Election

## 1. Load Packages

library(readr)  
library(ggplot2)  
library(gridExtra) # For multiple plots  
library(dplyr)  
library(tidyr)

## 2. Read the dataset

set.seed(1117)  
  
election\_data <- read.csv("census\_parlimen.csv")  
  
# N is the number of voters  
# Y is the number of voters who support the government  
# Notable Score is where that seat is contested by a famous politician  
election\_data$seat <- as.integer(election\_data$seat)  
election\_data$region <- as.factor(election\_data$region)  
election\_data$N <- as.integer(election\_data$N)  
election\_data$Y <- as.integer(election\_data$Y)  
election\_data$malay\_prop <- as.numeric(election\_data$malay\_prop)  
election\_data$indian\_prop <- as.numeric(election\_data$indian\_prop)  
election\_data$chinese\_prop <- as.numeric(election\_data$chinese\_prop)  
election\_data$notable\_score <- as.integer(election\_data$notable\_score)  
  
# Extract number of seats and regions  
C <- nrow(election\_data)  
R <- length(unique(election\_data$region))  
  
print(C)

[1] 222

print(R)

[1] 13

head(election\_data)

seat region state parlimen code\_parlimen N Y prop  
1 1 1 Perlis P.001 Padang Besar P.001 35520 21473 0.6045327  
2 2 1 Perlis P.002 Kangar P.002 42649 23343 0.5473282  
3 3 1 Perlis P.003 Arau P.003 37787 19376 0.5127689  
4 4 2 Kedah P.004 Langkawi P.004 31133 21407 0.6875984  
5 5 2 Kedah P.005 Jerlun P.005 45052 24161 0.5362914  
6 6 2 Kedah P.006 Kubang Pasu P.006 56224 33334 0.5928785  
 previous\_result malay\_prop chinese\_prop indian\_prop notable\_score pred\_prob  
1 1 0.8601 0.0814 0.0088 1 0.6031824  
2 1 0.8158 0.1473 0.0157 0 0.5445903  
3 1 0.8723 0.0781 0.0159 0 0.5120035  
4 1 0.8965 0.0675 0.0238 0 0.5318187  
5 1 0.9075 0.0678 0.0011 0 0.5382606  
6 1 0.8599 0.0872 0.0338 1 0.6075472  
 pred\_votes pred\_win  
1 21419 1  
2 23310 1  
3 19531 1  
4 16559 1  
5 24261 1  
6 34558 1

## 3. Define Logistic Regression Log-Posterior

Assuming all priors and hyperpriors are normally distributed:

$$
p(\eta \mid y) \propto p(y \mid \eta = \beta\_0, \beta\_{Ethnic,Region},\beta\_{Notable}) \cdot p(\beta\_{Ethnic,Region} \mid \mu\_{Ethnic},\tau\_{Ethnic}) \cdot p(\mu\_{Ethnic}) \\ \cdot p(\beta\_0) \cdot p(\beta\_{Notable})
\cdot p(\tau\_{Ethnic})
\\
\ln p(\eta \mid y) \propto \ln p(y \mid \eta = \beta\_0, \beta\_{Ethnic,Region},\beta\_{Notable}) + \ln p(\beta\_{Ethnic,Region} \mid \mu\_{Ethnic},\tau\_{Ethnic}) \\ + \ln p(\mu\_{Ethnic}) + \ln p(\beta\_0) + \ln p(\beta\_{Notable}) +
\ln p(\tau\_{Ethnic})
$$

The model distribution (Logistic Regression with Logit Link):

$$
f\_i = p\_i^{Y\_i} \cdot (1-p\_i)^{N\_i - Y\_i} \\
\ell\_i = Y\_i \cdot ln(p\_i) + (N\_i - Y\_i) \cdot ln(1 - p\_i) \\
\ell\_i =Y\_i \cdot \ln\!\left(\frac{e^{\eta\_i}}{1 + e^{\eta\_i}}\right) +
(N\_i - Y\_i) \cdot \ln\!\left(\frac{1}{1 + e^{\eta\_i}}\right) \\
\ell\_i =
Y\_i \cdot \left[-\ln\!\left(1 + e^{-\eta\_i}\right)\right] +
\left(N\_i - Y\_i\right) \cdot \left[-\ln\!\left(1 + e^{\eta\_i}\right)\right]
$$

The log-prior for intercept and beta\_notable (Normally distributed with mean 0 and standard deviation 2):

$$
p(\beta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\!\left(-\frac{(\beta-\mu)^2}{2\sigma^2}\right) \\
\text{Zero mean and dropping multiplicative constant}(\mu=0): \\
p(\beta) = \exp\!\left(-\frac{\beta^2}{2\sigma^2}\right) \\
\ln p(\beta) = - \!\left(\frac{\beta^2}{2\sigma^2}\right) \\
$$

The log-prior for region and ethnic-specific slope:

$$
p(\beta\_{Ethnic,Region\_i}) = \frac{1}{\sqrt{2\pi\sigma^2}}
\exp\!\left(-\frac{(\beta\_{Ethnic,Region\_i}-\mu\_{Ethnic})^2}{2\tau\_{Ethnic}^2}\right) \\
\ln p(\beta\_{Ethnic,Region\_i}) = - \!\left(\frac{(\beta\_{Ethnic,Region\_i}
- \mu\_{Ethnic})^2}{2\tau\_{Ethnic}^2}\right) \\
$$

The hyperprior of mu ethnic is normal distribution with mean 0 and standard deviation of 5:

$$
\ln p(\mu\_{Ethnic}) = - \!\left(\frac{\mu\_{ethnic}^2}{2\sigma^2}\right) \\
$$

The hyperprior of tau follows a half-cauchy distribution with scale parameter 2:

log\_posterior\_hier <- function(beta0, beta\_notable,  
 beta\_malay\_r, beta\_indian\_r, beta\_chinese\_r,  
 mu\_malay, mu\_indian, mu\_chinese, election\_data,  
 prior\_sd\_beta0 = 2,  
 prior\_sd\_notable = 2,  
 scale, tau\_malay, tau\_indian, tau\_chinese,  
 prior\_sd\_mu = 5) {  
   
 # The number of voters, the number of supporters of the BN goverment and ethnicity proportion of 222 constituencies  
 N\_vec <- election\_data$N  
 Y\_vec <- election\_data$Y  
 x\_malay <- election\_data$malay\_prop  
 x\_indian <- election\_data$indian\_prop  
 x\_chinese <- election\_data$chinese\_prop  
 x\_notable <- election\_data$notable\_score  
 region\_id <- as.integer(election\_data$region)  
   
 # 1) Log-likelihood of the model distribution   
 # eta is the systematic component  
 eta <- numeric(length(Y\_vec))  
 for (i in seq\_along((Y\_vec))) {  
 r <- region\_id[i]  
 eta[i] <- beta0 + beta\_notable \* x\_notable[i] +  
 beta\_malay\_r[r] \* x\_malay[i] +  
 beta\_indian\_r[r] \* x\_indian[i] +  
 beta\_chinese\_r[r] \* x\_chinese[i]  
 }  
   
 # Individual Log-likelihood function for a single data point  
 # log1p(x) = ln(1+x)  
 loglik\_i <- Y\_vec \* (-log1p(exp(-eta))) +  
 (N\_vec - Y\_vec) \* (-log1p(exp(eta)))  
   
 # All data points  
 loglik <- sum(loglik\_i)  
   
 # 2) Priors on β0 and β\_notable  
 logprior\_beta0 <- - (beta0^2) / (2 \* prior\_sd\_beta0^2)  
 logprior\_notable <- - (beta\_notable^2) / (2 \* prior\_sd\_notable^2)  
   
 # 3) Priors on region-specific slopes  
 logprior\_malay\_r <- - sum((beta\_malay\_r - mu\_malay)^2) / (2 \* tau\_malay^2)  
 logprior\_indian\_r <- - sum((beta\_indian\_r - mu\_indian)^2) / (2 \* tau\_indian^2)  
 logprior\_chinese\_r <- - sum((beta\_chinese\_r - mu\_chinese)^2) /   
 (2 \* tau\_chinese^2)  
   
 # 4) Hyperprior of prior mean  
 logprior\_mu\_malay <- - (mu\_malay^2) / (2 \* prior\_sd\_mu^2)  
 logprior\_mu\_indian <- - (mu\_indian^2) / (2 \* prior\_sd\_mu^2)  
 logprior\_mu\_chinese <- - (mu\_chinese^2) / (2 \* prior\_sd\_mu^2)  
   
 # 5) Hyperprior of tau\_ethnic  
 if (tau\_malay <= 0) {  
 logprior\_tau\_malay <- -Inf  
 } else {  
 logprior\_tau\_malay <- log(2) - log(pi \* scale \* (1 + (tau\_malay/scale)^2))  
 }  
   
 if (tau\_indian <= 0) {  
 logprior\_tau\_indian <- -Inf  
 } else {  
 logprior\_tau\_indian <- log(2) - log(pi \* scale \* (1 + (tau\_indian/scale)^2))  
 }  
   
 if (tau\_chinese <= 0) {  
 logprior\_tau\_chinese <- -Inf  
 } else {  
 logprior\_tau\_chinese <- log(2) - log(pi \* scale \* (1 + (tau\_chinese/scale)^2))  
 }  
   
 # Combine all components  
 return(loglik +  
 logprior\_beta0 + logprior\_notable +  
 logprior\_malay\_r + logprior\_indian\_r + logprior\_chinese\_r +  
 logprior\_mu\_malay + logprior\_mu\_indian + logprior\_mu\_chinese +  
 logprior\_tau\_malay + logprior\_tau\_indian + logprior\_tau\_chinese)  
}

## 4. Metropolis-Hasting Algorithm

We will use Normal Distribution as the proposal distribution.

set.seed(1)  
  
n\_iter <- 2500  
burn\_in <- 1500  
prior\_sd\_beta0 <- 2  
prior\_sd\_notable<- 2  
prior\_sd\_mu <- 5  
scale <- 2  
  
# This will determine the strength of random jump  
sigma\_beta0 <- 0.01  
sigma\_mu <- 0.30  
sigma\_region <- 0.01  
sigma\_tau <- 0.30  
  
R <- length(unique(election\_data$region))  
chain <- matrix(0, nrow = n\_iter, ncol = 2 + 3\*R + 3 + 3)  
colnames(chain) <- c(  
 "beta0", "beta\_notable",  
 paste0("beta\_malay\_r", 1:R),  
 paste0("beta\_indian\_r", 1:R),  
 paste0("beta\_chinese\_r", 1:R),  
 "mu\_malay", "mu\_indian", "mu\_chinese",  
 "tau\_malay", "tau\_indian", "tau\_chinese"  
)  
  
# Initial Starting Values  
chain[1, ] <- c(0, 0, rep(0, 3\*R), 0, 0, 0, 1, 1 ,1)  
  
# Calculate log-likelihood at initial point, we will update this after new parameter are accepted  
initial\_beta0 <- chain[1, "beta0"]  
initial\_beta\_notable <- chain[1, "beta\_notable"]  
initial\_beta\_malay\_r <- chain[1, paste0("beta\_malay\_r", 1:R)]  
initial\_beta\_indian\_r <- chain[1, paste0("beta\_indian\_r", 1:R)]  
initial\_beta\_chinese\_r <- chain[1, paste0("beta\_chinese\_r", 1:R)]  
initial\_mu\_malay <- chain[1, "mu\_malay"]  
initial\_mu\_indian <- chain[1, "mu\_indian"]  
initial\_mu\_chinese <- chain[1, "mu\_chinese"]  
initial\_tau\_malay <- chain[1, "tau\_malay"]  
initial\_tau\_chinese <- chain[1, "tau\_chinese"]  
initial\_tau\_indian <- chain[1, "tau\_indian"]  
  
logpost\_curr <- log\_posterior\_hier(  
 beta0 = initial\_beta0,  
 beta\_notable = initial\_beta\_notable,  
 beta\_malay\_r = initial\_beta\_malay\_r,  
 beta\_indian\_r = initial\_beta\_indian\_r,  
 beta\_chinese\_r = initial\_beta\_chinese\_r,  
 mu\_malay = initial\_mu\_malay,  
 mu\_indian = initial\_mu\_indian,  
 mu\_chinese = initial\_mu\_chinese,  
 election\_data = election\_data,  
 prior\_sd\_beta0 = prior\_sd\_beta0,  
 prior\_sd\_notable = prior\_sd\_notable,  
 scale = scale,  
 tau\_malay = initial\_tau\_malay,  
 tau\_chinese = initial\_tau\_chinese,  
 tau\_indian = initial\_tau\_indian,  
 prior\_sd\_mu = prior\_sd\_mu  
)  
  
# Track Acceptance  
move\_beta0 <- logical(n\_iter) # for block 1  
move\_region <- array(FALSE, dim = c(n\_iter, 3, R)) # for block 2  
move\_mu <- logical(n\_iter) # for block 3  
move\_tau <- logical(n\_iter) # for block 4  
  
# Main Loop  
for (t in 2:n\_iter) {  
 # Block 1: Update Beta 0 and Beta Notable  
 # The original value + The random jump  
 prev\_beta0 <- chain[t - 1, "beta0"]  
 prev\_beta\_notable <- chain[t - 1, "beta\_notable"]  
 prop2 <- c(prev\_beta0, prev\_beta\_notable) + rnorm(2, mean = 0, sd = sigma\_beta0)  
 prop\_beta0 <- prop2[1]  
 prop\_beta\_notable <- prop2[2]  
   
 prev\_beta\_malay\_r <- as.numeric(chain[t - 1, paste0("beta\_malay\_r", 1:R)])  
 prev\_beta\_indian\_r <- as.numeric(chain[t - 1, paste0("beta\_indian\_r", 1:R)])  
 prev\_beta\_chinese\_r <- as.numeric(chain[t - 1, paste0("beta\_chinese\_r", 1:R)])  
 prev\_mu\_malay <- chain[t - 1, "mu\_malay"]  
 prev\_mu\_indian <- chain[t - 1, "mu\_indian"]  
 prev\_mu\_chinese <- chain[t - 1, "mu\_chinese"]  
 prev\_tau\_malay <- chain[t - 1, "tau\_malay"]  
 prev\_tau\_indian <- chain[t - 1, "tau\_indian"]  
 prev\_tau\_chinese <- chain[t - 1, "tau\_chinese"]  
  
 logpost\_prop\_1 <- log\_posterior\_hier(  
 beta0 = prop\_beta0,  
 beta\_notable = prop\_beta\_notable,  
 beta\_malay\_r = prev\_beta\_malay\_r,  
 beta\_indian\_r = prev\_beta\_indian\_r,  
 beta\_chinese\_r = prev\_beta\_chinese\_r,  
 mu\_malay = prev\_mu\_malay,  
 mu\_indian = prev\_mu\_indian,  
 mu\_chinese = prev\_mu\_chinese,  
 election\_data = election\_data,  
 prior\_sd\_beta0 = prior\_sd\_beta0,  
 prior\_sd\_notable = prior\_sd\_notable,  
 scale = scale,  
 tau\_malay = prev\_tau\_malay,  
 tau\_chinese = prev\_tau\_chinese,  
 tau\_indian = prev\_tau\_indian,  
 prior\_sd\_mu = prior\_sd\_mu  
 )  
   
 # The core of this algorithm is that if the improvement in log-likelihood under the new parameter settings then the higher the chance we accept it  
 log\_r1 <- logpost\_prop\_1 - logpost\_curr  
 if (log(runif(1)) < log\_r1) {  
 # Write in the new values  
 chain[t, "beta0"] <- prop\_beta0  
 chain[t, "beta\_notable"] <- prop\_beta\_notable  
   
 # If the new setting is adopted, the value of the log-likelihood function has to be updated  
 logpost\_curr <- logpost\_prop\_1  
 # Record  
 move\_beta0[t] <- TRUE  
 } else {  
 chain[t, "beta0"] <- prev\_beta0  
 chain[t, "beta\_notable"] <- prev\_beta\_notable  
 move\_beta0[t] <- FALSE  
 }  
   
 new\_beta0 <- chain[t, "beta0"]  
 new\_beta\_notable <- chain[t, "beta\_notable"]  
   
   
 # Block 2: Update each region specific slope:  
 # We will use for loop to update the slope of each region for each ethnicity  
 # 2A) Slope of Malay of each region:  
 for (r in 1:R) {  
 bmr\_old <- prev\_beta\_malay\_r[r]  
 bmr\_prop <- bmr\_old + rnorm(1, mean = 0, sd = sigma\_region)  
   
 # We have I regions, we kept other I - 1 the same except for the one under the loop  
 prop\_beta\_malay\_r <- prev\_beta\_malay\_r  
 prop\_beta\_malay\_r[r] <- bmr\_prop  
   
 # Shall use the latest updated parameter (Intercept and beta\_notable)  
 logpost\_prop\_2a <- log\_posterior\_hier(  
 beta0 = new\_beta0,  
 beta\_notable = new\_beta\_notable,  
 beta\_malay\_r = prop\_beta\_malay\_r,  
 beta\_indian\_r = prev\_beta\_indian\_r,  
 beta\_chinese\_r = prev\_beta\_chinese\_r,  
 mu\_malay = prev\_mu\_malay,  
 mu\_indian = prev\_mu\_indian,  
 mu\_chinese = prev\_mu\_chinese,  
 election\_data = election\_data,  
 prior\_sd\_beta0 = prior\_sd\_beta0,  
 prior\_sd\_notable = prior\_sd\_notable,  
 scale = scale,  
 tau\_malay = prev\_tau\_malay,  
 tau\_chinese = prev\_tau\_chinese,  
 tau\_indian = prev\_tau\_indian,  
 prior\_sd\_mu = prior\_sd\_mu  
 )  
   
 log\_r2a <- logpost\_prop\_2a - logpost\_curr  
 if (log(runif(1)) < log\_r2a) {  
 chain[t, paste0("beta\_malay\_r", r)] <- bmr\_prop  
 logpost\_curr <- logpost\_prop\_2a  
 move\_region[t, 1, r] <- TRUE  
 } else {  
 chain[t, paste0("beta\_malay\_r", r)] <- bmr\_old  
 move\_region[t, 1, r] <- FALSE  
 }  
 }  
   
 new\_beta\_malay\_r <- as.numeric(chain[t, paste0("beta\_malay\_r", 1:R)])   
   
 # 2B) Slope of Indian of each region  
 for (r in 1:R){  
 bir\_old <- prev\_beta\_indian\_r[r]  
 bir\_prop <- bir\_old + rnorm(1, mean = 0, sd = sigma\_region)  
 prop\_beta\_indian\_r <- prev\_beta\_indian\_r  
 prop\_beta\_indian\_r[r] <- bir\_prop  
   
 logpost\_prop\_2b <- log\_posterior\_hier(  
 beta0 = new\_beta0,  
 beta\_notable = new\_beta\_notable,  
 beta\_malay\_r = new\_beta\_malay\_r,  
 beta\_indian\_r = prop\_beta\_indian\_r,  
 beta\_chinese\_r = prev\_beta\_chinese\_r,  
 mu\_malay = prev\_mu\_malay,  
 mu\_indian = prev\_mu\_indian,  
 mu\_chinese = prev\_mu\_chinese,  
 election\_data = election\_data,  
 prior\_sd\_beta0 = prior\_sd\_beta0,  
 prior\_sd\_notable = prior\_sd\_notable,  
 scale = scale,  
 tau\_malay = prev\_tau\_malay,  
 tau\_chinese = prev\_tau\_chinese,  
 tau\_indian = prev\_tau\_indian,  
 prior\_sd\_mu = prior\_sd\_mu  
 )  
   
 log\_r2b <- logpost\_prop\_2b - logpost\_curr  
 if (log(runif(1)) < log\_r2b) {  
 chain[t, paste0("beta\_indian\_r", r)] <- bir\_prop  
 logpost\_curr <- logpost\_prop\_2b  
 move\_region[t, 2, r] <- TRUE  
 } else {  
 chain[t, paste0("beta\_indian\_r", r)] <- bir\_old  
 move\_region[t, 2, r] <- FALSE  
 }  
 }  
   
 new\_beta\_indian\_r <- as.numeric(chain[t, paste0("beta\_indian\_r", 1:R)])   
   
 # 2C) Update the slope of chinese in each region  
 for (r in 1:R) {  
 bcr\_old <- prev\_beta\_chinese\_r[r]  
 bcr\_prop <- bcr\_old + rnorm(1, mean = 0, sd = sigma\_region)  
 prop\_beta\_chinese\_r <- prev\_beta\_chinese\_r  
 prop\_beta\_chinese\_r[r] <- bcr\_prop  
   
 logpost\_prop\_2c <- log\_posterior\_hier(  
 beta0 = new\_beta0,  
 beta\_notable = new\_beta\_notable,  
 beta\_malay\_r = new\_beta\_malay\_r,  
 beta\_indian\_r = new\_beta\_indian\_r,  
 beta\_chinese\_r = prop\_beta\_chinese\_r,  
 mu\_malay = prev\_mu\_malay,  
 mu\_indian = prev\_mu\_indian,  
 mu\_chinese = prev\_mu\_chinese,  
 election\_data = election\_data,  
 prior\_sd\_beta0 = prior\_sd\_beta0,  
 prior\_sd\_notable = prior\_sd\_notable,  
 scale = scale,  
 tau\_malay = prev\_tau\_malay,  
 tau\_chinese = prev\_tau\_chinese,  
 tau\_indian = prev\_tau\_indian,  
 prior\_sd\_mu = prior\_sd\_mu  
 )  
   
 log\_r2c <- logpost\_prop\_2c - logpost\_curr  
 if (log(runif(1)) < log\_r2c) {  
 chain[t, paste0("beta\_chinese\_r", r)] <- bcr\_prop  
 logpost\_curr <- logpost\_prop\_2c  
 move\_region[t, 3, r] <- TRUE  
 } else {  
 chain[t, paste0("beta\_chinese\_r", r)] <- bcr\_old  
 move\_region[t, 3, r] <- FALSE  
 }  
 }  
   
 new\_beta\_chinese\_r <- as.numeric(chain[t, paste0("beta\_chinese\_r", 1:R)])  
   
   
 # Block 3: We will update the hypermean mu for each ethnic  
 prev\_mu <- c(prev\_mu\_malay, prev\_mu\_indian, prev\_mu\_chinese)  
 prop3 <- prev\_mu + rnorm(3, mean = 0, sd = sigma\_mu)  
 prop\_mu\_malay <- prop3[1]  
 prop\_mu\_indian <- prop3[2]  
 prop\_mu\_chinese <- prop3[3]  
   
 logpost\_prop\_3 <- log\_posterior\_hier(  
 beta0 = new\_beta0,  
 beta\_notable = new\_beta\_notable,  
 beta\_malay\_r = new\_beta\_malay\_r,  
 beta\_indian\_r = new\_beta\_indian\_r,  
 beta\_chinese\_r = new\_beta\_chinese\_r,  
 mu\_malay = prop\_mu\_malay,  
 mu\_indian = prop\_mu\_indian,  
 mu\_chinese = prop\_mu\_chinese,  
 election\_data = election\_data,  
 prior\_sd\_beta0 = prior\_sd\_beta0,  
 prior\_sd\_notable = prior\_sd\_notable,  
 scale = scale,  
 tau\_malay = prev\_tau\_malay,  
 tau\_chinese = prev\_tau\_chinese,  
 tau\_indian = prev\_tau\_indian,  
 prior\_sd\_mu = prior\_sd\_mu  
 )  
   
 log\_r3 <- logpost\_prop\_3 - logpost\_curr  
 if (log(runif(1)) < log\_r3) {  
 chain[t, "mu\_malay"] <- prop\_mu\_malay  
 chain[t, "mu\_indian"] <- prop\_mu\_indian  
 chain[t, "mu\_chinese"] <- prop\_mu\_chinese  
 logpost\_curr <- logpost\_prop\_3  
 move\_mu[t] <- TRUE  
 } else {  
 chain[t, c("mu\_malay", "mu\_indian", "mu\_chinese")] <- prev\_mu  
 move\_mu[t] <- FALSE  
 }  
   
 new\_mu\_malay <- chain[t, "mu\_malay"]  
 new\_mu\_indian <- chain[t, "mu\_indian"]  
 new\_mu\_chinese <- chain[t, "mu\_chinese"]  
   
 # Block 4: Update tau\_ethnic  
 # The support of the distribution have to be >0  
 prev\_tau <- c(prev\_tau\_malay, prev\_tau\_indian, prev\_tau\_chinese)  
 repeat {  
 prop\_tau\_malay <- rnorm(1, mean = prev\_tau\_malay, sd = sigma\_tau)  
 if (prop\_tau\_malay > 0) break  
 }  
   
 repeat {  
 prop\_tau\_chinese <- rnorm(1, mean = prev\_tau\_chinese, sd = sigma\_tau)  
 if (prop\_tau\_chinese > 0) break  
 }  
   
 repeat {  
 prop\_tau\_indian <- rnorm(1, mean = prev\_tau\_indian, sd = sigma\_tau)  
 if (prop\_tau\_indian > 0) break  
 }  
   
 logpost\_prop\_4 <- log\_posterior\_hier(  
 beta0 = new\_beta0,  
 beta\_notable = new\_beta\_notable,  
 beta\_malay\_r = new\_beta\_malay\_r,  
 beta\_indian\_r = new\_beta\_indian\_r,  
 beta\_chinese\_r = new\_beta\_chinese\_r,  
 mu\_malay = new\_mu\_malay,  
 mu\_indian = new\_mu\_indian,  
 mu\_chinese = new\_mu\_chinese,  
 election\_data = election\_data,  
 prior\_sd\_beta0 = prior\_sd\_beta0,  
 prior\_sd\_notable = prior\_sd\_notable,  
 scale = scale,  
 tau\_malay = prop\_tau\_malay,  
 tau\_chinese = prop\_tau\_chinese,  
 tau\_indian = prop\_tau\_indian,  
 prior\_sd\_mu = prior\_sd\_mu  
 )  
   
 log\_r4 <- logpost\_prop\_4 - logpost\_curr  
 if (log(runif(1)) < log\_r4) {  
 chain[t, "tau\_malay"] <- prop\_tau\_malay  
 chain[t, "tau\_indian"] <- prop\_tau\_indian  
 chain[t, "tau\_chinese"] <- prop\_tau\_chinese  
 logpost\_curr <- logpost\_prop\_4  
 move\_tau[t] <- TRUE  
 } else {  
 chain[t, c("tau\_malay", "tau\_indian", "tau\_chinese")] <- prev\_tau  
 move\_tau[t] <- FALSE  
 }  
}

## 5. Inspect Acceptance Rate and Discard Burn-in

# Discard burn-in  
post <- as.data.frame(chain[(burn\_in + 1):n\_iter, ])  
post$iter <- seq(burn\_in + 1, n\_iter)  
  
# Acceptance rates  
acc\_beta0 <- mean(move\_beta0[(burn\_in + 1):n\_iter])  
acc\_mu <- mean(move\_mu[(burn\_in + 1):n\_iter])  
  
cat("Acceptance rate for beta0 and beta\_notable:", round(acc\_beta0, 3), "\n")

Acceptance rate for beta0 and beta\_notable: 0.011

cat("Acceptance rate for mu\_malay, mu\_indian, mu\_chinese:", round(acc\_mu, 3), "\n")

Acceptance rate for mu\_malay, mu\_indian, mu\_chinese: 0.216

for (r in 1:R) {  
 acc\_malay\_r <- mean(move\_region[(burn\_in + 1):n\_iter, 1, r])  
 acc\_indian\_r <- mean(move\_region[(burn\_in + 1):n\_iter, 2, r])  
 acc\_chinese\_r <- mean(move\_region[(burn\_in + 1):n\_iter, 3, r])  
 cat(sprintf("Region %d — beta\_malay: %.3f, beta\_indian: %.3f, beta\_chinese: %.3f\n", r, acc\_malay\_r, acc\_indian\_r, acc\_chinese\_r))  
}

Region 1 — beta\_malay: 0.174, beta\_indian: 0.222, beta\_chinese: 0.242  
Region 2 — beta\_malay: 0.089, beta\_indian: 0.227, beta\_chinese: 0.218  
Region 3 — beta\_malay: 0.066, beta\_indian: 0.234, beta\_chinese: 0.252  
Region 4 — beta\_malay: 0.089, beta\_indian: 0.252, beta\_chinese: 0.254  
Region 5 — beta\_malay: 0.147, beta\_indian: 0.236, beta\_chinese: 0.151  
Region 6 — beta\_malay: 0.106, beta\_indian: 0.232, beta\_chinese: 0.135  
Region 7 — beta\_malay: 0.117, beta\_indian: 0.246, beta\_chinese: 0.203  
Region 8 — beta\_malay: 0.076, beta\_indian: 0.191, beta\_chinese: 0.103  
Region 9 — beta\_malay: 0.133, beta\_indian: 0.218, beta\_chinese: 0.192  
Region 10 — beta\_malay: 0.125, beta\_indian: 0.251, beta\_chinese: 0.209  
Region 11 — beta\_malay: 0.096, beta\_indian: 0.235, beta\_chinese: 0.127  
Region 12 — beta\_malay: 0.210, beta\_indian: 0.251, beta\_chinese: 0.206  
Region 13 — beta\_malay: 0.158, beta\_indian: 0.247, beta\_chinese: 0.150

acc\_tau <- mean(move\_tau[(burn\_in + 1):n\_iter])  
cat("Acceptance rate for tau\_ethnic:", round(acc\_tau, 3), "\n")

Acceptance rate for tau\_ethnic: 0.242

head(post, 10)

beta0 beta\_notable beta\_malay\_r1 beta\_malay\_r2 beta\_malay\_r3  
1 0.3780744 0.3623436 -0.4792397 -0.1221287 -0.5096426  
2 0.3780744 0.3623436 -0.4792397 -0.1221287 -0.5096426  
3 0.3780744 0.3623436 -0.4792397 -0.1221287 -0.5096426  
4 0.3780744 0.3623436 -0.4792397 -0.1221287 -0.5096426  
5 0.3780744 0.3623436 -0.4792397 -0.1221287 -0.5096426  
6 0.3780744 0.3623436 -0.4792397 -0.1221287 -0.5096426  
7 0.3780744 0.3623436 -0.4792397 -0.1221287 -0.5096426  
8 0.3780744 0.3623436 -0.4792397 -0.1221287 -0.5096426  
9 0.3780744 0.3623436 -0.4792397 -0.1221287 -0.5096426  
10 0.3780744 0.3623436 -0.4792397 -0.1221287 -0.5096426  
 beta\_malay\_r4 beta\_malay\_r5 beta\_malay\_r6 beta\_malay\_r7 beta\_malay\_r8  
1 -0.289107 0.151127 0.005868628 0.1993636 -0.1825506  
2 -0.289107 0.151127 0.005868628 0.1993636 -0.1825506  
3 -0.289107 0.151127 0.005868628 0.1993636 -0.1825506  
4 -0.289107 0.151127 0.005868628 0.1993636 -0.1825506  
5 -0.289107 0.151127 0.005868628 0.1993636 -0.1825506  
6 -0.289107 0.151127 0.005868628 0.1993636 -0.1825506  
7 -0.289107 0.151127 0.005868628 0.1993636 -0.1825506  
8 -0.289107 0.151127 0.005868628 0.1993636 -0.1825506  
9 -0.289107 0.151127 0.005868628 0.1993636 -0.1825506  
10 -0.289107 0.151127 0.005868628 0.1993636 -0.1825506  
 beta\_malay\_r9 beta\_malay\_r10 beta\_malay\_r11 beta\_malay\_r12 beta\_malay\_r13  
1 0.6698417 0.5041466 0.9645776 2.003789 1.177913  
2 0.6698417 0.5041466 0.9645776 2.003789 1.177913  
3 0.6698417 0.5041466 0.9645776 2.003789 1.177913  
4 0.6698417 0.5041466 0.9645776 2.003789 1.177913  
5 0.6698417 0.5041466 0.9645776 2.003789 1.177913  
6 0.6698417 0.5041466 0.9645776 2.003789 1.177913  
7 0.6698417 0.5041466 0.9645776 2.003789 1.177913  
8 0.6698417 0.5041466 0.9645776 2.003789 1.177913  
9 0.6698417 0.5041466 0.9645776 2.003789 1.177913  
10 0.6698417 0.5041466 0.9645776 2.003789 1.177913  
 beta\_indian\_r1 beta\_indian\_r2 beta\_indian\_r3 beta\_indian\_r4 beta\_indian\_r5  
1 0.1151522 -1.184604 1.629457 -0.9438768 -0.9492146  
2 0.1151522 -1.184604 1.629457 -0.9438768 -0.9492146  
3 0.1151522 -1.184604 1.629457 -0.9438768 -0.9492146  
4 0.1151522 -1.184604 1.629457 -0.9438768 -0.9492146  
5 0.1151522 -1.184604 1.629457 -0.9438768 -0.9492146  
6 0.1151522 -1.184604 1.629457 -0.9438768 -0.9492146  
7 0.1151522 -1.184604 1.629457 -0.9438768 -0.9492146  
8 0.1151522 -1.184604 1.629457 -0.9438768 -0.9492146  
9 0.1151522 -1.184604 1.629457 -0.9438768 -0.9492146  
10 0.1151522 -1.184604 1.629457 -0.9438768 -0.9492146  
 beta\_indian\_r6 beta\_indian\_r7 beta\_indian\_r8 beta\_indian\_r9 beta\_indian\_r10  
1 -0.6561342 -0.6705368 -0.3972236 -2.175082 0.9734644  
2 -0.6561342 -0.6705368 -0.3972236 -2.175082 0.9734644  
3 -0.6561342 -0.6705368 -0.3972236 -2.175082 0.9734644  
4 -0.6561342 -0.6705368 -0.3972236 -2.175082 0.9734644  
5 -0.6561342 -0.6705368 -0.3972236 -2.175082 0.9734644  
6 -0.6561342 -0.6705368 -0.3972236 -2.175082 0.9734644  
7 -0.6561342 -0.6705368 -0.3972236 -2.175082 0.9734644  
8 -0.6561342 -0.6705368 -0.3972236 -2.175082 0.9734644  
9 -0.6561342 -0.6705368 -0.3972236 -2.175082 0.9734644  
10 -0.6561342 -0.6705368 -0.3972236 -2.175082 0.9734644  
 beta\_indian\_r11 beta\_indian\_r12 beta\_indian\_r13 beta\_chinese\_r1  
1 0.227651 -1.239458 -1.247319 1.158127  
2 0.227651 -1.239458 -1.247319 1.158127  
3 0.227651 -1.239458 -1.247319 1.158127  
4 0.227651 -1.239458 -1.247319 1.158127  
5 0.227651 -1.239458 -1.247319 1.158127  
6 0.227651 -1.239458 -1.247319 1.158127  
7 0.227651 -1.239458 -1.247319 1.158127  
8 0.227651 -1.239458 -1.247319 1.158127  
9 0.227651 -1.239458 -1.247319 1.158127  
10 0.227651 -1.239458 -1.247319 1.158127  
 beta\_chinese\_r2 beta\_chinese\_r3 beta\_chinese\_r4 beta\_chinese\_r5  
1 -1.628626 -2.200124 -4.005874 -2.268833  
2 -1.628626 -2.200124 -4.005874 -2.268833  
3 -1.628626 -2.200124 -4.005874 -2.268833  
4 -1.628626 -2.200124 -4.005874 -2.268833  
5 -1.628626 -2.200124 -4.005874 -2.268833  
6 -1.628626 -2.200124 -4.005874 -2.268833  
7 -1.628626 -2.200124 -4.005874 -2.268833  
8 -1.628626 -2.200124 -4.005874 -2.268833  
9 -1.628626 -2.200124 -4.005874 -2.268833  
10 -1.628626 -2.200124 -4.005874 -2.268833  
 beta\_chinese\_r6 beta\_chinese\_r7 beta\_chinese\_r8 beta\_chinese\_r9  
1 -1.762055 -1.826773 -2.339733 -1.725623  
2 -1.762055 -1.826773 -2.339733 -1.725623  
3 -1.762055 -1.826773 -2.339733 -1.725623  
4 -1.762055 -1.826773 -2.339733 -1.725623  
5 -1.762055 -1.826773 -2.339733 -1.725623  
6 -1.762055 -1.826773 -2.339733 -1.725623  
7 -1.762055 -1.826773 -2.339733 -1.725623  
8 -1.762055 -1.826773 -2.339733 -1.725623  
9 -1.762055 -1.826773 -2.339733 -1.725623  
10 -1.762055 -1.826773 -2.339733 -1.725623  
 beta\_chinese\_r10 beta\_chinese\_r11 beta\_chinese\_r12 beta\_chinese\_r13  
1 -2.202278 -2.071204 -2.635673 -1.348373  
2 -2.202278 -2.071204 -2.635673 -1.348373  
3 -2.202278 -2.071204 -2.635673 -1.348373  
4 -2.202278 -2.071204 -2.635673 -1.348373  
5 -2.202278 -2.071204 -2.635673 -1.348373  
6 -2.202278 -2.071204 -2.635673 -1.348373  
7 -2.202278 -2.071204 -2.635673 -1.348373  
8 -2.202278 -2.071204 -2.635673 -1.348373  
9 -2.202278 -2.071204 -2.635673 -1.348373  
10 -2.202278 -2.071204 -2.635673 -1.348373  
 mu\_malay mu\_indian mu\_chinese tau\_malay tau\_indian tau\_chinese iter  
1 -2.437986 -0.6594196 3.224635 13.21694 7.870056 17.52707 1501  
2 -2.437986 -0.6594196 3.224635 13.21694 7.870056 17.52707 1502  
3 -2.437986 -0.6594196 3.224635 13.21694 7.870056 17.52707 1503  
4 -2.437986 -0.6594196 3.224635 13.21694 7.870056 17.52707 1504  
5 -2.437986 -0.6594196 3.224635 13.21694 7.870056 17.52707 1505  
6 -2.437986 -0.6594196 3.224635 13.21694 7.870056 17.52707 1506  
7 -2.437986 -0.6594196 3.224635 13.21694 7.870056 17.52707 1507  
8 -2.437986 -0.6594196 3.224635 13.21694 7.870056 17.52707 1508  
9 -2.437986 -0.6594196 3.224635 13.21694 7.870056 17.52707 1509  
10 -2.437986 -0.6594196 3.224635 13.21694 7.870056 17.52707 1510

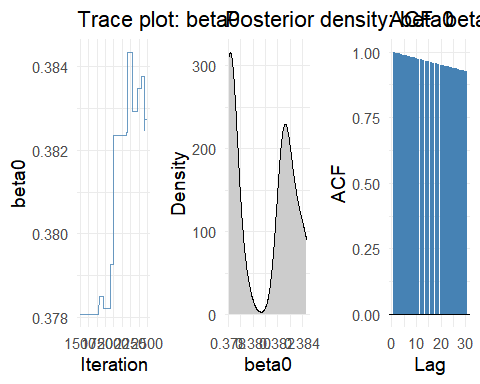
tail(post, 10)

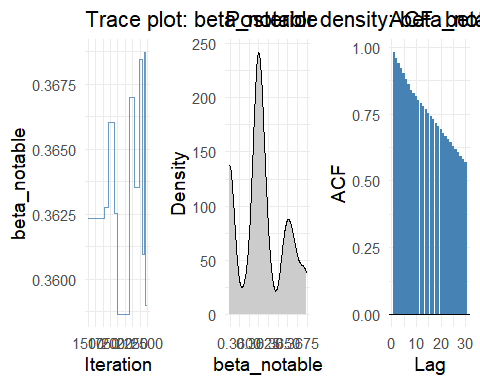
beta0 beta\_notable beta\_malay\_r1 beta\_malay\_r2 beta\_malay\_r3  
991 0.3827359 0.358975 -0.4467678 -0.1237265 -0.5120438  
992 0.3827359 0.358975 -0.4467678 -0.1237265 -0.5120438  
993 0.3827359 0.358975 -0.4467678 -0.1237265 -0.5120438  
994 0.3827359 0.358975 -0.4467678 -0.1237265 -0.5120438  
995 0.3827359 0.358975 -0.4467678 -0.1237265 -0.5120438  
996 0.3827359 0.358975 -0.4467678 -0.1237265 -0.5120438  
997 0.3827359 0.358975 -0.4467678 -0.1237265 -0.5120438  
998 0.3827359 0.358975 -0.4467678 -0.1237265 -0.5120438  
999 0.3827359 0.358975 -0.4467678 -0.1237265 -0.5120438  
1000 0.3827359 0.358975 -0.4467678 -0.1237265 -0.5120438  
 beta\_malay\_r4 beta\_malay\_r5 beta\_malay\_r6 beta\_malay\_r7 beta\_malay\_r8  
991 -0.2919454 0.1196706 0.01189675 0.2108109 -0.1863407  
992 -0.2919454 0.1196706 0.01189675 0.2108109 -0.1863407  
993 -0.2919454 0.1196706 0.01189675 0.2108109 -0.1863407  
994 -0.2919454 0.1196706 0.01189675 0.2108109 -0.1863407  
995 -0.2919454 0.1196706 0.01189675 0.2108109 -0.1863407  
996 -0.2919454 0.1196706 0.01189675 0.2108109 -0.1863407  
997 -0.2919454 0.1196706 0.01189675 0.2108109 -0.1863407  
998 -0.2919454 0.1196706 0.01189675 0.2108109 -0.1863407  
999 -0.2919454 0.1196706 0.01189675 0.2108109 -0.1863407  
1000 -0.2919454 0.1196706 0.01189675 0.2108109 -0.1863407  
 beta\_malay\_r9 beta\_malay\_r10 beta\_malay\_r11 beta\_malay\_r12 beta\_malay\_r13  
991 0.6927554 0.5175675 0.9542943 1.958937 1.193801  
992 0.6927554 0.5175675 0.9542943 1.958937 1.193801  
993 0.6927554 0.5175675 0.9542943 1.958937 1.193801  
994 0.6927554 0.5175675 0.9542943 1.958937 1.193801  
995 0.6927554 0.5175675 0.9542943 1.958937 1.193801  
996 0.6927554 0.5175675 0.9542943 1.958937 1.193801  
997 0.6927554 0.5175675 0.9542943 1.958937 1.193801  
998 0.6927554 0.5175675 0.9542943 1.958937 1.193801  
999 0.6927554 0.5175675 0.9542943 1.958937 1.193801  
1000 0.6927554 0.5175675 0.9542943 1.958937 1.193801  
 beta\_indian\_r1 beta\_indian\_r2 beta\_indian\_r3 beta\_indian\_r4 beta\_indian\_r5  
991 -0.03062407 -1.183644 1.721756 -0.9593447 -0.9960262  
992 -0.03062407 -1.183644 1.721756 -0.9593447 -0.9960262  
993 -0.03062407 -1.183644 1.721756 -0.9593447 -0.9960262  
994 -0.03062407 -1.183644 1.721756 -0.9593447 -0.9960262  
995 -0.03062407 -1.183644 1.721756 -0.9593447 -0.9960262  
996 -0.03062407 -1.183644 1.721756 -0.9593447 -0.9960262  
997 -0.03062407 -1.183644 1.721756 -0.9593447 -0.9960262  
998 -0.03062407 -1.183644 1.721756 -0.9593447 -0.9960262  
999 -0.03062407 -1.183644 1.721756 -0.9593447 -0.9960262  
1000 -0.03062407 -1.183644 1.721756 -0.9593447 -0.9960262  
 beta\_indian\_r6 beta\_indian\_r7 beta\_indian\_r8 beta\_indian\_r9  
991 -0.7270722 -0.7466762 -0.3880102 -2.2657  
992 -0.7270722 -0.7466762 -0.3880102 -2.2657  
993 -0.7270722 -0.7466762 -0.3880102 -2.2657  
994 -0.7270722 -0.7466762 -0.3880102 -2.2657  
995 -0.7270722 -0.7466762 -0.3880102 -2.2657  
996 -0.7270722 -0.7466762 -0.3880102 -2.2657  
997 -0.7270722 -0.7466762 -0.3880102 -2.2657  
998 -0.7270722 -0.7466762 -0.3880102 -2.2657  
999 -0.7270722 -0.7466762 -0.3880102 -2.2657  
1000 -0.7270722 -0.7466762 -0.3880102 -2.2657  
 beta\_indian\_r10 beta\_indian\_r11 beta\_indian\_r12 beta\_indian\_r13  
991 0.9539186 0.1795484 -1.768745 -1.417045  
992 0.9539186 0.1795484 -1.768745 -1.417045  
993 0.9539186 0.1795484 -1.768745 -1.417045  
994 0.9539186 0.1795484 -1.768745 -1.417045  
995 0.9539186 0.1795484 -1.768745 -1.417045  
996 0.9539186 0.1795484 -1.768745 -1.417045  
997 0.9539186 0.1795484 -1.768745 -1.417045  
998 0.9539186 0.1795484 -1.768745 -1.417045  
999 0.9539186 0.1795484 -1.768745 -1.417045  
1000 0.9539186 0.1795484 -1.768745 -1.417045  
 beta\_chinese\_r1 beta\_chinese\_r2 beta\_chinese\_r3 beta\_chinese\_r4  
991 1.083074 -1.628659 -2.183735 -4.390065  
992 1.083074 -1.628659 -2.183735 -4.390065  
993 1.083074 -1.628659 -2.183735 -4.390065  
994 1.083074 -1.628659 -2.183735 -4.390065  
995 1.083074 -1.628659 -2.183735 -4.390065  
996 1.083074 -1.628659 -2.183735 -4.390065  
997 1.083074 -1.628659 -2.183735 -4.390065  
998 1.083074 -1.628659 -2.183735 -4.390065  
999 1.083074 -1.628659 -2.183735 -4.390065  
1000 1.083074 -1.628659 -2.183735 -4.390065  
 beta\_chinese\_r5 beta\_chinese\_r6 beta\_chinese\_r7 beta\_chinese\_r8  
991 -2.252826 -1.757089 -1.846777 -2.351402  
992 -2.252826 -1.757089 -1.846777 -2.351402  
993 -2.252826 -1.757089 -1.846777 -2.351402  
994 -2.252826 -1.757089 -1.846777 -2.351402  
995 -2.252826 -1.757089 -1.846777 -2.351402  
996 -2.252826 -1.757089 -1.846777 -2.351402  
997 -2.252826 -1.757089 -1.846777 -2.351402  
998 -2.252826 -1.757089 -1.846777 -2.351402  
999 -2.252826 -1.757089 -1.846777 -2.351402  
1000 -2.252826 -1.757089 -1.846777 -2.351402  
 beta\_chinese\_r9 beta\_chinese\_r10 beta\_chinese\_r11 beta\_chinese\_r12  
991 -1.699035 -2.212723 -2.059377 -2.616757  
992 -1.699035 -2.212723 -2.059377 -2.616757  
993 -1.699035 -2.212723 -2.059377 -2.616757  
994 -1.699035 -2.212723 -2.059377 -2.616757  
995 -1.699035 -2.212723 -2.059377 -2.616757  
996 -1.699035 -2.212723 -2.059377 -2.616757  
997 -1.699035 -2.212723 -2.059377 -2.616757  
998 -1.699035 -2.212723 -2.059377 -2.616757  
999 -1.699035 -2.212723 -2.059377 -2.616757  
1000 -1.699035 -2.212723 -2.059377 -2.616757  
 beta\_chinese\_r13 mu\_malay mu\_indian mu\_chinese tau\_malay tau\_indian  
991 -1.352668 -0.3591503 0.7666997 -0.8821433 13.69328 6.85942  
992 -1.352668 -0.3591503 0.7666997 -0.8821433 13.69328 6.85942  
993 -1.352668 -0.3591503 0.7666997 -0.8821433 13.69328 6.85942  
994 -1.352668 -0.3591503 0.7666997 -0.8821433 13.69328 6.85942  
995 -1.352668 -0.3591503 0.7666997 -0.8821433 13.69328 6.85942  
996 -1.352668 -0.3591503 0.7666997 -0.8821433 13.69328 6.85942  
997 -1.352668 -0.3591503 0.7666997 -0.8821433 13.69328 6.85942  
998 -1.352668 -0.3591503 0.7666997 -0.8821433 13.69328 6.85942  
999 -1.352668 -0.3591503 0.7666997 -0.8821433 13.69328 6.85942  
1000 -1.352668 -0.3591503 0.7666997 -0.8821433 13.69328 6.85942  
 tau\_chinese iter  
991 15.47557 2491  
992 15.47557 2492  
993 15.47557 2493  
994 15.47557 2494  
995 15.47557 2495  
996 15.47557 2496  
997 15.47557 2497  
998 15.47557 2498  
999 15.47557 2499  
1000 15.47557 2500

## 6. Trace Plots, Posterior Densities and ACFs

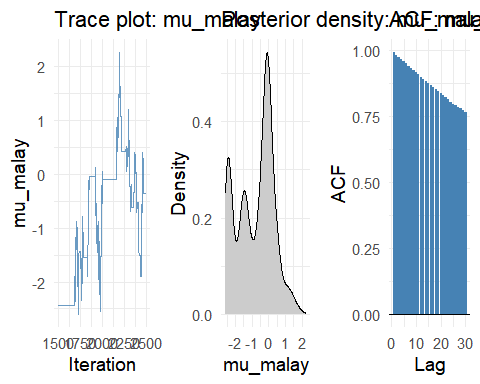
# ACF data for ggplot  
compute\_acf\_df <- function(x, L = 30) {  
 acf\_res <- acf(x, plot = FALSE, lag.max = L)  
 data.frame(lag = acf\_res$lag[-1], acf = acf\_res$acf[-1])  
}  
  
# Make trace, density, and acf plot  
plot\_trace\_dens\_acf <- function(df, varname) {  
 trace <- ggplot(df, aes\_string(x = "iter", y = varname)) +  
 geom\_line(color = "steelblue", alpha = 0.8) +  
 labs(title = paste("Trace plot:", varname), x = "Iteration", y = varname) +  
 theme\_minimal(base\_size = 14)  
   
 dens <- ggplot(df, aes\_string(x = varname)) +  
 geom\_density(fill = "grey80", color = "black") +  
 labs(title = paste("Posterior density:", varname), x = varname, y = "Density") +  
 theme\_minimal(base\_size = 14)  
   
 acf\_df <- compute\_acf\_df(df[[varname]], L = 30)  
 acf\_plot <- ggplot(acf\_df, aes(x = lag, y = acf)) +  
 geom\_bar(stat = "identity", fill = "steelblue") +  
 geom\_hline(yintercept = 0) +  
 labs(title = paste("ACF:", varname), x = "Lag", y = "ACF") +  
 theme\_minimal(base\_size = 14)  
   
 return(list(trace, dens, acf\_plot))  
}  
  
# Start Plotting  
for (var in c("beta0", "beta\_notable")) {  
 plots <- plot\_trace\_dens\_acf(post, var)  
 grid.arrange(grobs = plots, ncol = 3)  
}

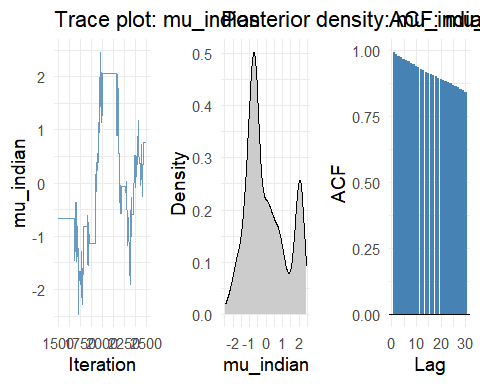
Warning: `aes\_string()` was deprecated in ggplot2 3.0.0.  
ℹ Please use tidy evaluation idioms with `aes()`.  
ℹ See also `vignette("ggplot2-in-packages")` for more information.

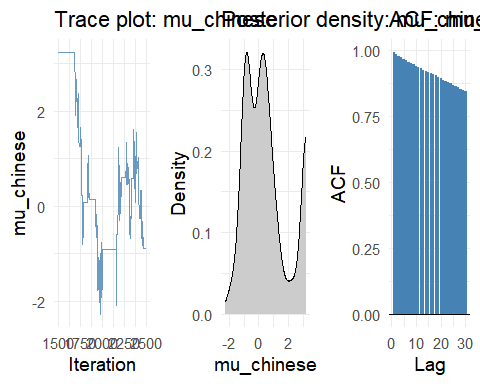




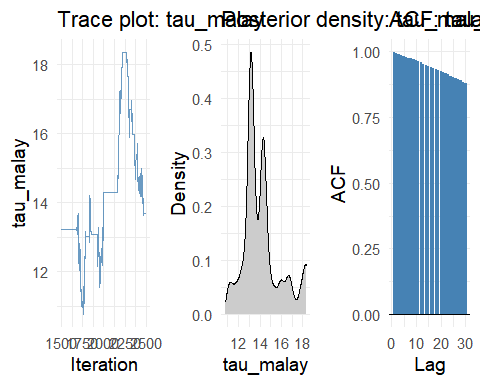
for (var in c("mu\_malay", "mu\_indian", "mu\_chinese")) {  
 plots <- plot\_trace\_dens\_acf(post, var)  
 grid.arrange(grobs = plots, ncol = 3)  
}

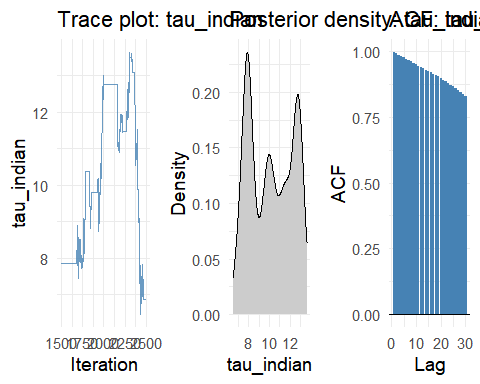


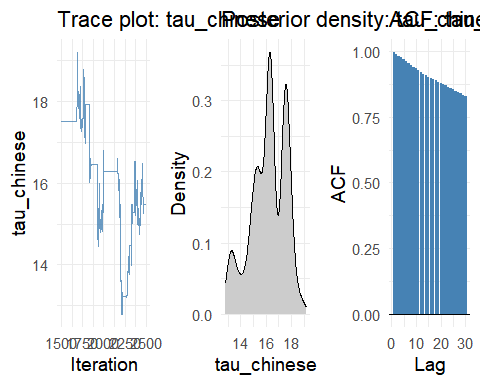




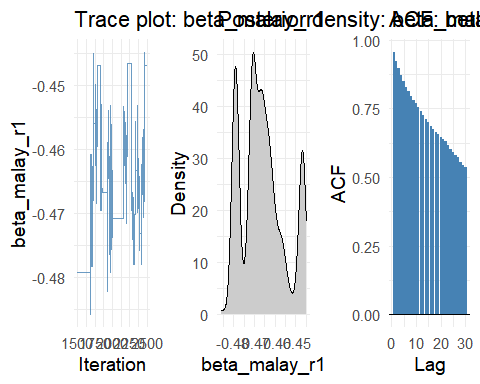
for (var in c("tau\_malay", "tau\_indian", "tau\_chinese")) {  
 plots <- plot\_trace\_dens\_acf(post, var)  
 grid.arrange(grobs = plots, ncol = 3)  
}

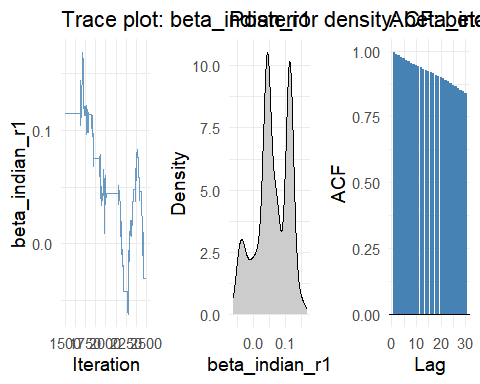


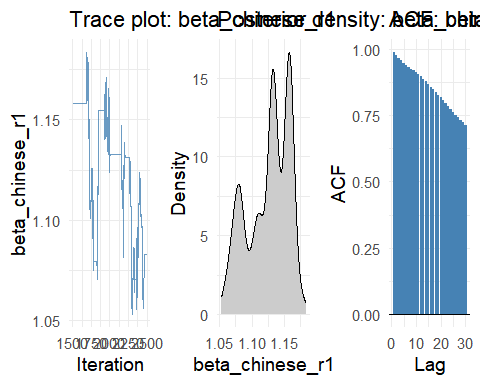


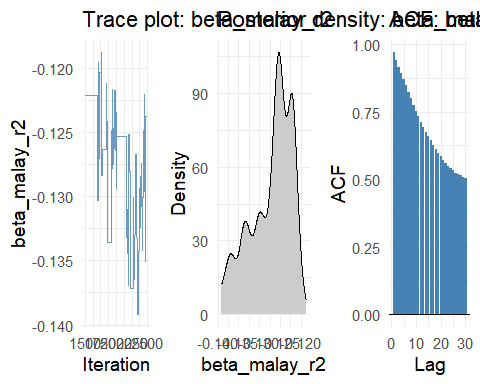


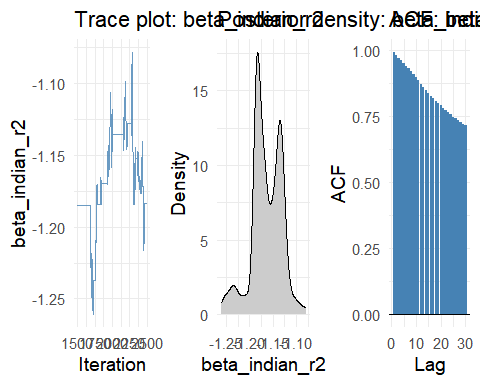
for (r in 1:R) {  
 for (eth in c("malay", "indian", "chinese")) {  
 var <- paste0("beta\_", eth, "\_r", r)  
 plots <- plot\_trace\_dens\_acf(post, var)  
 grid.arrange(grobs = plots, ncol = 3)  
 }  
}

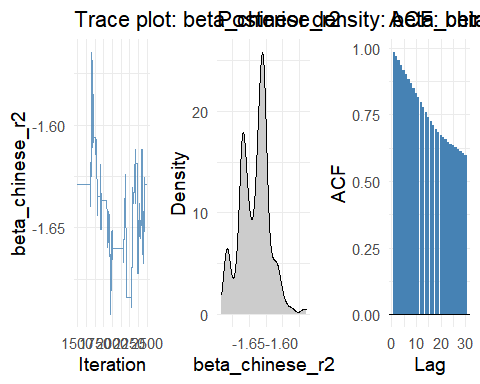


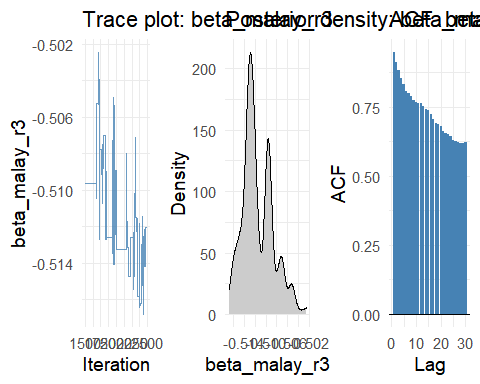


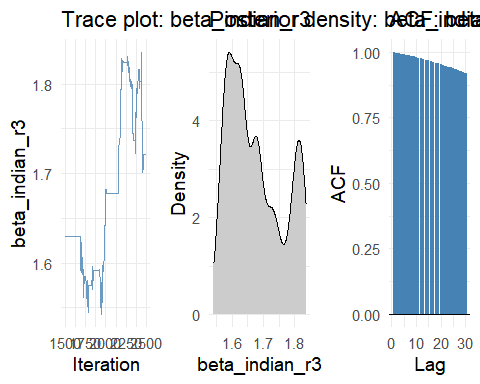


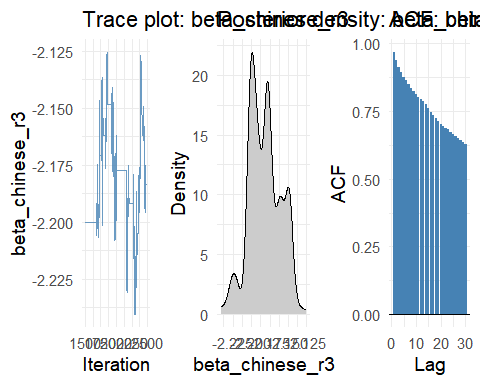


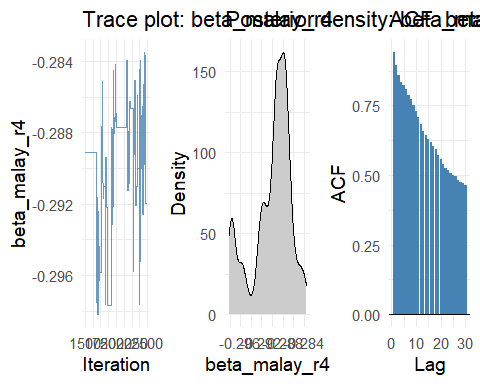


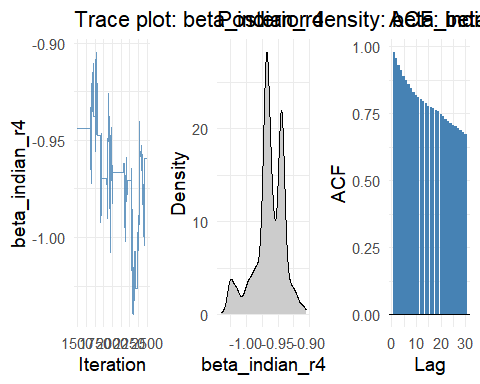


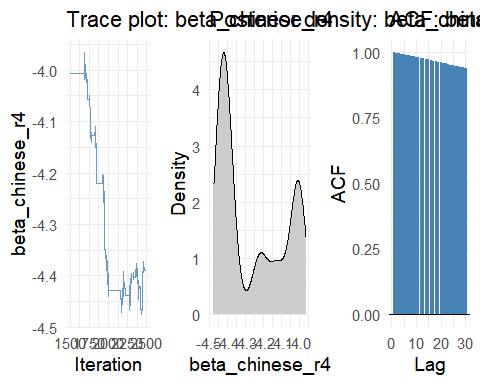


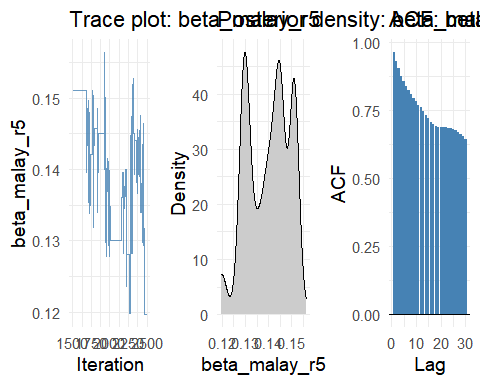


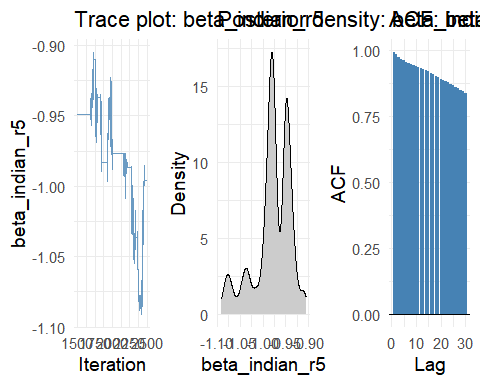


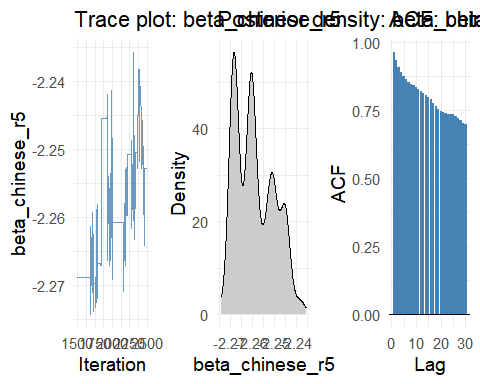


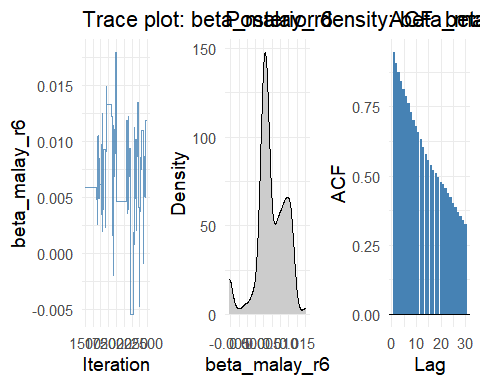


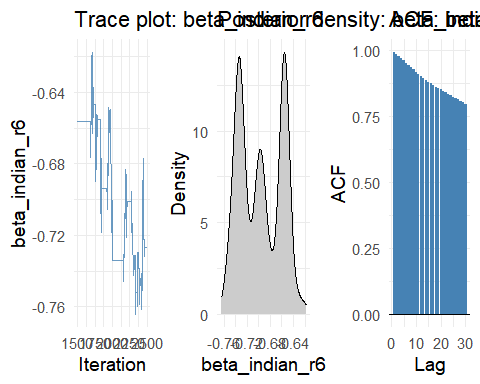


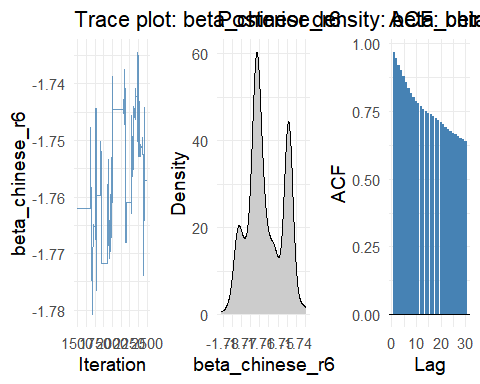


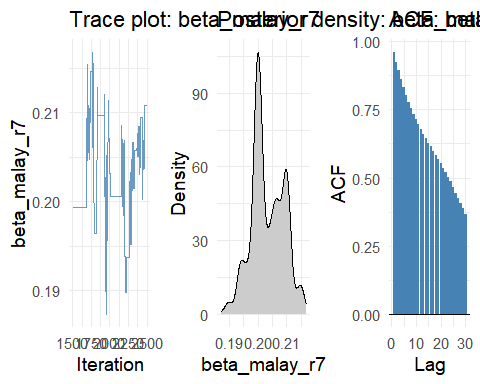


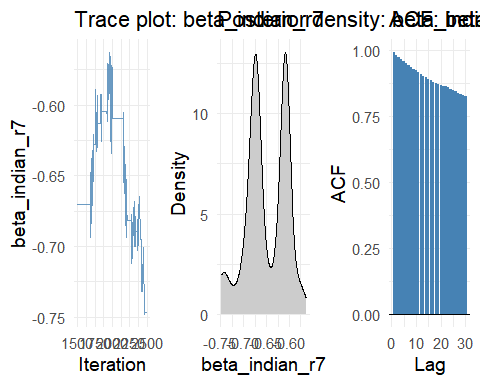


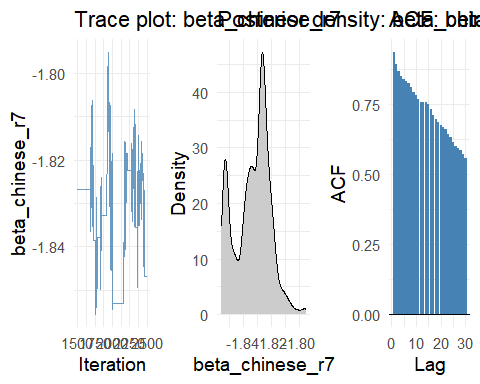


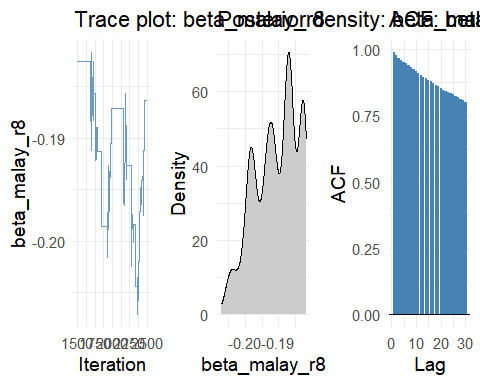


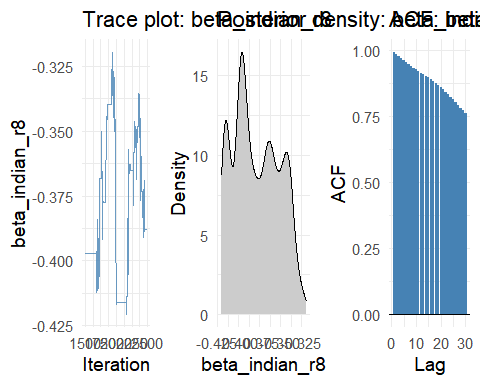


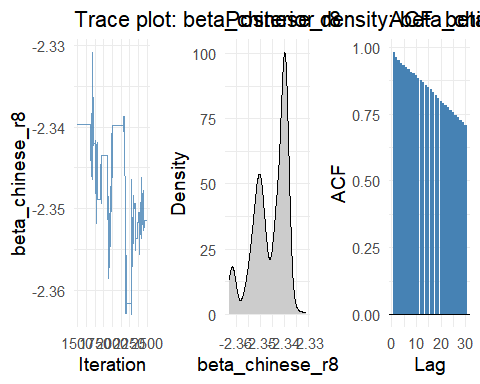


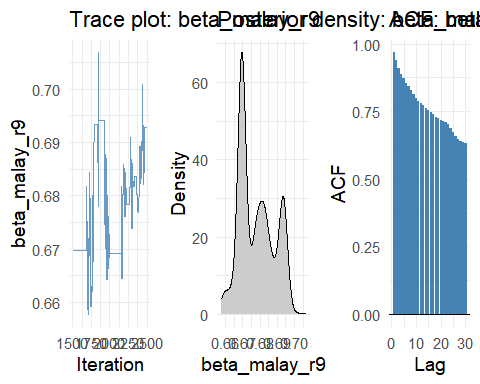


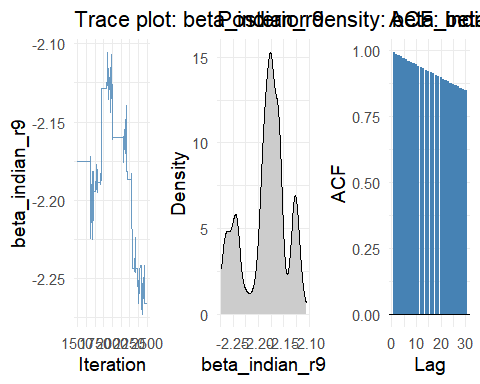


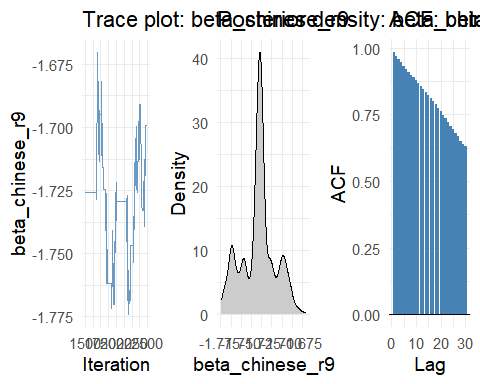


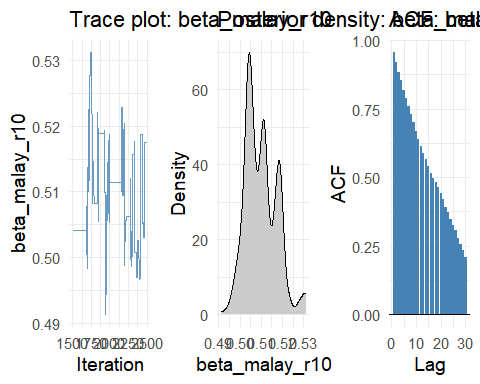


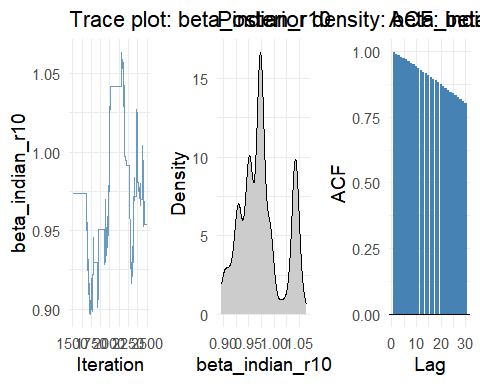


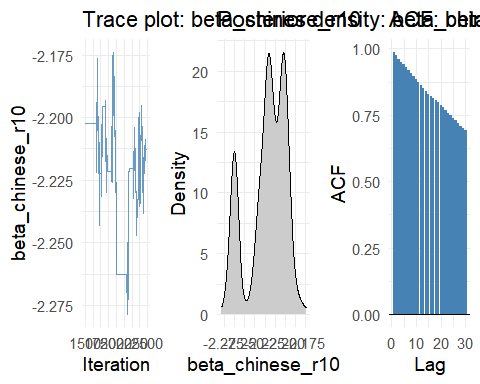


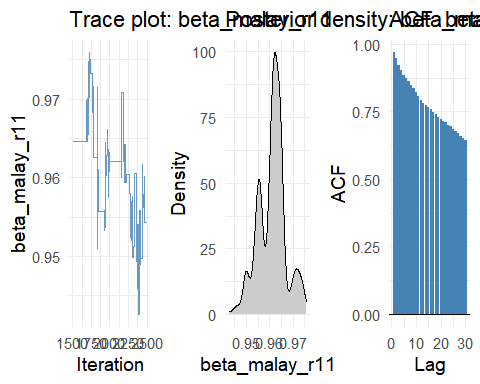


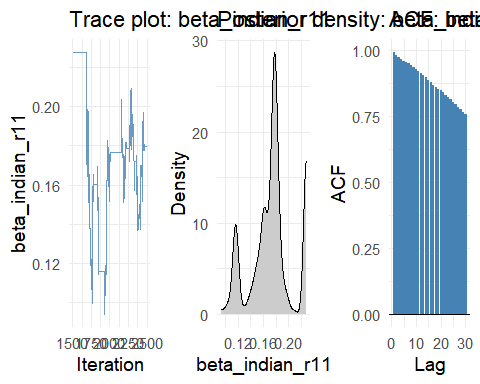


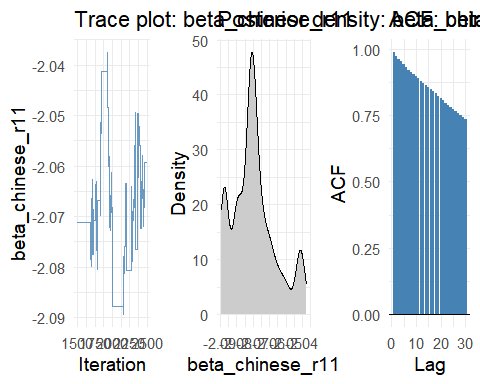


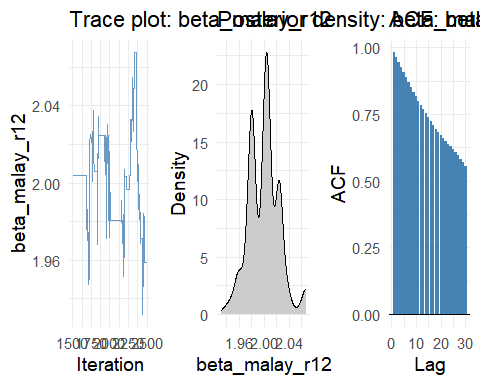


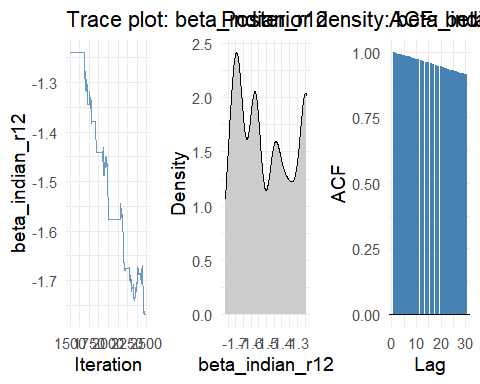


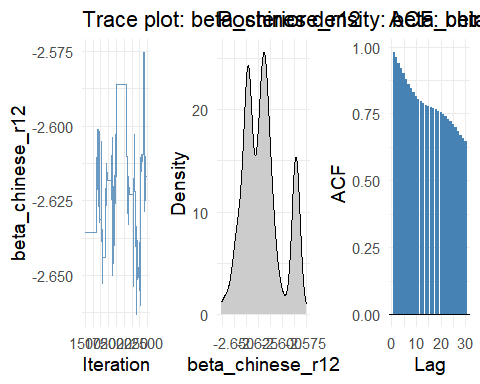


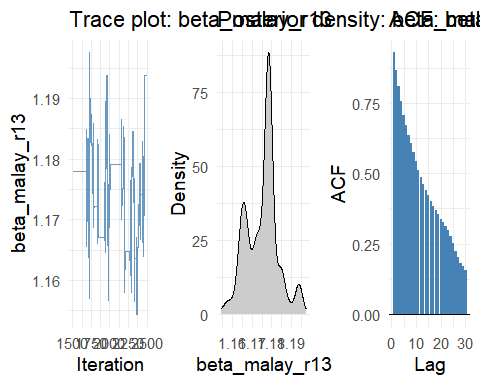


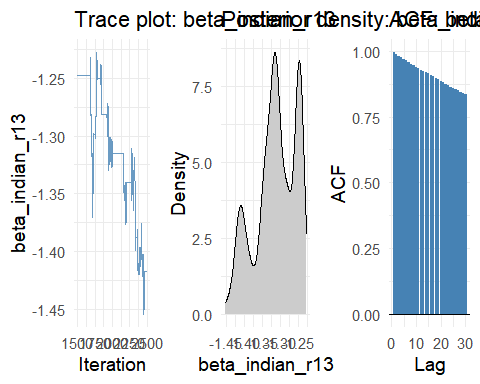


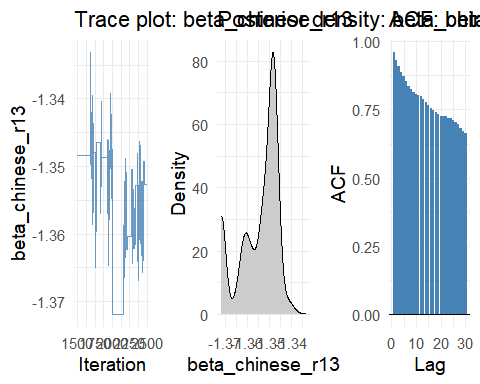












## 7. Posterior Summaries (Mean, SD, 95% Intervals)

posterior\_summary <- post %>%  
 summarise(  
 mean\_beta0 = mean(beta0),  
 sd\_beta0 = sd(beta0),  
 `2.5%\_beta0` = quantile(beta0, 0.025),  
 `97.5%\_beta0` = quantile(beta0, 0.975),  
   
 mean\_beta\_notable = mean(beta\_notable),  
 sd\_beta\_notable = sd(beta\_notable),  
 `2.5%\_beta\_notable` = quantile(beta\_notable, 0.025),  
 `97.5%\_beta\_notable` = quantile(beta\_notable, 0.975),  
   
 mean\_mu\_malay = mean(mu\_malay),  
 sd\_mu\_malay = sd(mu\_malay),  
 `2.5%\_mu\_malay` = quantile(mu\_malay, 0.025),  
 `97.5%\_mu\_malay`= quantile(mu\_malay, 0.975),  
   
 mean\_mu\_indian = mean(mu\_indian),  
 sd\_mu\_indian = sd(mu\_indian),  
 `2.5%\_mu\_indian` = quantile(mu\_indian, 0.025),  
 `97.5%\_mu\_indian`= quantile(mu\_indian, 0.975),  
   
 mean\_mu\_chinese = mean(mu\_chinese),  
 sd\_mu\_chinese = sd(mu\_chinese),  
 `2.5%\_mu\_chinese` = quantile(mu\_chinese, 0.025),  
 `97.5%\_mu\_chinese`= quantile(mu\_chinese, 0.975),  
   
 across(starts\_with("beta\_malay\_r"), list(mean = ~mean(.), sd = ~sd(.),  
 low\_2.5 = ~quantile(., 0.025),  
 hi\_97.5 = ~quantile(., 0.975))),  
 across(starts\_with("beta\_indian\_r"), list(mean = ~mean(.), sd = ~sd(.),  
 low\_2.5 = ~quantile(., 0.025),  
 hi\_97.5 = ~quantile(., 0.975))),  
 across(starts\_with("beta\_chinese\_r"), list(mean = ~mean(.), sd = ~sd(.),  
 low\_2.5 = ~quantile(., 0.025),  
 hi\_97.5 = ~quantile(., 0.975))),  
 mean\_malay\_tau = mean(tau\_malay),  
 sd\_malay\_tau = sd(tau\_malay),  
 `2.5%\_tau\_malay` = quantile(tau\_malay, 0.025),  
 `97.5%\_tau\_malay`= quantile(tau\_malay, 0.975),  
   
 mean\_chinese\_tau = mean(tau\_chinese),  
 sd\_chinese\_tau = sd(tau\_chinese),  
 `2.5%\_tau\_chinese` = quantile(tau\_chinese, 0.025),  
 `97.5%\_tau\_chinese`= quantile(tau\_chinese, 0.975),  
   
 mean\_indian\_tau = mean(tau\_indian),  
 sd\_indian\_tau = sd(tau\_indian),  
 `2.5%\_tau\_indian` = quantile(tau\_indian, 0.025),  
 `97.5%\_tau\_indian`= quantile(tau\_indian, 0.975),  
 ) %>%  
 pivot\_longer(cols = everything(),  
 names\_to = "parameter",  
 values\_to = "value")  
  
print(posterior\_summary, n = 100)

# A tibble: 188 × 2  
 parameter value  
 <chr> <dbl>  
 1 mean\_beta0 0.381   
 2 sd\_beta0 0.00244  
 3 2.5%\_beta0 0.378   
 4 97.5%\_beta0 0.384   
 5 mean\_beta\_notable 0.363   
 6 sd\_beta\_notable 0.00298  
 7 2.5%\_beta\_notable 0.359   
 8 97.5%\_beta\_notable 0.368   
 9 mean\_mu\_malay -0.804   
 10 sd\_mu\_malay 1.09   
 11 2.5%\_mu\_malay -2.44   
 12 97.5%\_mu\_malay 1.08   
 13 mean\_mu\_indian 0.00315  
 14 sd\_mu\_indian 1.22   
 15 2.5%\_mu\_indian -1.91   
 16 97.5%\_mu\_indian 2.07   
 17 mean\_mu\_chinese 0.582   
 18 sd\_mu\_chinese 1.52   
 19 2.5%\_mu\_chinese -1.53   
 20 97.5%\_mu\_chinese 3.22   
 21 beta\_malay\_r1\_mean -0.466   
 22 beta\_malay\_r1\_sd 0.0103   
 23 beta\_malay\_r1\_low\_2.5 -0.479   
 24 beta\_malay\_r1\_hi\_97.5 -0.447   
 25 beta\_malay\_r2\_mean -0.127   
 26 beta\_malay\_r2\_sd 0.00496  
 27 beta\_malay\_r2\_low\_2.5 -0.137   
 28 beta\_malay\_r2\_hi\_97.5 -0.120   
 29 beta\_malay\_r3\_mean -0.511   
 30 beta\_malay\_r3\_sd 0.00282  
 31 beta\_malay\_r3\_low\_2.5 -0.516   
 32 beta\_malay\_r3\_hi\_97.5 -0.505   
 33 beta\_malay\_r4\_mean -0.290   
 34 beta\_malay\_r4\_sd 0.00370  
 35 beta\_malay\_r4\_low\_2.5 -0.298   
 36 beta\_malay\_r4\_hi\_97.5 -0.284   
 37 beta\_malay\_r5\_mean 0.140   
 38 beta\_malay\_r5\_sd 0.00903  
 39 beta\_malay\_r5\_low\_2.5 0.120   
 40 beta\_malay\_r5\_hi\_97.5 0.151   
 41 beta\_malay\_r6\_mean 0.00706  
 42 beta\_malay\_r6\_sd 0.00458  
 43 beta\_malay\_r6\_low\_2.5 -0.00546  
 44 beta\_malay\_r6\_hi\_97.5 0.0135   
 45 beta\_malay\_r7\_mean 0.203   
 46 beta\_malay\_r7\_sd 0.00563  
 47 beta\_malay\_r7\_low\_2.5 0.194   
 48 beta\_malay\_r7\_hi\_97.5 0.215   
 49 beta\_malay\_r8\_mean -0.191   
 50 beta\_malay\_r8\_sd 0.00635  
 51 beta\_malay\_r8\_low\_2.5 -0.204   
 52 beta\_malay\_r8\_hi\_97.5 -0.183   
 53 beta\_malay\_r9\_mean 0.678   
 54 beta\_malay\_r9\_sd 0.0100   
 55 beta\_malay\_r9\_low\_2.5 0.661   
 56 beta\_malay\_r9\_hi\_97.5 0.694   
 57 beta\_malay\_r10\_mean 0.510   
 58 beta\_malay\_r10\_sd 0.00740  
 59 beta\_malay\_r10\_low\_2.5 0.498   
 60 beta\_malay\_r10\_hi\_97.5 0.528   
 61 beta\_malay\_r11\_mean 0.961   
 62 beta\_malay\_r11\_sd 0.00586  
 63 beta\_malay\_r11\_low\_2.5 0.950   
 64 beta\_malay\_r11\_hi\_97.5 0.973   
 65 beta\_malay\_r12\_mean 2.00   
 66 beta\_malay\_r12\_sd 0.0246   
 67 beta\_malay\_r12\_low\_2.5 1.95   
 68 beta\_malay\_r12\_hi\_97.5 2.07   
 69 beta\_malay\_r13\_mean 1.18   
 70 beta\_malay\_r13\_sd 0.00804  
 71 beta\_malay\_r13\_low\_2.5 1.16   
 72 beta\_malay\_r13\_hi\_97.5 1.19   
 73 beta\_indian\_r1\_mean 0.0601   
 74 beta\_indian\_r1\_sd 0.0511   
 75 beta\_indian\_r1\_low\_2.5 -0.0423   
 76 beta\_indian\_r1\_hi\_97.5 0.130   
 77 beta\_indian\_r2\_mean -1.16   
 78 beta\_indian\_r2\_sd 0.0325   
 79 beta\_indian\_r2\_low\_2.5 -1.24   
 80 beta\_indian\_r2\_hi\_97.5 -1.11   
 81 beta\_indian\_r3\_mean 1.68   
 82 beta\_indian\_r3\_sd 0.0859   
 83 beta\_indian\_r3\_low\_2.5 1.57   
 84 beta\_indian\_r3\_hi\_97.5 1.82   
 85 beta\_indian\_r4\_mean -0.964   
 86 beta\_indian\_r4\_sd 0.0239   
 87 beta\_indian\_r4\_low\_2.5 -1.03   
 88 beta\_indian\_r4\_hi\_97.5 -0.923   
 89 beta\_indian\_r5\_mean -0.979   
 90 beta\_indian\_r5\_sd 0.0395   
 91 beta\_indian\_r5\_low\_2.5 -1.08   
 92 beta\_indian\_r5\_hi\_97.5 -0.917   
 93 beta\_indian\_r6\_mean -0.697   
 94 beta\_indian\_r6\_sd 0.0361   
 95 beta\_indian\_r6\_low\_2.5 -0.752   
 96 beta\_indian\_r6\_hi\_97.5 -0.642   
 97 beta\_indian\_r7\_mean -0.650   
 98 beta\_indian\_r7\_sd 0.0434   
 99 beta\_indian\_r7\_low\_2.5 -0.747   
100 beta\_indian\_r7\_hi\_97.5 -0.580   
# ℹ 88 more rows