

Tarea 12

Dalia Camacho

12-MCMC convergencia

Implementaremos un modelo de regresión en JAGS, la base de datos que usaremos contiene información de mediciones de radón (activity) y del suelo en el que se hicieron las mediciones (floor = 0 casas con sótano, floor = 1 casas sin sótano), las mediciones corresponden a 919 hogares muestreados de 85 condados de Minnesota. El objetivo es construir un modelo de regresión en el que la medición de radón es la variable dependiente y el tipo de suelo es la covariable.

El modelo es como sigue:

$$y_i \sim N(\alpha + \beta x_i, \sigma^2)$$

La distribuciones iniciales que usaremos son:

$$\beta \sim N(0, 1000)$$

$$\sigma^2 \sim U(0, 1000)$$

```
set.seed(65445)
library(R2jags)
modelo_regresion.txt <-
'
  model{
    for(i in 1 : n) {
      y[i] ~ dnorm(y.hat[i], tau.y)
      y.hat[i] <- a + b * x[i]
    }
    a ~ dnorm(0, 0.001)
    b ~ dnorm(0, 0.001)
    tau.y <- pow(sigma.y, -2)
    sigma.y ~ dunif(0, 100)
  }
'

cat(modelo_regresion.txt, file = 'modelo_regresion.bugs')
# cargamos los datos con load radon
radon <- readr::read_csv("radon.csv")
# Iniciamos preparando los datos para el análisis, trabajaremos en
# escala logarítmica, hay algunos casos con medición cero, para éstos
# hacemos una pequeña corrección redondeándolos a 0.1.
y <- log(ifelse (radon$activity == 0, 0.1, radon$activity))
n <- nrow(radon)
x <- radon$floor
# jags
radon1_data      <- list("n", "y", "x")
radon1_parameters <- c("a", "b", "sigma.y")
```

El ejercicio consiste en que utilices la función `jags()` definiendo valores iniciales, número de cadenas, número de iteraciones y etapa de calentamiento. Asegurate de alcanzar convergencia y describe los diagnósticos que utilizaste para concluir que se convergió a la distribución posterior.

Para los valores iniciales de α y β consideramos los coeficientes de regresión lineal de una muestra de los datos de piso y actividad. Para el valor σ^2 tomamos una uniforme entre 0 y 100.

```
# Función de iniciales
jags_inits <- function(){
  sample_guide <- sample(1:n, replace = TRUE)
  model <- lm(y[sample_guide]~x[sample_guide])
  Coefs <- coef(model)
  return(list(a = rnorm(1,Coefs[1], 5),
             b = rnorm(1, Coefs[2]),
             sigma.y= runif(1,0,100)))
}
```

Corremos el modelo para 2000 iteraciones con un burnin de 200 para 5 cadenas

```
jags_fit <- jags(
  model.file = "modelo_regresion.bugs",    # modelo de JAGS
  inits = jags_inits,    # valores iniciales
  data = list(n=nrow(radon), y=radon$activity, x=radon$floor),    # lista con los datos
  parameters.to.save = c("a", "b", "sigma.y"),    # parámetros por guardar
  n.chains = 5,    # número de cadenas
  n.iter = 2000,    # número de pasos
  n.burnin = 200    # calentamiento de la cadena
)
```

```
## module glm loaded

## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 919
##   Unobserved stochastic nodes: 3
##   Total graph size: 1852
##
## Initializing model
```

El resultado del muestreo de Gibbs da como resultado los siguientes parámetros:

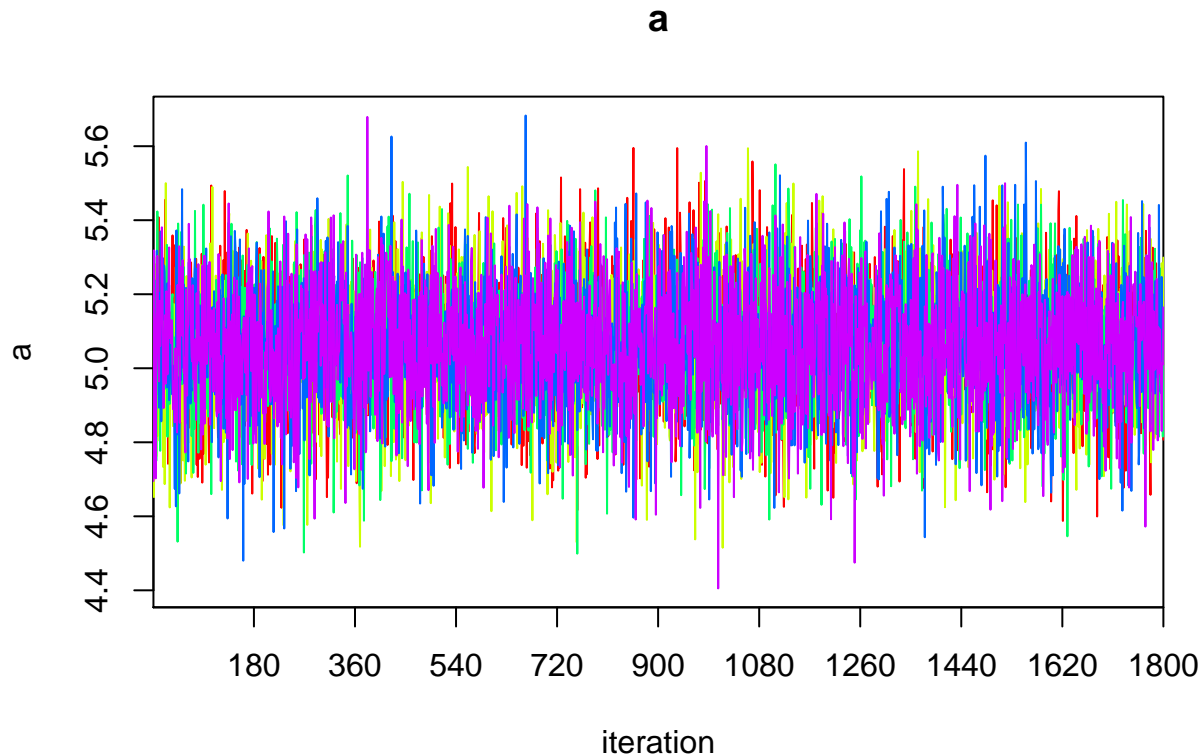
```
jags_fit

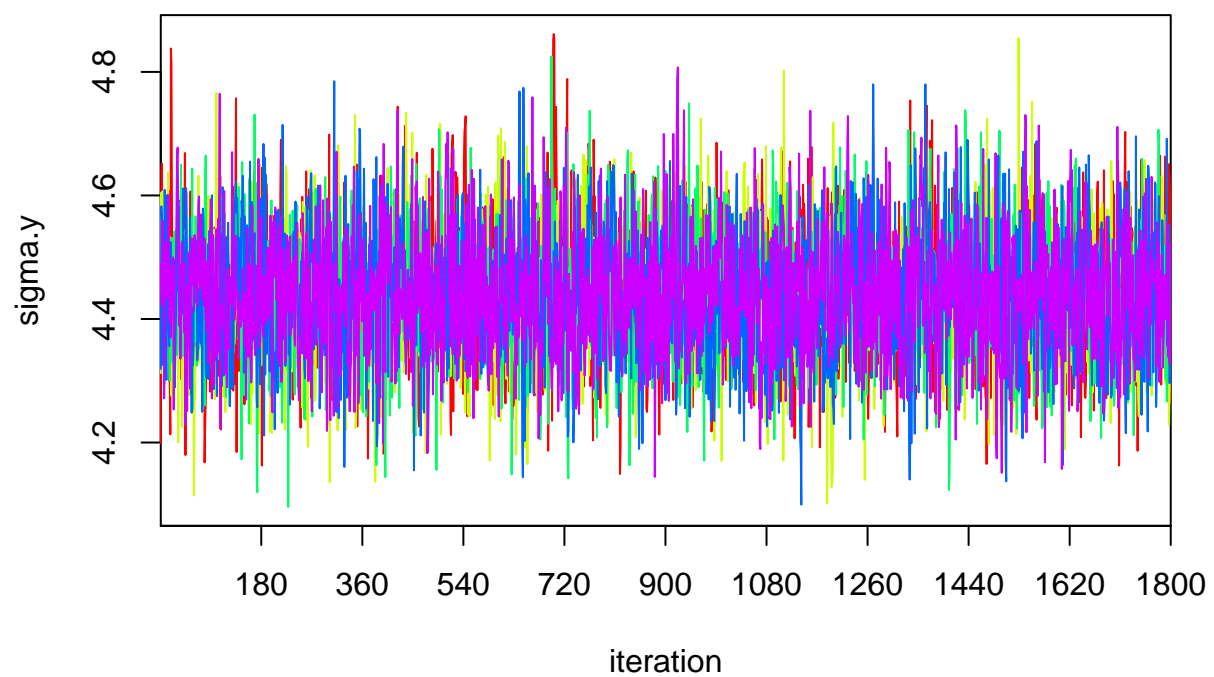
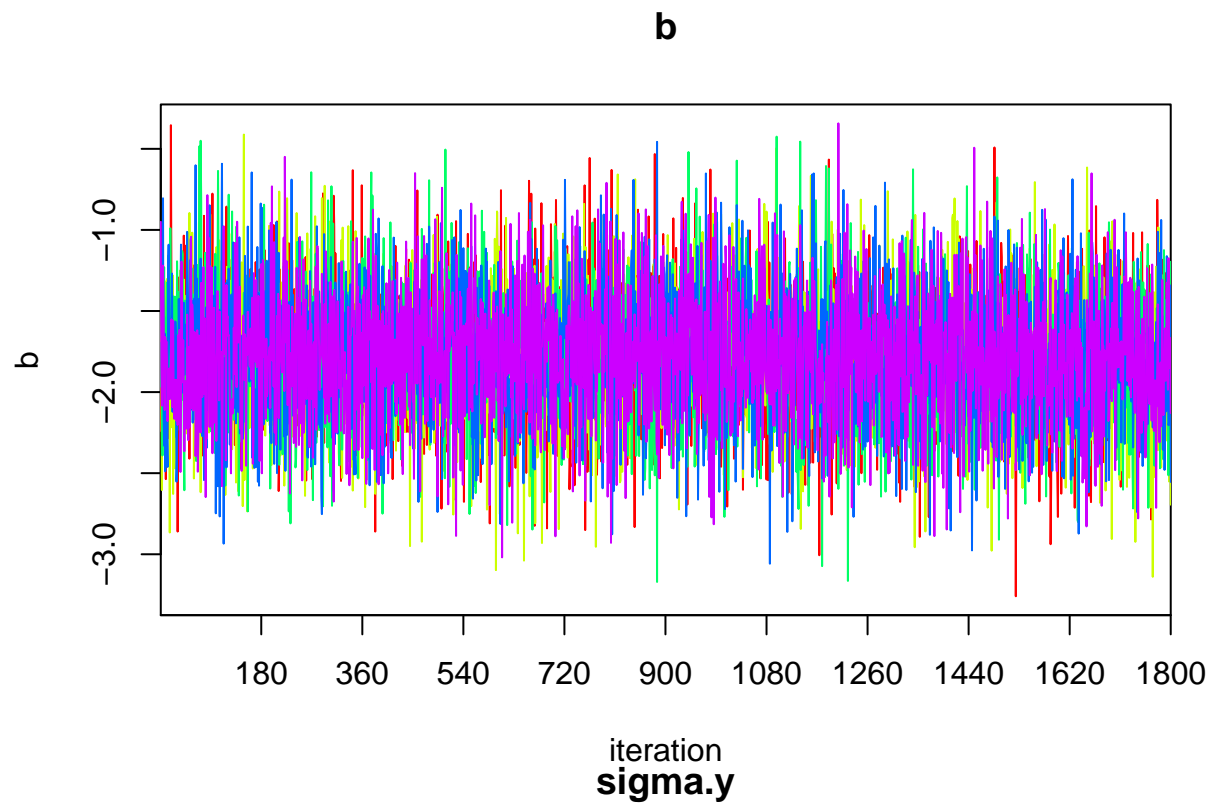
## Inference for Bugs model at "modelo_regresion.bugs", fit using jags,
## 5 chains, each with 2000 iterations (first 200 discarded)
## n.sims = 9000 iterations saved
##      mu.vect sd.vect   2.5%    25%    50%    75%   97.5%
## a         5.062   0.162   4.737   4.957   5.063   5.170   5.378
## b        -1.785   0.398  -2.558  -2.048  -1.790  -1.516  -1.000
## sigma.y    4.438   0.104   4.244   4.366   4.435   4.506   4.645
## deviance 5346.464   2.529 5343.642 5344.630 5345.780 5347.624 5352.948
##      Rhat n.eff
## a         1.001  9000
## b         1.001  9000
## sigma.y   1.001  9000
## deviance  1.001  9000
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
```

```
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 3.2 and DIC = 5349.7
## DIC is an estimate of expected predictive error (lower deviance is better).
```

Si graficamos los valores de α , β y σ podemos observar que las cuatro cadenas oscilan en valores cercanos y que no se observa que el comportamiento se modifique a través de las iteraciones más allá de ruido. Por lo que podemos concluir que el modelo convergió a la distribución original de los parámetros y los valores dados por el modelo son congruentes.

```
traceplot(jags_fit, varname = c("a", "b", "sigma.y"))
```





Instalar Stan y rstan, instrucciones aquí.