Project: Introduction to Bayesian Inference (3579)
Project 4
2023-2024

1st year Master of Statistics
Hasselt University

**Group members:**
Ronald Makanga (2365466)
Juan Vanegas Jadan (2366405)
Gianni Guarraci (2366293)
Charles Muiruri (2365043)

*Submission Date:* 01 June, 2024

**Lecturers:**
Prof. Dr. Christel FAES

# Contents

# Question 1:

```
## Abstracting beta0 ... 8000 valid values
## Abstracting beta1 ... 8000 valid values
## Abstracting beta2 ... 8000 valid values
## Abstracting deviance ... 8000 valid values
## Abstracting beta0 ... 8000 valid values
## Abstracting beta1 ... 8000 valid values
## Abstracting beta2 ... 8000 valid values
## Abstracting deviance ... 8000 valid values
## Abstracting beta0 ... 8000 valid values
## Abstracting beta1 ... 8000 valid values
## Abstracting beta2 ... 8000 valid values
## Abstracting deviance ... 8000 valid values
```

We ran a model using openBugs where we are estimating 3 unknown parameters $\beta_0$, $\beta_1$, and $\beta_2$ to model the probability of vaccination coverage for children in different poverty groups. The model is defined as follows:

## Bugs model

```
model
{
    for (i in 1:J) {
        Y[i] ~ dbin(p[i], N[i])
        logit(p[i]) <- beta0 + beta1 * btn133_400_pc[i] + beta2 *
            great_400pc[i]
    }
    beta0 ~ dnorm(0.00000E+00, 0.001)
    beta1 ~ dnorm(0.00000E+00, 0.001)
    beta2 ~ dnorm(0.00000E+00, 0.001)
}
```

```
##
## Iterations = 2001:10000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 8000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##              Mean      SD  Naive SE Time-series SE
## beta0      1.9208 0.05503 0.0003552      0.0004108
## beta1      0.3675 0.07584 0.0004895      0.0005638
## beta2      0.9763 0.08994 0.0005806      0.0006556
## deviance 254.0282 2.43933 0.0157458      0.0178846
##
## 2. Quantiles for each variable:
##
##              2.5%      25%      50%      75%    97.5%
## beta0      1.8140   1.8840   1.9200   1.9580   2.0300
## beta1      0.2189   0.3163   0.3671   0.4189   0.5157
## beta2      0.7994   0.9154   0.9766   1.0370   1.1530
## deviance 251.3000 252.2000 253.4000 255.1000 260.5000
```

The mean probability of vaccination coverage for each poverty group is as follows: * $<$ 133% FPL: 0.8722314

* 133% to <400% FPL: 0.9079025 * > 400% FPL: 0.9477026

# Question 2

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 27
##    Unobserved stochastic nodes: 3
##    Total graph size: 124
##
## Initializing model
```

### MCMC method and check converge of the MCMC chains Convergence tests

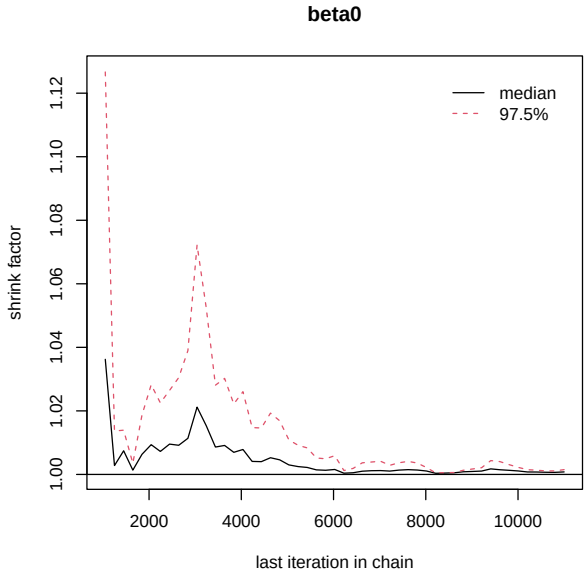
In our analysis, we used Gibbs Sampling to estimate the posterior distributions for our three unknown parameters (beta0, beta1 and beta2). To test for convergence for our MCMC chains, we used both Gelman-Rubin convergence diagnostic and trace plots.

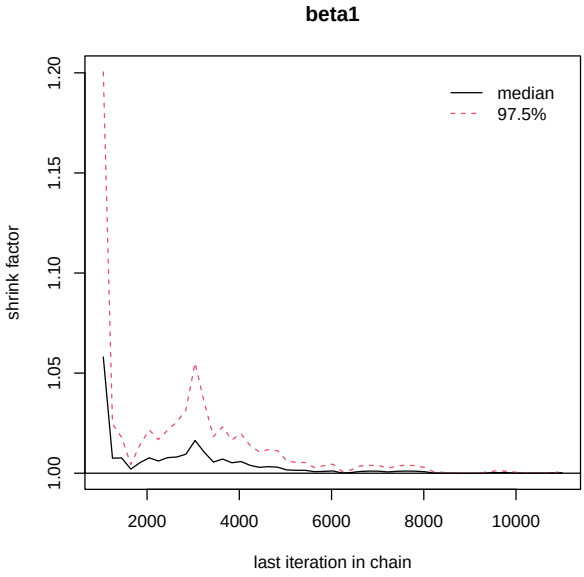
The Gelman-Rubin convergence diagnostic method allows us to compare within and between chain variances for each variable. Best results are obtained for parameters whose marginal posterior densities are approximately normal. To run the Gelman-Rubin convergence test, we set three MCMC chains and for each chain we set distinct starting values for our unknown parameters as shown in the code. We ran 10,000 MCMC trials with a burn-in value of 2,000 and a thining value of 1 for each chain.

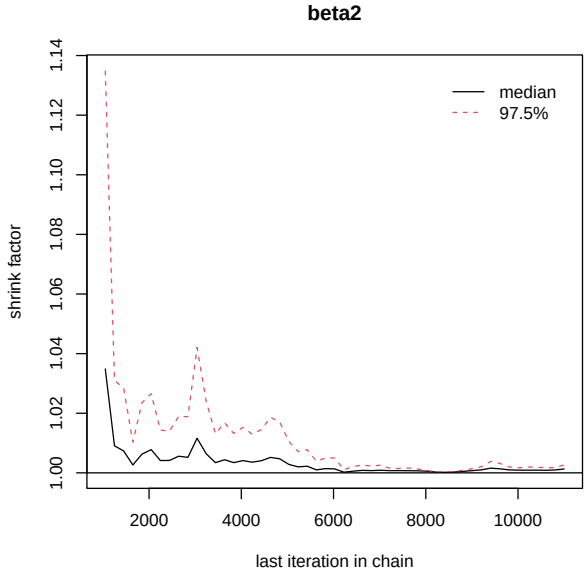
Then we produced gelman-rubin statistics and plot using the gelman.diag and gelman.plot functions in R. The gelman.diag function gives us the scale reduction factors for each parameter (beta0,beta1,and beta2). A factor of 1 means that the between variance and within chain variance are equal, larger values mean that there is still a notable difference between chains. on the other hand, gelman.plot shows if the shrink factor has really converged, or whether it still fluctuating. Both are results from both our functions.

### Gelman-Rubin Convergence Diagnostic

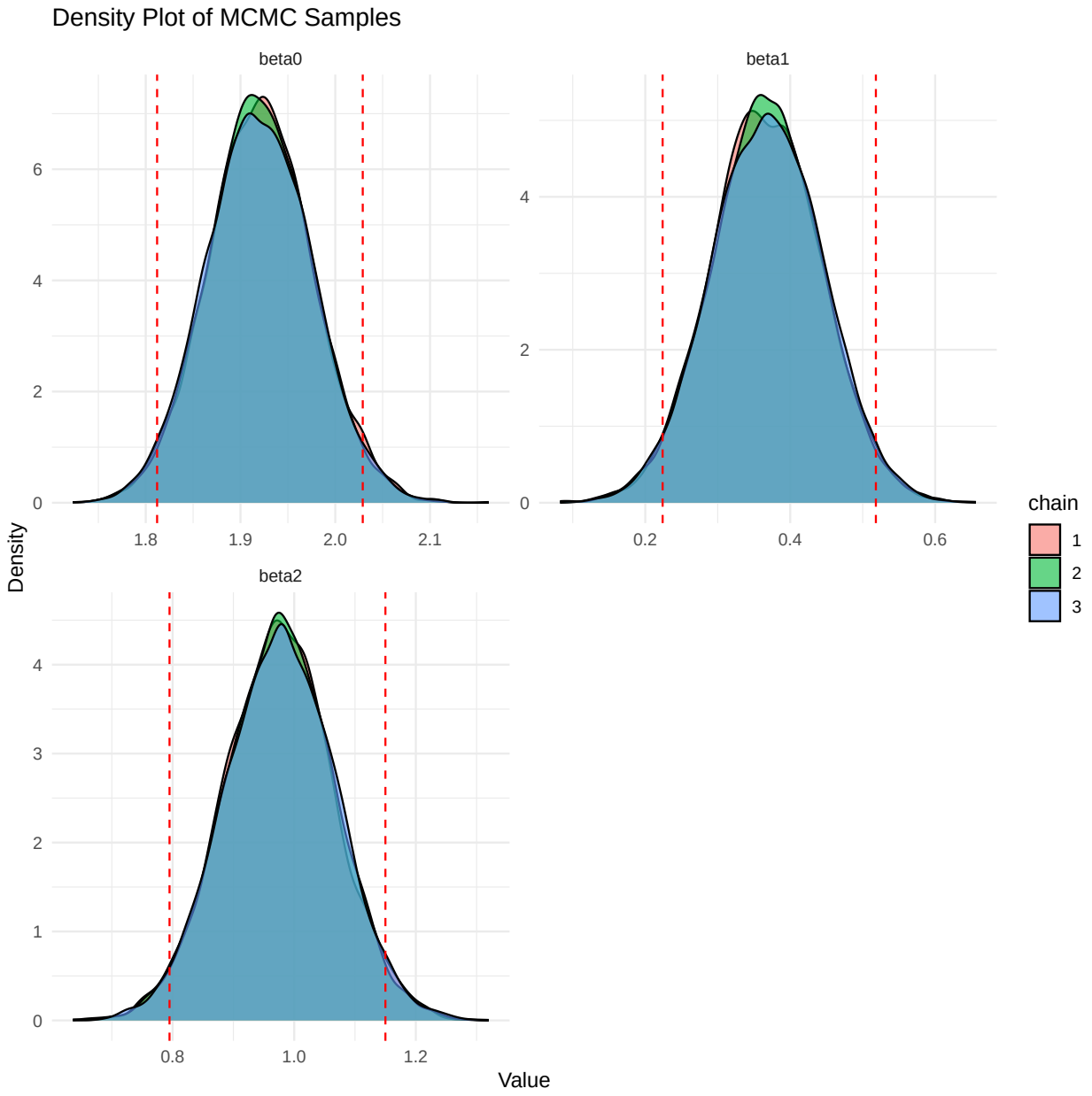
```
## Potential scale reduction factors:
##
##       Point est. Upper C.I.
## beta0          1          1
## beta1          1          1
## beta2          1          1
##
## Multivariate psrf
##
## 1
```


beta0
median
97.5%
shrink factor
1.00
1.02
1.04
1.06
1.08
1.10
1.12
2000
4000
6000
8000
10000
last iteration in chain


beta1
median
97.5%
shrink factor
1.00
1.05
1.10
1.15
1.20
2000
4000
6000
8000
10000
last iteration in chain


beta2
median
97.5%
shrink factor
1.00
1.02
1.04
1.06
1.08
1.10
1.12
1.14
2000
4000
6000
8000
10000
last iteration in chain

# Question 3

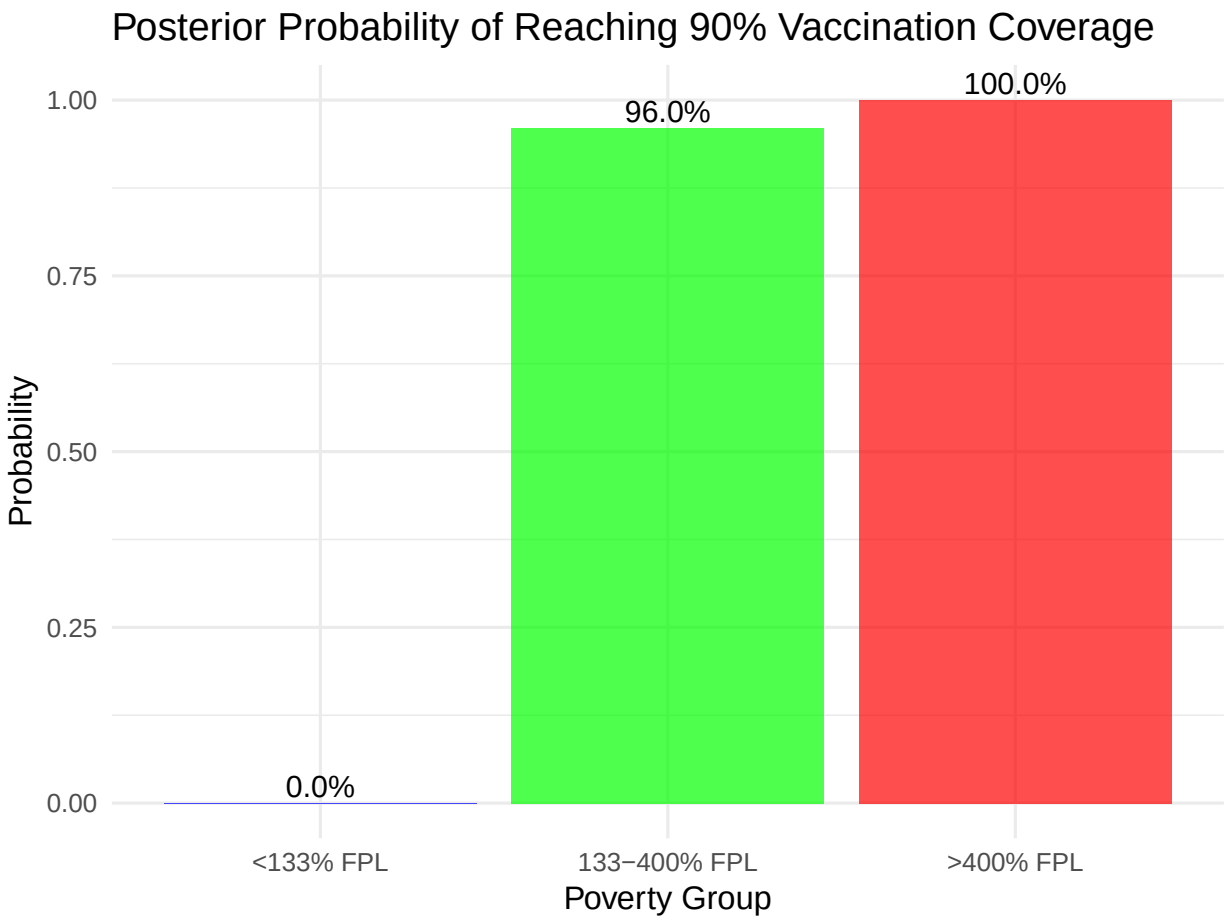


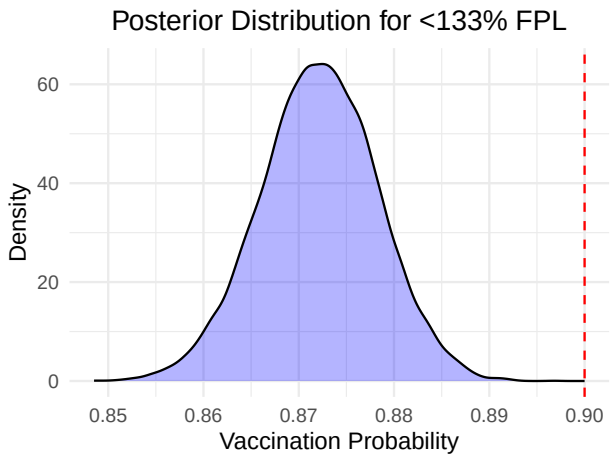

## Summary of posterior distribution

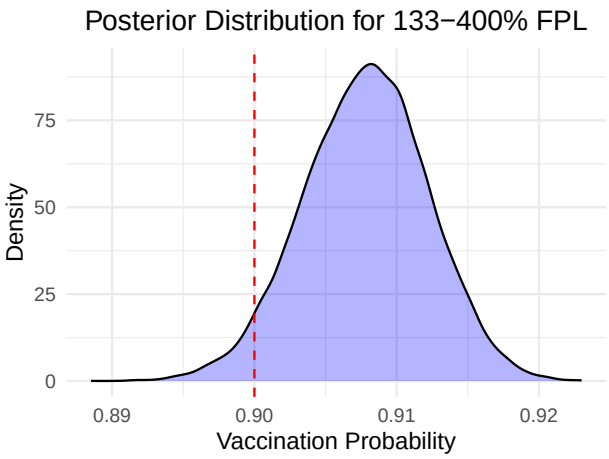
- The Posterior density with relation to $\beta_0$ has a mean of **0.8722314** which is the probability of children vaccinated from households whose income is less than 133%, with a 95% credible HPD interval of **(0.8596034, 0.8838084)** in which the proportion lies with 95% probability.
- The Posterior density with relation to $\beta_1$ has a mean of **0.9079025** which is the probability of children vaccinated from households whose income between 133% and 400% , with a 95% credible HPD interval of **(0.8845253, 0.9273852)** in which the proportion lies with 95% probability.
- The Posterior density with relation to $\beta_2$ has a mean of **0.9477026** which is the probability of children vaccinated from households whose income is more than 400%, with a 95% credible HPD interval of**(0.9313043, 0.9600363)** in which the proportion lies with 95% probability.

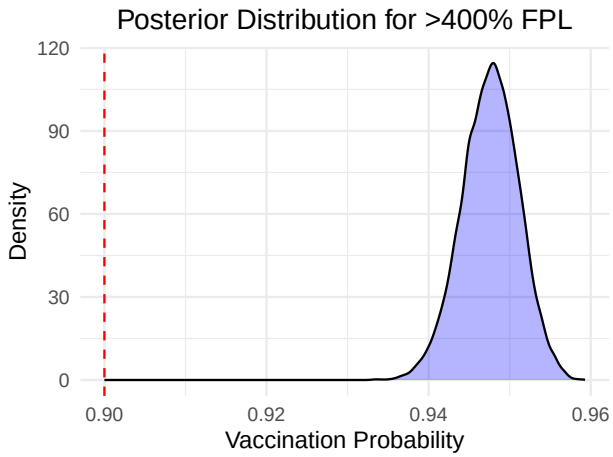
## Question 4

1:3, c(0, 0.960166666666667, 1)





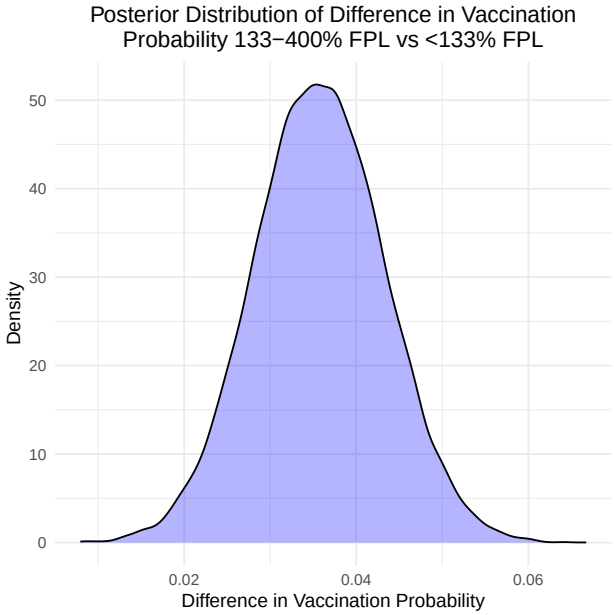



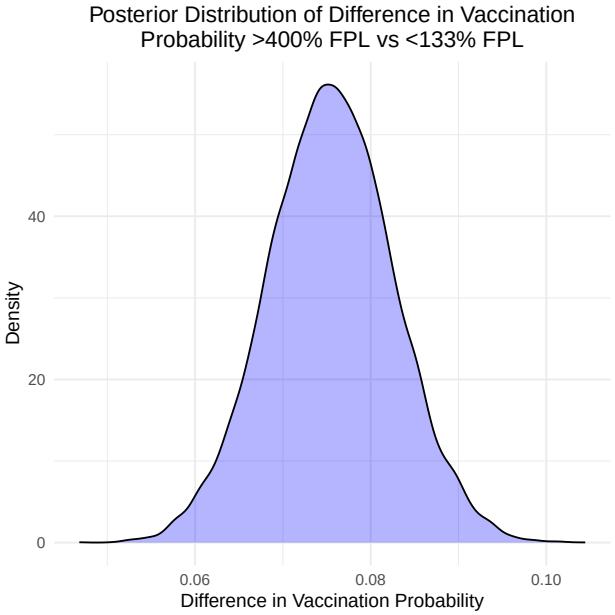



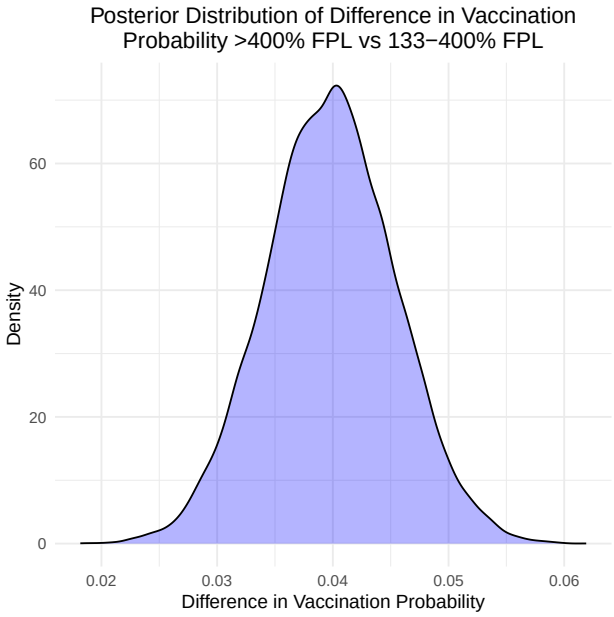

## Posterior Probabilities of Reaching 90% Vaccination Coverage:

- $<$ 133% FPL: Only 87.21% of the time does the vaccination coverage meet or exceed 90%, indicating lower vaccination rates in this poverty group.
- 133-400% FPL: With a 90.78% probability, this group has a better chance of meeting the 90% target, showing improvement compared to the <133% FPL group.
- $>$ 400% FPL: This group has a 94.76% probability of reaching or exceeding 90% coverage, indicating high vaccination rates.

# Question 5




## Impact of Poverty on Vaccination Coverage:

- 133-400% FPL vs <133% FPL: The mean difference in vaccination coverage is 0.0357, suggests that vaccination coverage is higher by about 3.57% in the 133-400% FPL group compared to the <133% FPL group.
- > 400% FPL vs <133% FPL: A mean difference of in vaccination coverage is 0.0755, indicates that the >400% FPL group has a higher vaccination coverage by about 7.55% compared to the <133% FPL group.
- > 400% FPL vs 133-400% FPL: The mean difference of in vaccination coverage is 0.0398, shows that the highest income group (>400% FPL) has about 3.98% higher coverage compared to the middle-income

group (133-400% FPL).

### Insights

Income Gradient in Vaccination Coverage:

- There is a clear gradient in vaccination coverage by poverty level, with higher-income groups achieving better vaccination rates. This suggests that income is a significant factor in vaccination coverage.

### Target Achievement:

- The highest income group (>400% FPL) consistently meets or exceeds the 90% vaccination coverage target, indicating successful vaccination efforts in this group. Lower-income groups, especially the <133% FPL group, struggle to meet the target. highlighting the need for targeted interventions.

## Question 6

Investigate whether the vaccination coverages are distinct at the different locations by adding a location-specific intercept.

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 27
##    Unobserved stochastic nodes: 11
##    Total graph size: 208
##
## Initializing model
```

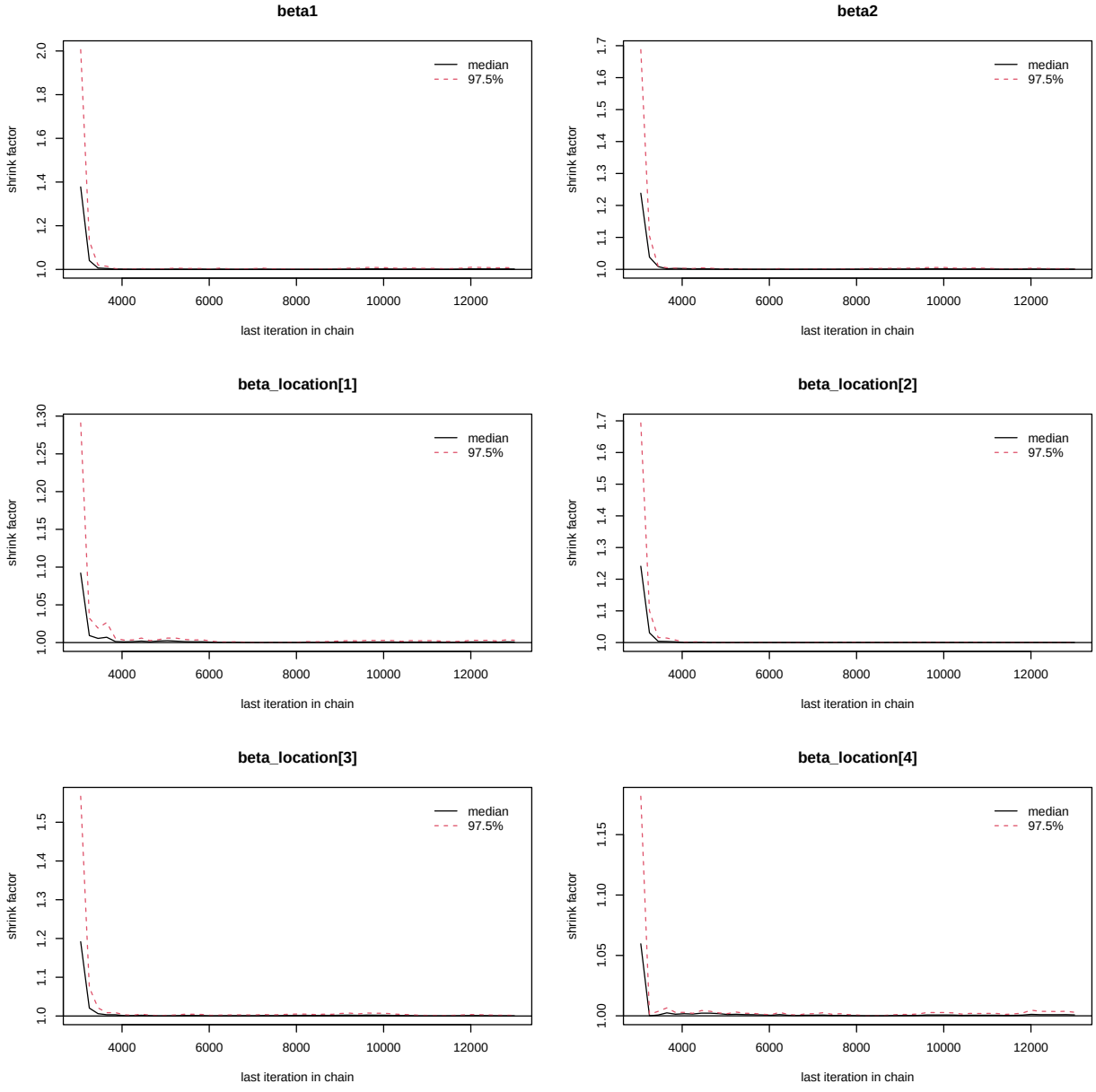
### Bugs model

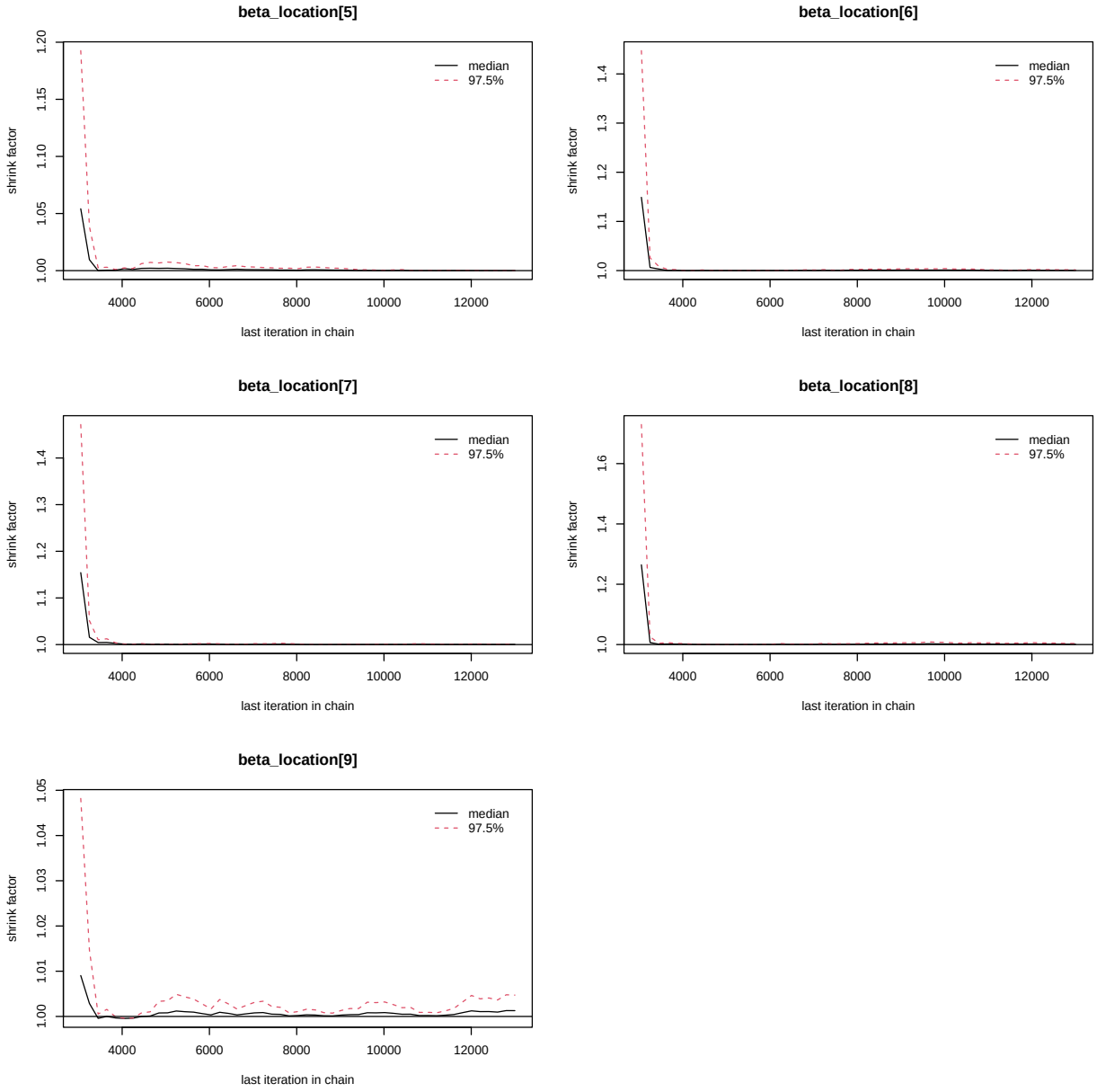
```
model
{
    for (i in 1:J) {
        Y[i] ~ dbin(p[i], N[i])
        logit(p[i]) <- beta1 * btn133_400_pc[i] + beta2 * great_400pc[i] +
            beta_location[location[i]]
    }
    beta1 ~ dnorm(0.00000E+00, 0.001)
    beta2 ~ dnorm(0.00000E+00, 0.001)
    for (j in 1:J_locations) {
        beta_location[j] ~ dnorm(0.00000E+00, 0.001)
    }
}
```

### Gelman-Rubin Convergence Diagnostic

```
## Potential scale reduction factors:
##
##                   Point est. Upper C.I.
## beta1                      1       1.01
## beta2                      1       1.00
## beta_location[1]           1       1.00
## beta_location[2]           1       1.00
```

```
## beta_location[3]          1       1.00
## beta_location[4]          1       1.00
## beta_location[5]          1       1.00
## beta_location[6]          1       1.00
## beta_location[7]          1       1.00
## beta_location[8]          1       1.00
## beta_location[9]          1       1.00
##
## Multivariate psrf
##
## 1
```

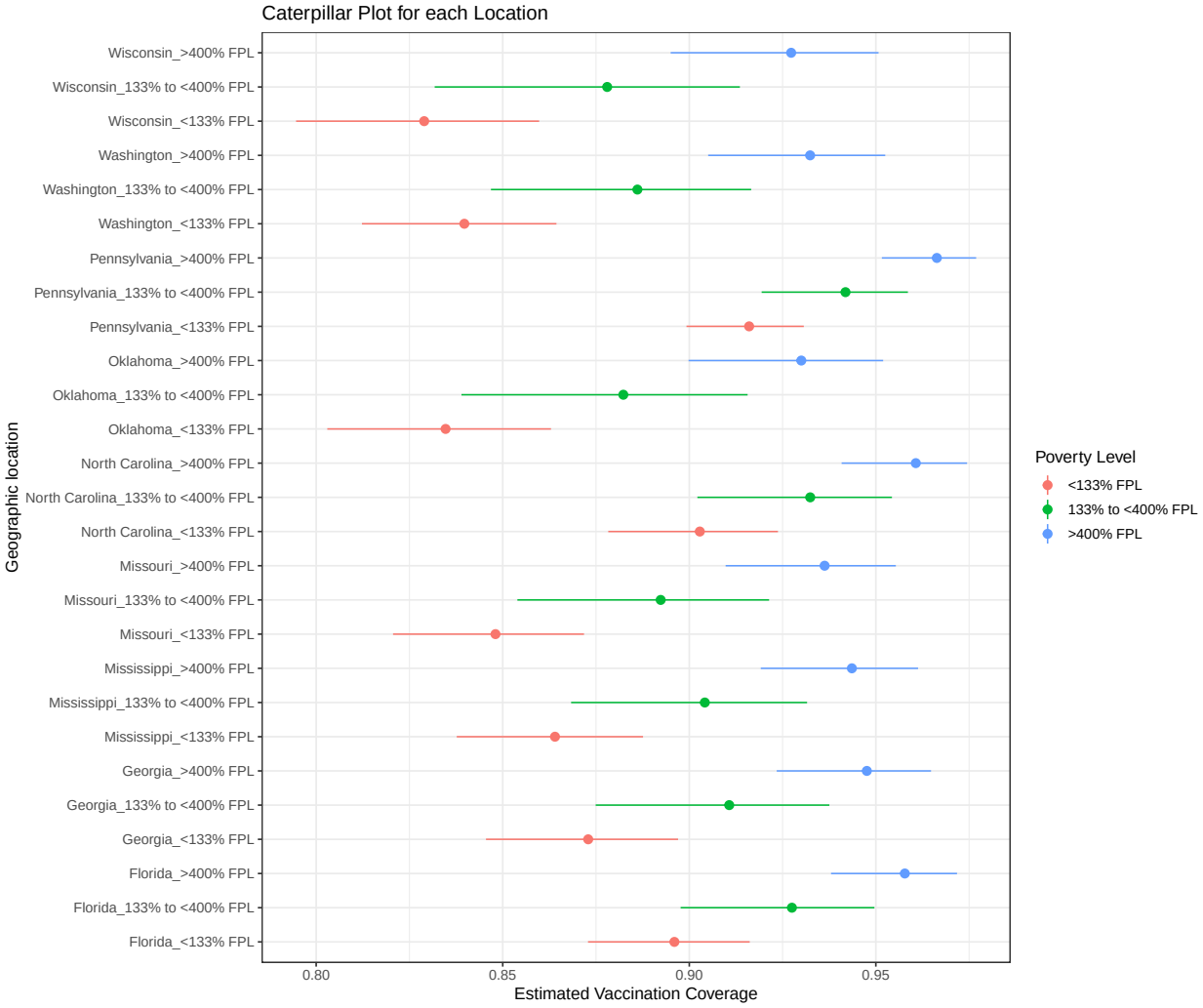





Convergence of the parameters is as shown in the Gelman-Rubin convergence diagnostic plot above. 1. $\beta_1$ and $\beta_2$ converge at 9000 iterations. 2. $\beta_{location_i}$ converges at different values, the least being 2000 and the highest being 9000 iterations.

In comparison to the model without location-specific intercepts, the model with location-specific intercepts has a higher number of iterations required for convergence. This is expected as the model complexity increases with the addition of location-specific intercepts, requiring more iterations to reach convergence.

# Question 7




- This caterpillar plot displays the expected coverage for each geographic region within our data. The point represents the mean coverage while the error bars represent the 95% credible interval. - The plot shows that the expected coverage for region with poverty index of less than 133% is the lowest across all the Geographic locations. - The region with poverty index between 133% and 400% has a higher expected coverage compared to the region with poverty index of less than 133%. - The region with poverty index greater than 400% has the highest expected coverage across all the Geographic locations.

# Question 8

| Poverty | Estimate |
|---|---|
| 133-400% FPL | 90 |
| >400% FPL | 94 |
| <133% FPL | 86 |
| Total | 270 |

This table shows the estimated number of vaccinated children in Mississippi by poverty group. The estimated number of vaccinated children from households is broken down as follows:
- From households with income less than 133% FPL is 86 children out of 100 children
- From households with income between 133% and 400% FPL is 90 children out of 100 children.
- From households with income greater than 400% FPL is 94 children out of 100 children.
- Overall vaccination coverage is 270 children out of 300 children.

# Appendix

```
# setup and Data

knitr::opts_chunk$set(
  echo = FALSE,
  message = FALSE,
  warning = FALSE,
  fig.height = 10,
  fig.width = 10
  )

if (!require(pacman)) install.packages("pacman")
p_load(rjags, coda, nimble, R2OpenBUGS, ggplot2, here, dplyr, ggpubr, tidyr, R2OpenBUGS)

projdata <- as.data.frame(read.csv(here("data/projectdata.txt")))|>
  mutate(less_133pc = ifelse(Poverty == "<133% FPL", 1,0),
         btn133_400_pc = ifelse(Poverty == "133% to <400% FPL", 1,0),
         great_400pc = ifelse(Poverty == ">400% FPL", 1,0))


# Question 1:

# Data prep for bugs
model_data <- list(
  Y = projdata$Vaccinated,
  N = projdata$Sample.Size,
  btn133_400_pc = projdata$btn133_400_pc,
  great_400pc = projdata$great_400pc,
  J = nrow(projdata)
)

parameters <- c("beta0", "beta1", "beta2")

model_inits <- list(
  list(beta0 = 1, beta1 = 0, beta2 = 0),
  list(beta0 = 1, beta1 = 1, beta2 = 0),
  list(beta0 = 1, beta1 = 0, beta2 = 1)
)

model1 <- function(){
  for (i in 1:J){
    Y[i] ~ dbin(p[i], N[i])
    logit(p[i]) <- beta0 + beta1*btn133_400_pc[i] + beta2*great_400pc[i]
  }
  #priors
```

```
  beta0 ~ dnorm(0, 0.001)
  beta1 ~ dnorm(0, 0.001)
  beta2 ~ dnorm(0, 0.001)
}

write.model(model1, here("models/model1code.txt"))

suppressMessages({
  suppressWarnings({
model.outq3 <- bugs(model_data, model_inits,
                  parameters = parameters, model.file = here("models/model1code.txt"),
                  n.chains = 3, n.iter = 10000, n.burnin = 2000, codaPkg = TRUE, debug = FALSE)


out2 <- read.bugs(model.outq3)

  })
})

inverse_logit <- function(x) {
  1 / (1 + exp(-x))
}
out.summary <- summary(out2)
beta0_mean <- out.summary[[1]]["beta0","Mean"]
beta1_mean <- out.summary[[1]]["beta1","Mean"]
beta2_mean <- out.summary[[1]]["beta2","Mean"]
pi_133 <- inverse_logit(beta0_mean)
pi_133_400 <- inverse_logit(beta1_mean + beta0_mean)
pi_400 <- inverse_logit(beta2_mean + beta0_mean)
out.summary

# Question 2


parames <- c("beta0", "beta1", "beta2")
set.seed(20240601)
# Run MCMC
suppressMessages({
  suppressWarnings({
    model_jag <- jags.model("model1code.txt", data = model_data, n.chains = 3)
    samples_jag <- coda.samples(model_jag, variable.names = parames, n.burnin = 1000, n.iter = 10000)
  })
})

gelman.diag(samples_jag)
gelman.plot(samples_jag,ask=FALSE)

# Question 3

mcmc_sample2 <- bind_rows(lapply(out2, as.data.frame), .id="chain")|>
  pivot_longer(cols=-chain, names_to = "Parameter", values_to = "Value")|>
  filter(Parameter != "deviance")

hpd_intervals2 <- HPDinterval(out2, prob = 0.95)
```

```
intvals <- data.frame(Parameter = c("beta0", "beta1", "beta2"),
                      Lower = c(mean(hpd_intervals2[[1]]["beta0", "lower"], hpd_intervals2[[2]]["beta0"
                      Upper = c(mean(hpd_intervals2[[1]]["beta0", "upper"], hpd_intervals2[[2]]["beta0"

mcmc_sample2fin <- left_join(mcmc_sample2, intvals, by = "Parameter")

ggplot(mcmc_sample2fin, aes(x = Value, fill = chain)) +
  geom_density(alpha = 0.6) +
  facet_wrap(~ Parameter, scales = "free", nrow = 2) +
  geom_vline(aes(xintercept = Lower), linetype = "dashed", color = "red")+
  geom_vline(aes(xintercept = Upper), linetype = "dashed", color = "red") +
  labs(x = "Value", y = "Density", title = "Density Plot of MCMC Samples") +
  theme_minimal()

# Question 4

# mcmc_samples contains the posterior samples
mcmc_samples <- as.mcmc.list(out2)

posterior_samples <- do.call(rbind.data.frame, mcmc_samples)

# Function to calculate probability from log-odds
logit_to_prob <- function(logit) {
  exp(logit) / (1 + exp(logit))
}

# Calculate the probabilities for each poverty group
posterior_samples <- posterior_samples %>%
  mutate(
    pi_133less = logit_to_prob(beta0),
    pi_133_400 = logit_to_prob(beta0 + beta1),
    pi_400more = logit_to_prob(beta0 + beta2)
  )

# Calculate the proportion of samples where vaccination probability is >= 90%
target_coverage <- 0.90

prob_133less_reached <- mean(posterior_samples$pi_133less >= target_coverage)
prob_133_400_reached <- mean(posterior_samples$pi_133_400 >= target_coverage)
prob_400more_reached <- mean(posterior_samples$pi_400more >= target_coverage)


# Combine the results into a data frame for plotting
posterior_probs_df <- data.frame(
  PovertyGroup = c("<133% FPL", "133-400% FPL", ">400% FPL"),
  Probability = c(prob_133less_reached, prob_133_400_reached, prob_400more_reached)
)|>mutate(PovertyGroup = factor(PovertyGroup, levels = c("<133% FPL", "133-400% FPL", ">400% FPL")))

# Plot the probabilities using ggplot2
ggplot(posterior_probs_df, aes(x = PovertyGroup, y = Probability)) +
  geom_bar(stat = "identity", fill = c("blue", "green", "red"), alpha = 0.7) +
  geom_text(aes(label = scales::percent(Probability, accuracy = 0.1)), vjust = -0.3, size = 5) +
  ylim(0, 1) +
  labs(
```

```
    title = "Posterior Probability of Reaching 90% Vaccination Coverage",
    x = "Poverty Group",
    y = "Probability"
  ) +
  theme_minimal(base_size = 15)

# Assuming posterior_samples already contains the calculated probabilities
posterior_samples <- posterior_samples %>%
  mutate(
    pi_133less = logit_to_prob(beta0),
    pi_133_400 = logit_to_prob(beta0 + beta1),
    pi_400more = logit_to_prob(beta0 + beta2)
  )

# Function to create density plot with 90% threshold line
create_density_plot <- function(data, var, group_name) {
  ggplot(data, aes_string(x = var)) +
    geom_density(fill = "blue", alpha = 0.3) +
    geom_vline(xintercept = 0.9, linetype = "dashed", color = "red") +
    labs(title = paste("Posterior Distribution for", group_name),
         x = "Vaccination Probability",
         y = "Density") +
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5))
}

# Create density plots for each group
p1 <- create_density_plot(posterior_samples, "pi_133less", "<133% FPL")
p2 <- create_density_plot(posterior_samples, "pi_133_400", "133-400% FPL")
p3 <- create_density_plot(posterior_samples, "pi_400more", ">400% FPL")

# Arrange the plots
library(gridExtra)
grid.arrange(p1, p2, p3, ncol = 2)

# Question 5


posterior_samples <- posterior_samples %>%
  mutate(
    diff_133_400_vs_133less = pi_133_400 - pi_133less,
    diff_400more_vs_133less = pi_400more - pi_133less,
    diff_400more_vs_133_400 = pi_400more - pi_133_400
  )

# Function to create density plot for differences
create_diff_density_plot <- function(data, var, group_comparison) {
  ggplot(data, aes_string(x = var)) +
    geom_density(fill = "blue", alpha = 0.3) +
    labs(title = stringr::str_wrap(paste("Posterior Distribution of Difference in Vaccination Probabili
         x = "Difference in Vaccination Probability",
         y = "Density") +
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5))
```

```
}

# Create density plots for differences in vaccination probabilities
p1 <- create_diff_density_plot(posterior_samples, "diff_133_400_vs_133less", "133-400% FPL vs <133% FPL
p2 <- create_diff_density_plot(posterior_samples, "diff_400more_vs_133less", ">400% FPL vs <133% FPL")
p3 <- create_diff_density_plot(posterior_samples, "diff_400more_vs_133_400", ">400% FPL vs 133-400% FPL


library(gridExtra)
grid.arrange(p1, p2, p3, ncol = 2)

# Question 6

projdata_jag <- projdata %>%
  mutate(location = as.numeric(factor(Geography)))

model_data_jags <- list(
  Y = projdata_jag$Vaccinated,
  N = projdata_jag$Sample.Size,
  btn133_400_pc = projdata_jag$btn133_400_pc,
  great_400pc = projdata_jag$great_400pc,
  location = projdata_jag$location,
  J = nrow(projdata_jag),
  J_locations = length(unique(projdata_jag$location))
)

modeljags <- function(){
  for (i in 1:J){
    Y[i] ~ dbin(p[i], N[i])
    logit(p[i]) <-  beta1*btn133_400_pc[i] + beta2*great_400pc[i] + beta_location[location[i]]
  }
  #priors
  beta1 ~ dnorm(0, 0.001)
  beta2 ~ dnorm(0, 0.001)
  for (j in 1:J_locations) {
    beta_location[j] ~ dnorm(0, 0.001)
  }
}

model.inits2 <- list(
  list(beta_location = rep(1, model_data_jags$J_locations), beta1 = 0, beta2 = 0),
  list(beta_location = rep(1, model_data_jags$J_locations), beta1 = 1, beta2 = 0),
  list(beta_location = rep(1, model_data_jags$J_locations), beta1 = 0, beta2 = 1)
)


params_loc_jags <- c("beta1", "beta2", "beta_location")

# Write model to file
write.model(modeljags, here("model3code.txt"))
suppressMessages({
  suppressWarnings({
    jags_mod2 <- jags.model("model3code.txt", data = model_data_jags,inits = model.inits2, n.chains = 3
    update(jags_mod2, 2000)
```

```
    jags_samples_extended <- coda.samples(jags_mod2, variable.names = params_loc_jags, n.iter = 10000)
  })
})


gelman.diag(jags_samples_extended)
gelman.plot(jags_samples_extended,ask=FALSE)

# Question 7

posterior_samples2 <- as.matrix(jags_samples_extended)

#  Calculate the mean and 95% credible intervals
coverage_estimates <- apply(posterior_samples2, 2, function(x) {
  mean_val <- mean(x)
  ci <- quantile(x, c(0.025, 0.975))
  c(mean = mean_val, lower = ci[1], upper = ci[2])
})

coverage_df <- data.frame(location = rep(colnames(coverage_estimates), each = 3),
                          estimate_type = rep(c("mean", "lower.2.5%", "upper.97.5%"), times = ncol(cove
                          coverage = as.vector(coverage_estimates),
                          stringsAsFactors = FALSE)

betas_loc <- coverage_df |>
  slice(1:6)

betas_loc_est <- coverage_df |>
  slice(7:nrow(coverage_df))|>
  mutate(coverage_133pc = plogis(coverage),
         coverage_133_400pc = case_when(
           estimate_type == "mean" ~ plogis(coverage + betas_loc$coverage[1]),
           estimate_type == "lower.2.5%" ~ plogis(coverage + betas_loc$coverage[2]),
           estimate_type == "upper.97.5%" ~ plogis(coverage + betas_loc$coverage[3])
         ),
         coverage_400pc = case_when(
           estimate_type == "mean" ~ plogis(coverage + betas_loc$coverage[4]),
           estimate_type == "lower.2.5%" ~ plogis(coverage + betas_loc$coverage[5]),
           estimate_type == "upper.97.5%" ~ plogis(coverage + betas_loc$coverage[6])
         ))|>select(-coverage)|>pivot_longer(cols = c(coverage_133pc, coverage_133_400pc, coverage_400p
  mutate(
         location = factor(location, levels = c("beta_location[1]", "beta_location[2]", "beta_location[3
         povlevel = factor(povlevel, levels = c("coverage_133pc", "coverage_133_400pc", "coverage_400pc
         location2 = paste0(location,"_", povlevel),
         location2 = factor(location2, levels = location2[order(-location)]))

#  Plot the data

ggplot(betas_loc_est, aes(x = location2, y = mean, ymin = `lower.2.5%`, ymax = `upper.97.5%`, colour = p
  geom_pointrange() +
  coord_flip() +
  labs(x = "Geographic location", y = "Estimated Vaccination Coverage", colour = "Poverty Level", title
  theme_bw(base_size = 12)
```

```
# Question 8

mississippi_index <- 3

jag7posterior_samples <- do.call(rbind, lapply(jags_samples_extended, as.data.frame))

mean_coverage_133_400FPL <- mean(plogis( jag7posterior_samples$beta1 + jag7posterior_samples[, grep("^be
mean_coverage_400FPL <- mean(plogis( jag7posterior_samples$beta2 + jag7posterior_samples[, grep("^beta_l
mean_coverage_133FPL <- mean(plogis( jag7posterior_samples[, grep("^beta_location", colnames(jag7poster

# Predict the number of vaccinated children
n_children <- 300
n_133_400FPL <- 100
n_400FPL <- 100
n_others <- 100

pred_vaccinated_133_400FPL <- n_133_400FPL * mean_coverage_133_400FPL
pred_vaccinated_400FPL <- n_400FPL * mean_coverage_400FPL
pred_vaccinated_133FPL <- n_others * mean_coverage_133FPL

datsr<-rbind(c("133-400% FPL", as.numeric(round(pred_vaccinated_133_400FPL,0))), c(">400% FPL", as.numer

dfe<-datsr|>data.table::as.data.table()|>data.table::setnames(c("Poverty", "Estimate"))|>dplyr::mutate(E
dfe|>flextable::flextable()|>flextable::autofit()
```