BAYESIAN PROJECT

Ana Mendoza

2025-04-08

load and clean the dataset

df <- read.csv("Autism Studies Dataset.csv")  
  
df\_clean <- df[!is.na(df$Sample.Size) & !is.na(df$Number.of.Cases), ]  
  
df\_clean$y <- df\_clean$Number.of.Cases  
df\_clean$n <- df\_clean$Sample.Size

prepare data for jags

data\_list <- list(  
 y = as.integer(df\_clean$y),  
 n = as.integer(df\_clean$n),  
 N = nrow(df\_clean)  
)

Here, we build the basic model where each study’s observed cases follow a binomial distribution. We assume the true prevalence rate for each study is unknown, and we give it a flat (uniform) prior, meaning we don’t assume anything ahead of time.

library(rjags)

## Loading required package: coda

## Linked to JAGS 4.3.2

## Loaded modules: basemod,bugs

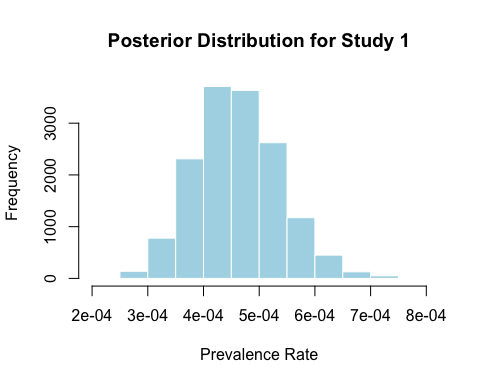
model\_string <- "  
model {  
 for (i in 1:N) {  
 y[i] ~ dbin(theta[i], n[i])  
 theta[i] ~ dbeta(1, 1)  
 }  
}  
"  
  
  
jags\_model <- jags.model(textConnection(model\_string), data = data\_list, n.chains = 3)

## Compiling model graph  
## Resolving undeclared variables  
## Allocating nodes  
## Graph information:  
## Observed stochastic nodes: 173  
## Unobserved stochastic nodes: 173  
## Total graph size: 521  
##   
## Initializing model

update(jags\_model, 1000) # Burn-in  
samples <- coda.samples(jags\_model, variable.names = c("theta"), n.iter = 5000)

This chunk plots the posterior distribution for just the first study. It shows what we believe the true prevalence rate is for that specific study after looking at the data.

posterior\_matrix <- as.matrix(samples)  
  
hist(posterior\_matrix[, 1],  
 main = "Posterior Distribution for Study 1",  
 xlab = "Prevalence Rate",  
 col = "lightblue", border = "white")



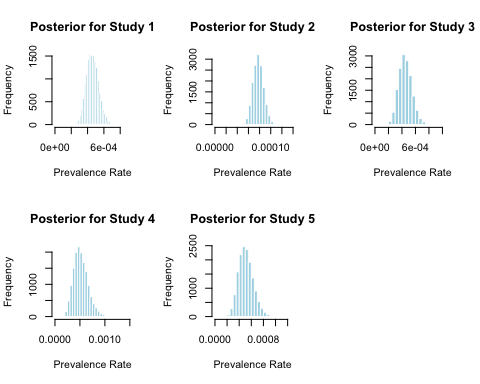
Posterior Summary for All Studies: Instead of looking at just one study, here we summarize the posterior results for all the studies by calculating the mean, standard deviation, and credible intervals for each estimated prevalence rate

library(coda)  
  
summary\_stats <- summary(samples)  
theta\_stats <- summary\_stats$statistics  
theta\_quantiles <- summary\_stats$quantiles  
  
posterior\_summary <- data.frame(  
 Mean = theta\_stats[, "Mean"],  
 SD = theta\_stats[, "SD"],  
 `2.5%` = theta\_quantiles[, "2.5%"],  
 `97.5%` = theta\_quantiles[, "97.5%"]  
)  
head(posterior\_summary)

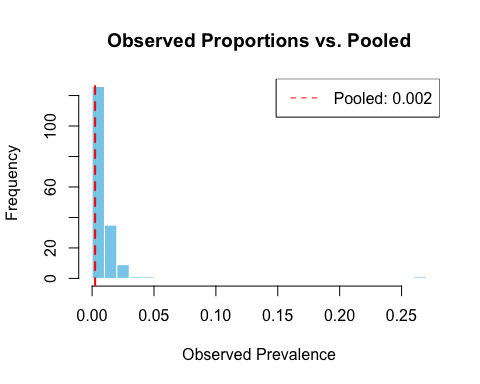
## Mean SD X2.5. X97.5.  
## theta[1] 4.610602e-04 7.594008e-05 3.233658e-04 6.215652e-04  
## theta[2] 7.773885e-05 9.398013e-06 6.031951e-05 9.700536e-05  
## theta[3] 4.515696e-04 9.789694e-05 2.769295e-04 6.583041e-04  
## theta[4] 5.201641e-04 1.468813e-04 2.740996e-04 8.520359e-04  
## theta[5] 5.125923e-04 1.216283e-04 3.031600e-04 7.755342e-04  
## theta[6] 5.258104e-04 2.959920e-05 4.688938e-04 5.849680e-04

Plot Posterior Distributions for Multiple Studies: This chunk creates a graph showing the density curves for the first five studies’ posterior distributions. It lets us compare how different (or similar) the prevalence rates seem across studies.

par(mfrow = c(2, 3))  
  
for (i in 1:5) {  
   
 max\_value <- max(posterior\_matrix[,i])  
   
 hist(posterior\_matrix[,i],   
 main = paste("Posterior for Study", i),  
 xlab = "Prevalence Rate",  
 col = "lightblue",  
 border = "white",  
 breaks = 20,  
 xlim = c(0, max\_value \* 1.2)) # Zoom based on data  
}

 Estimate and Plot Overall Prevalence Rate (Pooled):Here we calculate the overall prevalence rate by pooling all the data together. We also plot it on top of the observed proportions to compare individual studies versus the pooled estimate.

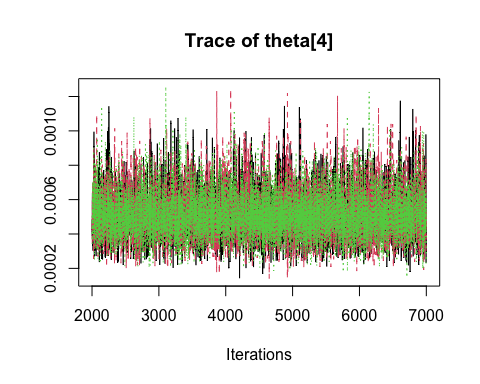
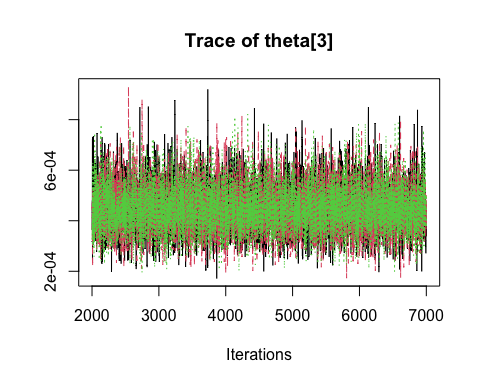
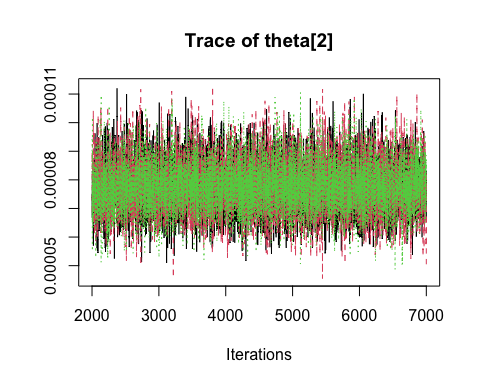
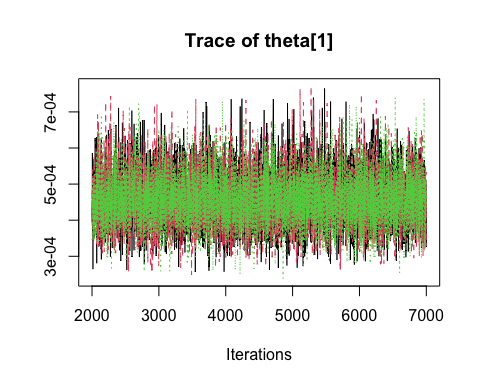
df\_clean$p\_hat <- df\_clean$y / df\_clean$n  
pooled\_p\_hat <- sum(df\_clean$y) / sum(df\_clean$n)  
  
hist(df\_clean$p\_hat, breaks=20, main="Observed Proportions vs. Pooled",  
 xlab="Observed Prevalence", col="skyblue", border="white")  
abline(v = pooled\_p\_hat, col="red", lwd=2, lty=2)  
legend("topright", legend=paste("Pooled:", round(pooled\_p\_hat, 3)), col="red", lty=2)

 Check Convergence Diagnostics: Before trusting our results, we use diagnostics to make sure the model actually converged and that the Markov chains are behaving nicely.

gelman\_diag <- gelman.diag(samples)  
print(gelman\_diag)

## Potential scale reduction factors:  
##   
## Point est. Upper C.I.  
## theta[1] 1 1.00  
## theta[2] 1 1.00  
## theta[3] 1 1.00  
## theta[4] 1 1.00  
## theta[5] 1 1.00  
## theta[6] 1 1.00  
## theta[7] 1 1.00  
## theta[8] 1 1.00  
## theta[9] 1 1.01  
## theta[10] 1 1.00  
## theta[11] 1 1.00  
## theta[12] 1 1.00  
## theta[13] 1 1.00  
## theta[14] 1 1.00  
## theta[15] 1 1.00  
## theta[16] 1 1.00  
## theta[17] 1 1.00  
## theta[18] 1 1.00  
## theta[19] 1 1.00  
## theta[20] 1 1.00  
## theta[21] 1 1.00  
## theta[22] 1 1.00  
## theta[23] 1 1.00  
## theta[24] 1 1.00  
## theta[25] 1 1.00  
## theta[26] 1 1.00  
## theta[27] 1 1.00  
## theta[28] 1 1.00  
## theta[29] 1 1.00  
## theta[30] 1 1.00  
## theta[31] 1 1.00  
## theta[32] 1 1.00  
## theta[33] 1 1.00  
## theta[34] 1 1.00  
## theta[35] 1 1.00  
## theta[36] 1 1.00  
## theta[37] 1 1.00  
## theta[38] 1 1.00  
## theta[39] 1 1.00  
## theta[40] 1 1.00  
## theta[41] 1 1.00  
## theta[42] 1 1.00  
## theta[43] 1 1.00  
## theta[44] 1 1.00  
## theta[45] 1 1.00  
## theta[46] 1 1.00  
## theta[47] 1 1.00  
## theta[48] 1 1.00  
## theta[49] 1 1.00  
## theta[50] 1 1.00  
## theta[51] 1 1.00  
## theta[52] 1 1.00  
## theta[53] 1 1.00  
## theta[54] 1 1.00  
## theta[55] 1 1.00  
## theta[56] 1 1.00  
## theta[57] 1 1.00  
## theta[58] 1 1.00  
## theta[59] 1 1.00  
## theta[60] 1 1.00  
## theta[61] 1 1.00  
## theta[62] 1 1.01  
## theta[63] 1 1.00  
## theta[64] 1 1.00  
## theta[65] 1 1.00  
## theta[66] 1 1.00  
## theta[67] 1 1.00  
## theta[68] 1 1.00  
## theta[69] 1 1.00  
## theta[70] 1 1.00  
## theta[71] 1 1.00  
## theta[72] 1 1.00  
## theta[73] 1 1.00  
## theta[74] 1 1.00  
## theta[75] 1 1.00  
## theta[76] 1 1.00  
## theta[77] 1 1.00  
## theta[78] 1 1.00  
## theta[79] 1 1.01  
## theta[80] 1 1.00  
## theta[81] 1 1.00  
## theta[82] 1 1.00  
## theta[83] 1 1.00  
## theta[84] 1 1.00  
## theta[85] 1 1.00  
## theta[86] 1 1.00  
## theta[87] 1 1.01  
## theta[88] 1 1.00  
## theta[89] 1 1.00  
## theta[90] 1 1.00  
## theta[91] 1 1.00  
## theta[92] 1 1.00  
## theta[93] 1 1.00  
## theta[94] 1 1.00  
## theta[95] 1 1.00  
## theta[96] 1 1.00  
## theta[97] 1 1.00  
## theta[98] 1 1.00  
## theta[99] 1 1.00  
## theta[100] 1 1.00  
## theta[101] 1 1.00  
## theta[102] 1 1.00  
## theta[103] 1 1.00  
## theta[104] 1 1.00  
## theta[105] 1 1.00  
## theta[106] 1 1.00  
## theta[107] 1 1.01  
## theta[108] 1 1.00  
## theta[109] 1 1.00  
## theta[110] 1 1.00  
## theta[111] 1 1.00  
## theta[112] 1 1.00  
## theta[113] 1 1.00  
## theta[114] 1 1.00  
## theta[115] 1 1.00  
## theta[116] 1 1.00  
## theta[117] 1 1.00  
## theta[118] 1 1.00  
## theta[119] 1 1.00  
## theta[120] 1 1.00  
## theta[121] 1 1.00  
## theta[122] 1 1.00  
## theta[123] 1 1.00  
## theta[124] 1 1.00  
## theta[125] 1 1.00  
## theta[126] 1 1.00  
## theta[127] 1 1.00  
## theta[128] 1 1.00  
## theta[129] 1 1.00  
## theta[130] 1 1.00  
## theta[131] 1 1.00  
## theta[132] 1 1.00  
## theta[133] 1 1.00  
## theta[134] 1 1.00  
## theta[135] 1 1.00  
## theta[136] 1 1.00  
## theta[137] 1 1.00  
## theta[138] 1 1.00  
## theta[139] 1 1.00  
## theta[140] 1 1.00  
## theta[141] 1 1.00  
## theta[142] 1 1.00  
## theta[143] 1 1.00  
## theta[144] 1 1.00  
## theta[145] 1 1.00  
## theta[146] 1 1.00  
## theta[147] 1 1.00  
## theta[148] 1 1.00  
## theta[149] 1 1.00  
## theta[150] 1 1.00  
## theta[151] 1 1.00  
## theta[152] 1 1.00  
## theta[153] 1 1.00  
## theta[154] 1 1.00  
## theta[155] 1 1.00  
## theta[156] 1 1.00  
## theta[157] 1 1.00  
## theta[158] 1 1.00  
## theta[159] 1 1.00  
## theta[160] 1 1.00  
## theta[161] 1 1.00  
## theta[162] 1 1.00  
## theta[163] 1 1.00  
## theta[164] 1 1.00  
## theta[165] 1 1.00  
## theta[166] 1 1.00  
## theta[167] 1 1.00  
## theta[168] 1 1.00  
## theta[169] 1 1.00  
## theta[170] 1 1.00  
## theta[171] 1 1.00  
## theta[172] 1 1.00  
## theta[173] 1 1.00  
##   
## Multivariate psrf  
##   
## 1.03

traceplot(samples[, 1:4])

 Add a Hierarchical Beta-Binomial Model (Hyperprior):Instead of treating each study as completely independent, here we build a hierarchical model where all studies share a common Beta distribution, and we estimate its parameters (alpha and beta) too.

model\_hierarchical <- "  
model {  
 for (i in 1:N) {  
 y[i] ~ dbin(theta[i], n[i])  
 theta[i] ~ dbeta(alpha, beta)  
 }  
   
 alpha ~ dgamma(1, 0.01)  
 beta ~ dgamma(1, 0.01)  
}  
"  
  
jags\_model\_hier <- jags.model(textConnection(model\_hierarchical), data = data\_list, n.chains = 3)

## Compiling model graph  
## Resolving undeclared variables  
## Allocating nodes  
## Graph information:  
## Observed stochastic nodes: 173  
## Unobserved stochastic nodes: 175  
## Total graph size: 524  
##   
## Initializing model

update(jags\_model\_hier, 1000)  
samples\_hier <- coda.samples(jags\_model\_hier, variable.names = c("theta", "alpha", "beta"), n.iter = 5000)

Analyzing Hyperparameters and Posterior Densities: After fitting the hierarchical model, this code looks at the estimated values of alpha and beta and plots the posterior distributions for a few studies under the new model.

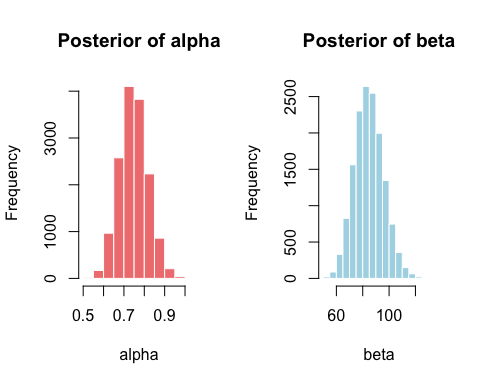
hier\_matrix <- as.matrix(samples\_hier)  
  
alpha\_post <- hier\_matrix[, "alpha"]  
beta\_post <- hier\_matrix[, "beta"]  
  
cat("Posterior mean of alpha:", round(mean(alpha\_post), 3), "\n")

## Posterior mean of alpha: 0.748

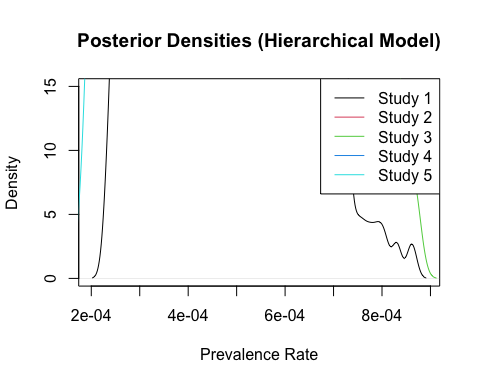
cat("Posterior mean of beta:", round(mean(beta\_post), 3), "\n")

## Posterior mean of beta: 84.885

# Plots  
par(mfrow=c(1,2))  
hist(alpha\_post, main="Posterior of alpha", col="lightcoral", xlab="alpha", border="white")  
hist(beta\_post, main="Posterior of beta", col="lightblue", xlab="beta", border="white")



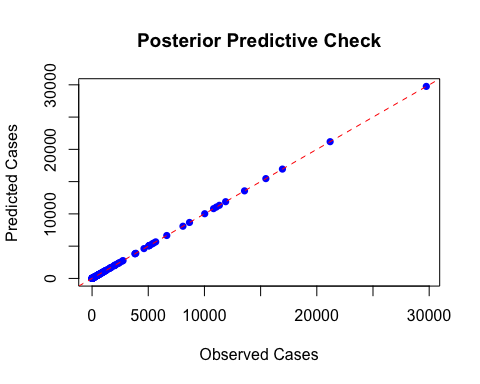
par(mfrow=c(1,1))  
plot(density(hier\_matrix[, "theta[1]"]), main="Posterior Densities (Hierarchical Model)",  
 xlab="Prevalence Rate", ylim=c(0, 15), col=1)  
for (i in 2:5) {  
 lines(density(hier\_matrix[, paste0("theta[", i, "]")]), col=i)  
}  
legend("topright", legend=paste("Study", 1:5), col=1:5, lty=1)

 Posterior Predictive Checks (PPC): Posterior predictive checks are like a “reality check” , we simulate new fake datasets from our model and see if they match the real-world data we observed.

model\_ppc <- "  
model {  
 for (i in 1:N) {  
 y[i] ~ dbin(theta[i], n[i])  
 y\_rep[i] ~ dbin(theta[i], n[i])  
 theta[i] ~ dbeta(1, 1)  
 }  
}  
"  
  
data\_list\_ppc <- data\_list  
jags\_ppc <- jags.model(textConnection(model\_ppc), data = data\_list\_ppc, n.chains = 3)

## Compiling model graph  
## Resolving undeclared variables  
## Allocating nodes  
## Graph information:  
## Observed stochastic nodes: 173  
## Unobserved stochastic nodes: 346  
## Total graph size: 694  
##   
## Initializing model

update(jags\_ppc, 1000)  
ppc\_samples <- coda.samples(jags\_ppc, variable.names = c("y\_rep"), n.iter = 5000)  
  
y\_rep\_matrix <- as.matrix(ppc\_samples)  
y\_rep\_means <- colMeans(y\_rep\_matrix)  
  
plot(df\_clean$y, y\_rep\_means, main="Posterior Predictive Check",  
 xlab="Observed Cases", ylab="Predicted Cases", col="blue", pch=16)  
abline(0, 1, col="red", lty=2)

 Mapping Autism Prevalence by Country: If our dataset includes country names, we can plot the estimated prevalence rates across the world to spot geographic patterns.

library(ggplot2)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(rnaturalearth)  
library(rnaturalearthdata)

##   
## Attaching package: 'rnaturalearthdata'

## The following object is masked from 'package:rnaturalearth':  
##   
## countries110

country\_means <- df\_clean %>%  
 mutate(theta\_mean = colMeans(posterior\_matrix)) %>%  
 group\_by(Country) %>%  
 summarise(MeanPrevalence = mean(theta\_mean))  
  
world <- ne\_countries(scale = "medium", returnclass = "sf")  
  
world\_map <- left\_join(world, country\_means, by = c("name" = "Country"))  
  
ggplot(data = world\_map) +  
 geom\_sf(aes(fill = MeanPrevalence)) +  
 scale\_fill\_viridis\_c(option = "plasma", na.value = "grey90") +  
 labs(title = "Estimated Autism Prevalence by Country",  
 fill = "Prevalence") +  
 theme\_minimal()

