Homework 7 – Solutions

by Hedibert Freitas Lopes.

Model A

Independent models for each patient, i.e., $I = 52$ separate models, one for each patient. With unkown logistic intercept β_{0i} and slope β_{1i}:

$$y_{ij} = \frac{exp(\beta_{0i} + \beta_{1i}t_{ij})}{1 + exp(\beta_{0i} + \beta_{1i}t_{ij})} + \epsilon_{ij}, \quad \epsilon_{ij} \sim NID(0, \sigma^2),$$

with $i = 1, \cdots, I$ and $j = 1, \cdots, n_i$. We estimate Model A by maximum likelihood estimation only. Figure 1 shows the fitted logistic curve for 40 out of 52 patients. See the Splus code in the appendix. Figure 2a presents the maximum likelihood estimator for (β_{0i}, β_{1i}), $i = 1, \dots, I$. Figure 2b shows a scatterplot of the estimated pairs $(\hat{\beta}_{0i}, \hat{\beta}_{1i})$ for each patient.

Model B

Now we will consider a longitudinal data model with normal/linear structure. Tomake a normal linear model meaningful we consider a transformation of y_{ij} to

$z_{ij} = log\left(y_{ij}/(1 - y_{ij})\right)$ and for this transformed data we consider the following hierarchical model,

$$\begin{aligned}
z_{ij} &= \gamma_{0i} + \gamma_{1i}t_{ij} + \epsilon_{ij} \\
\epsilon_{ij} &\sim N(0, \sigma^2) \quad i = 1, \cdots, I \text{and } j = 1, \cdots, n_i \\
\gamma_{0i} &\sim N(\phi_0, \tau_0) \\
\gamma_{1i} &\sim N(\phi_1, \tau_1) \\
\phi_k &\sim N(\mu_k, \rho_k) \qquad k = 0, 1,
\end{aligned}$$

with fixed hyperparameters $\sigma^2 = 7.5, \tau_0 = 0.151, \tau_1 = 0.032, \mu_0 = \mu_1 = 0, \rho_0 = 10000$ and $\rho_1 = 10000$.

Posterior distributions

The previous model can be rewritten in a more compact way as,

$$\begin{aligned}
z &\sim N(A_1\theta, C_1) \\
\theta &\sim N(A_2\phi, C_2) \\
\phi &\sim N(\mu, C_3).
\end{aligned}$$

where $n = \sum_{i=1}^{I} n_i$,$C_1 = \sigma^2 I_n$ and,

$$
z = \left(\begin{array}{c} z_{11} \\ z_{12} \\ \ddots \\ z_{1n_1} \\ z_{21} \\ z_{22} \\ \ddots \\ z_{2n_2} \\ \ddots\, z_{I1} \\ z_{I2} \\ \ddots \\ z_{In_I} \end{array} \right), A_1 = \left(\begin{array}{cccc|cccc}
1 & 0 & \cdots & 0 & t_{11} & 0 & \cdots & 0 \\
1 & 0 & \cdots & 0 & t_{12} & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
1 & 0 & \cdots & 0 & t_{1n_1} & 0 & \cdots & 0 \\
\hline
0 & 1 & \cdots & 0 & 0 & t_{21} & \cdots & 0 \\
0 & 1 & \cdots & 0 & 0 & t_{22} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 1 & \cdots & 0 & 0 & t_{2n_2} & \cdots & 0 \\
\hline
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\hline
0 & 0 & \cdots & 1 & 0 & 0 & \cdots & t_{I1} \\
0 & 0 & \cdots & 1 & 0 & 0 & \cdots & t_{I2} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 1 & 0 & 0 & \cdots & t_{In_I}
\end{array} \right), \theta = \left(\begin{array}{c} \gamma_{01} \\ \gamma_{11} \\ \vdots \\ \gamma_{0I} \\ \gamma_{1I} \end{array} \right),
$$

and

$$
A_2 = \left(\begin{array}{cc} 1_I & 0 \\ 0 & 1_I \end{array} \right), \phi = \left(\begin{array}{c} \phi_0 \\ \phi_1 \end{array} \right), \mu = \left(\begin{array}{c} \mu_0 \\ \mu_1 \end{array} \right), C_2 = \left(\begin{array}{cc} \tau_0 I_I & 0 \\ 0 & \tau_1 I_I) \end{array} \right), C_3 = \left(\begin{array}{cc} \rho_0 & 0 \\ 0 & \rho_1 \end{array} \right).
$$

Therefore, by using the result given in Lindley and Smith (1972) (hand-outs), the posterior for θ given $z, A_1, A_2, C_1, C_2, C_3$ and μ, is,

$$
(\theta | Data) \sim N(m, S)
$$

where

$$
\begin{aligned}
S &= \{A_1' C_1^{-1} A_1 + [C_2 + A_2 C_3 A_2']^{-1}\}^{-1} \\
m &= S\{A_1' C_1^{-1} z + [C_2 + A_2 C_3 A_2']^{-1} A_2 \mu\}
\end{aligned}
$$

It is worth to note that we do not need to create, for instance, the matrices A_1 (and some others), since we can use the following simplifications,[1]

$$
A_1' C_1^{-1} z = \frac{1}{\sigma^2} \left(\begin{array}{c} \sum z_{1j} \\ \sum z_{2j} \\ \vdots \\ \sum z_{Ij} \\ \hline \sum z_{1j} t_{1j} \\ \sum z_{1j} t_{1j} \\ \vdots \\ \sum z_{1j} t_{1j} \end{array} \right)
$$

[1]If you notice such simplifications – great, use them. But if you don't, it's okay to use the original matrices, like A_1 etc. and do the matrix algebra by Splus.

$$
C_2 + A_2 C_3 A_2' = \begin{pmatrix} c_{11} I_I + \rho_0 1_I 1_I' & 0 \\ 0 & c_{22} I_I + \rho_1 1_I 1_I' \end{pmatrix}
$$

$$
A_1' C_1^{-1} A_1 = \frac{1}{\sigma^2} \left(\begin{array}{cccc|cccc} n_1 & 0 & 0 & 0 & \sum t_{1j} & 0 & 0 & 0 \\ 0 & n_2 & 0 & 0 & 0 & \sum t_{2j} & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & n_I & 0 & 0 & 0 & \sum t_{Ij} \\ \hline \sum t_{1j} & 0 & 0 & 0 & \sum t_{1j}^2 & 0 & 0 & 0 \\ 0 & \sum t_{2j} & 0 & 0 & 0 & \sum t_{2j}^2 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \sum t_{Ij} & 0 & 0 & \sum t_{Ij}^2 & \end{array} \right)
$$

I wrote an S+ code to deal with these calculations. Figure 3a shows the scatterplot of the intercepts and slopes for each one of $I = 52$ patient. Figure 3b presents these intercepts and slopes along with 95% HPD intervals.

Posterior predictive Distributions

Analogously, it can be proved that the conditional posterior of ϕ given θ is given by,

$$
(\phi|\theta) \sim N(u, T)
$$

where,

$$
\begin{aligned}
u &= T[A_2' C_2^{-1} \theta + C_3^{-1} \mu] \\
T^{-1} &= [A_2' C_2^{-1} A_2 + C_3^{-1}]
\end{aligned}
$$

Applying this result to our case, we ended up with the following expressiosn for T and u, respectively,

$$
T = \begin{pmatrix} \frac{c_{11}\rho_0}{\rho_0 I + c_{11}} & 0 \\ 0 & \frac{c_{22}\rho_1}{\rho_1 I + c_{22}} \end{pmatrix}, u = \begin{pmatrix} \frac{\rho_0 \sum \gamma_{0i}}{\rho_0 I + c_{11}} \\ \frac{\rho_1 \sum \gamma_{1i}}{\rho_1 I + c_{22}} \end{pmatrix}
$$

Therefore, it is straightforward to obtain, by simulation, the posterior predictive distribuiton of y_{ij} for a new patient $i = 53$. Setting up $t_j = j \quad j = 0, 1, \cdots, 12$ (which corresponds to the range of t_{ij} observed for the previous $I = 52$ patients), and M(=1000 in my case) simulations, I have followed the steps below:

1. Generate $\theta^{(l)}$ from $p(\theta|data)$;

2. Generate $\phi^{(l)}$ from $p(\phi|\theta^{(l)}, data)$;

3. Generate $(\gamma_{0,53}^{(l)}, \gamma_{1,53}^{(l)})$ from $p(\gamma|\phi, Data)$;

4. Generate $z_{53,j}^{(l)}$ from $N(\gamma_{0,53}^{(l)} + \gamma_{1,53}^{(l)} t_{53,j}, \sigma^2)$;

5. Calculate $y_{53,j}^{(l)} = \frac{exp(z_{53,j}^{(l)})}{1+exp(z_{53,j}^{(l)})}$.

Using this algorithm and the results from previous sections, Figure 4 plots the predictive distribution for the proportion of White blood for a new patient, which looks like that one presented in class, fortunately.

Model C

Now we will consider a longitudinal data model with logistic likelihood and a single bivariate normal for the prior:

$$
\begin{aligned}
y_{ij} &= \frac{\beta_{0i} + \beta_{1i} t_{ij}}{1 + \beta_{0i} + \beta_{1i} t_{ij}} + \epsilon_{ij} \\
\epsilon_{ij} &\sim N(0, \sigma^2) \quad i = 1, \cdots, I \text{and } j = 1, \cdots, n_i \\
\beta_{0i} &\sim N(\phi_1, \tau_1) \\
(\beta_{1i}|\beta_{0i}) &\sim N(\phi_2 + \phi_3(\beta_{0i} - \phi_1), \tau_2) \\
\phi_k &\sim N(\mu_k, \rho_k) \qquad k = 1, 2, 3.
\end{aligned}
$$

Again I fixed, as suggested, $\sigma^2 = 7.5, \tau_1 = 0.151, \tau_2 = 0.032, \mu_1 = \mu_2 = \mu_3 = 0$ and $\rho_1 = \rho_2 = \rho_3 = 1$. I have simulated 2000 times in bugs (see the appendix), and I kept the last 1000 to make inference. The trace plots for the intercepts and slopes for some patients are plotted in Figure 5. Figure 6 presents the trace plot for ϕ's. It seems that the chains converged to their stationary distribution, so we can use them to make our inference.

The scatterplot containing the intercepts and slopes for all patients is plotted in Figure 7a, while these coefficients and HPD intervals are plotted in Figure 7b.

Using the ϕ's simulated in BUGS (It took 1 hour) I have reentered in the algorithm described in the previous section through step (3) .The proportion of white bood for a new patient is plotted in Figure 8.

Model D

Now we will consider a longitudinal data model with logistic likelihood and a mixture prior for the first stage, i.e.

$$
\begin{aligned}
y_{ij} &= \frac{\beta_{0i} + \beta_{1i} t_{ij}}{1 + \beta_{0i} + \beta_{1i} t_{ij}} + \epsilon_{ij} \\
\epsilon_{ij} &\sim N(0, \sigma^2) \quad i = 1, \cdots, I \text{and } j = 1, \cdots, n_i
\end{aligned}
$$

$$
\begin{aligned}
\beta_{0i} &\sim \sum_{m=1}^{3} \omega_m N(\phi_{1m}, \tau_1) \\
(\beta_{1i}|\beta_{0i}) &\sim \sum_{m=1}^{3} \omega_m N(\phi_{2m} + \phi_{3m}(\beta_{0i} - \phi_{1m}), \tau_2) \\
\phi_{km} &\sim N(\mu_m, \rho_m) \qquad k = 1, 2, 3.
\end{aligned}
$$

Again I fixed, as suggested, $\sigma^2 = 7.5, \tau_1 = 0.151, \tau_2 = 0.032, \mu_1 = -3, \mu_2 = 0, \mu_3 = 3$ and $\rho_1 = \rho_2 = \rho_3 = 1$. As in model C, I have simulated 2000 times in bugs (see the appendix), and I kept the last 1000 to make inference (It took 2.2 hours). The trace plots for the intercepts and slopes for some patients are plotted in Figure 9. Figure 10a presents the trace plot for ϕ's. The trace plot for the mixture weights are plotted in Figure 10b, which are not so constant, indicating that the chain is jumping from one to another component of the mixture.

Figure 11a shows the scatterplot of the intercept and slopes for each patient and figure 11b shows these coefficientes with HPD intervals.

The posterior distribuiton for β_0 and β_1 for a new patient is shown in Figure 12. These pictures presents, at least, bimodality, with can be supported by the model, but can also be an identification problem. I have set up ϕ's quite differents to avoid, or to minimize identifiability problems.

Finally, figure 13 show the S-curve for a new patient. Actually, S-shape is not the form of this curve, and I think the multimodality is the cause of this problem. Unfortunately I do not know the cure for this problem.

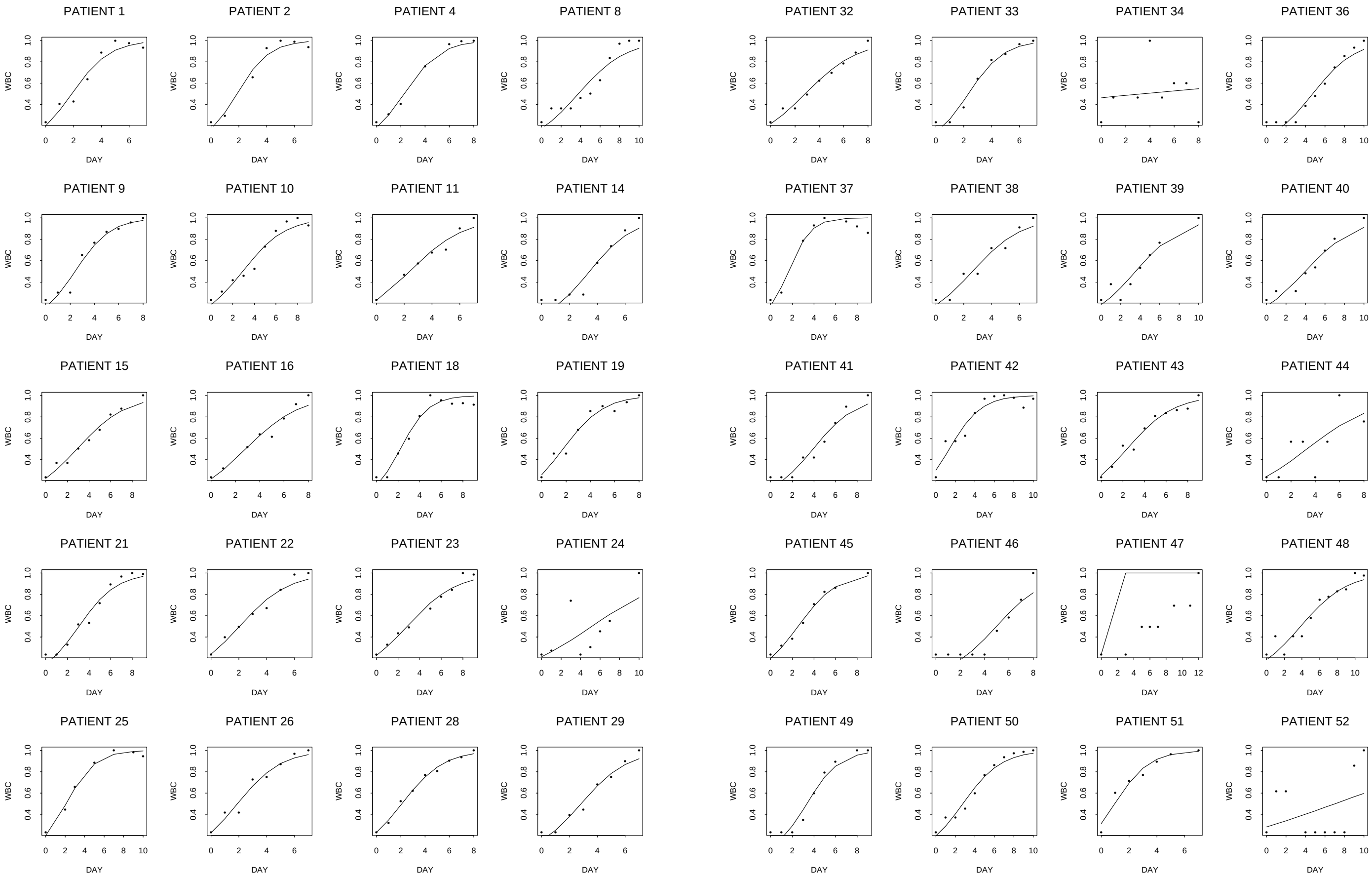

Figure 1: Model A. Fitted curves : Classical Analysis for each patient

newpage

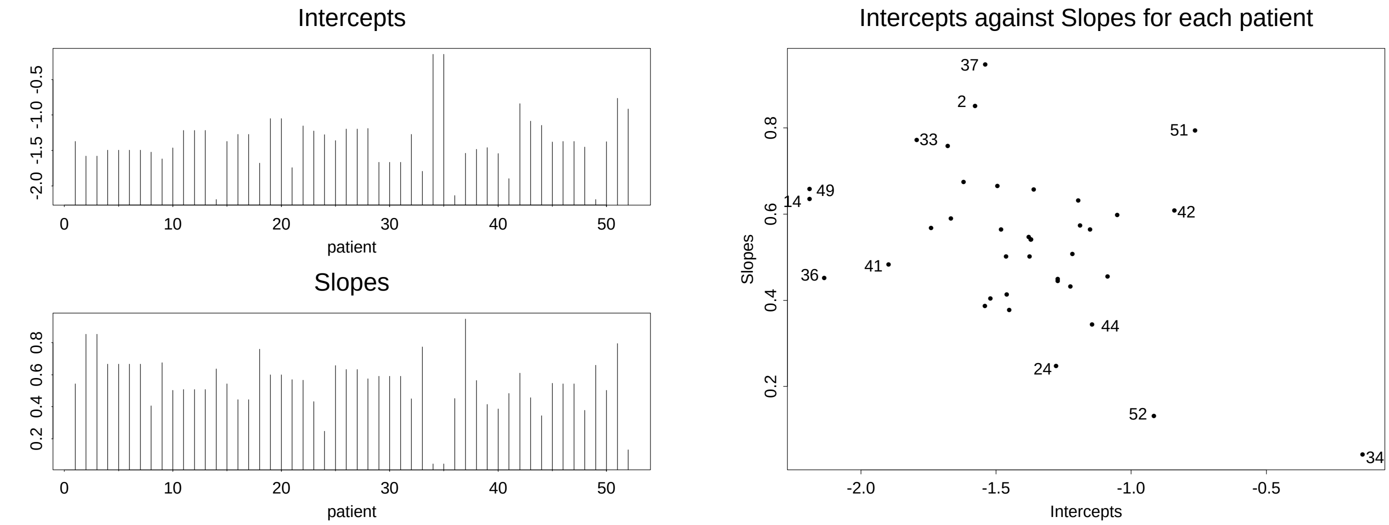

Figure 2: Model A. M.l.e.'s for intercept and slope for each patient (left panel) and scatterplot for slopes against the intercepts (right panel).

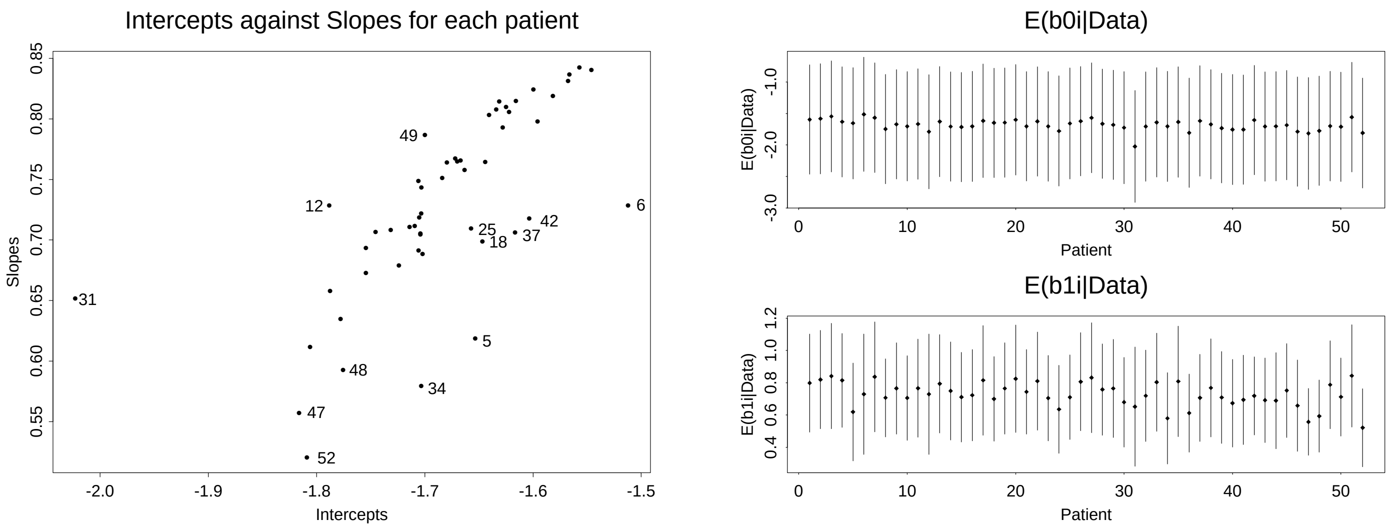

Figure 3: Model B. Posterior means: Scatterplot of slopes against the intercept (left panel). Intercept and Slopes for each patient (right panel).

Figure 4: Fitted logistic curve and quantiles for a new patient (Model B)

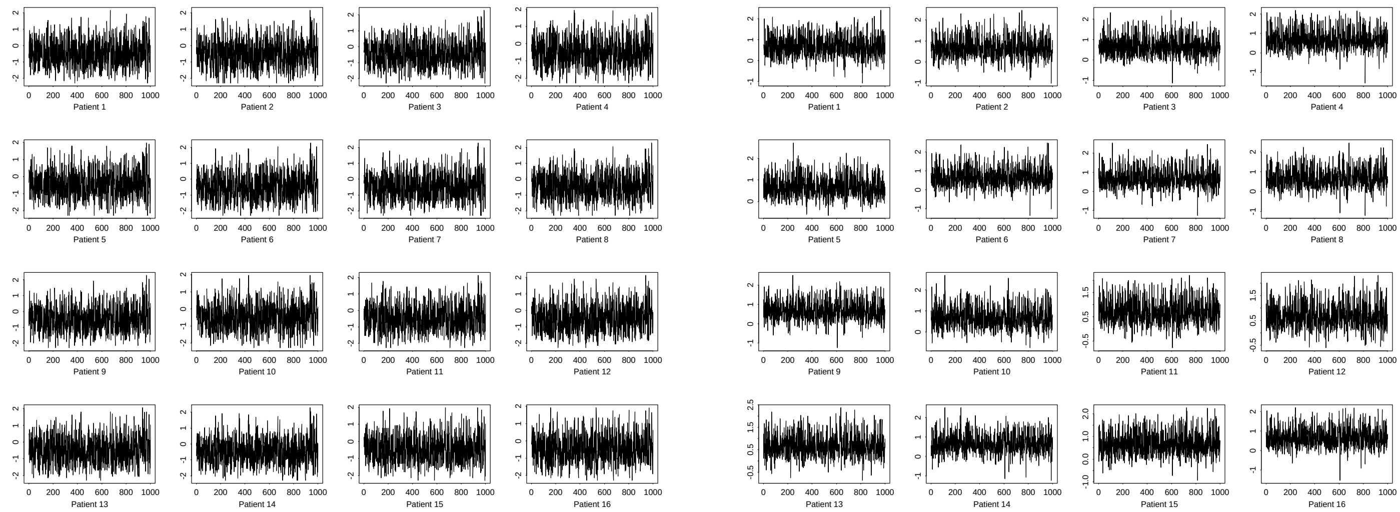

Figure 5: Model C. Simulation trajectories: β_{0i} (left panel) and β_{1i} (right panel) for some selectetd patients. Lack of any obvious trend gives evidence that the MCMC simulations have practically converged.

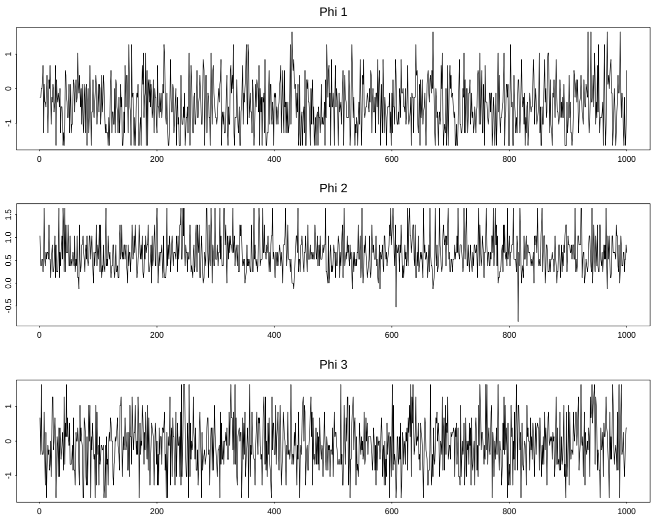

Figure 6: Model C. Simulation trajectories for ϕ.

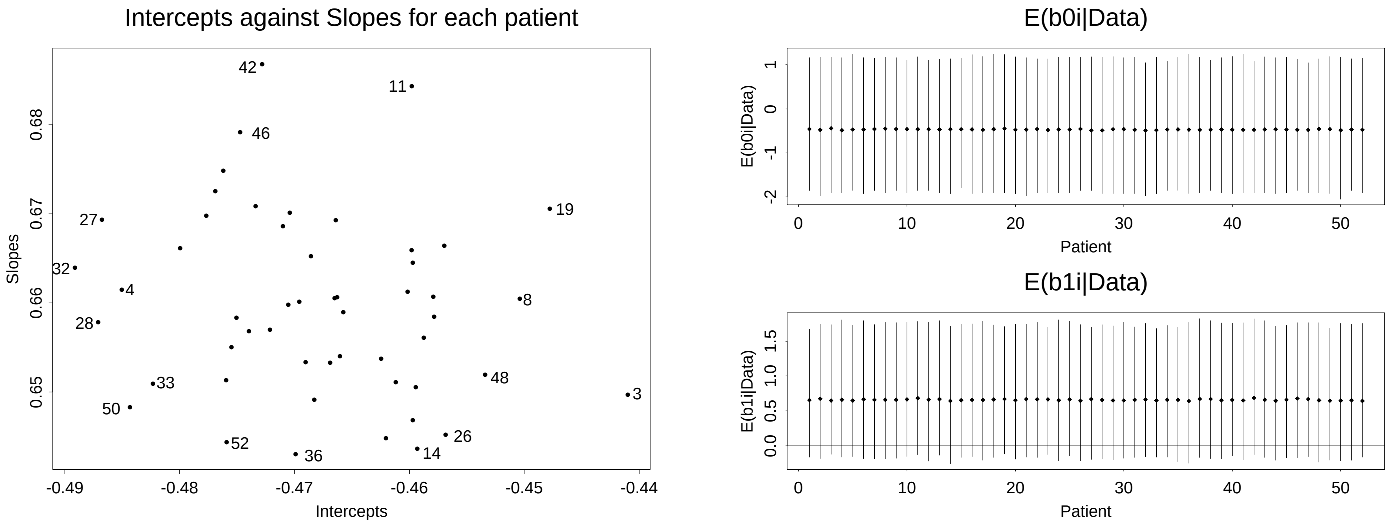

Figure 7: Model C. Scatterplot of posterior means $(\bar{\beta}_{0i}, \bar{\beta}_{1i})$, $i = 1, \ldots, I$ (left panel). Boxplots of $p(\beta_{0i}|y)$ and $p(\beta_{1i}|y)$ (right panel).

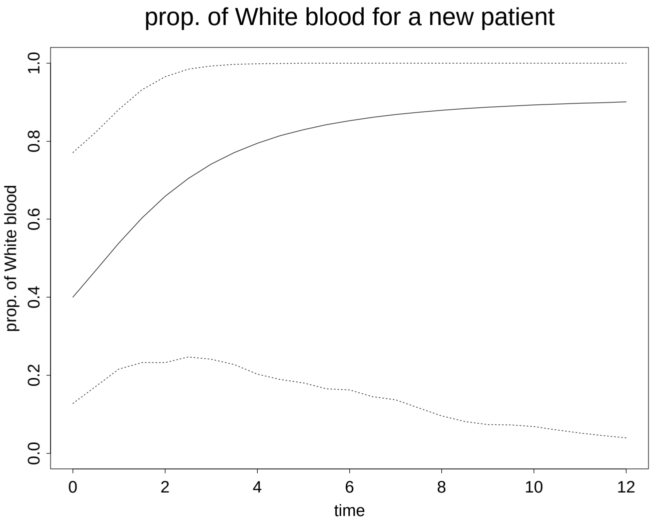

Figure 8: Model C. Fitted logistic curve and quantiles for a new patient (solid curve) with one posterior standard deviation margins (dotted line). Note: Something's wrong with the lower one s.d. margin ..

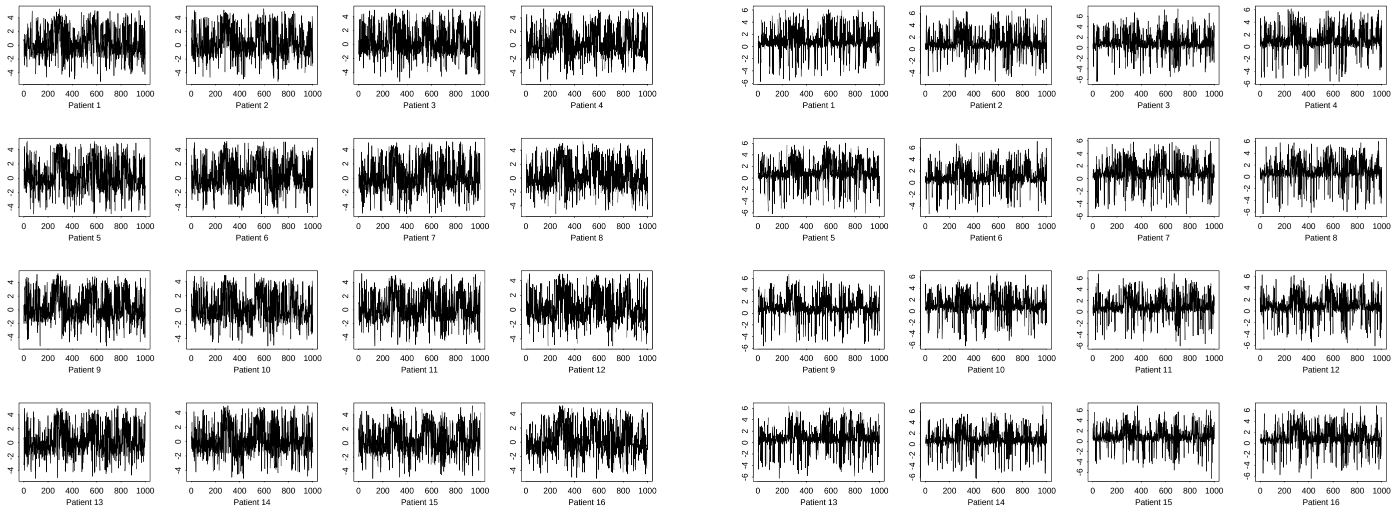

Figure 9: Model D. Simulation trajectories: β_{0i} (left panel) and β_{1i} (right panel) for some selectetd patients. Lack of any obvious trend gives evidence that the MCMC simulations have practically converged.

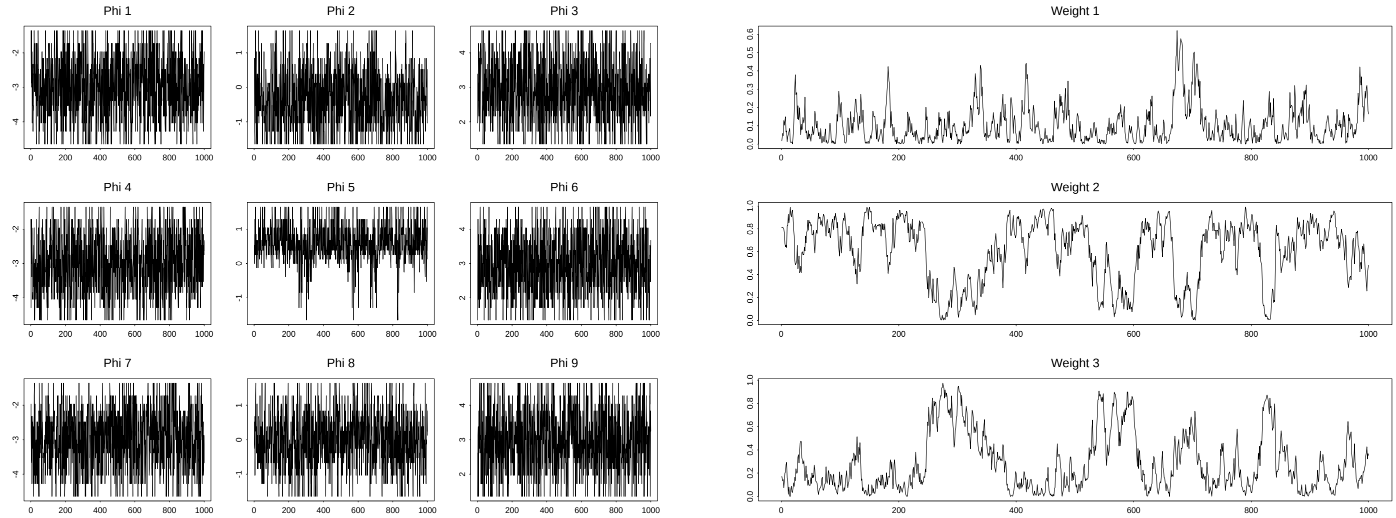

Figure 10: Model D. Simulation trajectories for ϕ and the weights w_j.

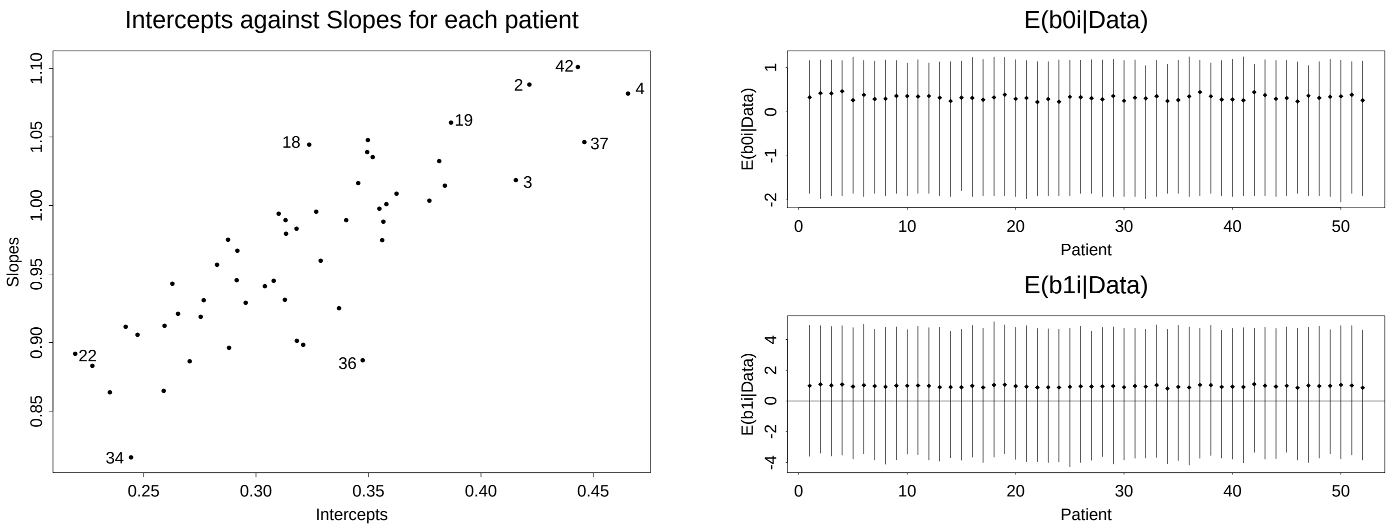

Figure 11: Scatterplot for slopes against the intercept (Model D)

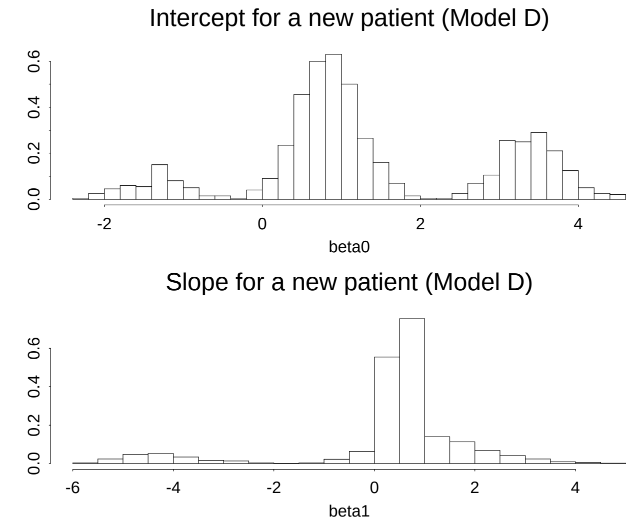

Figure 12: Posterior distribuiton of β_0 and β_1 (Model D)

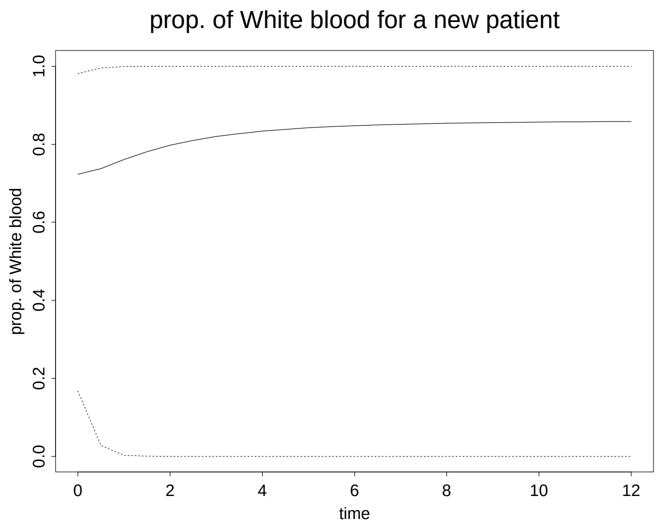

Figure 13: Fitted logistic curve and quantiles for a new patient (Model D)

Bugs Code

Model C

```
model hw7c;

const
  T=12,       # maximum number of observations per patient
  I=52;       # number of patients
var
  ni[I],      # number of observations per patient
  Y[I,T],     # White Cells
  time[I,T],  # instante of time
  qnorm[19],  # 10%,30%,50%,70%,90% normal quantiles.
  p[19],      # Discrete distribuiton on qnorm
  mu[I,T],    # Mean of the Yij's
  ibeta0[I],ibeta1[I],beta0[I],beta1[I],
  tau1,tau2,mu1,mu2,mu3,inv.sig2,
  phi1,phi2,phi3,rho1,rho2,rho3,iphi1,iphi2,iphi3;

data ni in "ni", Y in "Y", time in "day", qnorm in "qnorm";
{
 for (i in 1:19){
    p[i] <- 1/19;
 }
 # Model specification
 for (i in 1:I){
   for (j in 1:ni[i]){
      logit(mu[i,j]) <- beta0[i]+beta1[i]*time[i,j];
      Y[i,j] ~ dnorm(mu[i,j],inv.sig2);
   }
   # Prior specification : 1st level
   ibeta0[i] ~ dcat(p[]);
   ibeta1[i] ~ dcat(p[]);
   beta0[i] <- qnorm[ibeta0[i]]*sqrt(tau1)+phi1;
   beta1[i] <- qnorm[ibeta1[i]]*sqrt(tau2)+phi2+phi3*(beta0[i]-phi1);
 }
 # Prior Specification : 2nd level
 iphi1 ~ dcat(p[]);
 iphi2 ~ dcat(p[]);
 iphi3 ~ dcat(p[]);
 phi1 <- qnorm[iphi1]*sqrt(rho1)+mu1;
 phi2 <- qnorm[iphi2]*sqrt(rho2)+mu2;
 phi3 <- qnorm[iphi3]*sqrt(rho3)+mu3;

 # Constants
 inv.sig2 <- 1/7.490159;
 tau1 <- 0.1506923;
 tau2 <- 0.0317020;
 mu1  <- 0;
 mu2  <- 0;
 mu3  <- 0;
 rho1 <- 1;
 rho2 <- 1;
 rho3 <- 1;
}
```

Model D

```
model hw7d;

const
  T=12,       # maximum number of observations per patient
  I=52,       # number of patients
  K=3,        # number of components in the mixture prior
  N=19;       # number of points for discretization
```

```
var
  ni[I],      # number of observations per patient
  Y[I,T],     # White Cells
  time[I,T],  # instante of time
  qnorm[N],   # 10%,30%,50%,70%,90% normal quantiles.
  p[N],       # Discrete distribuiton on qnorm
  mu[I,T],    # Mean of the Yij's
  alpha[K],   # Prior of the Dirichlet
  Z[I],       # latent variable indicating the true normal
  P[K],       # Distribuiton of Z[K]
  ibeta0[I],ibeta1[I],beta0[I],beta1[I],
  phi1[K],phi2[K],phi3[K],iphi1[K],iphi2[K],iphi3[K],
  tau1,tau2,mu1[K],mu2[K],mu3[K],inv.sig2;

data ni in "ni", Y in "Y", time in "day", qnorm in "qnorm";
{
 for (i in 1:N){
    p[i] <- 1/N;
 }
 alpha[1] <- 1;alpha[2] <- 1;alpha[3] <- 1;

 # Model specification
 for (i in 1:I){
   for (j in 1:ni[i]){
      logit(mu[i,j]) <- beta0[i]+beta1[i]*time[i,j];
      Y[i,j] ~ dnorm(mu[i,j],inv.sig2);
   }
   # Latent variable
   Z[i] ~ dcat(P[]);   # Prior specification : 1st level
   ibeta0[i] ~ dcat(p[]);
   ibeta1[i] ~ dcat(p[]);
   beta0[i] <- qnorm[ibeta0[i]]*sqrt(tau1)+phi1[Z[i]];
   beta1[i] <- qnorm[ibeta1[i]]*sqrt(tau2)+phi2[Z[i]]+
               phi3[Z[i]]*(beta0[i]-phi1[Z[i]]);
 }
 # Prior for the latent variables
 P[] ~ ddirch(alpha[]);
 # Prior Specification : 2nd level
 for (k in 1:K){
   iphi1[k] ~ dcat(p[]);
   phi1[k] <- qnorm[iphi1[k]]+mu1[k];
   iphi2[k] ~ dcat(p[]);
   phi2[k] <- qnorm[iphi2[k]]+mu2[k];
   iphi3[k] ~ dcat(p[]);
   phi3[k] <- qnorm[iphi3[k]]+mu3[k];
 }

 # Constants
 inv.sig2 <- 1/7.490159;
 tau1 <- 0.1506923;tau2 <- 0.0317020;
 mu1[1]  <- -3;mu2[1]  <- -3;mu3[1]  <- -3;
 mu1[2]  <-  0;mu2[2]  <-  0;mu3[2]  <-  0;
 mu1[3]  <-  3;mu2[3]  <-  3;mu3[3]  <-  3;
}
```

S+ Code

```
##################################################
# STA 214 - Probability and Statistical Models #
# HOMEWORK 7                                   #
# DUE : April 22nd, 1997.                      #
# Phd Student : Hedibert Freitas Lopes         #
##################################################
#pscript(onefile=F,print.it=F)
motif()
par(mfrow=c(8,5))
par(cex=0.5)

####################################################
# Function to calculate Quantiles 2.5% and 97.5%  #
####################################################
quant0025_function(x){quantile(x,0.025)}
quant0975_function(x){quantile(x,0.975)}
quant0100_function(x){quantile(x,0.100)}
quant0900_function(x){quantile(x,0.900)}

#####################################################
# Function for reading dataset  (Peter's Function) #
#####################################################
read.dta <- function()
{
  dta <<- scan(skip=2,
       file="214.dta",what=list(patid=0,y=0,ctx=0,gm=0,day=0))
  x <<- dta[[5]]
  ctx <<- dta[[3]]
  gm <<- dta[[4]]
  y <<- dta[[2]]
  patient <<- dta[[1]]
}
##############################################################
# Loop over all patients, plot data for i-th patient
# and compute m.l.e. for logistic fit
# return matrix with logistic regr par:
#     (Nx2) matrix b
# and covariate matrices:  (NxT) X, (Nx2): X2=[ctx,gm]
#     matrix of responses: (NxT) Y
# where T=12 is the max number responses per patient
# missing values are filled with 0
##############################################################
plot.all <- function()
{
  b <<- NULL # will record m.l.e. for each patient in i-th row
  ni <<-NULL # will record number obs per patient
  day <<- matrix(0,nrow=52,ncol=12)
  X <<- matrix(0,nrow=52,ncol=2)
  Y <<- matrix(0,nrow=52,ncol=12)
  i _ 0
  repeat{
    i _ i + 1
    idx _ patient==i
    n <- sum(idx)
    ni <<- c(ni,n)
    day[i,1:n] <<- x[idx]
    X[i,1] <<- ctx[idx][1]
    X[i,2] <<- gm[idx][1]
    Y[i,1:n] <<- y[idx]
    if (sum(idx)<7)
      b <<- rbind(b,b[nrow(b),])
    else{
      plot(x[idx],y[idx],xlab="DAY",ylab="WBC")
      title(paste("PATIENT",i))
      g _ glim(x[idx],y[idx],error="gaussian", iter.max=100,
         link="logit",intercept=T, init = c(-0.7,0.55))
      ehat <- x[idx]*g$coef[2]+g$coef[1]
      yhat <- 1/(1+exp(-ehat))
      lines(x[idx],yhat)
      b <<- rbind(b,g$coef)
      cat("i=",i,"\t b=",format(b[i,]),"\t n=",sum(idx),"\n")
      if (i==52) break
    }
  }
  mb _ apply(b[-c(1,15,46,47),],2,mean)
  b[1,] <<- mb
  b[15,] <<- mb
  b[46,] <<- mb
  b[47,] <<- mb
  NULL # values are returned in global variables b,ni,day,X,Y
}

init.bugs <- function()
{ # prepares data files for bugs
  options(digits=3)
  write(format(t(X)),file="X",ncolumns=ncol(X))
  write(format(t(Y)),file="Y",ncolumns=ncol(Y))
  write(format(t(day)),file="day",ncolumns=ncol(day))
  write(format(ni),file="ni",ncolumns=1)
  write(format(t(b)),file="b",ncolumns=ncol(b))
}

####################################################
# MODEL A : LOGISTIC REGRESSION PER PACIENT       #
# Classical GLM with logit-link and normal errors #
####################################################
read.dta()
plot.all()
init.bugs()

## Plot all intercepts and slopes
#par(mfrow=c(2,1))
#tsplot(b[,1],type="h",xlab="patient")
#title("Intercepts")
#tsplot(b[,2],type="h",xlab="patient")
#title("Slopes")
#
## Plotting the pairs (b0i,b1i) for each pacient
# from i=1,2,...,I=52 (Maximum Likelihood Estimates)
par(mfrow=c(1,1))
plot(b[1:I],b[(I+1):k],xlab="Intercepts",ylab="Slopes")
identify(b[1:I],b[(I+1):k])
title("Intercepts against Slopes for each patient")

####################################################
# MODEL B : NORMAL LINEAR HIERARCHICAL MODEL      #
# Bayesian Analysis (Lindley and Smith (1972))    #
# response = logit(yij)                           #
####################################################
n_sum(ni)
I_52
k_2*I
z_y
z[(y!=0)&(y!=1)]_log(y[(y!=0)&(y!=1)]/(1-y[(y!=0)&(y!=1)]))
z[y==0]_-7
z[y==1]_7

#Hyperparameters
sigma2_var(z)
c11_var(b[,1])
c22_var(b[,2])
mu0_0
```

```
mu1_0
rho0_10000
rho1_10000

#  Calculating the marginal posterior for (b0i,b1i i=1,2,...,I=52)
# Matrix t(X)*X
XtX_matrix(0,k,k)
X1_cbind(diag(ni),diag(apply(day,1,sum)))
X2_cbind(diag(apply(day,1,sum)),diag(apply(day^2,1,sum)))
XtX_rbind(X1,X2)
rm(X1,X2)
Xz_matrix(0,k,1)
day[,1]_99
xxx_t(day)
xxx_xxx[xxx!=0]
xxx[xxx==99]_0
day[,1]_0

# Matrix t(X)*z
for (i in (1:I)){Xz[i]_sum(z[patient==i])}
for (i in (1:I)){Xz[I+i]_sum(z[patient==i]*xxx[patient==i])}
rm(xxx)

#Vbeta+A2*Vphi*t(A2)
xxx_matrix(rho0,I,I)+diag(c11,ncol=I,nrow=I)
yyy_matrix(rho1,I,I)+diag(c22,ncol=I,nrow=I)
zzz_matrix(0,I,I)
V_rbind(cbind(xxx,zzz),cbind(zzz,yyy))
S_solve((XtX/sigma2)+solve(V))
rm(xxx,yyy,zzz,V)
m_S%*%Xz/sigma2
sii_sqrt(diag(S))

# 2a. Plotting the pairs (E(b0i|data),E(b1i|data)) for each patient
# from i=1,2,...,I=52
par(mfrow=c(1,1))
plot(m[1:I],m[(I+1):k],xlab="Intercepts",ylab="Slopes")
identify(m[1:I],m[(I+1):k])
title("Intercepts against Slopes for each patient")

# Plotting E(b0i|Data) for each pacient and 2 standard deviations
# Since de posterior is normal, this corresponds to 95% of the
# distribuiton, approximately.
par(mfrow=c(2,1))
plot(1:I,m[1:I],xlab="Patient",ylab="E(b0i|Data)",main="E(b0i|Data)",pch=18,
     ylim=c(min(m[1:I]-2*sii[1:I]),max(m[1:I]+2*sii[1:I])))
for (i in 1:I){segments(i,m[i]-2*sii[i],i,m[i]+2*sii[i])}
plot(1:I,m[(I+1):k],xlab="Patient",ylab="E(b1i|Data)",
     main="E(b1i|Data)",pch=18,
     ylim=c(min(m[(I+1):k]-2*sii[(I+1):k]),max(m[(I+1):k]+2*sii[(I+1):k])))
for (i in (I+1):k){segments(i-I,m[i]-2*sii[i],i-I,m[i]+2*sii[i])}

#################################################################
# Simulations to obtain the posterior distribuiton of yij
# for i=53 (a new patient) and tij=0,1,2,3,....,12.
#################################################################
M_1000     #number of draws
#--------------------------------------
# STEP [1]
# Genereating (b0i,b1i) for i=1,2,...,53
#--------------------------------------
# Symmetrizing S because S+ complains about it when calculating the
# Cholesky decomposition (don't even think to ask me why) !
for (i in 1:(2*I)){for (j in 1:i){S[j,i]_S[i,j]}}
cholS_t(chol(S))
znorm_matrix(c(rnorm(M*2*I)),2*I,M)
betadraw_matrix(m,2*I,M)+cholS%*%znorm
```

```
#--------------------------------------
# STEP [2]
# Generating (phi1,phi2)
#--------------------------------------
u1_apply(betadraw[1:I,],2,sum)
u2_apply(betadraw[(I+1):k,],2,sum)
u1_u1*rho0/(rho0*I+c11)
u2_u2*rho1/(rho1*I+c22)
T1_diag(c(c11*rho0/(rho0*I+c11),c22*rho1/(rho1*I+c22)))
phi1_u1+rnorm(M)*sqrt(T1[1,1])
phi2_u2+rnorm(M)*sqrt(T1[2,2])
#----------------------------------------------
# STEP [3,4]
# Genereating (b0i,b1i) for i=53 given phi1,phi2
#----------------------------------------------
b053_phi1+rnorm(M)*sqrt(c11)
b153_phi2+rnorm(M)*sqrt(c22)
#----------------------------------------------
# STEP [5]
# Genereating zij given (b0i,b1i) for i=53
# and give t=(0,12)
#----------------------------------------------
tim_seq(0,12,0.5)
lt_length(tim)
z53_matrix(0,M,length(tim))
for (i in 1:lt){z53[,i]_b053+b153*tim[i]}
#----------------------------------------------
# STEP [6]
# Calculating yij=exp(zij)/(1+exp(zij) i=53
# and give t=(0,12)
#----------------------------------------------
z53_exp(z53)/(1+exp(z53))
meanz53_apply(z53,2,mean)
q0025z53_apply(z53,2,quant0025)
q0975z53_apply(z53,2,quant0975)
par(mfrow=c(1,1))
plot(tim,meanz53,xlab="time",ylab="prop. of White blood",
     main="prop. of White blood for a new patient",
     ylim=c(0,1),type="l")
lines(tim,q0025z53,lty=2)
lines(tim,q0975z53,lty=2)

##################################################
# MODEL C: Yij=exp(nuij)/(1+exp(nuij)+eij        #
# Data structured hierarchical prior             #
##################################################
BUGS_matrix(scan('run2.out'),ncol=2,byrow=T)
k_2*I
M_1000
draws_matrix(BUGS[,2],M,k+3)

# Checking convergence for the intercept for the first 16 patients
par(mfrow=c(4,4))
#for (i in c(1:16)){
#  tsplot(draws[,i],xlab=paste("Patient",i))
#}
# Checking convergence for the slopes for the first 16 patients
par(mfrow=c(4,4))
for (i in c(1:16)){
   tsplot(draws[,(I+i)],xlab=paste("Patient",i))
}
# Checking convergence for the phi's
par(mfrow=c(3,1))
for (i in c(1:3)){
   tsplot(draws[,(k+i)])
   title(paste("Phi",i))
}
```

```
# Calculating Summaries

means_apply(draws,2,mean)
q0025int_apply(draws,2,quant0025)
q0975int_apply(draws,2,quant0975)

# 2a. Plotting the pairs (E(b0i|data),E(b1i|data)) for each pacient
# from i=1,2,...,I=52
par(mfrow=c(1,1))
plot(means[1:I],means[(I+1):k],xlab="Intercepts",ylab="Slopes")
identify(means[1:I],means[(I+1):k])
title("Intercepts against Slopes for each patient")

# Plotting E(b0i|Data) for each pacient and 95% HPD intervals.
par(mfrow=c(2,1))
plot(1:I,means[1:I],xlab="Patient",
ylab="E(b0i|Data)",main="E(b0i|Data)",pch=18,
     ylim=c(min(q0025int[1:I]),max(q0975int[1:I])))
for (i in 1:I){segments(i,q0025int[i],i,q0975int[i])}
plot(1:I,means[(I+1):k],xlab="Patient",ylab="E(b1i|Data)",
     main="E(b1i|Data)",pch=18,
     ylim=c(min(q0025int[(I+1):k]),max(q0975int[(I+1):k])))
for (i in (I+1):k){segments(i-I,q0025int[i],i-I,q0975int[i])}
abline(h=0)

#############################################################
# Simulations to obtain the posterior distribuiton of yij    #
# for i=53 (a new patient) and tij=0,1,2,3,....,12.          #
#############################################################
M_1000     #number of draws
#-----------------------------------------------
# STEP [3,4]
# Genereating (b0i,b1i) for i=53 given phi1,phi2
#-----------------------------------------------
draws1_rnorm(M)*sqrt(c11)
b053_draws[,(k+1)]+draws1
b153_draws[,(k+2)]+draws[,(k+3)]*draws1 + rnorm(M)*sqrt(c22)
rm(draws1)
#-----------------------------------------------
# STEP [5]
# Genereating zij given (b0i,b1i) for i=53
# and give t=(0,12)
#-----------------------------------------------
y53_matrix(0,M,length(tim))
for (i in 1:lt){y53[,i]_exp(b053+b153*tim[i])/(1+exp(b053+b153*tim[i]))}
#-----------------------------------------------
# STEP [6]
# Calculating yij=exp(zij)/(1+exp(zij) i=53
# and give t=(0,12)
#-----------------------------------------------
meany53_apply(y53,2,mean)
q0025y53_apply(y53,2,quant0025)
q0975y53_apply(y53,2,quant0975)
par(mfrow=c(1,1))
plot(tim,meany53,xlab="time",ylab="prop. of White blood",
     main="prop. of White blood for a new patient",
     ylim=c(0,1),type="l")
lines(tim,q0025y53,lty=2)
lines(tim,q0975y53,lty=2)


##################################################
# MODEL D: Yij=exp(nuij)/(1+exp(nuij)+eij        #
# and structured hierarchical prior              #
# and eij ~ mixture of three normals             #
##################################################
```

```
BUGS_matrix(scan('run4.out'),ncol=2,byrow=T)
k_2*I
M_1000
drawsd_matrix(BUGS[,2],M,k+12)

# Checking convergence for the intercept for the first 16 patients
par(mfrow=c(4,4))
for (i in c(1:16)){
   tsplot(drawsd[,i],xlab=paste("Patient",i))
}
# Checking convergence for the slopes for the first 16 patients
par(mfrow=c(4,4))
for (i in c(1:16)){
   tsplot(drawsd[,(I+i)],xlab=paste("Patient",i))
}
# Checking convergence for the phi's
par(mfrow=c(3,3))
for (i in c(1:9)){
   tsplot(drawsd[,(k+i)])
   title(paste("Phi",i))
 }
# Checking convergence for the Weights
par(mfrow=c(3,1))
for (i in c(1:3)){
   tsplot(drawsd[,(k+9+i)])
   title(paste("Weight",i))
}
# Calculating Summaries
 #
dmeans_apply(drawsd,2,mean)
dq0025int_apply(drawsd,2,quant0025)
dq0975int_apply(drawsd,2,quant0975)

# 2a. Plotting the pairs (E(b0i|data),E(b1i|data)) for each pacient
# from i=1,2,...,I=52
par(mfrow=c(1,1))
plot(dmeans[1:I],dmeans[(I+1):k],xlab="Intercepts",ylab="Slopes")
identify(dmeans[1:I],dmeans[(I+1):k])
title("Intercepts against Slopes for each patient")

# Plotting E(b0i|Data) for each pacient and 95% HPD intervals.
par(mfrow=c(2,1))
plot(1:I,dmeans[1:I],xlab="Patient",
ylab="E(b0i|Data)",main="E(b0i|Data)",pch=18,
     ylim=c(min(q0025int[1:I]),max(q0975int[1:I])))
for (i in 1:I){segments(i,q0025int[i],i,q0975int[i])}
plot(1:I,dmeans[(I+1):k],xlab="Patient",ylab="E(b1i|Data)",
     main="E(b1i|Data)",pch=18,
     ylim=c(min(dq0025int[(I+1):k]),max(dq0975int[(I+1):k])))
for (i in (I+1):k){segments(i-I,dq0025int[i],i-I,dq0975int[i])}
abline(h=0)

#############################################################
# Simulations to obtain the posterior distribuiton of yij    #
# for i=53 (a new patient) and tij=0,1,2,3,....,12.          #
#############################################################
M_1000     #number of draws
#-----------------------------------------------
# STEP [3,4]
# Genereating (b0i,b1i) for i=53 given phi1,phi2
#-----------------------------------------------
latent_matrix(0,M,1)
indicator_c(1,2,3)
for (i in 1:M){
    latent[i]_sample(indicator,1,drawsd[i,114:116])
  }
```

```
beta0_beta1_matrix(0,M,1)
for (i in 1:M){
    normdraw_rnorm(1)
    beta0[i]_normdraw*sqrt(c11)+drawsd[,(104+latent[i])]
    beta1[i]_rnorm(1)*sqrt(c22)+drawsd[,(107+latent[i])]+
             drawsd[,(110+latent[i])]*normdraw*sqrt(c11)
  }

# Histogram of beta0 and beta1 for the new patient.
par(mfrow=c(2,1))
hist(beta0,col=0,main="Intercept for a new patient (Model D)",
     nclass=30,probability=TRUE)
hist(beta1,col=0,main="Slope for a new patient (Model D)",
     nclass=30,probability=TRUE)

#------------------------------------------------
# STEP [5]
# Genereating zij given (b0i,b1i) for i=53
# and give t=(0,12)
#------------------------------------------------
y53_matrix(0,M,length(tim))
for (i in 1:lt){y53[,i]_exp(beta0+beta1*tim[i])/(1+exp(beta0+beta1*tim[i]))}
#------------------------------------------------
# STEP [6]
# Calculating yij=exp(zij)/(1+exp(zij) i=53
# and give t=(0,12)
#------------------------------------------------
meany53_apply(y53,2,mean)
q0025y53_apply(y53,2,quant0025)
q0975y53_apply(y53,2,quant0975)
par(mfrow=c(1,1))
plot(tim,meany53,xlab="time",ylab="prop. of White blood",
     main="prop. of White blood for a new patient",
     ylim=c(0,1),type="l")
lines(tim,q0025y53,lty=2)
lines(tim,q0975y53,lty=2)
```