Spatial Econometrics Models with INLA and MCMC

Virgilio Gómez-Rubio

Introduction

In this example we compare the implementatin of several Spatial Econometrics models using INLA and MCMC. In particular, the implementation of these methods with INLA is based on the *slm* latent class, whilst the implementation with MCMC has been done with *jags* and it is available in package *SEjags*.

```
library(SEjags)
library(INLA)
```

MCMC models

```
data(columbus)
d <- columbus
W <- nb2mat(col.gal.nb, style = "W")
m.form <-  CRIME ~ INC + HOVAL

#Fit models with SEjags
if(!file.exists("INLAvsMCMC-MCMC.Rdata") ) {
 sem.mcmc <- SEjags(m.form, data = d, W = W, model = "sem",
   n.burnin = 5000, n.iter = 10000, n.thin = 20, linear.predictor = TRUE)
 slm.mcmc <- SEjags(m.form, data = d, W = W, model = "slm",
   n.burnin = 5000, n.iter = 10000, n.thin = 20, linear.predictor = TRUE)
 sdm.mcmc <- SEjags(m.form, data = d, W = W, model = "sdm",
   n.burnin = 5000, n.iter = 10000, n.thin = 20, linear.predictor = TRUE)

 save(file = "INLAvsMCMC-MCMC.Rdata",
   list = c("sem.mcmc", "slm.mcmc", "sdm.mcmc"))
} else {
  load("INLAvsMCMC-MCMC.Rdata")
}
```

INLA models

```
#Area index
columbus$idx <- 1:nrow(columbus)
#Adjacency matrix as sparse matrix
W.inla <- as(W, "CsparseMatrix")

#Model matrix for SLM models
mmatrix <- model.matrix(m.form, columbus)
mmatrix2 <-  cBind(mmatrix, W.inla %*% mmatrix[,-1])
colnames(mmatrix2)[4:5]<- paste("lag", colnames(mmatrix2)[2:3],sep="")

#Zero-variance for error term
```

```
#Zero-variance to remove effect in linear predictor: DOES NOT WORK
zero.variance = list(prec=list(initial = 25, fixed=TRUE))
#Large variance to allow for an uncnstrained estimation of the other parameters
#zero.variance = list(prec=list(initial = 1/100, fixed=TRUE))

#Compute eigenvalues for SLM model (as in Havard's code)
e = eigen(W.inla)$values
re.idx = which(abs(Im(e)) < 1e-6)
rho.max = 1/max(Re(e[re.idx]))
rho.min = 1/min(Re(e[re.idx]))
rho = mean(c(rho.min, rho.max))

#
#Variance-covarinace matrix for beta coeffients' prior
#
betaprec <- .001
#Standard regression model
Q.beta = Diagonal(n = ncol(mmatrix), x = 1)
Q.beta = betaprec * Q.beta
#Regression model with lagged covariates
Q.beta2 = Diagonal(n = ncol(mmatrix2), x = 1)
Q.beta2 = betaprec * Q.beta2


#Arguments for slm latent model
args.slm = list(
   rho.min = rho.min,
   rho.max = rho.max,
   W = W.inla,#as(W.inla,"dgTMatrix"),
   X = matrix(0, nrow(mmatrix),0),
   Q.beta = matrix(1,0,0)
)

#Hyperparameters
hyper.slm = list(
   prec = list(prior = "loggamma", param = c(0.01, 0.01)),
      rho = list(initial = 0, prior = "logitbeta", param = c(1,1))
)


#Control fixed
c.fixed <- list(prec = 0.001, prec.intercept = 0.001)
```

Fitting models with INLA:

```
#SEM model
hyper.sem <- hyper.slm
hyper.sem$rho$initial <- 0.85 #Fixed to posterior mode from MCMC
hyper.sem$rho$fixed <- TRUE

#Change zero variance
zero.variance$prec$initial <- 1/100

#Control inla
```

```
c.inla <- list(strategy = "laplace", fast = FALSE,
  tolerance = 0.001,
     int.strategy = "ccd", h = 0.001, dz = 0.05, stencil = 9)
#  int.strategy = 'grid', diff.logdens = 0.1, h = 0.001, dz = 0.01, stencil = 9)

# Create linear combinations on the covariates to estimate
# linear predictor (and fitted values).

n <- nrow(columbus)

#Test how the structure of the linear combinatios should be
lc1 <- inla.make.lincomb(list("(Intercept)" = 1,
  INC = columbus$INC[1], HOVAL = columbus$HOVAL[1],
  idx = c(1, rep(NA, n-1))
))

lc.linpred <- lapply(1:n, function(X) {
  idx.lc <- rep(NA, 49)
  idx.lc[X] <- 1
  aux <- as.list(mmatrix[X, ])
  aux$idx = idx.lc
  inla.make.lincomb(aux)
})


lc.linpred <- do.call(c, lc.linpred)
names(lc.linpred) <- paste("lc.linpred", 1:n, sep = "")

#Linear combination of the fixed effects
lc.fixed <- inla.make.lincombs(INLA:::inla.uncbind(mmatrix))
names(lc.fixed) <- paste("lc.fixed", 1:n, sep = "")


#Linear combinations for SLM and SDM
lc.linpred2 <- lapply(1:n, function(X) {
  idx.lc <- rep(NA, 49)
  idx.lc[X] <- 1
  inla.make.lincomb(list(idx = idx.lc))
})
lc.linpred2 <- do.call(c, lc.linpred2)
names(lc.linpred2) <- paste("lc", 1:n, sep = "")

sem.inla<-inla(CRIME ~ INC + HOVAL +
   f(idx, model = "slm", args.slm = args.slm, hyper = hyper.slm),
   data = as.data.frame(columbus), family = "gaussian",
   lincomb = c(lc.linpred, lc.fixed), control.predictor = list(compute=TRUE),
   control.fixed = c.fixed,
   control.inla = c.inla,
   control.family = list(hyper = zero.variance),
   control.compute = list(dic = TRUE, cpo = TRUE)
)
```

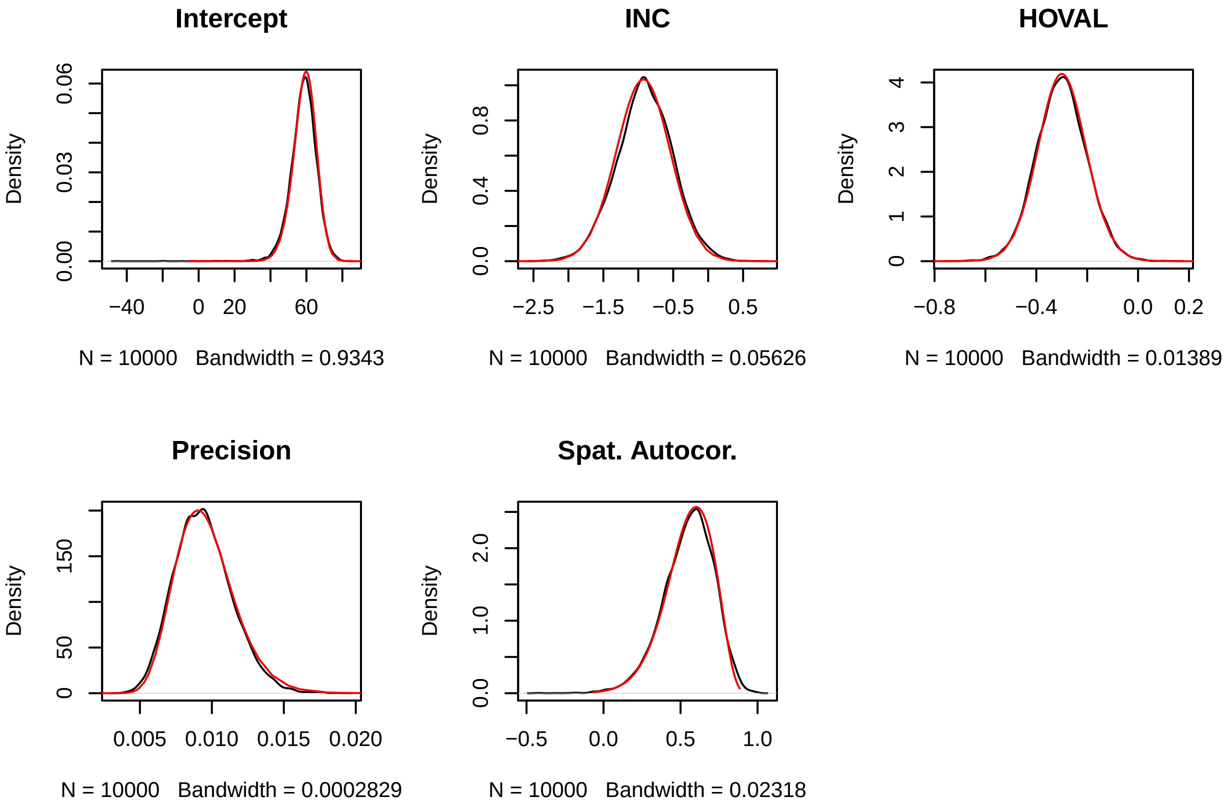
```r
#SLM model
slm.inla<-inla( CRIME ~ -1 +
   f(idx, model="slm",
      args.slm=list(rho.min = rho.min, rho.max = rho.max, W = W.inla, X=mmatrix,
         Q.beta = Q.beta),
      hyper=hyper.slm),
   data=as.data.frame(columbus), family="gaussian",
   lincomb = lc.linpred2, control.predictor = list(compute=TRUE),
   control.fixed = c.fixed,
   control.inla = c.inla,
   control.family = list(hyper=zero.variance),
   control.compute=list(dic=TRUE, cpo=TRUE)
)

#SDM model
hyper.sdm <- hyper.slm

#Fix rho
hyper.sdm$rho$initial <- 0.77 #Fixed to posterior mode from MCMC
#hyper.sdm$rho$fixed <- TRUE

#Use stronger prior
hyper.sdm$rho$param <- c(140, 60)


sdm.inla <- inla( CRIME ~ -1 +
   f(idx, model = "slm",
      args.slm = list(rho.min = rho.min, rho.max = rho.max, W = W.inla,
        X = mmatrix2, Q.beta = Q.beta2),
      hyper = hyper.sdm),
   data = as.data.frame(columbus), family = "gaussian",
   lincomb = lc.linpred2, control.predictor = list(compute=TRUE),
   control.fixed = c.fixed,
   control.inla = c.inla,
   control.family = list(hyper = zero.variance),
   control.compute = list(dic = TRUE, cpo = TRUE)
)
```

Results

Transform estimate of spatial autocorrelation provided by INLA:

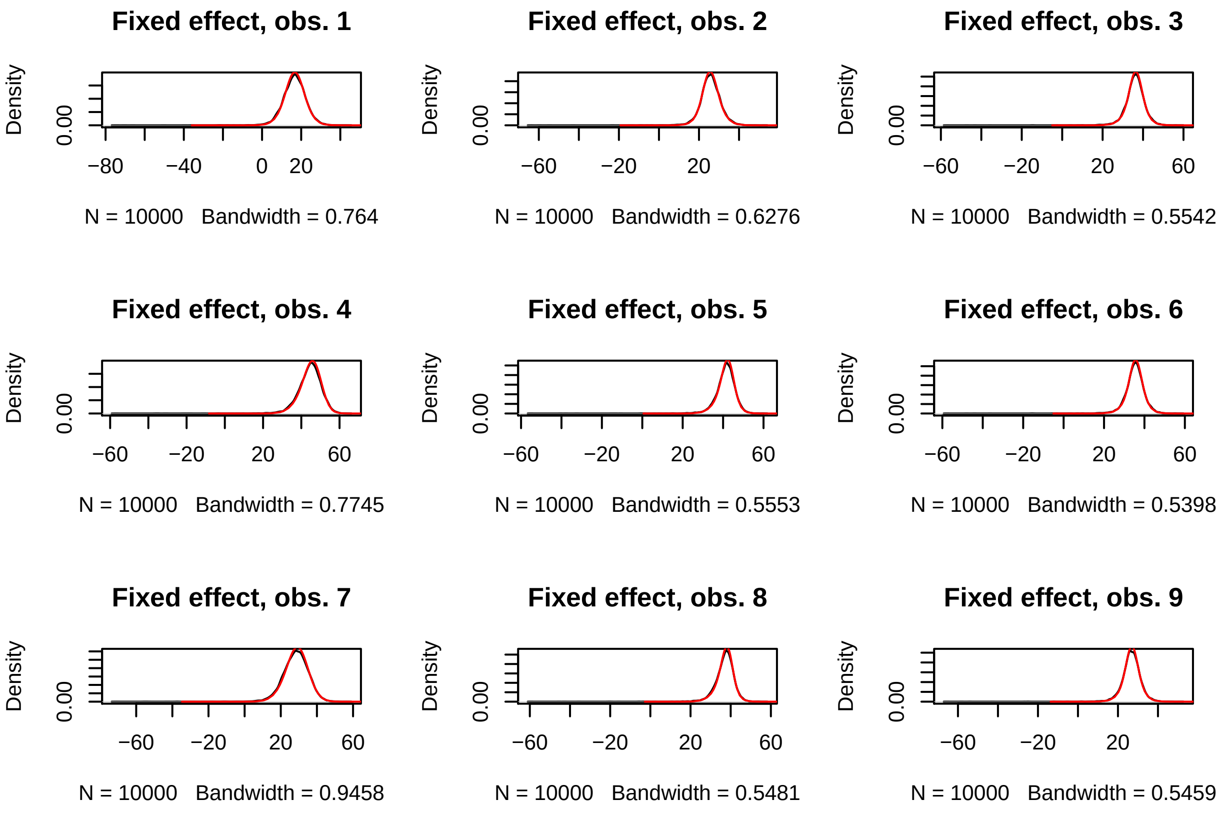
```r
  ff <- function(z){z * (rho.max - rho.min) + rho.min}
  semmarg <- inla.tmarginal(ff, sem.inla$marginals.hyperpar[[2]])
  slmmarg <- inla.tmarginal(ff, slm.inla$marginals.hyperpar[[2]])
  sdmmarg <- inla.tmarginal(ff, sdm.inla$marginals.hyperpar[[2]])
```

Spatial Error Model (SEM)



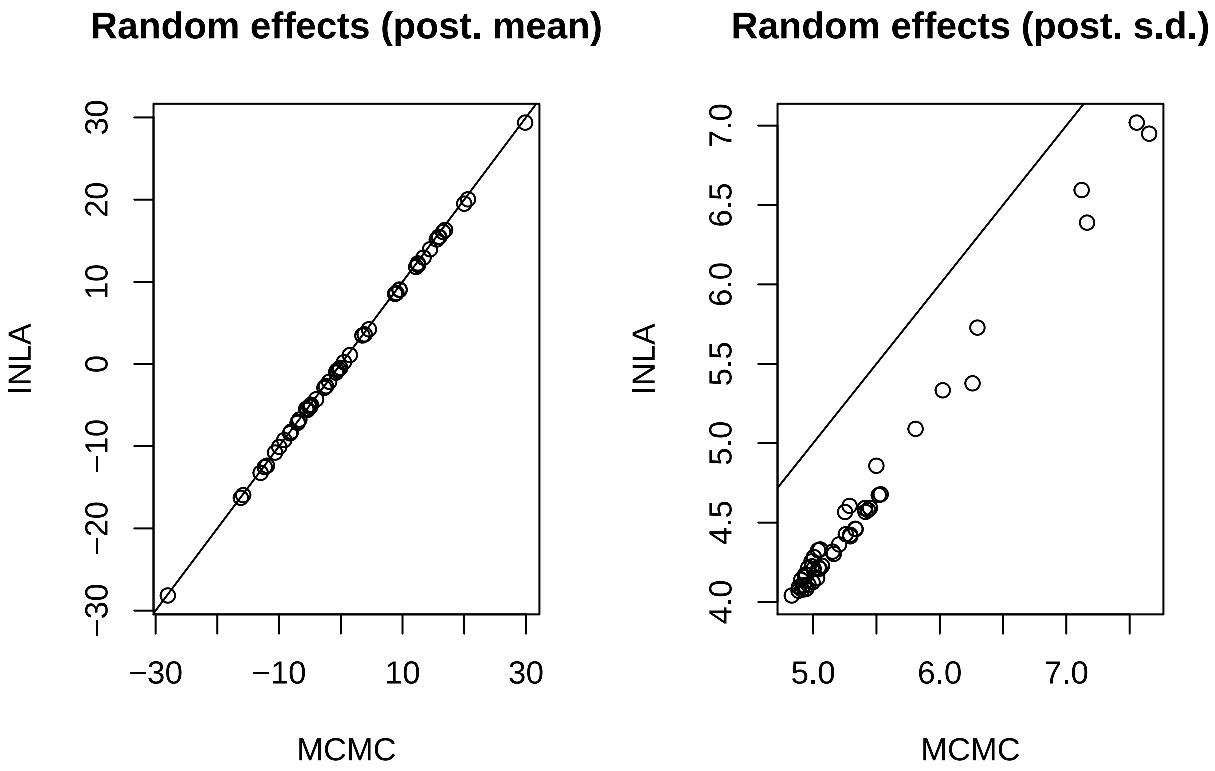
Effect of the covariates

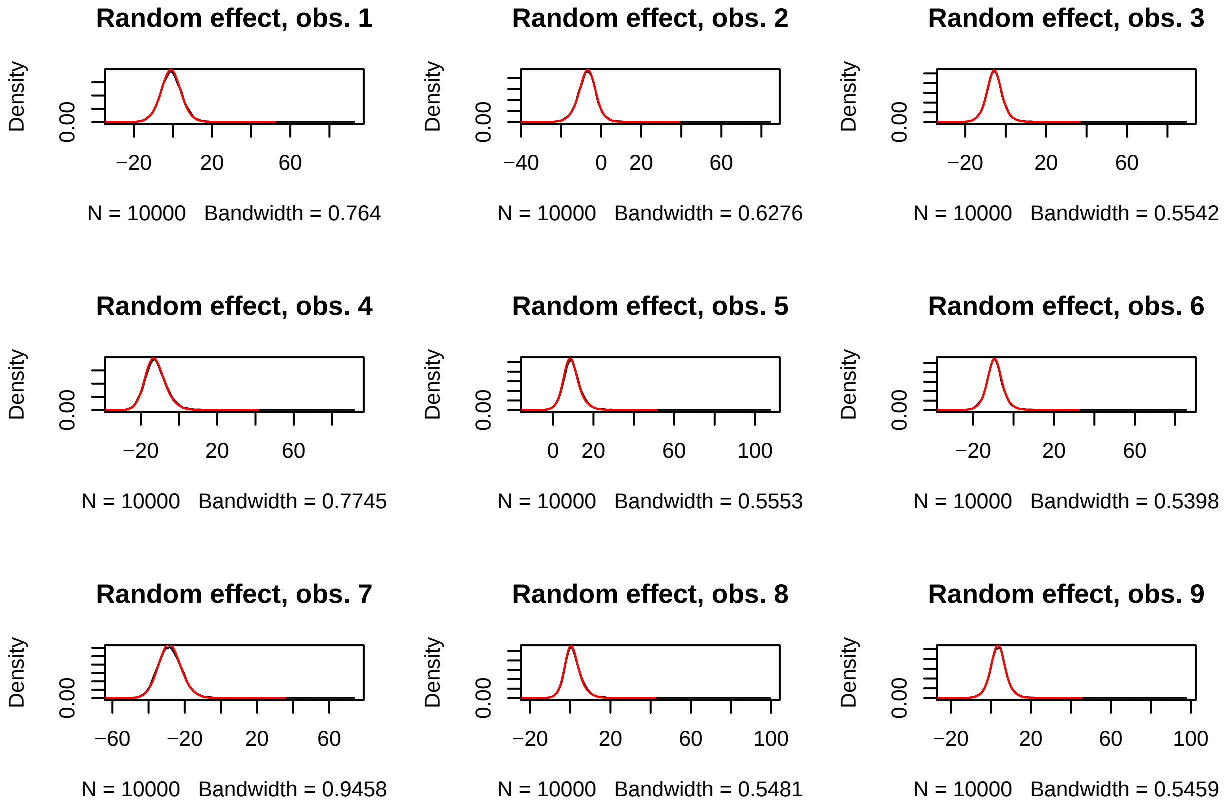
The next plots display the estimated effect of the covariates alone to see if there is agreement.



Estimate of the random effects

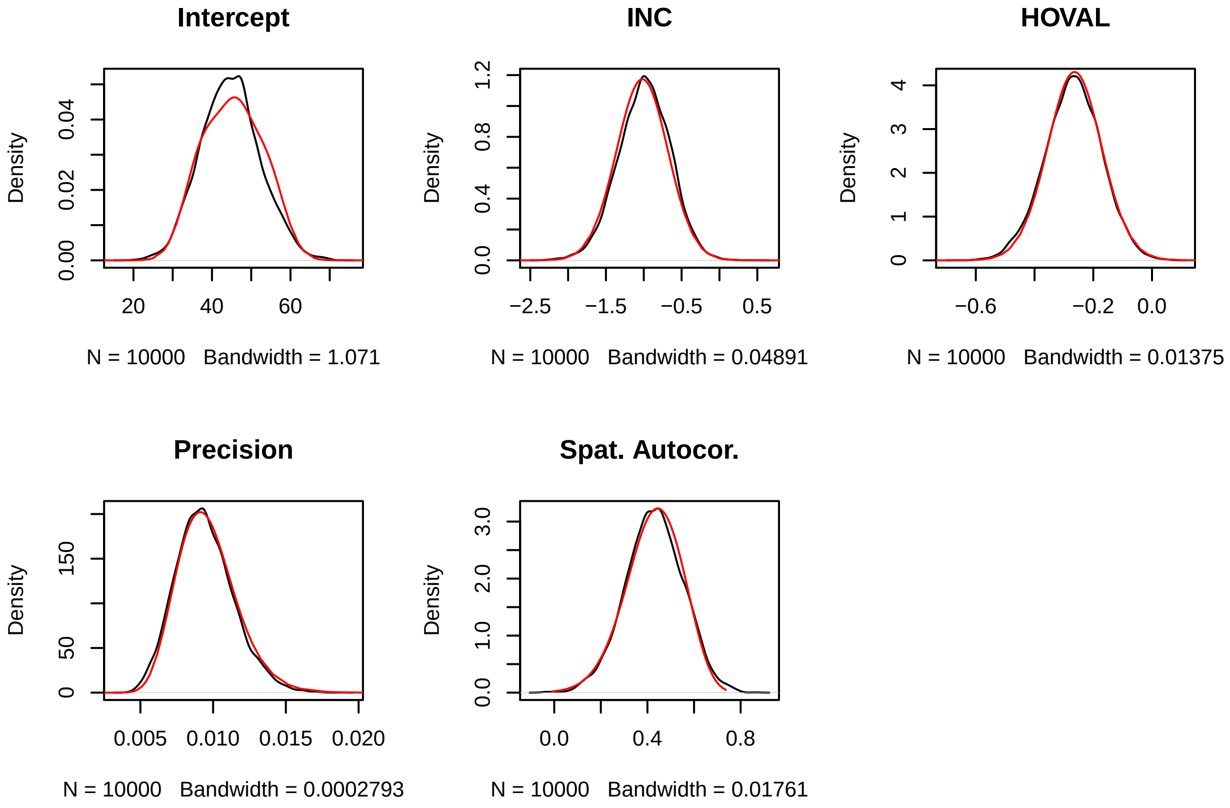
```r
sem.raneff.mcmc <- apply(sem.mcmc$mu, 2, function(X) {columbus$CRIME - X})
sem.raneff.mcmc <- matrix(sem.raneff.mcmc, ncol = 49, byrow = TRUE)

sem.raneff.mean <- apply(sem.raneff.mcmc, 2, mean)
sem.raneff.sd <- apply(sem.raneff.mcmc, 2, sd)
```


Random effects (post. mean)
Random effects (post. s.d.)
INLA
30
20
10
0
−10
−20
−30
−30
−10
10
30
MCMC
INLA
7.0
6.5
6.0
5.5
5.0
4.5
4.0
5.0
6.0
7.0
MCMC


Random effect, obs. 1
Density
0.00
−20
20
60
N = 10000 Bandwidth = 0.764
Random effect, obs. 2
Density
0.00
−40
0
20
60
N = 10000 Bandwidth = 0.6276
Random effect, obs. 3
Density
0.00
−20
20
60
N = 10000 Bandwidth = 0.5542
Random effect, obs. 4
Density
0.00
−20
20
60
N = 10000 Bandwidth = 0.7745
Random effect, obs. 5
Density
0.00
0
20
60
100
N = 10000 Bandwidth = 0.5553
Random effect, obs. 6
Density
0.00
−20
20
60
N = 10000 Bandwidth = 0.5398
Random effect, obs. 7
Density
0.00
−60
−20
20
60
N = 10000 Bandwidth = 0.9458
Random effect, obs. 8
Density
0.00
−20
20
60
100
N = 10000 Bandwidth = 0.5481
Random effect, obs. 9
Density
0.00
−20
20
60
100
N = 10000 Bandwidth = 0.5459

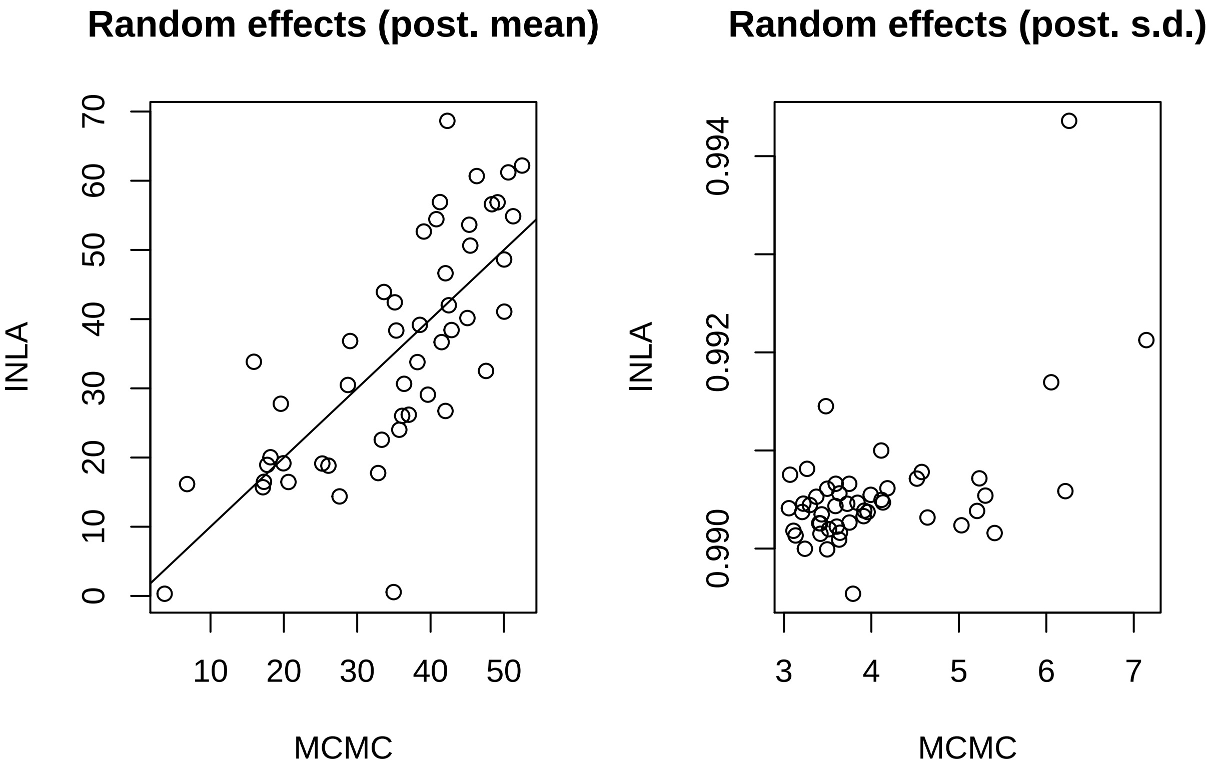
Spatial Lag Model (SLM)

Effect of the covariates

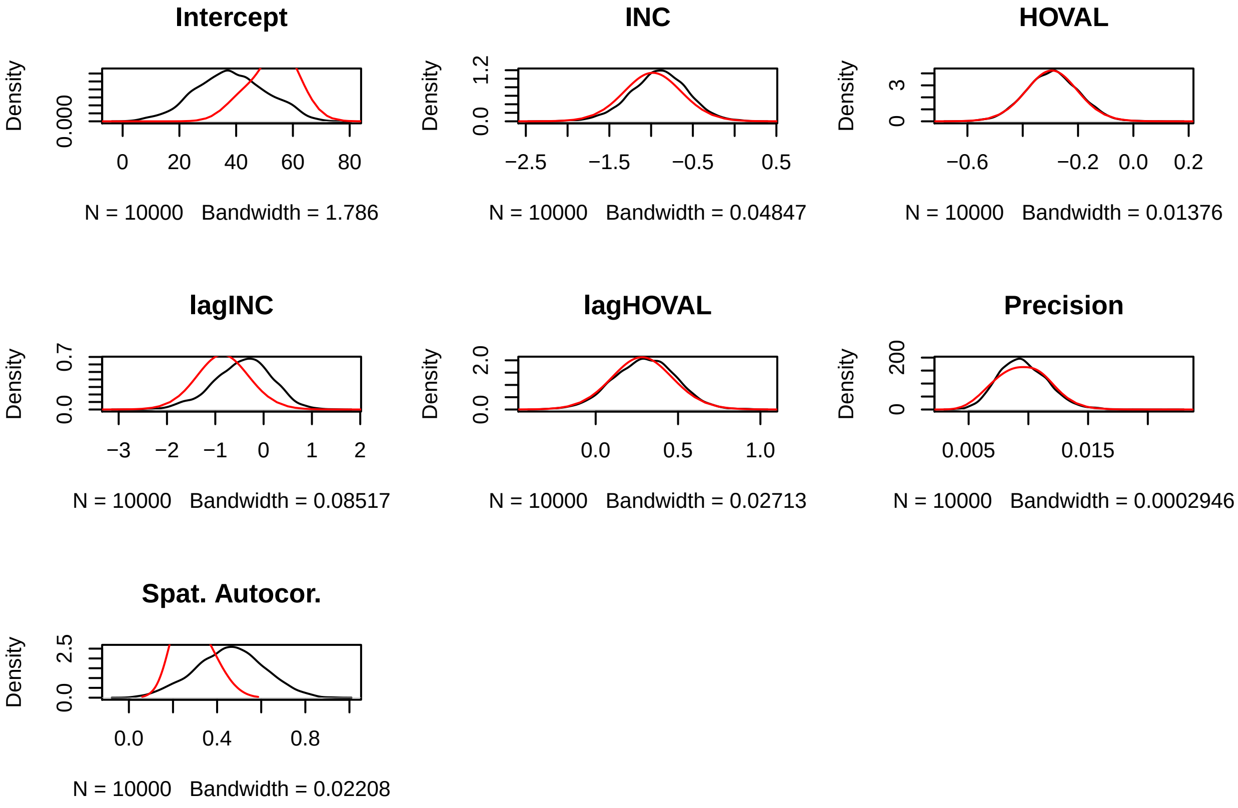


Estimate of the linear predictor

```r
slm.raneff.mcmc <- apply(slm.mcmc$mu, 2, function(X) {columbus$CRIME - X})
slm.raneff.mcmc <- matrix(slm.raneff.mcmc, ncol = 49, byrow = TRUE)

slm.raneff.mean <- apply(slm.raneff.mcmc, 2, mean)
slm.raneff.sd <- apply(slm.raneff.mcmc, 2, sd)
```



Spatial Durbin Model (SDM)



Discussion