A Bayesian zero-inflated beta-binomial model for longitudinal data with group-specific changepoints

2024-08-13

Package Installed

```
library(VGAM)       # for rbetabinom
library(mvtnorm)    # for rmvt proposal
library(truncnorm)  # for rtruncnorm()
library(aod)
library(msm)
library(tmvtnorm)   # for trmvt()
library(ggplot2)
library(tidyr)
library(dplyr)
library(ggplot2)
library(BayesLogit) # for rpg function
library(MCMCpack)   # for Iwish update of Sigmab
library(gamlss)       # for ZIBB functions
```

Data Generating

```
set.seed(2202023)
ntrial<-7                        # number of quest. in one ACEs score (fixed)
n<-500                           # total subjects
nis<-sample(1:12,n,T)            # repeated measure per subject (unbalanced)
id<-rep(1:n,nis)                 # id
N<-length(id)                    # total observations

t<-rep(0,N)
for (i in 1:n) t[id==i]<-sort(sample(1:12,nis[i]))  # time variable
t<-t-5                                              # centered time
trt<-rbinom(n,1,.5)
tx<-rep(trt,nis)                                    # trt indicaotr


truekappa1<-kappa1<-3                 # true changepoint (control)
xg1<-(t-kappa1)*(t>kappa1)*(tx==0)    # 1(t>cp1)*1(trt=control)
truekappa2<-kappa2<-2                 # true changepoint (tx)
xg2<-(t-kappa2)*(t>kappa2)*(tx==1)    # 1(t>cp2)*1(trt=treatment)
xg<-xg1+xg2                           # random slope after corr. cps

Xs<-cbind(1,t,tx,t*tx)                # Sub design matrix for "Binary"/"BB component"
X1<-cbind(Xs,xg1,xg2)                 # design matrix for *binary component*
truealpha<-alpha<-c(0.59,0.86,0.45,-0.35,-1.15,-0.54) # true alpha (binary)
```

```r
X2<-cbind(Xs,xg1,xg2)                            # design matrix for *BB component*
truebeta<-beta<-c(-0.94,0.59,0.89,0.43,.68,-.66) # true beta (count)

p1<-ncol(X1)                          # number of parameters (binary)
p2<-ncol(X2)                          # number of parameters (count)

truesigmab<-sigmab<-matrix(c(0.5,0.05,0.05,0.05,0.05,0.05,
                             0.05,0.35,0.05,0.05,0.05,0.05,
                             0.05,0.05,0.35,0.05,0.05,0.05,
                             0.05,0.05,0.05,0.5,0.15,0.05,
                             0.05,0.05,0.05,0.15,0.5,0.05,
                             0.05,0.05,0.05,0.05,0.05,0.35),6,6,byrow = T) # 6 by 6 random effect varia

trueB<-B<-rmvnorm(n,rep(0,6),sigmab)          # true random effects

trueb11<-b11<-B[,1]
trueb12<-b12<-B[,2]
trueb13<-b13<-B[,3]
trueb21<-b21<-B[,4]
trueb22<-b22<-B[,5]
trueb23<-b23<-B[,6]

B11<-rep(b11,nis)
B12<-rep(b12,nis)
B13<-rep(b13,nis)
B21<-rep(b21,nis)
B22<-rep(b22,nis)
B23<-rep(b23,nis)

### Binary component ###
eta1<-X1%*%alpha+B11+B12*t+B13*xg
mu1<-exp(eta1)/(1+exp(eta1))
u<-rbinom(N,1,mu1)                 # Abs. latent variable
N1<-sum(u)                         # true num of ppl at risk

### Count component ###
eta2<-X2%*%beta+B21+B22*t+B23*xg
mu2<-exp(eta2)/(1+exp(eta2))
truerho<-rho<-0.25                # Correlation parameter
y<-rep(0,N)
y[u==1]<-rbetabinom(N1,size=ntrial,prob=mu2[u==1],rho=rho) # outcome
```

```r
cat("Proportion of obs at risk=",N1/N)
```

```
## Proportion of obs at risk= 0.6153846
```

```r
cat("Proportion of structural zeros=",1-mean(u))
```

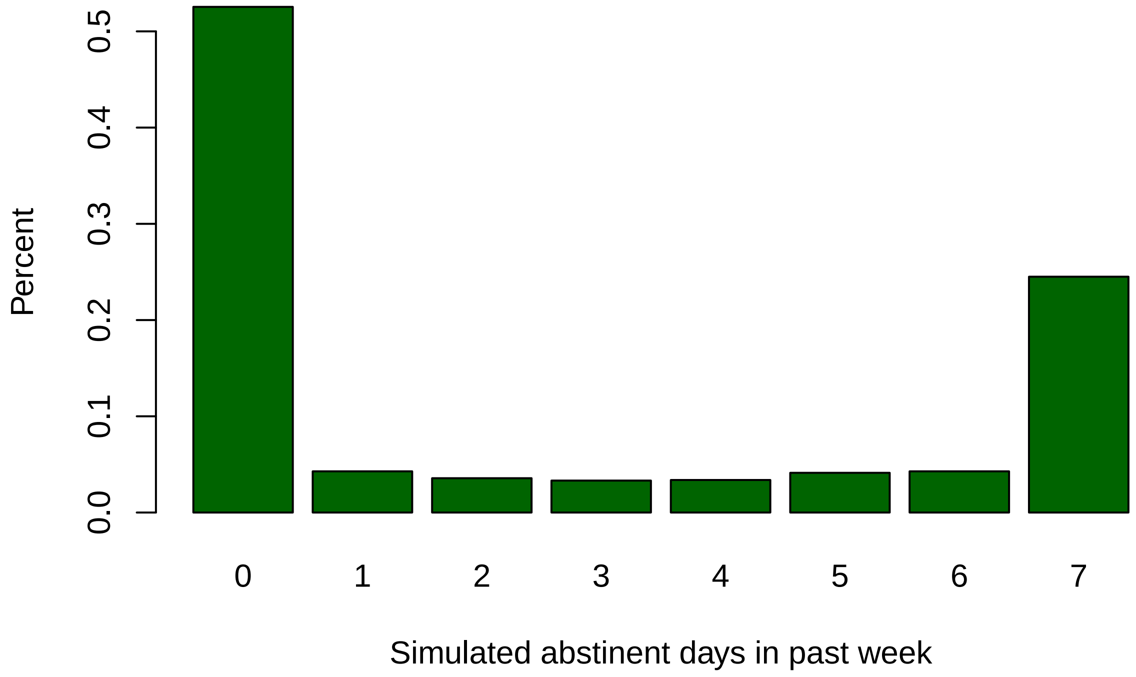
```
## Proportion of structural zeros= 0.3846154
```

```r
cat("Proportion of zeros=",length(y[y==0])/N)
```

```
## Proportion of zeros= 0.5254342
```

Histogram of the simulated data

```
tmp<-table(y)
barplot(tmp/N,col="darkgreen",xlab="Simulated abstinent days in past week",ylab="Percent")
```



MCMC Set-Up

```
#-------------------------------------------------------------------------------#
# Bayesian ZIBB model #
# MCMC-MH Setting     #
#---------------------#

# Priors
alpha0<-rep(0,p1)         # prior mean for alpha
beta0<-rep(0,p2)          # prior mena for beta
T0a<-diag(0.01,p1)        # prior precision for alpha
T0b<-diag(0.01,p2)        # prior precision for beta
d0<-7                     # prior inverse Wishart: IW(7,I_6)
C0<-diag(6)

kappa0<-0                 # prior mean for kappa
sigmak0<-10^4             # prior var for kaapa

nk<-1
L0<--3                    # CP from week -3 (2) to 6 (11)
U0<-6

# proposal
sigmar0<-0.01            # proposal var for rho
sigmakap10<-0.1          # proposal var for kappa1
sigmakap20<-0.1          # proposal var for kappa2
sigmab210<-0.1           # proposal var for b21
sigmab220<-0.1           # proposal var for b22
sigmab230<-0.1           # proposal var for b23

# Inists
alpha<-rep(0,p1)
beta<-rep(0,p2)
covb<-diag(0.01,p2)        # Proposal covariance
rho<-0.5                   # over-dispersed
A1<-0                      # Acceptance counter (beta)
A2<-0                      # Acceptance counter (rho)
Akap1<-0                   # Acceptance counter (kappa) -> control
Akap2<-0                   # Acceptance counter (kappa) -> tx
y1<-rbinom(N,1,.5)         # At risk indicator
y1[y>0]<-1                 # If y>0, then patient is at risk w.p. 1
n0<-length(y[y==0])        # Number of observed 0's

bmat<-rmvnorm(n,rep(0,6),diag(6))
b11<-bmat[,1]
b12<-bmat[,2]
b13<-bmat[,3]
b21<-bmat[,4]
b22<-bmat[,5]
b23<-bmat[,6]

B11<-rep(b11,nis)
B12<-rep(b12,nis)
B13<-rep(b13,nis)
```

```
B21<-rep(b21,nis)
B22<-rep(b22,nis)
B23<-rep(b23,nis)

Bmat<-cbind(b11,b12,b13,b21,b22,b23)
sigmab<-cov(Bmat)

#----------------#
# Initial matrix #
#----------------#
kappa1<-0
kappa2<-0                           # Init changepoints
xg1<-(t-kappa1)*(t>kappa1)*(tx==0)  # Initialize design matrix
xg2<-(t-kappa2)*(t>kappa2)*(tx==1)  # Initialize design matrix
xg<-xg1+xg2
X1<-X2<-cbind(Xs,xg1,xg2)


#################
# Store Samples #
#################
nsim<-30000                   # Number of MCMC Iterations
thin<-5                             # Thinnisng interval
burn<-10000                   # Burnisn
lastit<-(nsim-burn)/thin        # Last stored value
Betatmp<-matrix(0,nsim,p2)
Alpha<-matrix(0,lastit,p1)
Beta<-matrix(0,lastit,p2)
Rho<-rep(0,lastit)
Kappa1<-rep(0,lastit)
Kappa2<-rep(0,lastit)
Sigmab<-matrix(0,lastit,36)
B11s<-matrix(0,lastit,n)
B12s<-matrix(0,lastit,n)
B13s<-matrix(0,lastit,n)
B21s<-matrix(0,lastit,n)
B22s<-matrix(0,lastit,n)
B23s<-matrix(0,lastit,n)
L<-matrix(0,lastit,N)        # Likelihood function
```

MCMC Algorithm (not run)

```
#-----------#
# MCMC + MH #
#-----------#
set.seed(1234)
time.start<-proc.time()
for (i in 1:nsim) {
  ### BINARY: update alpha ####
  mu<-X1%*%alpha+B11+B12*t+B13*xg
  omega<-rpg(N,1,mu)
  z<-(y1-1/2)/omega
  v<-solve(T0a+crossprod(X1*sqrt(omega)))
```

```
m<-v%*%(T0a%*%alpha0+t(sqrt(omega)*X1)%*%c(sqrt(omega)*(z-B11-B12*t-B13*xg)))
alpha<-c(rmvnorm(1,m,v))

### RANDOM: binary (b1i1) ###
priorprec<-c(1/(sigmab[1,1]-sigmab[1,-1]%*%solve(sigmab[-1,-1])%*%sigmab[-1,1]))
priormean<-Bmat[,-1]%*%t(sigmab[1,-1]%*%solve(sigmab[-1,-1]))
v<-1/(priorprec+c(tapply(omega,id,sum)))
m<-v*(priorprec*priormean+c(tapply(omega*(z-X1%*%alpha-B12*t-B13*xg),id,sum)))
b11<-rnorm(n,m,sqrt(v))
B11<-rep(b11,nis)

### RANDOM: binary (b1i2) ###
priorprec<-c(1/(sigmab[2,2]-sigmab[2,-2]%*%solve(sigmab[-2,-2])%*%sigmab[-2,2]))
priormean<-Bmat[,-2]%*%t(sigmab[2,-2]%*%solve(sigmab[-2,-2]))
v<-1/(priorprec+c(tapply(omega*t^2,id,sum)))
m<-v*(priorprec*priormean+c(tapply(t*omega*(z-X1%*%alpha-B11-B13*xg),id,sum)))
b12<-rnorm(n,m,sqrt(v))
B12<-rep(b12,nis)

### RANDOM: binary (b1i3) ###
priorprec<-c(1/(sigmab[3,3]-sigmab[3,-3]%*%solve(sigmab[-3,-3])%*%sigmab[-3,3]))
priormean<-Bmat[,-3]%*%t(sigmab[3,-3]%*%solve(sigmab[-3,-3]))
v<-1/(priorprec+c(tapply(omega*xg^2,id,sum)))
m<-v*(priorprec*priormean+c(tapply(xg*omega*(z-X1%*%alpha-B11-B12*t),id,sum)))
b13<-rnorm(n,m,sqrt(v))
B13<-rep(b13,nis)

### update at-risk ind ###
eta1<-X1%*%alpha+B11+B12*t+B13*xg
eta2<-X2%*%beta+B21+B22*t+B23*xg
pi<-pmax(0.001,pmin(0.999,1/(1+exp(-eta1))))    # at-risk probability
pr<-1/(1+exp(-eta2))
q<-dbetabinom(0,size=ntrial,prob=pr,rho=rho)
theta<-pi*q/(pi*q+1-pi)   # Conditional prob that y1=1 given y=0 -- i.e. Pr(chance zero|observed zero
y1[y==0]<-rbinom(n0,1,theta[y==0])  # If y=0, then draw a "chance zero" w.p. theta, if y=1, then y1=1
nis1<-tapply(y1,id,sum)


### COUNT: update beta ###
# Current likelihood
eta<-X2%*%beta+B21+B22*t+B23*xg
mu<-1/(1+exp(-eta))
lold<-sum(dbetabinom(y[y1==1],size=ntrial,prob=mu[y1==1],rho=rho,log=T))

# Draw candidate beta and compute likelihood
betanew<-beta +rmvnorm(1,sigma=.15*covb)   # Draw from "symmetric" MV t_3 dist
eta<-X2%*%c(betanew)+B21+B22*t+B23*xg
mu<-1/(1+exp(-eta))
lnew<-sum(dbetabinom(y[y1==1],size=ntrial,prob=mu[y1==1],rho=rho,log=T))

# Acceptance prob on log scale =log(lnew x prior) - log (lold x prior)
r1<-lnew+dmvnorm(betanew,beta0,solve(T0b),log=T)-(lold+dmvnorm(beta,beta0,solve(T0b),log=T))
if(log(runif(1))<r1) {
  beta<-c(betanew)
  if (i> burn & i%%thin==0) A1<-A1+1
}

Betatmp[i,]<-beta
if (i==nsim/2) covb<-cov(Betatmp[(nsim/4+1):nsim/2,])  # Update proposal cov

### RANDOM: binary (b2i1) ###
priorprec<-c(1/(sigmab[4,4]-sigmab[4,-4]%*%solve(sigmab[-4,-4])%*%sigmab[-4,4]))
priormean<-Bmat[,-4]%*%t(sigmab[4,-4]%*%solve(sigmab[-4,-4]))

b21new<-rnorm(n,b21,sqrt(sigmab210))

eta<-X2%*%beta+rep(b21new,nis)+B22*t+B23*xg
mu<-1/(1+exp(-eta))

lnew<-rep(1,n)     # log-likelihood if empty block
lnew[nis1>0]<-tapply(dbetabinom(y[y1==1],size=ntrial,prob=mu[y1==1],rho=rho,log=T),id[y1==1],sum)

eta<-X2%*%beta+B21+B22*t+B23*xg
mu<-1/(1+exp(-eta))

lold<-rep(1,n)    # log-likelihood if empty block
lold[nis1>0]<-tapply(dbetabinom(y[y1==1],size=ntrial,prob=mu[y1==1],rho=rho,log=T),id[y1==1],sum)

ratio<-lnew+dnorm(b21new,priormean,sqrt(1/priorprec),log=T)-(lold+dnorm(b21,priormean,sqrt(1/priorpre

utmp<-1*(log(runif(n))<ratio)
b21[utmp==1]<-b21new[utmp==1]
B21<-rep(b21,nis)

### RANDOM: binary (b2i2) ###
priorprec<-c(1/(sigmab[5,5]-sigmab[5,-5]%*%solve(sigmab[-5,-5])%*%sigmab[-5,5]))
priormean<-Bmat[,-5]%*%t(sigmab[5,-5]%*%solve(sigmab[-5,-5]))

b22new<-rnorm(n,b22,sqrt(sigmab220))

eta<-X2%*%beta+B21+rep(b22new,nis)*t+B23*xg
mu<-1/(1+exp(-eta))

lnew<-rep(1,n)     # log-likelihood if empty block
lnew[nis1>0]<-tapply(dbetabinom(y[y1==1],size=ntrial,prob=mu[y1==1],rho=rho,log=T),id[y1==1],sum)

eta<-X2%*%beta+B21+B22*t+B23*xg
mu<-1/(1+exp(-eta))

lold<-rep(1,n)    # log-likelihood if empty block
lold[nis1>0]<-tapply(dbetabinom(y[y1==1],size=ntrial,prob=mu[y1==1],rho=rho,log=T),id[y1==1],sum)

ratio<-lnew+dnorm(b22new,priormean,sqrt(1/priorprec),log=T)-(lold+dnorm(b22,priormean,sqrt(1/priorpre

utmp<-1*(log(runif(n))<ratio)
b22[utmp==1]<-b22new[utmp==1]
```

```
B22<-rep(b22,nis)

### RANDOM: binary (b2i3) ###
priorprec<-c(1/(sigmab[6,6]-sigmab[6,-6]%*%solve(sigmab[-6,-6])%*%sigmab[-6,6]))
priormean<-Bmat[,-6]%*%t(sigmab[6,-6]%*%solve(sigmab[-6,-6]))

b23new<-rnorm(n,b23,sqrt(sigmab230))

eta<-X2%*%beta+B21+B22*t+rep(b23new,nis)*xg
mu<-1/(1+exp(-eta))

lnew<-rep(1,n)     # log-likelihood if empty block
lnew[nis1>0]<-tapply(dbetabinom(y[y1==1],size=ntrial,prob=mu[y1==1],rho=rho,log=T),id[y1==1],sum)

eta<-X2%*%beta+B21+B22*t+B23*xg
mu<-1/(1+exp(-eta))

lold<-rep(1,n)    # log-likelihood if empty block
lold[nis1>0]<-tapply(dbetabinom(y[y1==1],size=ntrial,prob=mu[y1==1],rho=rho,log=T),id[y1==1],sum)

ratio<-lnew+dnorm(b23new,priormean,sqrt(1/priorprec),log=T)-(lold+dnorm(b23,priormean,sqrt(1/priorpre

utmp<-1*(log(runif(n))<ratio)
b23[utmp==1]<-b23new[utmp==1]
B23<-rep(b23,nis)

### COUNT: update rho ###
# Current likelihood
eta<-X2%*%beta+B21+B22*t+B23*xg
mu<-1/(1+exp(-eta))
lold<-sum(dbetabinom(y[y1==1],size=ntrial,prob=mu[y1==1],rho=rho,log=T))

# Draw candidate rho and compute likelihood from truncated noraml
rhonew<-rtnorm(1,rho,sqrt(sigmar0),0,1)           # Draw from truncated normal
lnew<-sum(dbetabinom(y[y1==1],size=ntrial,prob=mu[y1==1],rho=rhonew,log=T))

# Acceptance prob on log scale =log(lnew x prior) - log (lold x prior)
rrho<-lnew-lold+dtnorm(rho,rhonew,sqrt(sigmar0),0,1,log=T)-dtnorm(rhonew,rho,sqrt(sigmar0),0,1,log=T)
if(log(runif(1))<rrho) {
  rho<-rhonew
  if (i> burn & i%%thin==0) A2<-A2+1
}

# Update sigmab
Bmat<-cbind(b11,b12,b13,b21,b22,b23)
sigmab<-riwish(d0+n,C0+crossprod(Bmat))

# update changepoint (kaapa) --> tx
kappa2new<-rtnorm(1,kappa2,sqrt(sigmakap20),L0,U0) # propose new cp

xg2new<-(t-kappa2new)*(t>kappa2new)*(tx==1)
xgnew<-xg1+xg2new
```

```
X1new<-cbind(Xs,xg1,xg2new)                     # new design matrix (binary)
eta1<-X1new%*%alpha+B11+B12*t+B13*xgnew
mu1<-pmax(0.001,pmin(0.999,1/(1+exp(-eta1))))

X2new<-cbind(Xs,xg1,xg2new)                     # new design matrix (BB)
eta2<-X2new%*%beta+B21+B22*t+B23*xgnew
mu2<-pmax(0.001,pmin(0.999,1/(1+exp(-eta2))))

lnew<-sum(dZIBB(y[tx==1],mu=mu2[tx==1],sigma=rho/(1-rho),nu=1-mu1[tx==1],bd=ntrial,log=T))

eta1<-X1%*%alpha+B11+B12*t+B13*xg
mu1<-pmax(0.001,pmin(0.999,1/(1+exp(-eta1))))

eta2<-X2%*%beta+B21+B22*t+B23*xg                           # old design matrix (BB)
mu2<-pmax(0.001,pmin(0.999,1/(1+exp(-eta2))))

lold<-sum(dZIBB(y[tx==1],mu=mu2[tx==1],sigma=rho/(1-rho),nu=1-mu1[tx==1],bd=ntrial,log=T))

# (lnew-lold)+(priornew-priorold)+(proposalnew-proposalold)
rkap2<-(lnew-lold)+
  (dtnorm(kappa2,kappa2new,sqrt(sigmakap20),L0,U0,log=T)-dtnorm(kappa2new,kappa2,sqrt(sigmakap20),L0,U

if(log(runif(1))<rkap2) {
  kappa2<-kappa2new
  xg2<-xg2new    # update slope after cp for tx
  xg<-xgnew      # update slope after cp
  X1<-X1new      # update design matrix (binary)
  X2<-X2new      # update design matrix (bb)
  if (i> burn & i%%thin==0) Akap2<-Akap2+1
}



# update changepoint (kaapa) --> control
kappa1new<-rtnorm(1,kappa1,sqrt(sigmakap10),L0,U0) # propose new cp

xg1new<-(t-kappa1new)*(t>kappa1new)*(tx==0)
xgnew<-xg1new+xg2

X1new<-cbind(Xs,xg1new,xg2)                     # new design matrix (binary)
eta1<-X1new%*%alpha+B11+B12*t+B13*xgnew
mu1<-pmax(0.001,pmin(0.999,1/(1+exp(-eta1))))

X2new<-cbind(Xs,xg1new,xg2)                     # new design matrix (BB)
eta2<-X2new%*%beta+B21+B22*t+B23*xgnew
mu2<-pmax(0.001,pmin(0.999,1/(1+exp(-eta2))))

lnew<-sum(dZIBB(y[tx==0],mu=mu2[tx==0],sigma=rho/(1-rho),nu=1-mu1[tx==0],bd=ntrial,log=T))

eta1<-X1%*%alpha+B11+B12*t+B13*xg
mu1<-pmax(0.001,pmin(0.999,1/(1+exp(-eta1))))

eta2<-X2%*%beta+B21+B22*t+B23*xg                           # old design matrix (BB)
```

```
mu2<-pmax(0.001,pmin(0.999,1/(1+exp(-eta2))))

lold<-sum(dZIBB(y[tx==0],mu=mu2[tx==0],sigma=rho/(1-rho),nu=1-mu1[tx==0],bd=ntrial,log=T))

# (lnew-lold)+(priornew-priorold)+(proposalnew-proposalold)
rkap1<-(lnew-lold)+
  (dtnorm(kappa1,kappa1new,sqrt(sigmakap10),L0,U0,log=T)-dtnorm(kappa1new,kappa1,sqrt(sigmakap10),L0,U

if(log(runif(1))<rkap1) {
  kappa1<-kappa1new
  xg1<-xg1new     # update slope after cp (control)
  xg<-xgnew       # update slope after cp
  X1<-X1new       # update design matrix (binary)
  X2<-X2new       # update design matrix (bb)
  if (i> burn & i%%thin==0) Akap1<-Akap1+1
}

#################
# Store Results #
#################
if (i> burn & i%%thin==0) {
  j<-(i-burn)/thin
  Alpha[j,]<-alpha
  Beta[j,]<-beta
  Rho[j]<-rho
  Kappa1[j]<-kappa1
  Kappa2[j]<-kappa2
  Sigmab[j,]<-c(sigmab)
  B11s[j,]<-b11
  B12s[j,]<-b12
  B13s[j,]<-b13
  B21s[j,]<-b21
  B22s[j,]<-b22
  B23s[j,]<-b23

  # likelihood function for ZIBB
  eta1<-X1%*%alpha+B11+B12*t+B13*xg
  mu1<- pmax(0.001,pmin(0.999,1/(1+exp(-eta1))))

  eta2<-X2%*%beta+B21+B22*t+B23*xg
  mu2<- pmax(0.001,pmin(0.999,1/(1+exp(-eta2))))

  L[j,]<-dZIBB(y,mu=mu2,sigma=rho/(1-rho),nu=1-mu1,bd=ntrial)
}

if (i%%5==0) {
  print(i)
  print(truealpha)
  print(alpha)
  print(truebeta)
  print(beta)
  print(truerho)
  print(rho)
```

```
    print(truekappa1)
    print(kappa1)
    print(truekappa2)
    print(kappa2)
    print(truesigmab)
    print(sigmab)
  }
}
(time.tol<-proc.time()-time.start)
```

Save/Load MCMC Samples

```
# samples<-list(Alpha=Alpha,
#                Beta=Beta,
#                Rho=Rho,
#                Sigmab=Sigmab,
#                Kappa1=Kappa1,
#                Kappa2=Kappa2,
#                B11s=B11s,
#                B12s=B12s,
#                B13s=B13s,
#                B21s=B21s,
#                B22s=B22s,
#                B23s=B23s)

dir.sav<-"C:\\Users\\chech\\OneDrive - Medical University of South Carolina\\Research\\Bayes ZIBB\\Githu

# save(samples,file=paste(dir.sav,"ZIBBCP-Simulation-N500.Rda",sep=""))
load(file=paste(dir.sav,"ZIBBCP-Simulation-N500.Rda",sep=""))

Alpha<-samples$Alpha
Beta<-samples$Beta
Rho<-samples$Rho
Sigmab<-samples$Sigmab
Kappa1<-samples$Kappa1
Kappa2<-samples$Kappa2
B11s<-samples$B11s
B12s<-samples$B12s
B13s<-samples$B13s
B21s<-samples$B21s
B22s<-samples$B22s
B23s<-samples$B23s
```

Results

```
# Results
malpha<-colMeans(Alpha)
qalpha<-apply(Alpha,2,quantile,c(0.025,0.975))

mbeta<-colMeans(Beta)
qbeta<-apply(Beta,2,quantile,c(0.025,0.975))
```

```r
mrho<-mean(Rho)
qrho<-quantile(Rho,c(0.025,0.975))

msigmab<-colMeans(Sigmab)
qsigmab<-apply(Sigmab,2,quantile,c(0.025,0.975))

mkap1<-mean(Kappa1)
qkap1<-quantile(Kappa1,c(0.025,0.975))

mkap2<-mean(Kappa2)
qkap2<-quantile(Kappa2,c(0.025,0.975))

truealpha
```

```
## [1]  0.59  0.86  0.45 -0.35 -1.15 -0.54
```

```r
print(malpha)
```

```
## [1]  0.3632604  0.9253200  0.6731569 -0.3319699 -1.1373650 -0.6780192
```

```r
print(qalpha)
```

```
##             [,1]      [,2]       [,3]        [,4]       [,5]       [,6]
## 2.5%  -0.1221006 0.6650762 0.03597402 -0.69174539 -1.6069616 -1.0670925
## 97.5%  0.9754308 1.2056293 1.25938617  0.02731948 -0.6503024 -0.2719076
```

```r
truebeta
```

```
## [1] -0.94  0.59  0.89  0.43  0.68 -0.66
```

```r
print(mbeta)
```

```
## [1] -0.8584806  0.5469668  0.8109234  0.4719279  0.5975064 -0.3931986
```

```r
print(qbeta)
```

```
##             [,1]      [,2]     [,3]      [,4]      [,5]        [,6]
## 2.5%  -1.2448380 0.3807418 0.419972 0.2481761 0.1825027 -0.67161412
## 97.5% -0.4772675 0.7256284 1.195628 0.6838851 0.9877138 -0.05173769
```

```r
truerho
```

```
## [1] 0.25
```

```r
print(mrho)
```

```
## [1] 0.278412
```

```r
print(qrho)
```

```
##      2.5%     97.5%
## 0.2312829 0.3223268
```

```r
ind=upper.tri(truesigmab,diag = T)
print(truesigmab[ind])
```

```
##  [1] 0.50 0.05 0.35 0.05 0.05 0.35 0.05 0.05 0.05 0.50 0.05 0.05 0.05 0.15 0.50
## [16] 0.05 0.05 0.05 0.05 0.05 0.35
```

```r
print(msigmab[c(ind)])
```

```
##  [1]  1.191406237  0.015455297  0.573733561  0.013769515 -0.011645816
##  [6]  0.451519022 -0.008580402  0.028673486  0.010977867  0.257743738
## [11] -0.004870581  0.061324053 -0.027774004  0.018652947  0.441966948
## [16]  0.052737247  0.029913022  0.009933817  0.032549082 -0.020688467
## [21]  0.307494560
```

```
print(qsigmab[,c(ind)])
```

```
##           [,1]       [,2]      [,3]       [,4]        [,5]      [,6]       [,7]
## 2.5%  0.455959 -0.1667685 0.3373984 -0.1156159 -0.14550638 0.1249357 -0.2053484
## 97.5% 2.279762  0.1834361 0.8940911  0.1526152  0.09558679 1.2336610  0.1321427
##              [,8]        [,9]      [,10]      [,11]       [,12]      [,13]
## 2.5%  -0.07734927 -0.09202636 0.08246784 -0.2076451 -0.05270105 -0.1785281
## 97.5%  0.16164710  0.12368251 0.54660632  0.2026450  0.16117973  0.1156774
##             [,14]     [,15]      [,16]      [,17]      [,18]       [,19]
## 2.5%  -0.09147108 0.3055626 -0.1245842 -0.1215617 -0.1298095 -0.05404482
## 97.5%  0.11663806 0.6139320  0.2636779  0.2478683  0.1387012  0.20576788
##            [,20]     [,21]
## 2.5%  -0.1871606 0.1020287
## 97.5%  0.1398025 0.8820946
```

```
truekappa1
```

```
## [1] 3
```

```
print(mkap1)
```

```
## [1] 3.075909
```

```
print(qkap1)
```

```
##     2.5%    97.5%
## 2.459150 3.715393
```

```
truekappa2
```

```
## [1] 2
```

```
print(mkap2)
```

```
## [1] 1.83943
```

```
print(qkap2)
```

```
##     2.5%    97.5%
## 1.123590 2.604459
```

```
print(truekappa1-truekappa2)
```

```
## [1] 1
```

```
mean(Kappa1-Kappa2)
```

```
## [1] 1.236479
```

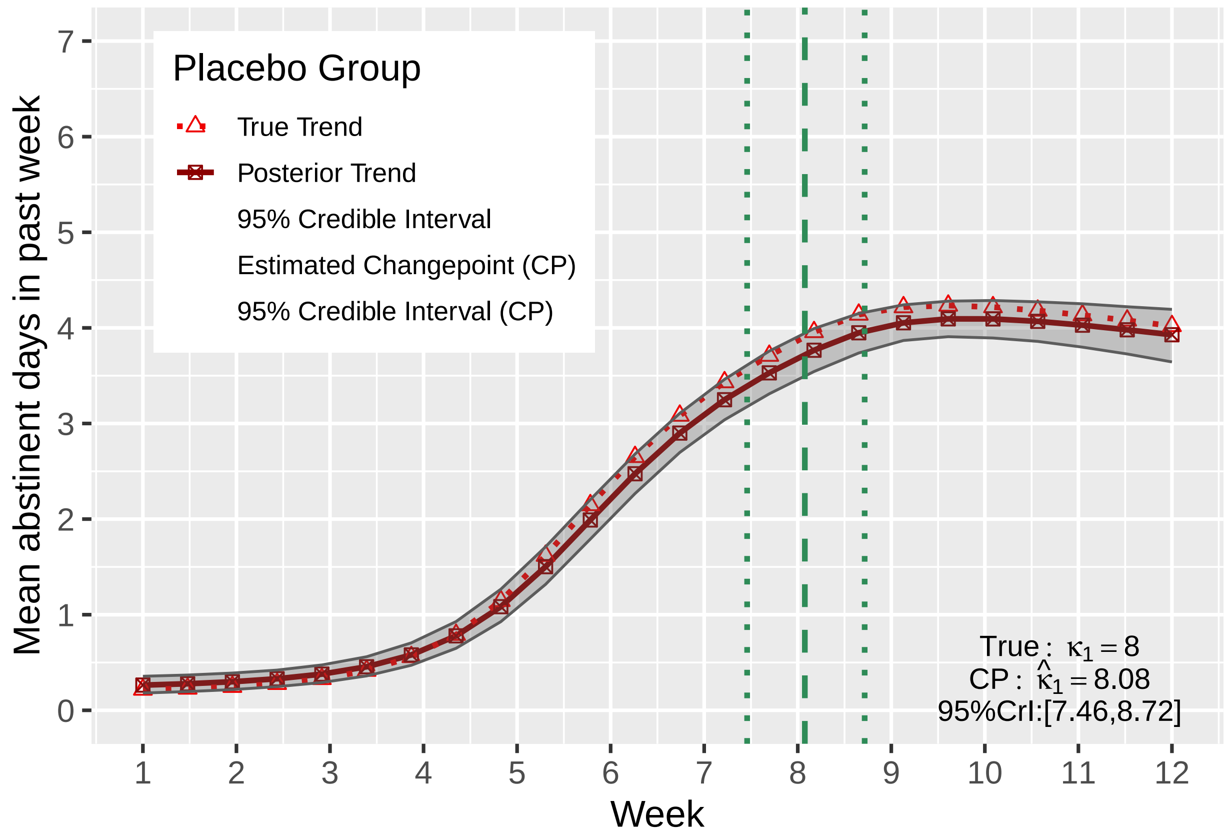
```
quantile(Kappa1-Kappa2,c(.025,.975))
```

```
##      2.5%     97.5%
## 0.2470876 2.2039351
```

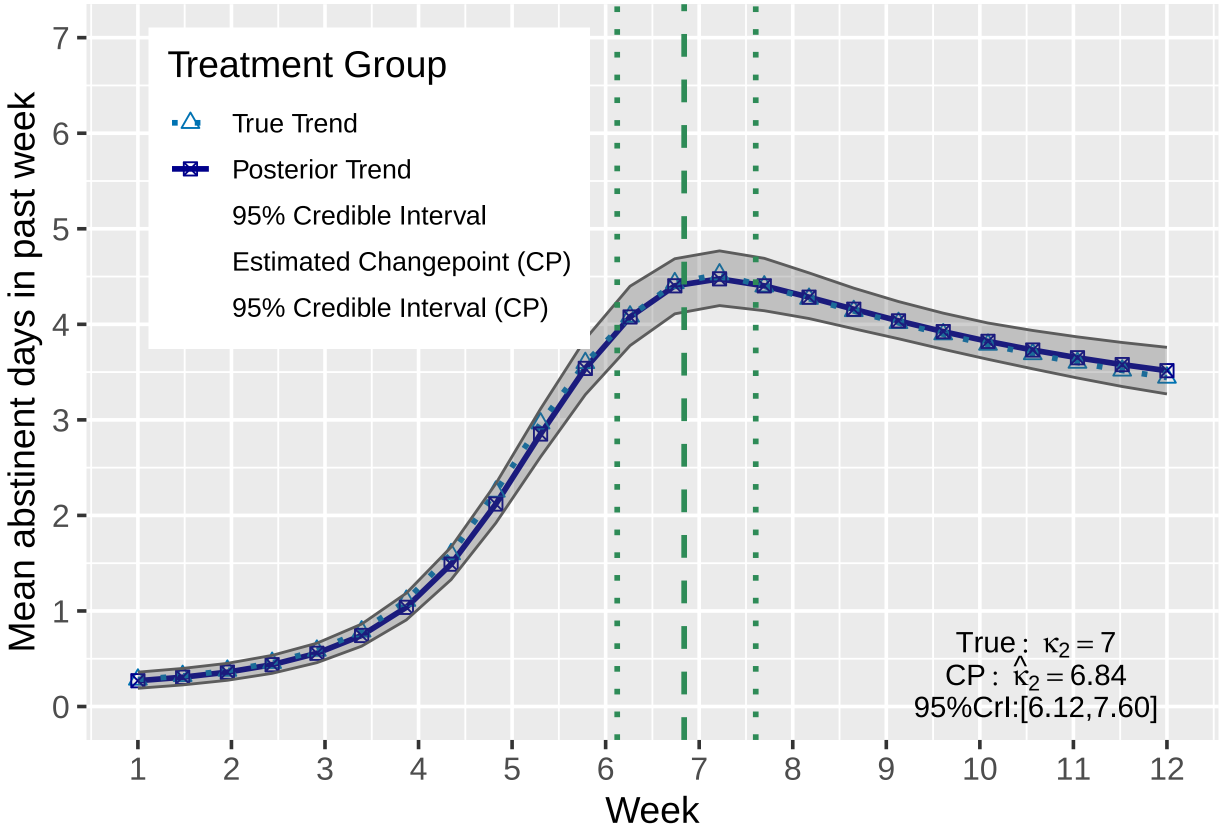
Figures

Anbsinence trends (placebo vs treatment)

```r
#-------------------------------------------------------------------------------#
# Figures #
#---------#
num=24    # 24 time-point grids
grid<-seq(-4,7,length.out=num)

n1<-sum(trt==0)     # number of subjects (Placebo)
n2<-sum(trt==1)     # number of subjects (Treatment)

tp<-rep(grid,n)
nisp<-rep(num,n)
txp<-rep(trt,nisp)

# True trajectory (placebo and tx groups)
xgp1<-(tp-truekappa1)*(tp>truekappa1)*(txp==0)
xgp2<-(tp-truekappa2)*(tp>truekappa2)*(txp==1)
xgp<-xgp1+xgp2
Xp<-cbind(1,tp,txp,tp*txp,xgp1,xgp2)
eta1obs<-Xp%*%truealpha+rep(trueb11,nisp)+rep(trueb12,nisp)*tp+rep(trueb13,nisp)*xgp
eta2obs<-Xp%*%truebeta+rep(trueb21,nisp)+rep(trueb22,nisp)*tp+rep(trueb23,nisp)*xgp
piobs<-1/(1+exp(-eta1obs))
muobs<-1/(1+exp(-eta2obs))
yobs<-piobs*muobs*ntrial


dat<-data.frame(yobs=yobs,tp=tp,txp=txp)
tmp<-dat %>%
  group_by(txp,tp)%>%
  summarise(yobs=mean(yobs))


YPOS1<-array(0,dim=c(n1,num,lastit))
YPOS2<-array(0,dim=c(n2,num,lastit))
YDIFF<-matrix(0,lastit,num)
for (j in 1:lastit){
  b11<-B11s[j,];B11<-rep(b11,nisp)
  b12<-B12s[j,];B12<-rep(b12,nisp)
  b13<-B13s[j,];B13<-rep(b13,nisp)
  b21<-B21s[j,];B21<-rep(b21,nisp)
  b22<-B22s[j,];B22<-rep(b22,nisp)
  b23<-B23s[j,];B23<-rep(b23,nisp)
  alpha<-Alpha[j,]
  beta<-Beta[j,]
  kappa1<-Kappa1[j]
  kappa2<-Kappa2[j]

  spgp1<-(tp-kappa1)*(tp>kappa1)*(txp==0)
  spgp2<-(tp-kappa2)*(tp>kappa2)*(txp==1)
  spgp<-spgp1+spgp2
  X<-cbind(1,tp,txp,tp*txp,spgp1,spgp2)

  eta1<-X%*%alpha+B11+B12*tp+B13*spgp
  pi<-1/(1+exp(-eta1))

  eta2<-X%*%beta+B21+B22*tp+B23*spgp
  mu<-1/(1+exp(-eta2))

  mmu<-matrix(pi*mu*ntrial,n,num,byrow=TRUE)


  YPOS1[,,j]<-mmu[trt==0,]
  YPOS2[,,j]<-mmu[trt==1,]

  YDIFF[j,]<-colMeans(YPOS2[,,j])-colMeans(YPOS1[,,j])

  #print(j)
}


ypos1<-colMeans(t(colMeans(YPOS1[,,1:lastit])))
ypos2<-colMeans(t(colMeans(YPOS2[,,1:lastit])))


yposlower1<-apply(t(colMeans(YPOS1[,,1:lastit])),2,quantile,prob=0.025)
yposupper1<-apply(t(colMeans(YPOS1[,,1:lastit])),2,quantile,prob=0.975)

yposlower2<-apply(t(colMeans(YPOS2[,,1:lastit])),2,quantile,prob=0.025)
yposupper2<-apply(t(colMeans(YPOS2[,,1:lastit])),2,quantile,prob=0.975)



dplot<-data.frame(grid=rep(grid+5,2),
                  mmu1=c(tmp$yobs[1:num],ypos1),
                  lb1=c(rep(NA,num),yposlower1),
                  ub1=c(rep(NA,num),yposupper1),
                  gp=c(rep("True Trend",num),rep("Posterior Trend",num)),
                  mmu2=c(tmp$yobs[((num+1):(2*num))],ypos2),
                  lb2=c(rep(NA,num),yposlower2),
                  ub2=c(rep(NA,num),yposupper2))

ggplot(dplot,aes(x=grid,y=mmu1,col=gp,shape=gp))+
  geom_line(linetype=c(rep("solid",num),rep("dotted",num)),size=1)+
  geom_point(size=2)+
  geom_ribbon(aes(ymin = lb1, ymax = ub1,col="95% Credible Interval",
                  fill="95% Credible Interval"),
              linetype=1,alpha=0.3,show.legend = F)+
  geom_vline(aes(xintercept=mkap1+5,col="Estimated Changepoint (CP)"),
             linetype="dashed",size=1,show.legend = F)+
  geom_vline(aes(xintercept=qkap1[1]+5,col="95% Credible Interval (CP)"),
             linetype="dotted",size=1,show.legend = F)+
  geom_vline(aes(xintercept=qkap1[2]+5,col="95% Credible Interval (CP)"),
             linetype="dotted",size=1,show.legend = F)+
  scale_x_continuous(breaks = 1:12,limits=c(1,12))+
  scale_y_continuous(breaks = 0:7,limits=c(0,7))+
  scale_color_manual(breaks = c("True Trend","Posterior Trend",
                                "95% Credible Interval",
                                "Estimated Changepoint (CP)",
                                "95% Credible Interval (CP)"),
                     values = c("red2","red4","grey36","seagreen","seagreen"))+
  scale_shape_manual(breaks = c("True Trend","Posterior Trend"),
                     values = c(2,7))+
  scale_fill_manual(breaks = c("95% Credible Interval"),
                    values = c("grey36"))+
  xlab("Week")+ylab("Mean abstinent days in past week")+
  guides(color = guide_legend(title="Placebo Group",
                              override.aes = list(
                                linetype = c(3,1,1,5,3),
                                shape=c(2,7,NA,NA,NA)),
                              reverse = F), fill="none",
         shape="none",linetype="none")+
  theme_gray(base_size = 14)+
  theme(legend.key = element_rect(fill = "white"),
        legend.position = c(0.25,0.75),legend.text=element_text(size=10),
        axis.text=element_text(size=12),
        axis.title=element_text(size=14))+
  annotate('text', x = 10.8, y = .65,
           label = "True:~kappa[1]==8.0",parse = TRUE,size=4)+
  annotate('text', x = 10.8, y = 0.35,
           label = "CP:~hat(kappa)[1]==8.08",parse = TRUE,size=4)+
  annotate(geom="text", x=10.8, y=0, label="95%CrI:[7.46,8.72]",
           color="black",size=4)
```

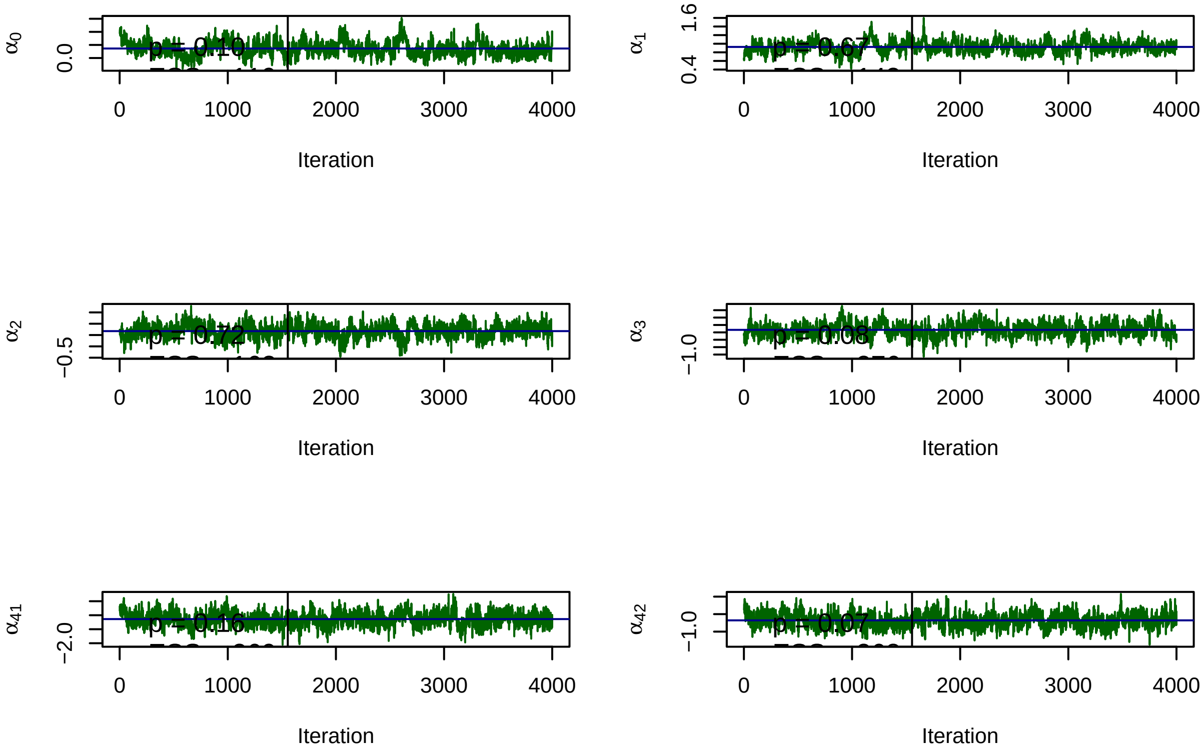


```
ggplot(dplot,aes(x=grid,y=mmu2,col=gp,shape=gp))+
  geom_line(linetype=c(rep("solid",num),rep("dotted",num)),size=1)+
  geom_point(size=2)+
  geom_ribbon(aes(ymin = lb2, ymax = ub2,col="95% Credible Interval",
                fill="95% Credible Interval"),
            linetype=1,alpha=0.3,show.legend = F)+
  geom_vline(aes(xintercept=mkap2+5,col="Estimated Changepoint (CP)"),
            linetype="dashed",size=1,show.legend = F)+
  geom_vline(aes(xintercept=qkap2[1]+5,col="95% Credible Interval (CP)"),
            linetype="dotted",size=1,show.legend = F)+
  geom_vline(aes(xintercept=qkap2[2]+5,col="95% Credible Interval (CP)"),
            linetype="dotted",size=1,show.legend = F)+
  scale_x_continuous(breaks = 1:12,limits=c(1,12))+
  scale_y_continuous(breaks = 0:7,limits=c(0,7))+
  scale_color_manual(breaks = c("True Trend","Posterior Trend",
                                "95% Credible Interval",
                                "Estimated Changepoint (CP)",
                                "95% Credible Interval (CP)"),
                     values = c("#0072B2","darkblue","grey36",
                                "seagreen","seagreen"))+
  scale_shape_manual(breaks = c("True Trend","Posterior Trend"),
                     values = c(2,7))+
  scale_fill_manual(breaks = c("95% Credible Interval"),
                    values = c("grey36"))+
  xlab("Week")+ylab("Mean abstinent days in past week")+
  guides(color = guide_legend(title="Treatment Group",
                            override.aes = list(
                              linetype = c(3,1,1,5,3),
                              shape=c(2,7,NA,NA,NA)),
                            reverse = F), fill="none",shape="none",linetype="none")+
theme_gray(base_size = 14)+
theme(legend.key = element_rect(fill = "white"),
      legend.position = c(0.25,0.75),legend.text=element_text(size=10),
      axis.text=element_text(size=12),
      axis.title=element_text(size=14))+
annotate('text', x = 10.6, y = 0.65,
         label = "True:~kappa[2]==7.0",parse = TRUE,size=4)+
annotate('text', x = 10.6, y = 0.35,
         label = "CP:~hat(kappa)[2]==6.84",parse = TRUE,size=4)+
annotate(geom="text", x=10.6, y=0, label="95%CrI:[6.12,7.60]",
         color="black",size=4)
```



Traceplots

```
par(mfrow=c(3,2))
# binary part
plot(1:lastit,Alpha[,1],type="l",col="darkgreen",
     xlab="Iteration",ylab=expression(alpha[0]))
abline(h=malpha[1],col="blue4")
legend(cex=1.2,"topleft",legend=c("p = 0.10", "ESS = 119"))

plot(1:lastit,Alpha[,2],type="l",col="darkgreen",
     xlab="Iteration",ylab=expression(alpha[1]))
abline(h=malpha[2],col="blue4")
legend(cex=1.2,"topleft",legend=c("p = 0.67", "ESS = 143"))

plot(1:lastit,Alpha[,3],type="l",col="darkgreen",
     xlab="Iteration",ylab=expression(alpha[2]))
abline(h=malpha[3],col="blue4")
legend(cex=1.2,"topleft",legend=c("p = 0.72", "ESS = 163"))

plot(1:lastit,Alpha[,4],type="l",col="darkgreen",
     xlab="Iteration",ylab=expression(alpha[3]))
abline(h=malpha[4],col="blue4")
legend(cex=1.2,"topleft",legend=c("p = 0.08","ESS = 276"))

plot(1:lastit,Alpha[,5],type="l",col="darkgreen",
     xlab="Iteration",ylab=expression(alpha[41]))
abline(h=malpha[5],col="blue4")
legend(cex=1.2,"topleft",legend=c("p = 0.16","ESS = 269"))

plot(1:lastit,Alpha[,6],type="l",col="darkgreen",
     xlab="Iteration",ylab=expression(alpha[42]))
abline(h=malpha[6],col="blue4")
legend(cex=1.2,"topleft",legend=c("p = 0.07","ESS = 268"))
```



α_0
0.0
p = 0.10
α_1
1.6
0.4
p = 0.67
α_2
−0.5
p = 0.72
α_3
−1.0
p = 0.08
α_41
−2.0
p = 0.16
α_42
−1.0
p = 0.07
0
1000
2000
3000
4000
Iteration