

Concepts of Bayesian Data Analysis: Project4

Charles Muiruri* Gianni Guarraci† Juan Vanegas Jadan‡ Ronald Makanga§

Report generated - May 31, 2024

Contents

Modeling using Open Bugs	2
Model	2
Question 1:	2
Question 2	2
Question 3: Density plots	3
QUESTION 4: Calculate the posterior probability that the vaccination coverage target (90%) is reached for each poverty group.	4
Question 5	5
Question 6	5
Question 7	10
Question 8	10
Appendix	11

*Hasselt University, charles.muiruri@student.uhasselt.be

†Hasselt University, gianni.guarraci@student.uhasselt.be

‡Hasselt University, juan.vanegasjadan@student.uhasselt.be

§Hasselt University, ronald.makanga@student.uhasselt.be

Modeling using Open Bugs

Model

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

Question 1:

The bugs model is as shown below

```
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)
}
```

The mean probability of vaccination coverage for each poverty group is as follows:

<133% FPL: 0.8723253

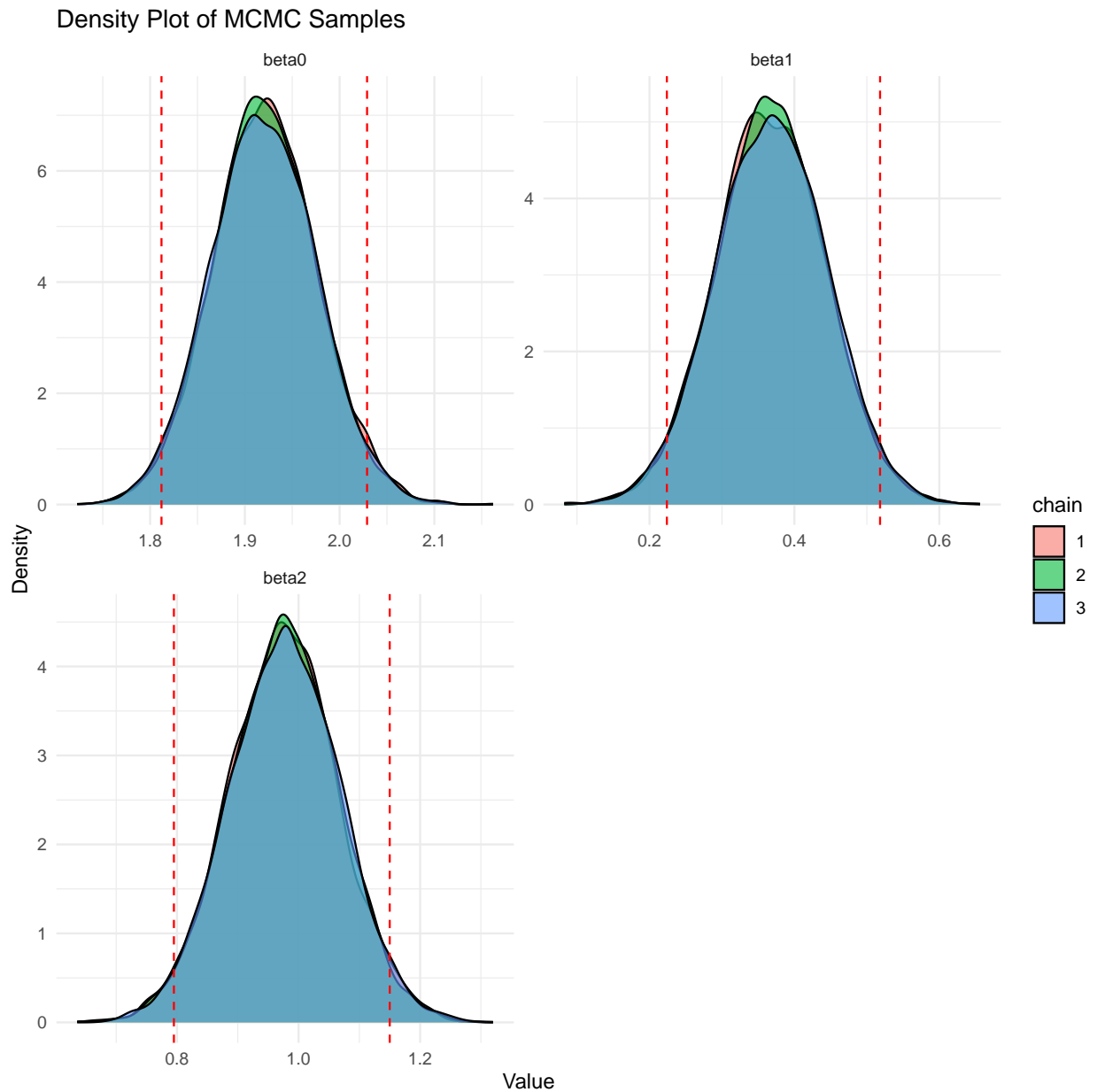
133% to <400% FPL: 0.9078979

>400% FPL: 0.9476732

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
```

Question 3: Density plots



Summary of posterior distribution

- The Posterior density with relation to β_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 β_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 β_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: Calculate the posterior probability that the vaccination coverage target (90%) is reached for each poverty group.

```
## Posterior probability that vaccination coverage target (90%) is reached:
```

```
## <133% FPL: 0
```

```
## 133-400% FPL: 0.96025
```

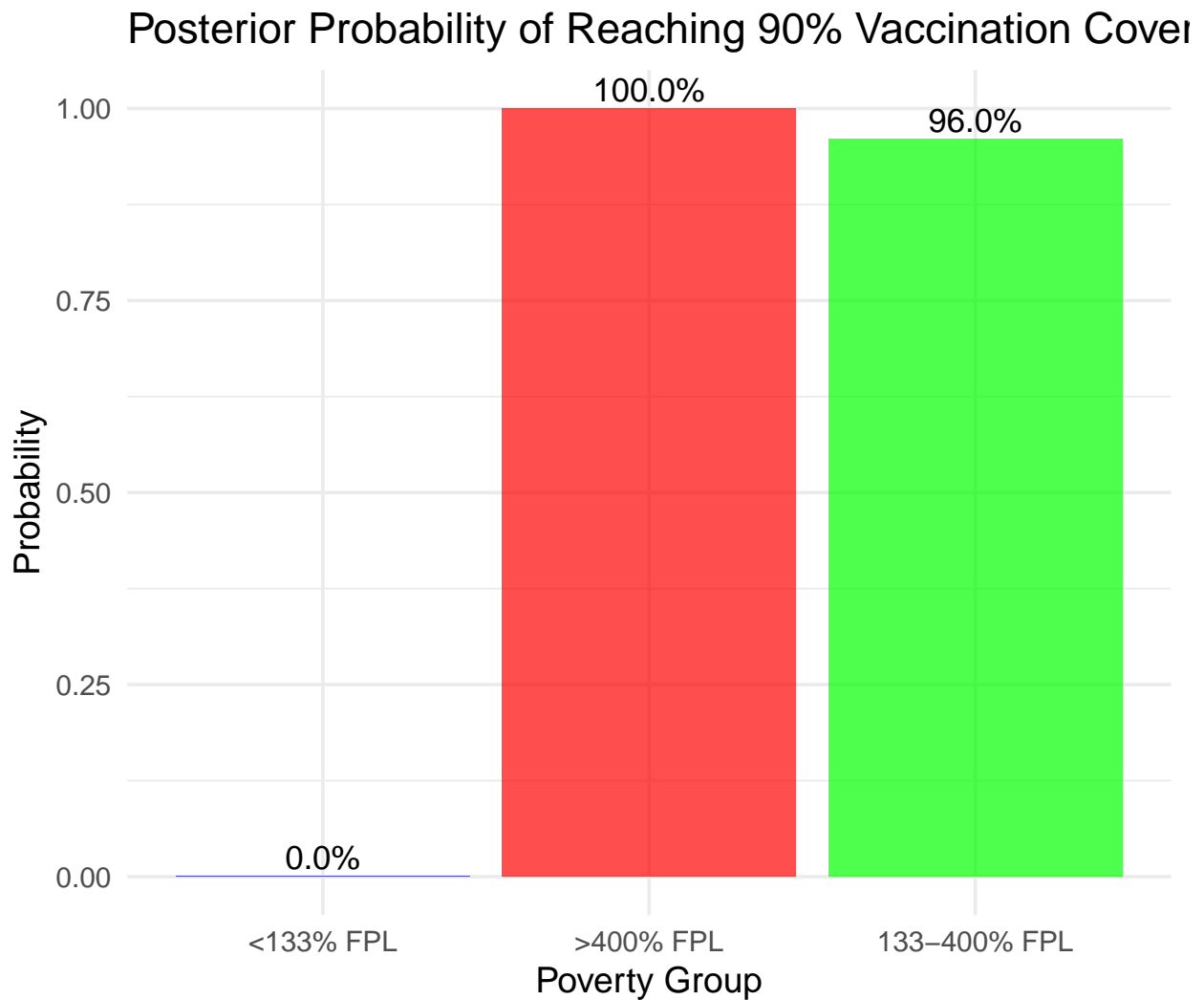
```
## >400% FPL: 1
```

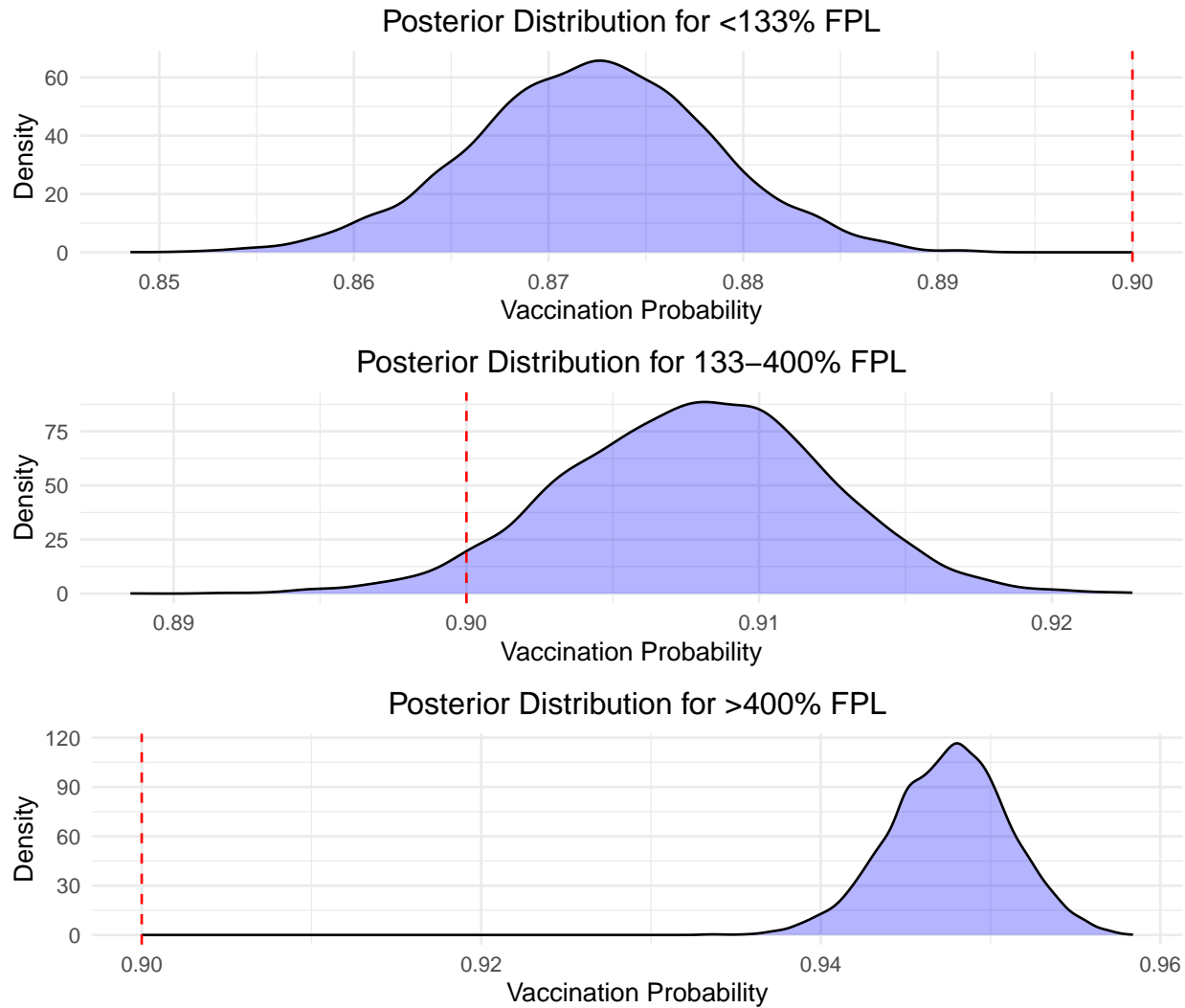
```
## PovertyGroup Probability
```

```
## 1 <133% FPL 0.00000
```

```
## 2 133-400% FPL 0.96025
```

```
## 3 >400% FPL 1.00000
```





Question 5

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.2089 -0.1486 -0.1355 -0.1355 -0.1226 -0.0632
```

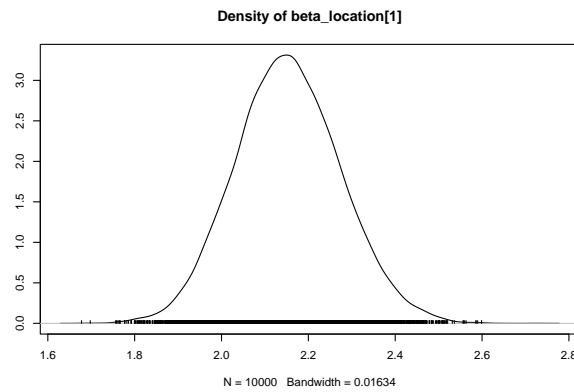
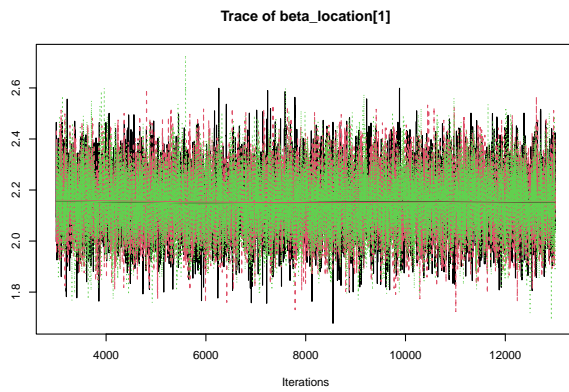
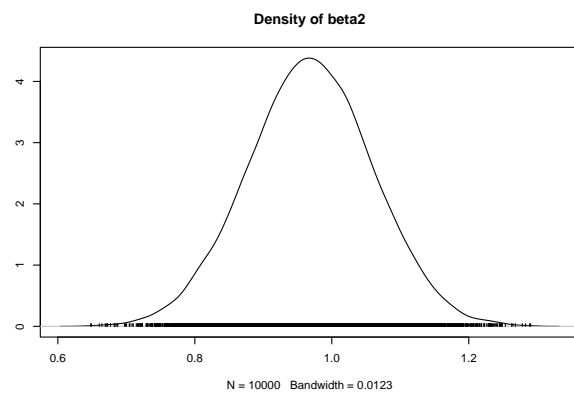
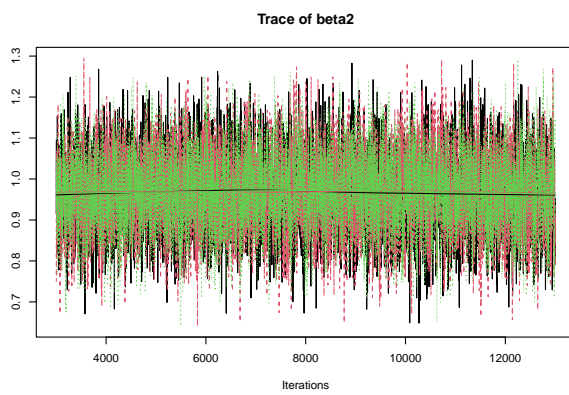
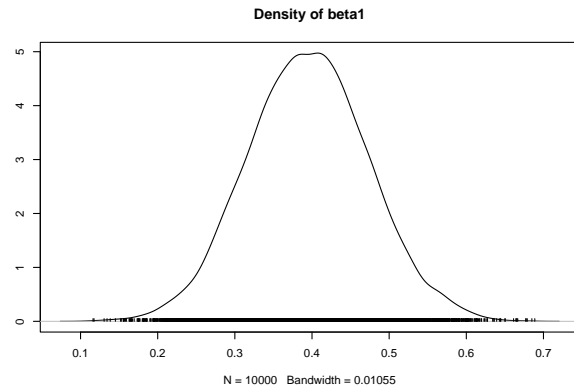
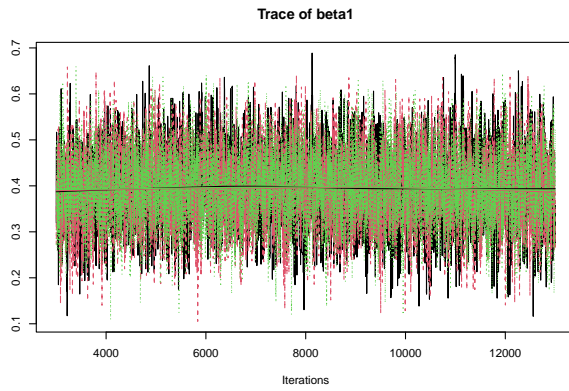
Question 6

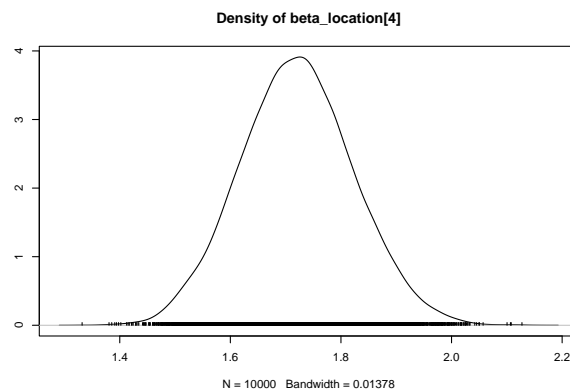
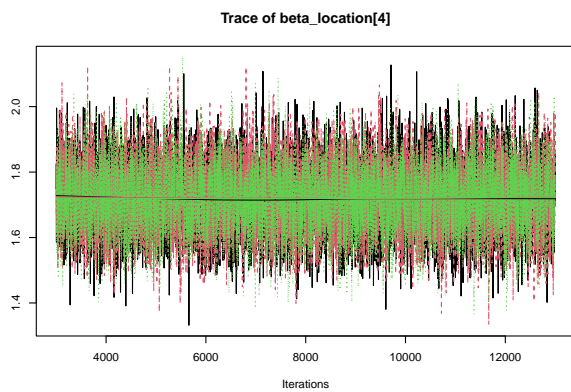
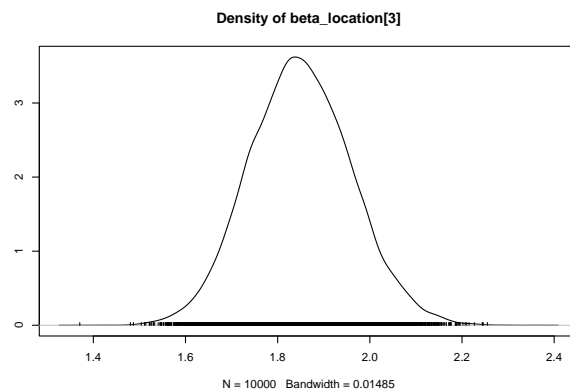
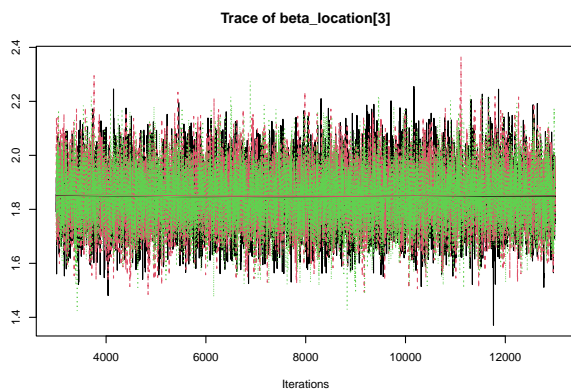
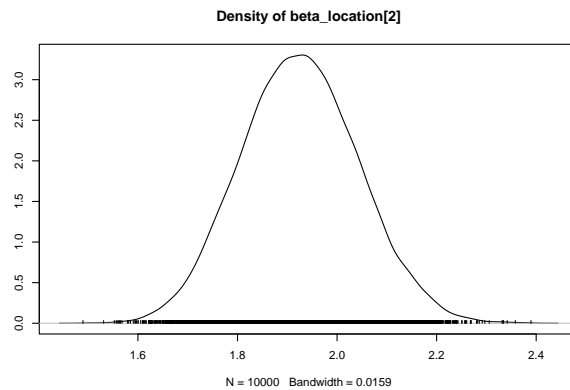
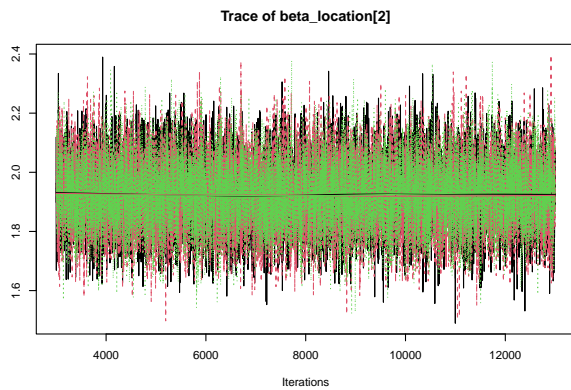
Investigate whether the vaccination coverages are distinct at the different locations by adding a location-specific intercept.

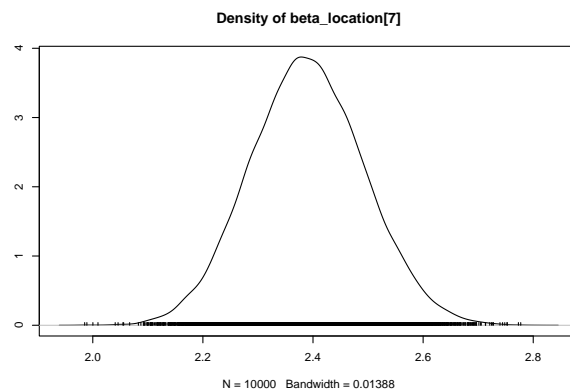
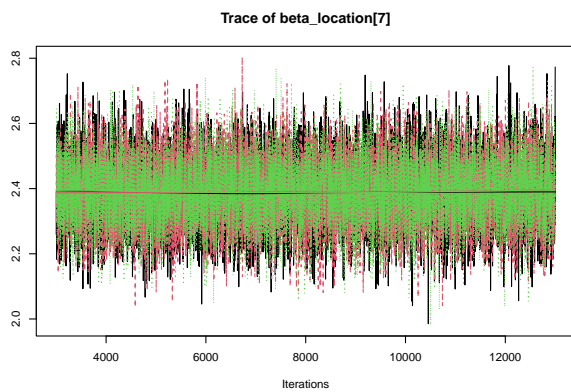
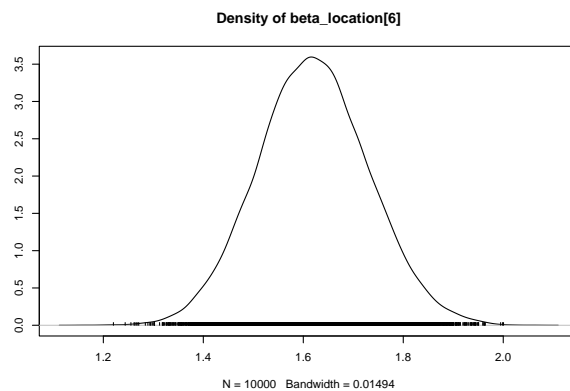
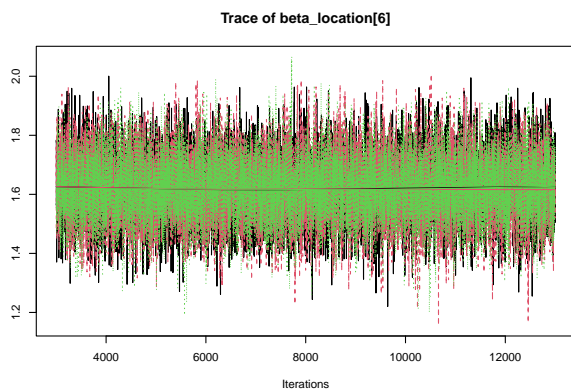
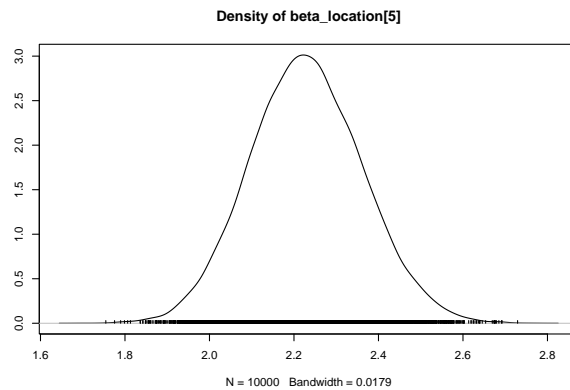
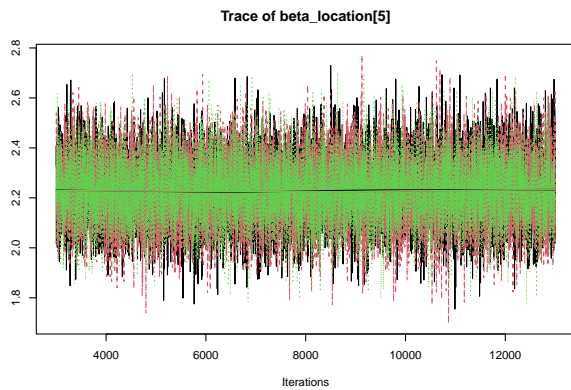
For this, β_0 is assumed to be the logit of the proportion of vaccinated individuals for each location in the group less than 133% FPL, so we need to calculate it. So we will find just the proportion for groups

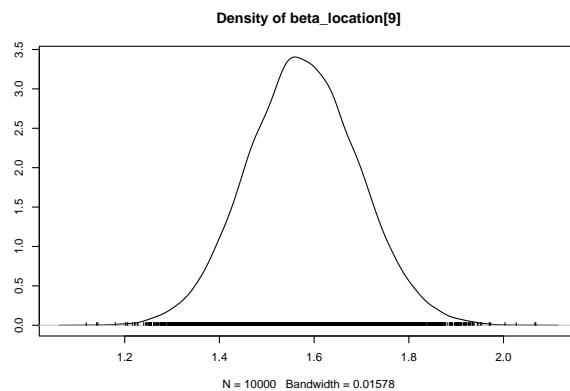
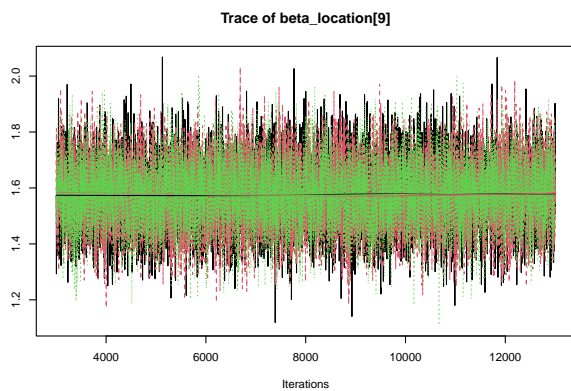
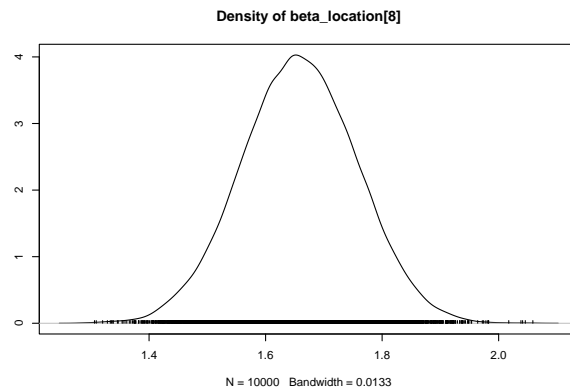
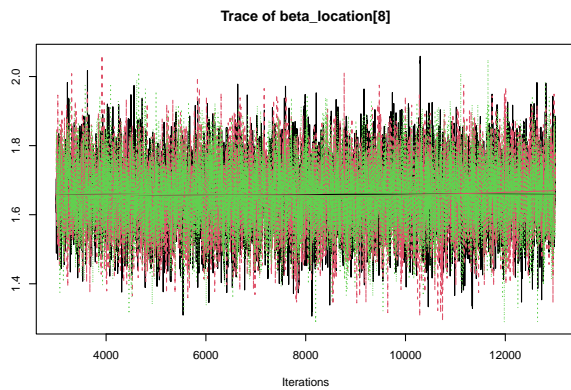
```
## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 27
##   Unobserved stochastic nodes: 12
##   Total graph size: 209
##
```

```
## Initializing model
```

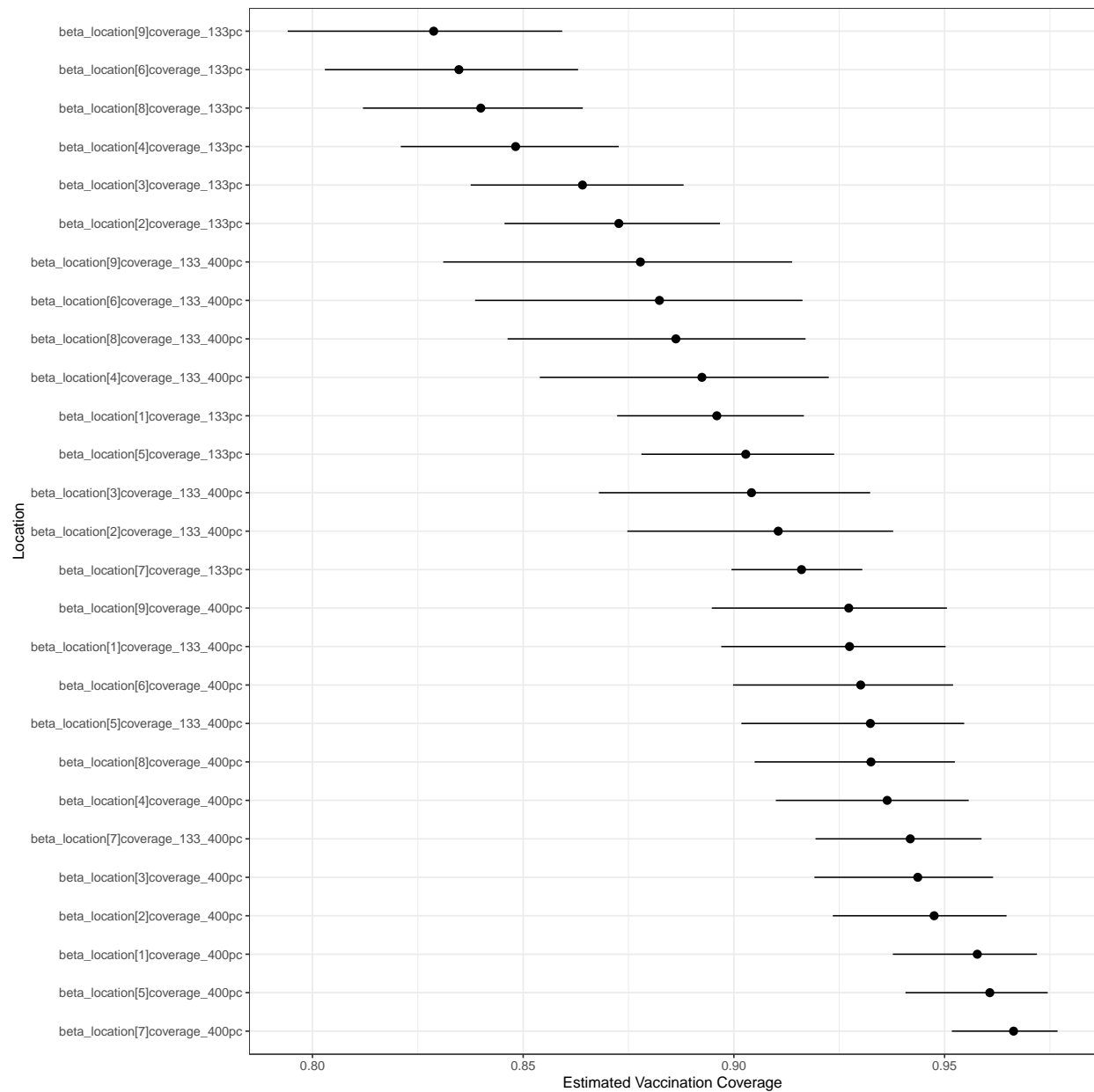








Question 7



Question 8

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

Appendix

R Code

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

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))
# 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)
)

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

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

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 to file
write.model(model1, here("models/model1code.txt"))
# View file
file.show(here("models/model1code.txt"))

model.out <- bugs(model_data, model_inits,
  parameters = parameters, model.file = here("models/model1code.txt"),
  n.chains = 1, n.iter = 10000, n.burnin = 5000, codaPkg = TRUE,
  debug = FALSE)

# debug=TRUE opens openBug and displays traceplots and summaries

# Model output
out <- read.bugs(model.out)
summary(out)

# Prep data for density and trace plots
```

```

mcmc_samples <- as.mcmc(out)
mcmc_df <- as.data.frame(mcmc_samples)
mcmc_df$iteration <- 1:nrow(mcmc_df)
mcmc_long <- pivot_longer(mcmc_df, cols = -iteration, names_to = "Parameter",
                          values_to = "Value")

points_data <- data.frame(x = mcmc_df$beta0, x1 = mcmc_df$beta1, x2 = mcmc_df$beta2,
                          x3 = mcmc_df$deviance, y = rep(0, nrow(mcmc_df)))

# Density plots
ggarrange(p_beta0 <- ggplot(mcmc_df, aes(x = beta0)) +
  geom_density(fill = "blue", alpha = 0.1) +
  geom_point(data = points_data, aes(x = x, y = y)) +
  labs(title = "Posterior Distribution of beta0", x = "beta0", y = "Density")+
  theme_bw(base_size = 12)+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()),

p_beta1 <- ggplot(mcmc_df, aes(x = beta1)) +
  geom_density(fill = "green", alpha = 0.1) +
  geom_point(data = points_data, aes(x = x1, y = y)) +
  labs(title = "Posterior Distribution of beta1", x = "beta1", y = "Density") +
  theme_bw(base_size = 12)+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()),

p_beta2 <- ggplot(mcmc_df, aes(x = beta2)) +
  geom_density(fill = "red", alpha = 0.1) +
  geom_point(data = points_data, aes(x = x2, y = y)) +
  labs(title = "Posterior Distribution of beta2", x = "beta2", y = "Density")+
  theme_bw(base_size = 12)+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()),

p_dev <- ggplot(mcmc_df, aes(x = deviance)) +
  geom_density(fill = "yellow", alpha = 0.1) +
  geom_point(data = points_data, aes(x = x3, y = y)) +
  labs(title = "Posterior Distribution of deviance", x = "Deviance", y = "Density")+
  theme_bw(base_size = 12)+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()),
nrow = 2, ncol = 2)

# Trace plots
ggplot(mcmc_long, aes(x = iteration, y = Value, color = Parameter)) +
  geom_line() +
  scale_color_manual(values = c("blue", "green", "red", "yellow")) +
  facet_wrap(~ Parameter, scales = "free_y") +
  labs(title = "Trace Plots of MCMC Samples", x = "Iteration", y = "Parameter Value") +
  theme_bw(base_size = 12)+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        legend.position = "none")

# Autocorrelation and crosscorrelation plots
crosscorr.plot(out)
autocorr.plot(out)

```