

```

path <-getwd()
trm.dat1 <-
read.csv(paste(path, "/Physician_TRM_MB_NewRegions_matrix_withSex.csv",
sep=""),header = T,skip = 3)
head(trm.dat1)

x<-names(trm.dat1)
trm.dat2 <- trm.dat1[, (substr(x,1,1) %in% c("X"))=="FALSE"]

trm.dat3 <-trm.dat2 [1:443,]
trm.dat4 <-trm.dat3 [-65,]

dim(trm.dat4)
rowSums(is.na(trm.dat4))

trm.dat5 <- trm.dat4[-c(27,93,120,202,248,314,341,423),]
#which(rowSums(is.na(trm.dat5))>100)
#rowSums(is.na(trm.dat5))

#row.names(trm.dat4)<- c(1:dim(trm.dat4)[1])
trm.dat6 <- trm.dat5[-c(1,2)]
trm.dat7 <- as.matrix(trm.dat6)
dim(trm.dat7)

x<-colnames(trm.dat7)
Sum.of.med_people <- trm.dat7[, (substr(x,1,16) %in%
c("Sum.of.med_peopl"))]
Average.of.avg_med_age <- trm.dat6[, (substr(x,1,18) %in%
c("Average.of.avg_med"))]
Average.of.avg_pop_age <- trm.dat6[, (substr(x,1,18) %in%
c("Average.of.avg_pop"))]
Sum.of.sum_pop <- trm.dat6[, (substr(x,1,10) %in% c("Sum.of.sum"))]

sum(is.na(Sum.of.med_people))
Sum.of.med_people[is.na(Sum.of.med_people)]<-0
sum(is.na(Sum.of.med_people))

sum(is.na(Average.of.avg_med_age))
Average.of.avg_med_age[is.na(Average.of.avg_med_age)]<-
Average.of.avg_pop_age[is.na(Average.of.avg_med_age)]
sum(is.na(Average.of.avg_med_age))
Average.of.avg_med_age[is.na(Average.of.avg_med_age)]<-
mean(Average.of.avg_med_age[!is.na(Average.of.avg_med_age)])
sum(is.na(Average.of.avg_med_age))

sum(is.na(Sum.of.sum_pop))
sum(is.na(Sum.of.sum_pop[,50]))
for(i in 1:dim(Sum.of.sum_pop)[2])
  Sum.of.sum_pop[,i] <- Sum.of.sum_pop[,50]
sum(is.na(Sum.of.sum_pop))

avg_med_age <- Average.of.avg_med_age

```

```

med_people <- Sum.of.med_people
sum_pop <- Sum.of.sum_pop

# aggregating data
months<-c(rep(2000,9),rep(c(2001:2009),each=12))
years<-c(rep(2000,9),rep(c(2001:2009),each=12))
list.x <- rep(list(1:12),9)
list.x2 <- (sapply(1:9, function(i) {list.x[[i]]+(i-1)*12}))+9)

med_people.final <- matrix(rowSums(med_people[,1:9]),nrow =
dim(med_people)[1],ncol = 1)
for(i in 1:9){
  med_people.final<- cbind(med_people.final,
matrix(rowSums(med_people[,list.x2[,i]]),nrow =
dim(med_people)[1],ncol = 1))
}

sum_pop.final <- matrix(rowSums(sum_pop[,1:9]),nrow =
dim(sum_pop)[1],ncol = 1)
for(i in 1:9){
  sum_pop.final<- cbind(sum_pop.final,
matrix(rowSums(sum_pop[,list.x2[,i]]),nrow = dim(sum_pop)[1],ncol =
1))
}

avg_med_age.final <- matrix(rowMeans(avg_med_age[,1:9]),nrow =
dim(avg_med_age)[1],ncol = 1)
for(i in 1:9){
  avg_med_age.final<- cbind(avg_med_age.final,
matrix(rowMeans(avg_med_age[,list.x2[,i]]),nrow =
dim(avg_med_age)[1],ncol = 1))
}

library("coda")
library("rbugs")
library("rjags")
library("R2WinBUGS")
library("dclone")
library("cluster")
library("nlme")
library("MASS")
library("dclone")
library("splines")

I=dim(avg_med_age.final)[1]/2 #nubmer of
areas
T=dim(avg_med_age.final)[2] # time series
length
#S=length(table(asthma1$SEX)) # SEX length
S<-2

```

```

Nsp=40      # number of splines fixed knots
degree=1    # degree of Penalized spline
#asthma<-as.data.frame(asthma)
# ----- B-spline bases -----
matXZ      <-
bs(avg_med_age.final,df=Nsp+degree,degree=degree,intercept=FALSE)

X<-cbind(1,rep(1:2,each=I*T),matXZ[,1])

nbetas<-dim(X)[2]

Z<-matXZ[,-c(1)]
colnames(Z)<-1:Nsp
#Z

#----- P-spline bases -----
#knots <- seq(0,1,length=(Nsp+2))[-c(1,(Nsp+2))]
#knots <- quantile(unique(c(avg_med_age.final)),knots)
#Z1 <- outer(c(avg_med_age.final),knots,"-")
#Z1 <- Z1*(Z1>0)
#colnames(Z1)<-1:Nsp

#----- preparing data -----
y<- med_people.final #matrix(c(asthma$med_people),I,T,byrow=TRUE)
m<- sum_pop.final   #matrix(c(asthma$sum_pop),I,T,byrow=TRUE)

K=1
parameters.hb <- c("beta","py", "rho", "sigma2.e", "sigma2.nu",
"sigma2.sp")
parameters.dc <- c("beta", "rho", "sigma2.e", "sigma2.nu",
"sigma2.sp")
parameters.p <- c("py")

model <- function() {
  for(k in 1:K){
    for(s in 1:S){
      for(i in 1:I){
        for(j in 1:T){
          y[((k-1)*I*S+(s-1)*I+i),j] ~ dbin(py[((k-1)*I*S+(s-
1)*I+i),j],m[(s-1)*I+i,j])
          logit(py[((k-1)*I*S+(s-1)*I+i),j]) <- mupy[((k-1)*I*S+(s-
1)*I+i),j]
        }
        mupy[((k-1)*I*S+(s-1)*I+i),1:T] ~ dmnorm(mmupy[((k-1)*I*S+(s-
1)*I+i),1:T],tauG[,])
        for(j in 1:T){mmupy[((k-1)*I*S+(s-1)*I+i),j]<-inprod(X[((s-
1)*I*T+(j-1)*I+i),],beta[1,])+inprod(Z[((s-1)*I*T+(j-
1)*I+i),],sp[,k])+nu[i,k]}
      }
    }
  }
}

```

```

    }}
  }

  for(l in 1:nbetas){beta[1,l]~dnorm(0.0,1.0E-6)}

  tau.e ~ dgamma(1.0E-3,1.0E-3)
  sigma2.e<-1/tau.e

  tau.nu~dgamma(1.0E-3,1.0E-3)
  sigma2.nu<-1/tau.nu

  tau.sp~dgamma(1.0E-3,1.0E-3)
  sigma2.sp<-1/tau.sp

  rho ~ dunif(-1,1)

  for(k in 1:K){
    for(i in 1:I){
      nu[i,k]~dnorm(0,tau.nu)
    }
  }

  for(k in 1:K){
    for(i in 1:Nsp){
      sp[i,k]~dnorm(0,tau.sp)
    }
  }

  for(i in 1:T){gam[i,i]<-1/(1-rho*rho)}
  for(i in 1:(T-1)){gam[i,i+1]<-rho/(1-rho*rho)}
  gam[i+1,i]<-rho/(1-rho*rho)}
  for(i in 1:(T-2)){gam[i,i+2]<-pow(rho,2)/(1-rho*rho)}
  gam[i+2,i]<-pow(rho,2)/(1-rho*rho)}
  for(i in 1:(T-3)){gam[i,i+3]<-pow(rho,3)/(1-rho*rho)}
  gam[i+3,i]<-pow(rho,3)/(1-rho*rho)}
  for(i in 1:(T-4)){gam[i,i+4]<-pow(rho,4)/(1-rho*rho)}
  gam[i+4,i]<-pow(rho,4)/(1-rho*rho)}
  for(i in 1:(T-5)){gam[i,i+5]<-pow(rho,5)/(1-rho*rho)}
  gam[i+5,i]<-pow(rho,5)/(1-rho*rho)}
  for(i in 1:(T-6)){gam[i,i+6]<-pow(rho,6)/(1-rho*rho)}
  gam[i+6,i]<-pow(rho,6)/(1-rho*rho)}
  for(i in 1:(T-7)){gam[i,i+7]<-pow(rho,7)/(1-rho*rho)}
  gam[i+7,i]<-pow(rho,7)/(1-rho*rho)}
  for(i in 1:(T-8)){gam[i,i+8]<-pow(rho,8)/(1-rho*rho)}
  gam[i+8,i]<-pow(rho,8)/(1-rho*rho)}
  gam[1,T]<-pow(rho,9)/(1-rho*rho)
  gam[T,1]<-pow(rho,9)/(1-rho*rho)

  sigmaG<-sigma2.e*gam
  invG<-inverse(sigmaG)

  for(i in 1:T){tauG[i,i]<-invG[i,i]}
  for(i in 1:(T-1)){tauG[i,i+1]<-(invG[i,i+1]+invG[i+1,i])/2}
  tauG[i+1,i]<-(invG[i,i+1]+invG[i+1,i])/2}

```

```

for(i in 1:(T-2)){tauG[i,i+2]<-(invG[i,i+2]+invG[i+2,i])/2
tauG[i+2,i]<-(invG[i,i+2]+invG[i+2,i])/2}
for(i in 1:(T-3)){tauG[i,i+3]<-(invG[i,i+3]+invG[i+3,i])/2
tauG[i+3,i]<-(invG[i,i+3]+invG[i+3,i])/2}
for(i in 1:(T-4)){tauG[i,i+4]<-(invG[i,i+4]+invG[i+4,i])/2
tauG[i+4,i]<-(invG[i,i+4]+invG[i+4,i])/2}
for(i in 1:(T-5)){tauG[i,i+5]<-(invG[i,i+5]+invG[i+5,i])/2
tauG[i+5,i]<-(invG[i,i+5]+invG[i+5,i])/2}
for(i in 1:(T-6)){tauG[i,i+6]<-(invG[i,i+6]+invG[i+6,i])/2
tauG[i+6,i]<-(invG[i,i+6]+invG[i+6,i])/2}
for(i in 1:(T-7)){tauG[i,i+7]<-(invG[i,i+7]+invG[i+7,i])/2
tauG[i+7,i]<-(invG[i,i+7]+invG[i+7,i])/2}
for(i in 1:(T-8)){tauG[i,i+8]<-(invG[i,i+8]+invG[i+8,i])/2
tauG[i+8,i]<-(invG[i,i+8]+invG[i+8,i])/2}
tauG[1,T]<-(invG[1,T]+invG[T,1])/2
tauG[T,1]<-(invG[1,T]+invG[T,1])/2
}

#----- model prediction
model.p <- function() {
  for(k in 1:K){
    for(s in 1:S){
      for(i in 1:I){
        for(j in 1:T){
          y[((k-1)*I*S+(s-1)*I+i),j] ~ dbin(py[((k-1)*I*S+(s-
1)*I+i),j],m[(s-1)*I+i,j])
          logit(py[((k-1)*I*S+(s-1)*I+i),j]) <- mupy[((k-1)*I*S+(s-
1)*I+i),j]
        }
        mupy[((k-1)*I*S+(s-1)*I+i),1:T] ~ dmnorm(mmupy[((k-1)*I*S+(s-
1)*I+i),1:T],tauG[,])
        for(j in 1:T){mmupy[((k-1)*I*S+(s-1)*I+i),j]<-inprod(X[((s-
1)*I*T+(j-1)*I+i),],beta[1,])+inprod(Z[((s-1)*I*T+(j-
1)*I+i),],sp[,k])+nu[i,k]}
      }}
    }

    tmp[1:(nbetas+4)] ~ dmnorm(param[,], prec[,])

    for(l in 1:nbetas){beta[1,l]<- tmp[l]}
    rho <- (1-step(tmp[(nbetas+1)]+1))*(-0.9999)+step(tmp[(nbetas+1)]-
1)*(0.9999)+(1-step(abs(tmp[(nbetas+1)])-1))*(tmp[(nbetas+1)])
    sigma2.e <- abs(tmp[(nbetas+2)])
    sigma2.nu <- abs(tmp[(nbetas+3)])
    sigma2.sp <- abs(tmp[(nbetas+4)])

    tau.sp<-1/sigma2.sp
    tau.nu<-1/sigma2.nu

    for(k in 1:K){
      for(i in 1:Nsp){

```

```

    sp[i,k] ~ dnorm(0,tau.sp)
  }}

for(k in 1:K){
  for(i in 1:I){
    nu[i,k]~dnorm(0,tau.nu)
  }}

for(i in 1:T){gam[i,i]<-1/(1-rho*rho)}
for(i in 1:(T-1)){gam[i,i+1]<-rho/(1-rho*rho)}
gam[i+1,i]<-rho/(1-rho*rho)}
for(i in 1:(T-2)){gam[i,i+2]<-pow(rho,2)/(1-rho*rho)}
gam[i+2,i]<-pow(rho,2)/(1-rho*rho)}
for(i in 1:(T-3)){gam[i,i+3]<-pow(rho,3)/(1-rho*rho)}
gam[i+3,i]<-pow(rho,3)/(1-rho*rho)}
for(i in 1:(T-4)){gam[i,i+4]<-pow(rho,4)/(1-rho*rho)}
gam[i+4,i]<-pow(rho,4)/(1-rho*rho)}
for(i in 1:(T-5)){gam[i,i+5]<-pow(rho,5)/(1-rho*rho)}
gam[i+5,i]<-pow(rho,5)/(1-rho*rho)}
for(i in 1:(T-6)){gam[i,i+6]<-pow(rho,6)/(1-rho*rho)}
gam[i+6,i]<-pow(rho,6)/(1-rho*rho)}
for(i in 1:(T-7)){gam[i,i+7]<-pow(rho,7)/(1-rho*rho)}
gam[i+7,i]<-pow(rho,7)/(1-rho*rho)}
for(i in 1:(T-8)){gam[i,i+8]<-pow(rho,8)/(1-rho*rho)}
gam[i+8,i]<-pow(rho,8)/(1-rho*rho)}
gam[1,T]<-pow(rho,9)/(1-rho*rho)
gam[T,1]<-pow(rho,9)/(1-rho*rho)

sigmaG<-sigma2.e*gam
invG<-inverse(sigmaG)

for(i in 1:T){tauG[i,i]<-invG[i,i]}
for(i in 1:(T-1)){tauG[i,i+1]<-(invG[i,i+1]+invG[i+1,i])/2}
tauG[i+1,i]<-(invG[i,i+1]+invG[i+1,i])/2}
for(i in 1:(T-2)){tauG[i,i+2]<-(invG[i,i+2]+invG[i+2,i])/2}
tauG[i+2,i]<-(invG[i,i+2]+invG[i+2,i])/2}
for(i in 1:(T-3)){tauG[i,i+3]<-(invG[i,i+3]+invG[i+3,i])/2}
tauG[i+3,i]<-(invG[i,i+3]+invG[i+3,i])/2}
for(i in 1:(T-4)){tauG[i,i+4]<-(invG[i,i+4]+invG[i+4,i])/2}
tauG[i+4,i]<-(invG[i,i+4]+invG[i+4,i])/2}
for(i in 1:(T-5)){tauG[i,i+5]<-(invG[i,i+5]+invG[i+5,i])/2}
tauG[i+5,i]<-(invG[i,i+5]+invG[i+5,i])/2}
for(i in 1:(T-6)){tauG[i,i+6]<-(invG[i,i+6]+invG[i+6,i])/2}
tauG[i+6,i]<-(invG[i,i+6]+invG[i+6,i])/2}
for(i in 1:(T-7)){tauG[i,i+7]<-(invG[i,i+7]+invG[i+7,i])/2}
tauG[i+7,i]<-(invG[i,i+7]+invG[i+7,i])/2}
for(i in 1:(T-8)){tauG[i,i+8]<-(invG[i,i+8]+invG[i+8,i])/2}
tauG[i+8,i]<-(invG[i,i+8]+invG[i+8,i])/2}
tauG[1,T]<-(invG[1,T]+invG[T,1])/2
tauG[T,1]<-(invG[1,T]+invG[T,1])/2
}

```

```

#-----
stats.hb<-list()
stats.dc<-list()
stats.p<-list()

modelout.sim<-modelout.mcmc<-list()

data.sim<-list(y=y, m=m, X=X, Z=Z, I=I, T=T, S=S,
Nsp=Nsp,nbetas=nbetas, K=1)
print(paste("----- HB model-----"))
modelout.sim<- jags.fit(data=data.sim,parameters.hb,
model,n.chains=2,thin = 5, n.iter=50000, n.adapt=50000)
update(updated.model(modelout.sim),50000)
modelout.mcmc<-
coda.samples(updated.model(modelout.sim),parameters.hb,n.iter=50000,thin=5)
num<-0
while( (max ((gelman.diag(modelout.mcmc))$psrf[,1]) >1.05) & num<40){
  update(updated.model(modelout.sim),50000)
  modelout.mcmc<-
coda.samples(updated.model(modelout.sim),parameters.hb,n.iter=50000,thin=5)
  num<-num+1
}

print(paste("Gelman"))
print(max((gelman.diag(modelout.mcmc))$psrf[,1]))
stats.hb<-
summary(modelout.mcmc,quantiles=c(0.005,0.01,0.025,0.05,0.95,0.975,0.99,0.995))
print(stats.hb)
print(stats.hb$statistics)

lambda.hb<-as.numeric(lambdamax.diag(modelout.mcmc))
#lambda.hb<-0.0288606776485366
print(paste("lambda.hb",lambda.hb))

print(paste("----- Data cloning number-----
-----"))

K1<-0 #initial number of clones
r2<-1
mse<-1
lambda.ratio<-1
r.hat<-2
num<-0
while( ( r2>0.01 | mse>0.01 ) & ( lambda.ratio>=0.05 & num<5)){
  K1<-K1+20
  dcmoc<-dc.fit(list(y=dclone(y,K1), m=m, X=X, Z=Z, I=I, T=T, S=S,
Nsp=Nsp, nbetas=nbetas, K=1), parameters.dc, model,

```

```

n.clones=c(K1),multiply=c("K"),unchanged=c("I","T","m","Nsp","nbetas",
"S"), n.chains=2, n.adapt=50000, n.update=50000,n.iter=50000, thin=5)
dcd<-dcdiag(dcmmod)
print(dcd)
r2<-dcd[1,4]
mse<-dcd[1,3]
lambda.ratio<-dcd[1,2]/lambda.hb
num<-num+1
}
print(paste("num",num))
stats.dc<-
summary(dcmmod,quantiles=c(0.005,0.01,0.025,0.05,0.95,0.975,0.99,0.995)
)
print(stats.dc)

```

```

#-----Prediction of random effects based on MLE:
print(paste("Prediction for Data cloning model="))
prec <- make.symmetric((K1)*solve(vcov(dcmmod)))
prdat <-list(y=y, m=m, X=X, Z=Z, I=I, T=T, S=S, Nsp=Nsp,
nbetas=nbetas, param = coef(dcmmod), prec = prec, K=1)
mod1.1.pr <- jags.fit(prdat, parameters.p, model.p, n.chains=2,
n.adapt=50000, n.update=50000, n.iter=50000, thin=5)

update(updated.model(mod1.1.pr), 50000)
mod1.2.pr <- coda.samples(updated.model(mod1.1.pr), parameters.p,
n.iter=50000, thin = 5)

num<-0
while( (max ((gelman.diag(mod1.2.pr))$psrf[,1]) >1.05) & num<60){
  update(updated.model(mod1.1.pr), 50000)
  mod1.2.pr <- coda.samples(updated.model(mod1.1.pr), parameters.p,
n.iter=50000, thin = 5)
  num<-num+1
}
print(paste("num",num))
stats.p<-
summary(mod1.2.pr,quantiles=c(0.005,0.01,0.025,0.05,0.95,0.975,0.99,0.
995))

date()

save(list = ls(all=TRUE), file =
paste(path,"/asthma_physicianVisit_217region.RData",sep=""))

```

```

library(ggplot2)
attributes(data.sim)

```

```

data.sim2 <- data.frame(paste("R",1:217, sep = ""),
rep(c("Female","Male"), each=217) ,log((data.sim$y/data.sim$m+1e-
10)/(1-(data.sim$y/data.sim$m+1e-10))))
colnames(data.sim2) <- c("Region", "Gender" ,rep(c(2001:2010),sep =
""))
library(reshape2)
data.sim3 <- melt(data.sim2, id.vars=1:2)
data.sim3$Region= factor(data.sim3$Region, levels=paste("R",1:217,sep
= ""))
colnames(data.sim3) <- c("Region", "Gender", "Year", "logitProp")
data.sim3

data.sim4 = data.sim3[data.sim3$Gender=="Female",]
data.sim5 = data.sim4[which(data.sim4$Region %in%
c(paste("R",c(1:16),sep = ""))),]

pdf("../physRate.pdf")
ggplot(data.sim5, aes(x = Year, y = logitProp, colour=Region)) +
  geom_line(aes(group=Region)) +
  facet_wrap(~ Region) +
  theme(legend.position="none", text = element_text(size=10),
        axis.text.x = element_text(angle=45, hjust=1))+
  ggtitle("logit of Asthma physician rate of females for some Regions,
2001-2010")
dev.off()

```