ebj + t(chol(Vbj))%*%rnorm(p)
    B[j,]<-bj

    SSE<-SSE + sum( (yj-Xj%*%bj)^2 )
  }
  s2<-1/rgamma(1,(nu0+length(y))/2, (nu0*s20 + SSE)/2 )

  ## update across-group parameters
  Vbeta<-solve( iV0 + m*iPsi )
  Ebeta<-Vbeta%*%( iV0%*%beta0 + m*iPsi%*%apply(B,2,mean) )
  beta<-c(Ebeta + t(chol(Vbeta))%*%rnorm(p) )

  SSB<-crossprod( sweep(B,2,beta,"-") )
  iPsi<-rWishart(1,eta0+m,solve( Psi0 + SSB ) )[,,1]

  if(s%%10==0){
    S2<-c(S2,s2)
    BETA<-rbind(BETA,beta)
    PSI<-rbind(PSI,c(solve(iPsi)))
```

Posterior diagnostics and summaries



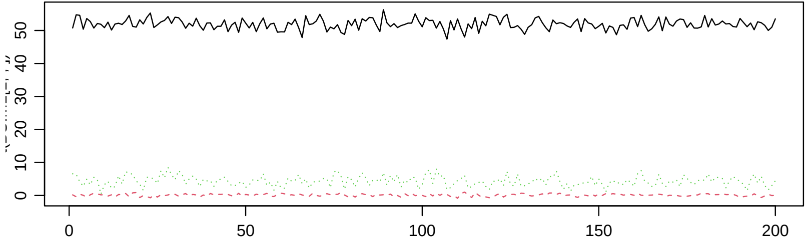

```r
apply(BETA,2,mean)

## [1] 50.1574792 -3.6547129  0.1125781  3.3308918

apply(BETA,2,sd)

## [1] 0.61384791 0.80857057 0.06179939 0.29236705
```

Within-group heterogeneity

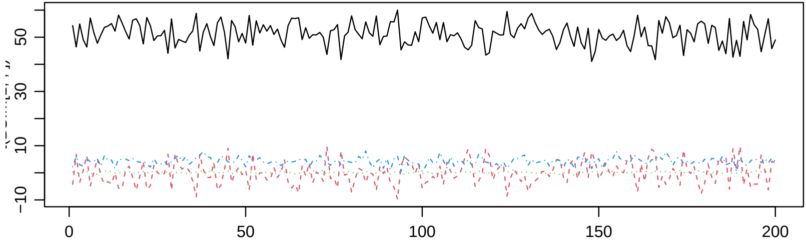
```
matrix( apply(PSI,2,mean),p,p)

##              [,1]        [,2]        [,3]      [,4]
## [1,]  16.52595382 -13.04101241 -0.08555646 0.8143489
## [2,] -13.04101241  22.22442567  0.06608246 0.9628302
## [3,]  -0.08555646   0.06608246  0.16250711 0.1010201
## [4,]   0.81434891   0.96283017  0.10102014 2.9211896
```

Group-level uncertainty

```
matplot(t(BSIM1[1,,]),type="l")
```




```
matplot(t(BSIM[1,,]),type="l")
```



What does lmer do?

```
WX0<-WX[,-1]
WX[1:10,]

##           hflp hwh   ses
##   [1,] 1     1   2 -0.23
##   [2,] 1     1   0  0.69
##   [3,] 1     1   4 -0.68
##   [4,] 1     1   5 -0.89
##   [5,] 1     1   3 -1.28
##   [6,] 1     1   5 -0.93
##   [7,] 1     1   1  0.36
##   [8,] 1     1   4 -0.24
##   [9,] 1     1   8 -1.07
##  [10,] 1     1   2 -0.10

fit<-lmer( y ~ WX0 + (WX0|groups) )

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
Model failed to converge with max|grad| = 0.00529659 (tol = 0.002, component 1)
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
Model is nearly unidentifiable:  large eigenvalue ratio
##  - Rescale variables?
```

Model specification with macro effects

Recall that

- a macro effect is constant within a group;
- the group-level intercept term is constant within a group, so

macro effects and group level intercepts are confounded.

Including macro fixed effect but no random effect requires a reparametrization:

$$
\begin{aligned}
y_{i,j} = & \beta_0 + \beta_1 \times hwh_{i,j} + \beta_2 \times ses_{i,j} + \beta_3 \times flp_j \\
& a_{0,j} + a_{1,j} \times hwh_{i,j} + a_{2,j} \times ses_{i,j} + \\
& \epsilon_{i,j} \\
= & \boldsymbol{\beta}^T \mathbf{x}_{i,j} + \mathbf{a}_j{}^T \mathbf{z}_{i,j} + \epsilon_{i,j}
\end{aligned}
$$

- $\mathbf{x}_{i,j} = (1, hwh_{i,j}, ses_{i,j}, flp_j)$
- $\mathbf{z}_{i,j} = (1, hwh_{i,j}, ses_{i,j})$

Prior specification

$$\mathbf{y}_j = \mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{a}_j + \mathbf{e}_j$$

Hierarchical model

- Across-groups: $a_1, \ldots, a_m \sim$ i.i.d. $N_q(0, \Psi)$;
- Within-groups: $e_1, \ldots, e_m \sim$ i.i.d. $N_n(0, \sigma^2 I)$;

Prior distributions: (same as before)

- $\boldsymbol{\beta} \sim N_p(\boldsymbol{\beta}_0, \boldsymbol{V}_0)$;
- $\Psi^{-1} \sim \text{Wishart}(\eta_0, \Psi_0^{-1})$;
- $1/\sigma^2 \sim \text{gamma}(\nu_0/2, \nu_0\sigma_0^2/2)$;

Full conditional distributions

Random effects:

$$\begin{aligned}(\mathbf{y}_j - \mathbf{X}_j\boldsymbol{\beta}) \equiv \tilde{\mathbf{y}}_j &= \mathbf{Z}_j\mathbf{a}_j + \mathbf{e}_j \\ \mathbf{a}_j &\sim N(0, \Psi) \\ \mathbf{a}_j | \cdots &\sim N(E_j, V_j)\end{aligned}$$

where

$$\begin{aligned}V_j &= (\Psi^{-1} + \mathbf{X}_j^\top \mathbf{X}_j/\sigma^2)^{-1} \\ E_j &= (\Psi^{-1} + \mathbf{X}_j^\top \mathbf{X}_j/\sigma^2)^{-1}\mathbf{X}_j^\top \tilde{\mathbf{y}}_j/\sigma^2\end{aligned}$$

Full conditional distributions

A similar trick can be used to update fixed effects. Recall the "combined" regression across all groups:

$$\begin{aligned}
\mathbf{y} &= \mathbf{X}\boldsymbol{\beta} + \mathbf{Za} + \epsilon \\
\mathbf{y} - \mathbf{Za} \equiv \tilde{\mathbf{y}} &= \mathbf{X}\boldsymbol{\beta} + \mathbf{e} \\
\boldsymbol{\beta} &\sim N(\boldsymbol{\beta}_0, \mathbf{V}_0) \\
\boldsymbol{\beta}| \cdots &\sim N(E, V)
\end{aligned}$$

where

$$\begin{aligned}
V &= (V_0^{-1} + \mathbf{X}^\top \mathbf{X}/\sigma^2)^{-1} \\
E &= (V_0^{-1} + \mathbf{X}^\top \mathbf{X}/\sigma^2)^{-1} \mathbf{X}^\top \tilde{\mathbf{y}}/\sigma^2
\end{aligned}$$

Variance components

$$1/\sigma^2 | y_1, \ldots, y_m, \boldsymbol{\beta}, a_1, \ldots, a_m \sim \text{gamma}((\nu_0 + N)/2, (\nu_0 \sigma_0^2 + SSE)/2)$$

$$SSE = \sum_j \sum_i (y_{i,j} - x_{i,j}^\top \boldsymbol{\beta} - z_{i,j}^\top a_j)^2$$

$$= \sum_j ||y_j - X_j b_j - Z_j a_j||^2$$

$$\Psi^{-1} | a_1, \ldots, a_m \sim \text{Wishart}(\eta_0 + m, [\Psi_0 + SSA]^{-1})$$

$$SSB(\beta) = \sum_j a_j a_j^\top$$

Gibbs sampler

```
##           hwh   ses hflp
##  [1,] 1   2 -0.23    1
##  [2,] 1   0  0.69    1
##  [3,] 1   4 -0.68    1
##  [4,] 1   5 -0.89    1
##  [5,] 1   3 -1.28    1
##  [6,] 1   5 -0.93    1
##  [7,] 1   1  0.36    1
##  [8,] 1   4 -0.24    1
##  [9,] 1   8 -1.07    1
## [10,] 1   2 -0.10    1
##           hwh   ses
##  [1,] 1   2 -0.23
##  [2,] 1   0  0.69
##  [3,] 1   4 -0.68
##  [4,] 1   5 -0.89
##  [5,] 1   3 -1.28
##  [6,] 1   5 -0.93
##  [7,] 1   1  0.36
##  [8,] 1   4 -0.24
##  [9,] 1   8 -1.07
## [10,] 1   2 -0.10
```

Some starting values

```
beta<-lm( y ~ -1+ X )$coef
beta0<-beta ; V0<-diag(50,pf) ; iV0<-solve(V0)

yt<- y-X%*%beta

A<-NULL
SSE<-df<-0
for(j in 1:m){
  ytj<-yt[groups==j,]
  Zj<-Z[groups==j,]
  aj<-c(solve( t(Zj)%*%Zj + diag(pr) )%*%(t(Zj)%*%ytj))

  A<-rbind(A,aj)
  SSE<-SSE + sum( (ytj-Zj%*%aj)^2 ) ; df<-df+max(0,length(ytj)-pr)
}

s2<-s20<-SSE/df ; nu0<-2

Psi<-Psi0<-t(A)%*%A/m ; iPsi<-iPsi0<-solve(Psi0) ; eta0<-pr+1
```

Sampler

```
BETA<-S2<-PSI<-NULL
for(s in 1:2000){

  ## update within-groups parameters
  SSE<-0
  yt<-y - X%*%beta
  for(j in 1:m){
    ytj<-yt[groups==j]
    Zj<-Z[groups==j,,drop=FALSE]

    Vaj<-solve( iPsi + t(Zj)%*%Zj/s2 )
    Eaj<-Vaj%*%t(Zj)%*%ytj/s2

    aj<-Eaj + t(chol(Vaj))%*%rnorm(pr)
    A[j,]<-aj

    SSE<-SSE + sum( (ytj-Zj%*%aj)^2 )
  }
  s2<-1/rgamma(1,(nu0+length(y))/2, (nu0*s20 + SSE)/2 )

  ## update across-group variance
  iPsi<-rWishart(1,eta0+m,solve( Psi0 + crossprod(A) ) )[,,1]
```

Sampler

```
  ## update fixed effects
  yt<-y
  for(j in 1:m){
    ij<-which(groups==j)
    yt[ij]<-y[ij] -Z[ij,]%*%A[j,]
  }
  Vbeta<-solve( iV0 + t(X)%*%X/s2 )
  Ebeta<-Vbeta%*%( iV0%*%beta0 + t(X)%*%yt/s2 )
  beta<-c(Ebeta + t(chol(Vbeta))%*%rnorm(pf) )


  if(s%%10==0){
    S2<-c(S2,s2)
    BETA<-rbind(BETA,beta)
    PSI<-rbind(PSI,c(solve(iPsi)))
  }
}
```

```
apply(BETA,2,mean)

## [1] 50.2640553  0.1035717  3.4020541 -3.8575231

apply(BETA,2,sd)

## [1] 0.5157828 0.0565537 0.2910383 0.6846160

matrix( apply(PSI,2,mean),pr,pr)

##            [,1]       [,2]       [,3]
## [1,] 12.0940490 0.15422535 0.74599385
## [2,]  0.1542254 0.09926969 0.08826288
## [3,]  0.7459938 0.08826288 2.35987261

mean(S2)

## [1] 70.38666
```

What would lmer do?

```
fit<-lmer( y ~  X[,-1] + (Z[,-1]  | groups) )

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
Model failed to converge with max|grad| = 0.0144188 (tol = 0.002, component 1)

summary(fit)

## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ X[, -1] + (Z[, -1] | groups)
##
## REML criterion at convergence: 19714.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0676 -0.6361  0.0098  0.6454  4.4824
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  groups   (Intercept) 11.68770 3.4187
##           Z[, -1]hwh   0.03081 0.1755   0.56
##           Z[, -1]ses   2.08751 1.4448   0.08 0.65
##  Residual             70.63271 8.4043
## Number of obs: 2742, groups:  groups, 146
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) 50.26935    0.51868  96.919
## X[, -1]hwh   0.10647    0.05304   2.008
## X[, -1]ses   3.40612    0.27575  12.352
## X[, -1]hflp -3.85800    0.69855  -5.523
```