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
```

```r
#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:

```r
#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
```

```r
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)
)
```

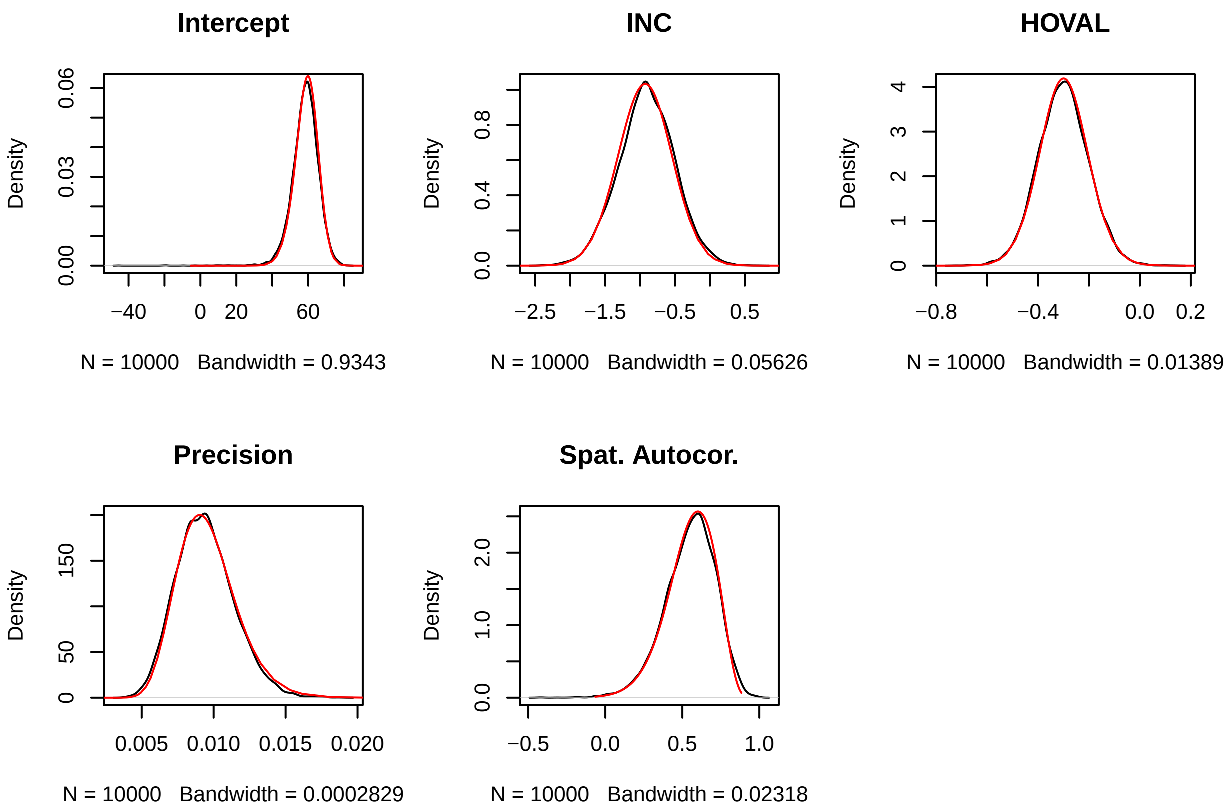
Results

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]])
```

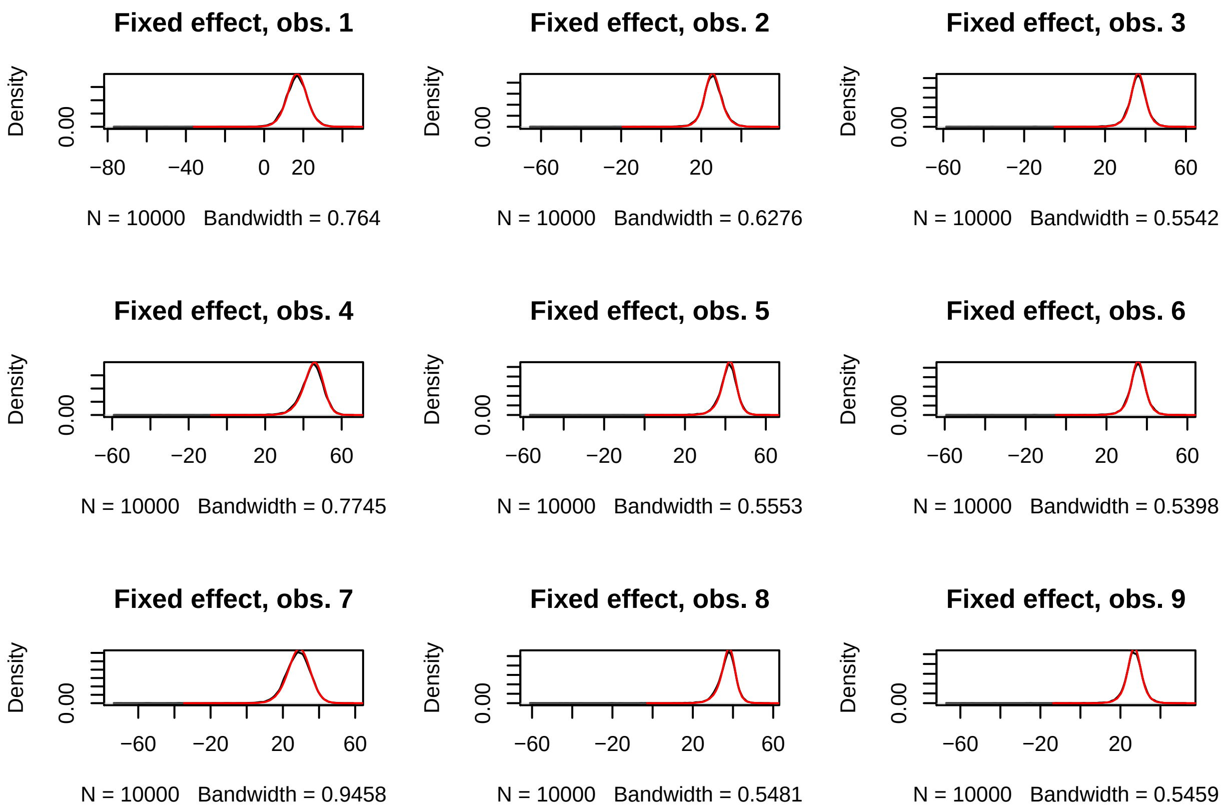
Spatial Error Model (SEM)

The following plot displays the posterior marginals of all the model parameters. As it can be seen, the agreement between INLA and MCMC is quite high.



Effect of the covariates

The next plots display the estimated effect of the covariates alone to see if there is agreement. Instead of the fitted values reported by INLA, we have used the fitted values computed using a linear combination on the values of the covariates using the fixed effects parameters as coefficients.



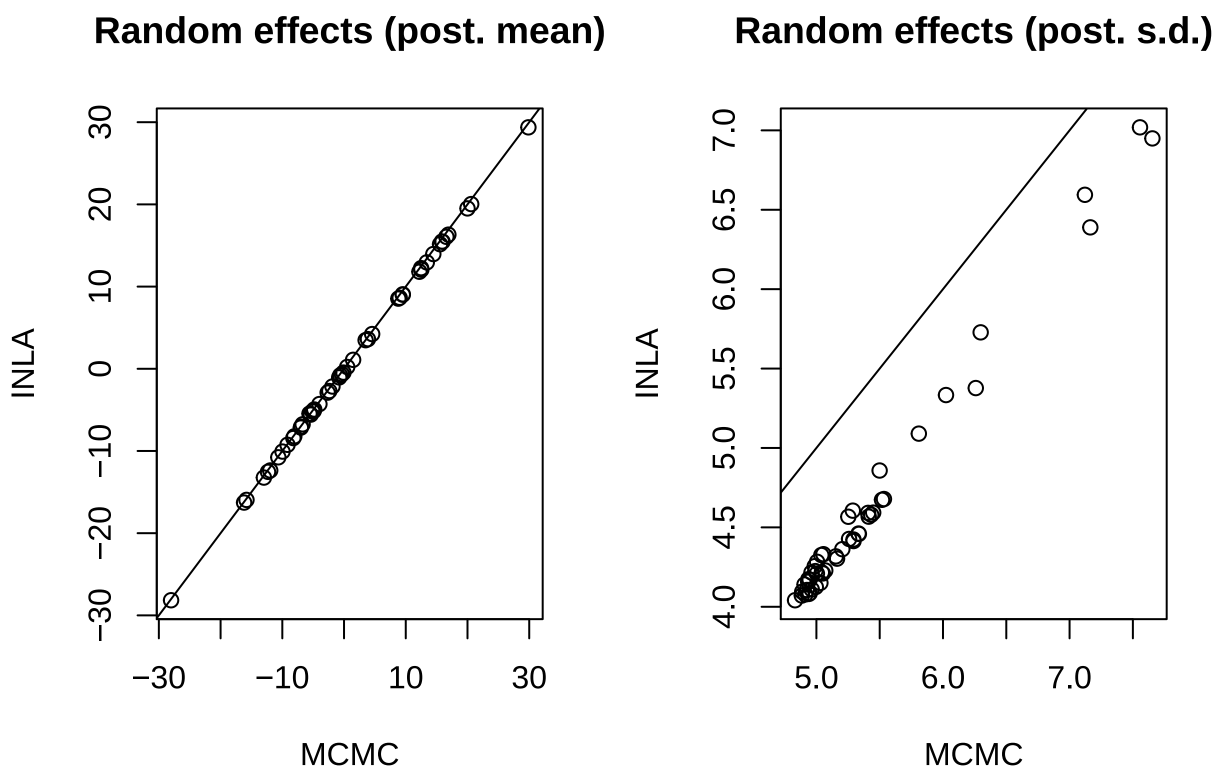
Estimates of the random effects

We compare now the estimates of the random effects by INLA and MCMC. For MCMC, the values of the random effects is computed by taking the difference between the observed value and the fixed effects. Then, the posterior mean and standard deviations are computed. For INLA, we have used the values reported.

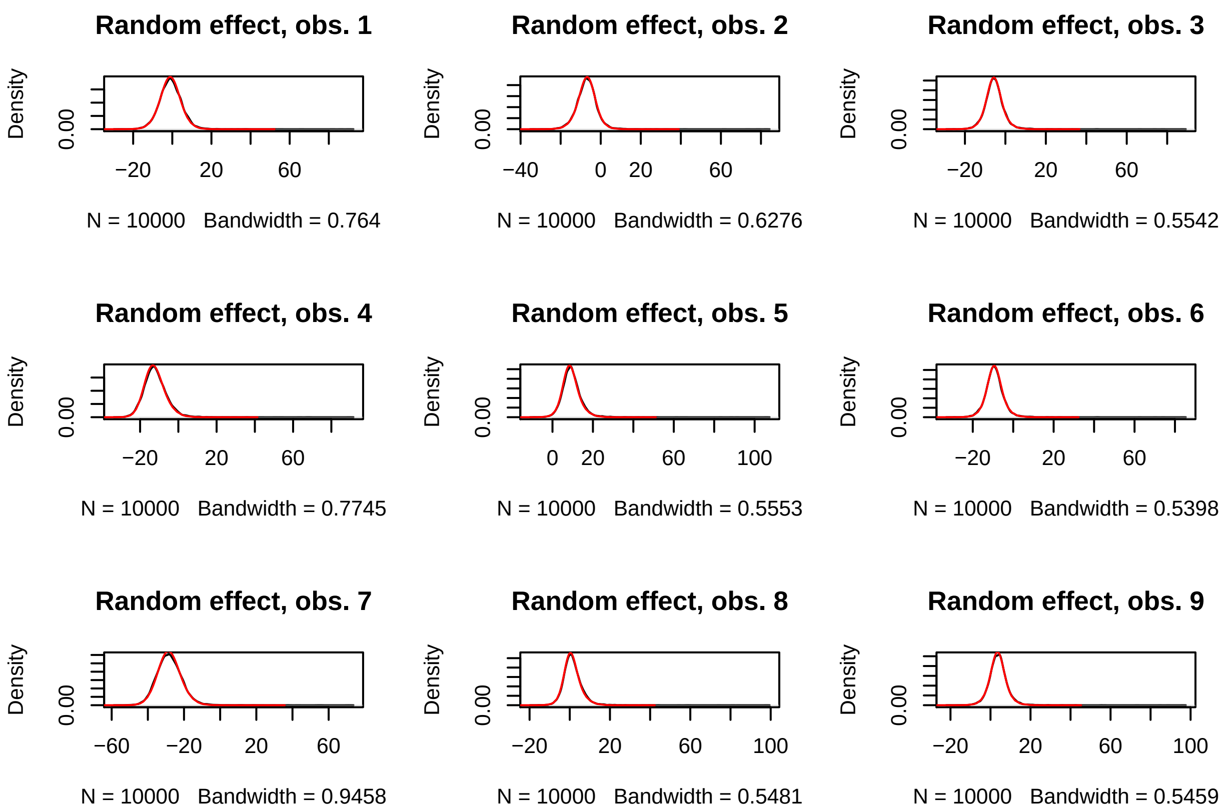
```
sem.raneff.mcmc <- apply(sem.mcmc$mu, 2, function(X) {columbus$CRIME - X})
sem.raneff.mcmc <- matrix(sem.raneff.mcmc, ncol = 49, byrow = TRUE)

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

It seems as if the standard deviations reported by MCMC are smaller than the ones reported by INLA but both provide very similar estimates of the posterior means.

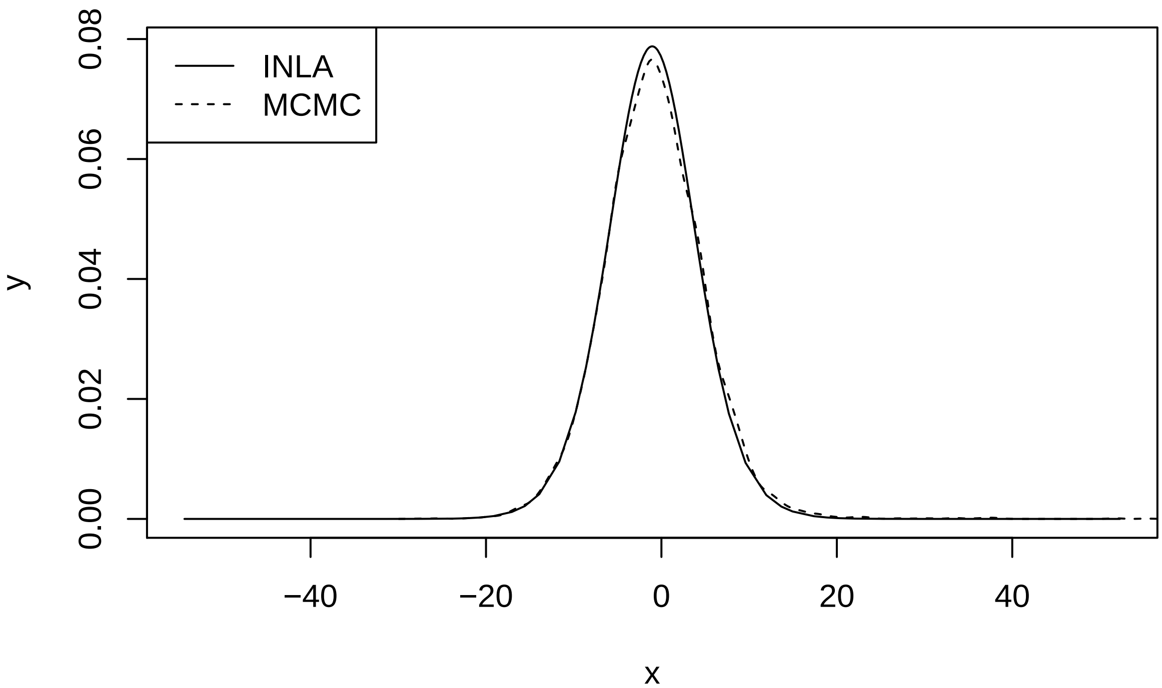


HOWEVER, the marginals seem to agree. The MCMC output seems to have different estimates of the s.d. as comparted to the INLA output, BUT the marginals agree!!!! :O



Checking observation 1

Here we develop a bit more on the differences between INLA and MCMC using observation 1. The next Figure shows the marginal of the random effect with INLA and MCMC:



Now, we check the posterior means and variances:

```
#As reported by INLA
sem.inla$summary.random$idx[1,]
```

```
##   ID      mean       sd 0.025quant  0.5quant 0.975quant     mode
## 1  1 -1.049073 5.333231  -11.64157 -1.054999   9.589537 -1.05113
##            kld
## 1 1.440997e-10
```

```
#As computed using the posterior marginal
inla.zmarginal(sem.inla$marginals.random$idx[[1]], FALSE)
```

```
## Mean            -1.04907
## Stdev           5.33294
## Quantile  0.025 -11.6643
## Quantile  0.25  -4.53728
## Quantile  0.5   -1.09652
## Quantile  0.75  2.34331
## Quantile  0.975 9.53888
```

```
#Using the MCMC output
mean(sem.raneff.mcmc[, 1])
```

```
## [1] -0.7850188
```

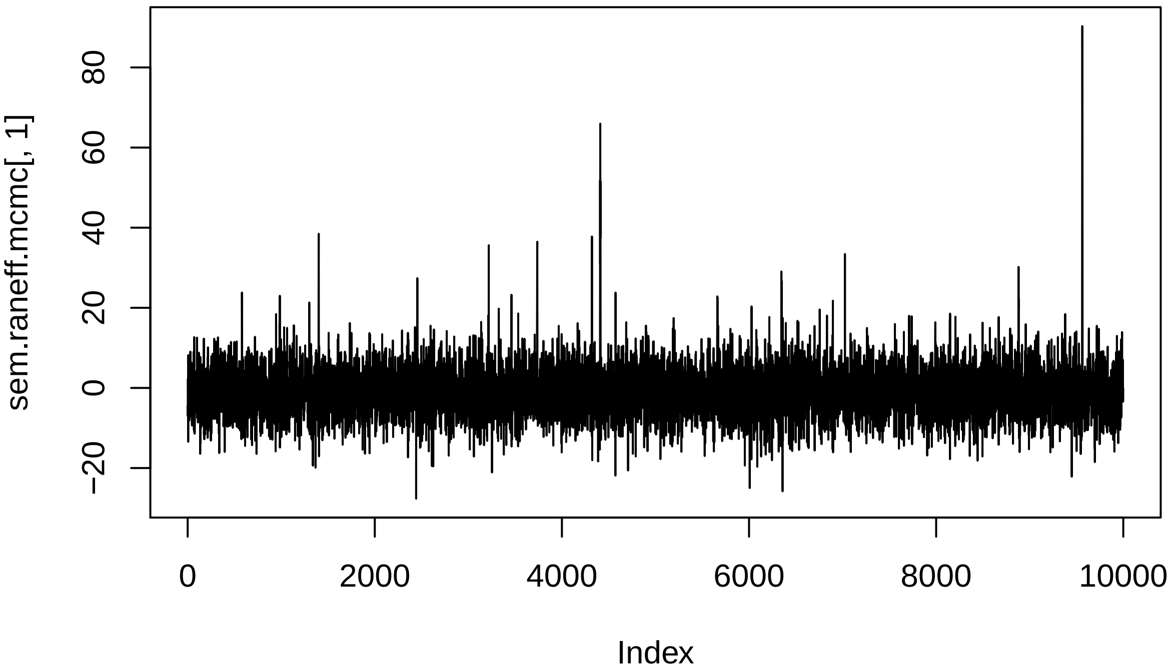
```
sd(sem.raneff.mcmc[, 1])
```

```
## [1] 6.023839
```

```r
quantile(sem.raneff.mcmc[, 1], c(0.025, 0.25, 0.5, 0.75, 0.975))
```

```
##        2.5%         25%         50%         75%       97.5%
## -11.8075878  -4.4975029  -0.9645656   2.6793699  10.5347752
```

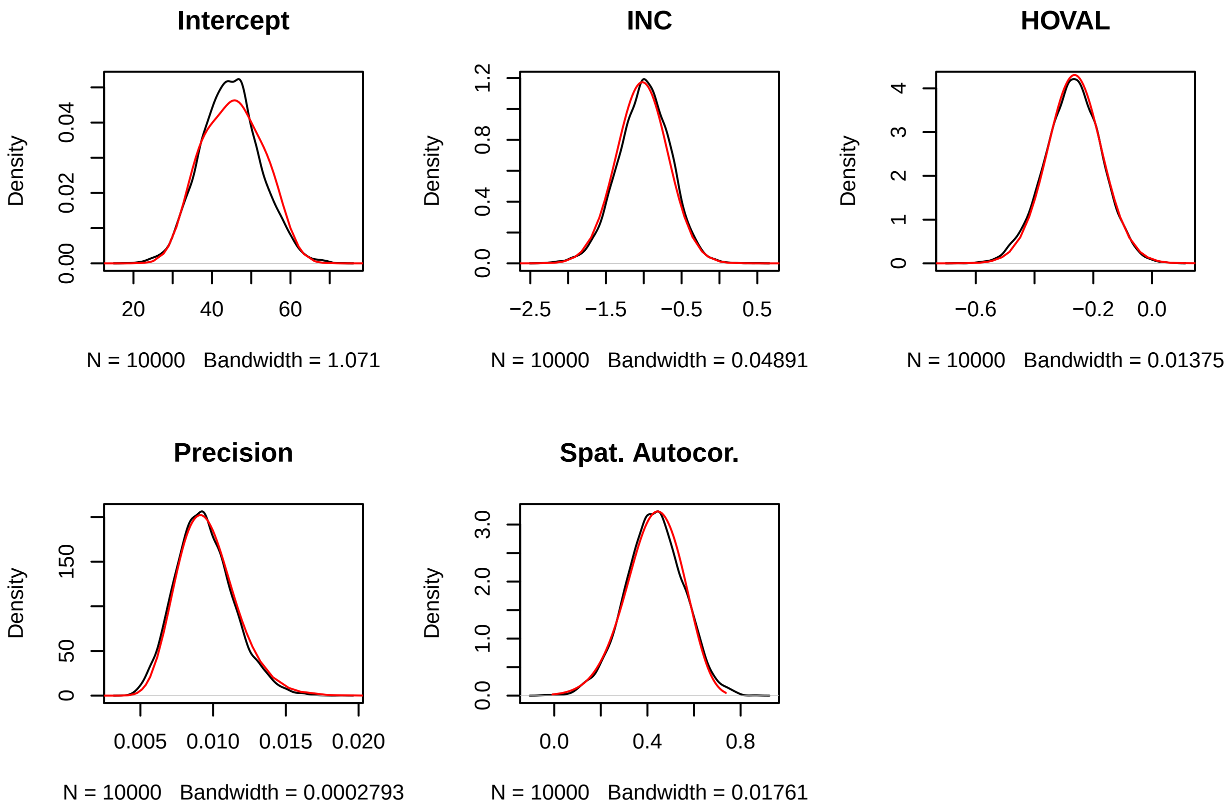
Quantiles look very similar, but the mean and variance of from the MCMC output look a bit larger. If we plot the MCMC output, we get:



There are a few samples that look larger than they should. These will not affect the density plots or quantiles but they have a an impact when computing the posterior mean and standard deviation.

Spatial Lag Model (SLM)

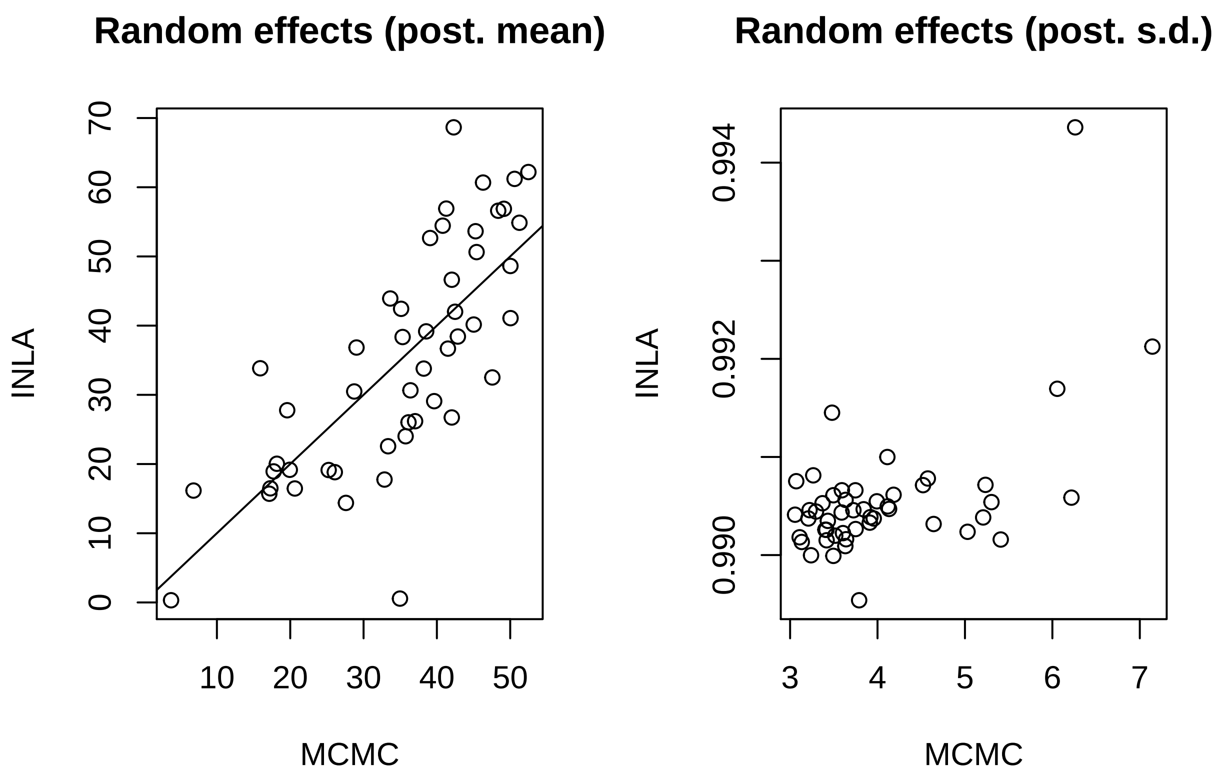
Effect of the covariates



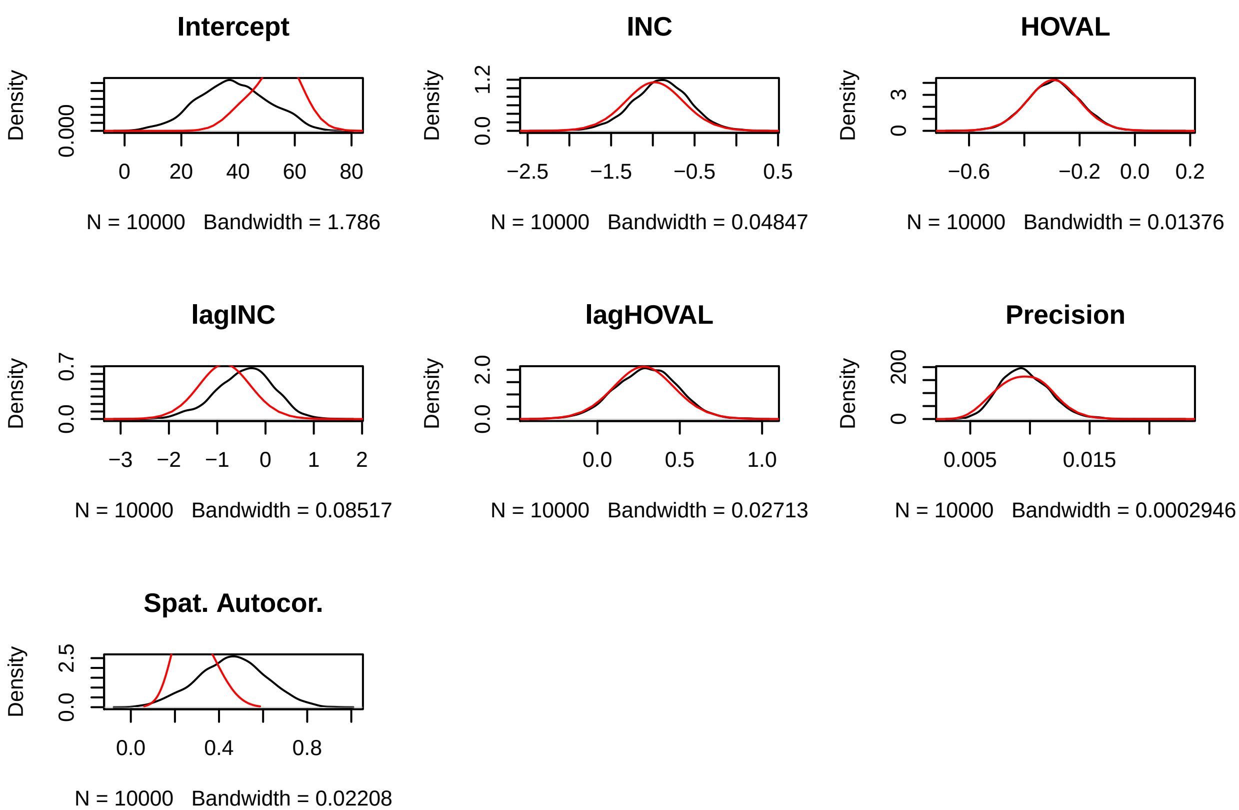
Estimate of the linear predictor

```
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