Supplement 9.3. R/WinBUGS code for evaluations summarized in Table 9.3 (MCMC estimation)

Brian R. Gray, US Geological Survey, Upper Midwest Environmental Sciences Center, La Crosse, Wisconsin, USA.

This supplement provides R/WinBUGS code for evaluating MCMC estimation of variance components using logit-normal models of binomial data derived from two-way cross-classification designs. Corresponding routines were used for all scenarios summarized in Table 9.3 of Chapter 9.

The following code was created by Sherwin Toribio, University Wisconsin – La Crosse, July 2010.

Created by Sherwin Toribio, University Wisconsin – La Crosse, July 2010

1. Simulation code for scenario 8.

```
iter=200; nv=5; nsites=20; nyears=5
v_s=.3;v_y=1;v_s_y=.15; b0=-2
result_case8=matrix(rep(0,iter*4),ncol=4)

for(i in 1:iter){
data.gen=data.gen.2F.bin.random.effects(nsites,nyears,nv,v_s,v_y,v_s_y,b0)
       obs=data.gen$data
       results=est_2F_bin_random_effects(obs,nsites,nyears,nv)
       result_case8[i,]=results$summary[1:4,5] # b0,sigma.b,sigma.site,sigma.yr
       }

mean(results_case8[,1]);sd(results_case8[,1])  # yields the mean and sd for b_0.
apply(results_case8[,2:4]^2,2,mean)     # mean(var_components)- [s*yr,site,yr].
apply(results_case8[,2:4]^2,2,sd)       # sd(var_components) – [s*yr,site,yr].
```

2. An R function to simulate data.

```
data.gen.2F.bin.random.effects=function(sites,years,nv,var_site,var_yr,var_site_year,b0)
{
year=rep(1:years,sites)
site=rep(1:sites,years)
year.eff.v=round(rnorm(years,mean=0,sd=sqrt(var_yr)),digits=4)
site.eff.v=round(rnorm(sites,mean=0,sd=sqrt(var_site)),digits=4)
site.year.eff=round(rnorm(sites*years,mean=0,sd=sqrt(var_site_year)),digits=4)

data=data.frame(site=sort(site),year=year,year.eff=rep(year.eff.v,sites),
site.eff=rep(site.eff.v,each=years),site.year.eff=site.year.eff)

logit.p=b0 + data$year.eff + data$site.eff + data$site.year.eff
p=1/(1+exp(-logit.p))
y_ij=rbinom(sites*years,nv,p)
data=data.frame(data,prob=round(p,digits=4),count=y_ij)

list(data=data,site.eff.v=site.eff.v,year.eff.v=year.eff.v)
}
```

3. An R function to obtain the Bayesian estimates. This function uses WinBUGS to obtain posterior samples for model parameters and saves it as an R object.

```
est_2F_bin_random_effects<-function(obs,nsites,nyears,nv)
{
y = obs$count
site=obs$site
year=obs$year
N = length(y)

data=list("y","nv","N","site","year","nyears","nsites")

inits<-function(){list(b0=0, tau.b=0.5, tau.b1=0.5, tau.b2=0.5,
b1=rep(0,nsites), b2=rep(0,nyears), b=rep(0,N))}

parameters<-c("b0","sigma.b","sigma.b1","sigma.b2")
```

```
results<-bugs(data,inits,parameters,"2F.bin.random.effects.bug",n.chains=2,
n.iter=11000,n.burnin=1000,n.thin=5)
}
```

4. Specifying the model for WinBUGS (needed for the previous function to work)

```
# save as "2F.bin.random.effects.bug"
model
      {
      for(i in 1:nsites){b1[i]  ~ dnorm(0.0, tau.b1)}
      for(j in 1:nyears){b2[j]  ~ dnorm(0.0, tau.b2)}
      for(k in 1 : N) {
            logit(p[k]) <- b0 + b1[site[k]] + b2[year[k]] + b[k]
            y[k] ~ dbin(p[k],nv);      # nv - no. of visits
            b[k] ~ dnorm(0.0, tau.b);  # site*year random effects
            }

      # priors:
      b0 ~ dnorm(0.0,1.0E-4)
      tau.b1     ~ dgamma(1.0E-3,1.0E-3); sigma.b1 <- 1.0 / sqrt(tau.b1)
      tau.b2     ~ dgamma(1.0E-3,1.0E-3); sigma.b2 <- 1.0 / sqrt(tau.b2)
      tau.b      ~ dgamma(1.0E-3,1.0E-3); sigma.b  <- 1.0/  sqrt(tau.b)
      }
```