

# SpMeta: Spatial-Meta Analysis/Regression Modeling

## SpMeta\_Example

[1] Load neighborhood adjacency list:

```
url_path<-"https://raw.githubusercontent.com/warrenjl/SpMeta/main/SpMeta_Example/neighbors.RData"  
load(url(url_path))
```

[2] Simulate data from the proposed model:

- Setting the reproducibility seed and initializing packages for data simulation:

```
set.seed(2165)  
  
library(SpMeta)  
library(mnormt) #Multivariate normal distribution
```

- Setting the global data values:

```
n<-length(W) #Number of separated spatial regions  
m<-rep(NA, #Number of contiguous spatial units in each region  
      times = n)  
for(i in 1:n){  
  m[i]<-nrow(W[[i]])  
}
```

- Setting the values for the statistical model parameters and simulating data:

```
sigma2_true<-rgamma(n = n,  
                    shape = 10.00,  
                    rate = 100.00)  
  
beta_true<-c(1.00,  
             -2.00)  
  
tau2_true<-rgamma(n = n,  
                  shape = 10.00,  
                  rate = 10.00)  
  
rho_true<-runif(n = n)  
  
phi_true<-list(0)  
theta_true<-list(0)  
theta_hat<-list(0)  
X<-list(0)  
Q<-list(0)  
SE<-list(0)  
for(i in 1:n){  
  
  X[[i]]<-matrix(1,  
                 nrow = m[i],  
                 ncol = 2)  
  X[[i]][,2]<-rnorm(n = m[i])  
  
  Q[[i]]<-rho_true[i]*(diag(rowSums(W[[i]])) - W[[i]]) +  
    (1.00 - rho_true[i])*diag(m[i])  
  
  phi_true[[i]]<-rmnorm(n = 1,  
                        mean = 0.00,  
                        varcov = (tau2_true[i]*chol2inv(chol(Q[[i]]))))
```

```

phi_true[[i]]<-phi_true[[i]] -
  mean(phi_true[[i]])

theta_true[[i]]<-rmnorm(n = 1,
  mean = (X[[i]]%*%beta_true + phi_true[[i]]),
  varcov = (sigma2_true[i]*diag(m[i])))
SE[[i]]<-rgamma(n = m[i],
  shape = 1.00,
  rate = 10.00)
theta_hat[[i]]<-rnorm(n = m[i],
  mean = theta_true[[i]],
  sd = SE[[i]])

}

```

[2] Fit SpMeta:

```

results<-SpMeta(mcmc_samples = 11000,
  theta_hat = theta_hat,
  se = SE,
  x = X,
  model_indicator = 2,
  neighbors = W,
  metrop_var_rho_trans = rep(3.00, time = n))

```

```

## Progress: 10%
## rho Acceptance (min): 37%
## rho Acceptance (max): 55%
## *****
## Progress: 20%
## rho Acceptance (min): 36%
## rho Acceptance (max): 54%
## *****
## Progress: 30%
## rho Acceptance (min): 35%
## rho Acceptance (max): 53%
## *****
## Progress: 40%
## rho Acceptance (min): 34%
## rho Acceptance (max): 53%
## *****
## Progress: 50%
## rho Acceptance (min): 34%
## rho Acceptance (max): 53%
## *****
## Progress: 60%
## rho Acceptance (min): 33%
## rho Acceptance (max): 53%
## *****
## Progress: 70%
## rho Acceptance (min): 34%
## rho Acceptance (max): 53%
## *****
## Progress: 80%

```

```
## rho Acceptance (min): 34%
## rho Acceptance (max): 54%
## *****
## Progress: 90%
## rho Acceptance (min): 34%
## rho Acceptance (max): 54%
## *****
## Progress: 100%
## rho Acceptance (min): 34%
## rho Acceptance (max): 54%
## *****
```

[3] Analyzing Output:

```
par(mfrow=c(2,2))
plot(results$beta[1, 1001:11000],
     type="l",
     ylab="beta",
     xlab="Sample")
abline(h=beta_true[1],
       col="red",
       lwd=2) #True value
plot(results$beta[2, 1001:11000],
     type="l",
     ylab="beta",
     xlab="Sample")
abline(h=beta_true[2],
       col="red",
       lwd=2) #True value

phi_samps<-matrix(NA,
                  nrow = 11000,
                  ncol = sum(m))
for(j in 1:11000){
  phi_samps[j,]<-unlist(results$phi[[j]])
}
plot(colMeans(phi_samps[1001:11000,]),
     unlist(phi_true),
     ylab = "phi_true",
     xlab = "phi_est")
abline(0,1,
       col = "red")

theta_true_samps<-matrix(NA,
                        nrow = 11000,
                        ncol = sum(m))
for(j in 1:11000){
  theta_true_samps[j,]<-unlist(results$theta_true[[j]])
}
plot(colMeans(theta_true_samps[1001:11000,]),
     unlist(theta_true),
     ylab = "theta_true_true",
     xlab = "theta_true_est")
abline(0,1,
       col = "red")
```

