

Concepts of Bayesian Data Analysis: Project4

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

Report generated - May 29, 2024

Contents

Modeling using Open Bugs	2
Model	2
Question 1:	2
Question 2	3
Question 3	5
Question 4	10
Question 5	11
Question 6	11
Question 7	18
Question 8	18
Appendix	19

*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 ... 5000 valid values
## Abstracting beta1 ... 5000 valid values
## Abstracting beta2 ... 5000 valid values
## Abstracting deviance ... 5000 valid values

##
## Iterations = 5001:10000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 5000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean        SD   Naive SE Time-series SE
## beta0     1.9187  0.05379  0.0007607      0.0008041
## beta1     0.3698  0.07410  0.0010480      0.0011437
## beta2     0.9811  0.08931  0.0012630      0.0014289
## deviance 253.9800 2.39063  0.0338086      0.0383106
##
## 2. Quantiles for each variable:
##
##           2.5%       25%       50%       75%     97.5%
## beta0     1.8120  1.8830  1.9170  1.9560  2.0240
## beta1     0.2239  0.3197  0.3700  0.4184  0.5164
## beta2     0.8058  0.9216  0.9797  1.0430  1.1620
## deviance 251.3000 252.2000 253.3000 255.1000 260.1000

## [[1]]
##           lower      upper
## beta0     1.8120  2.0240
## beta1     0.2346  0.5242
## beta2     0.7952  1.1480
## deviance 251.1000 258.8000
## attr(,"Probability")
## [1] 0.95
```

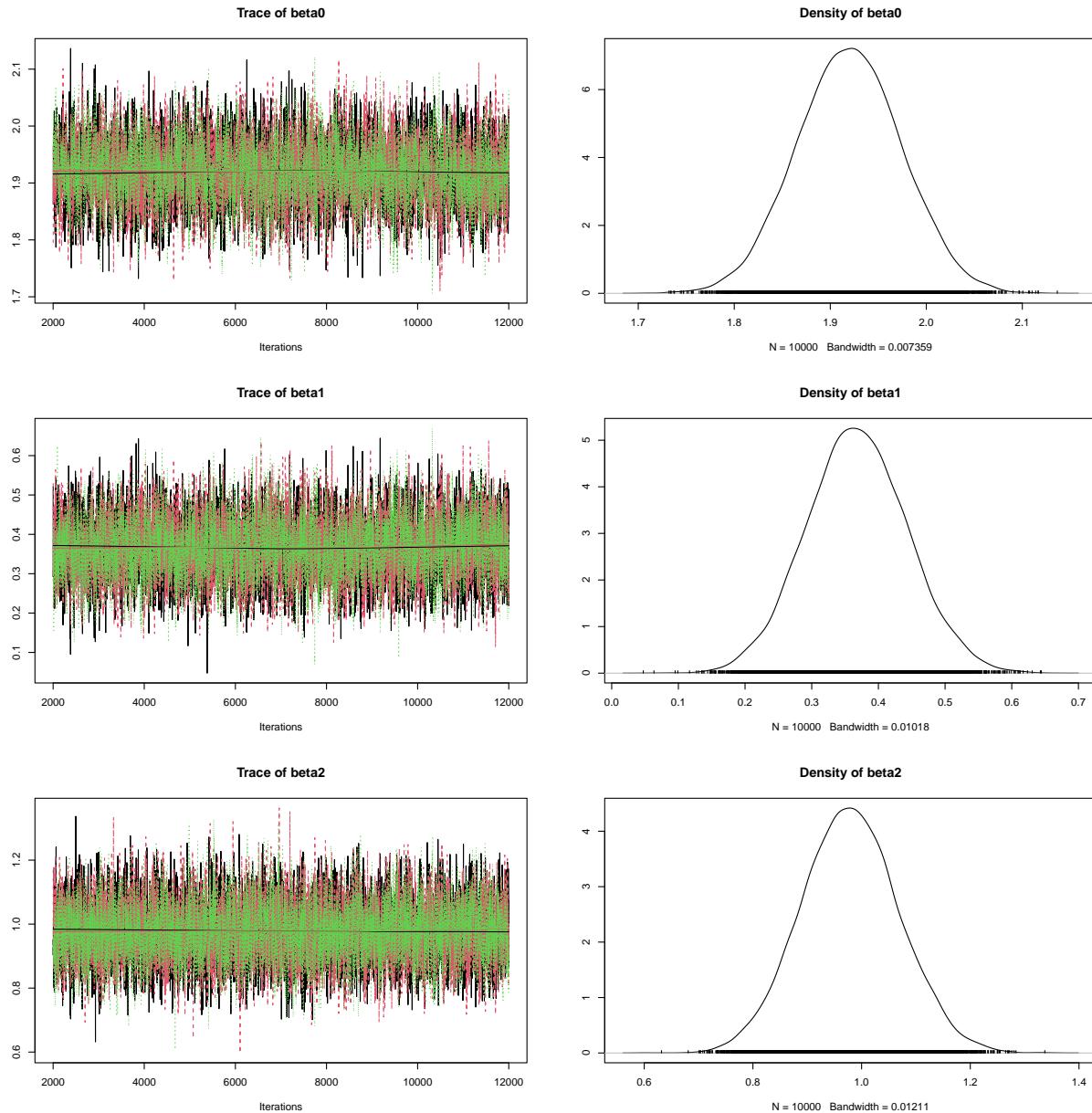
Question 1:

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)
}
```

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
##
## Iterations = 2001:12000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean      SD  Naive SE Time-series SE
## beta0 1.9198 0.05457 0.0003151      0.000842
## beta1 0.3679 0.07547 0.0004357      0.001071
## beta2 0.9789 0.08983 0.0005186      0.001140
##
## 2. Quantiles for each variable:
##
##           2.5%     25%     50%     75%   97.5%
## beta0 1.8137 1.8831 1.9199 1.9566 2.0263
## beta1 0.2206 0.3172 0.3673 0.4183 0.5192
## beta2 0.8044 0.9179 0.9781 1.0384 1.1563
```



```

## Potential scale reduction factors:
##
##          Point est. Upper C.I.
## beta0      1      1
## beta1      1      1
## beta2      1      1
##
## Multivariate psrf
##
## 1
##
##    beta0    beta1    beta2
## 4206.117 4976.121 6219.251
##
## The mean probability of vaccination coverage for each poverty group is as follows:

```

```

## <133% FPL:  0.8719881
## 133% to <400% FPL:  0.9079151
## >400% FPL:  0.9478346

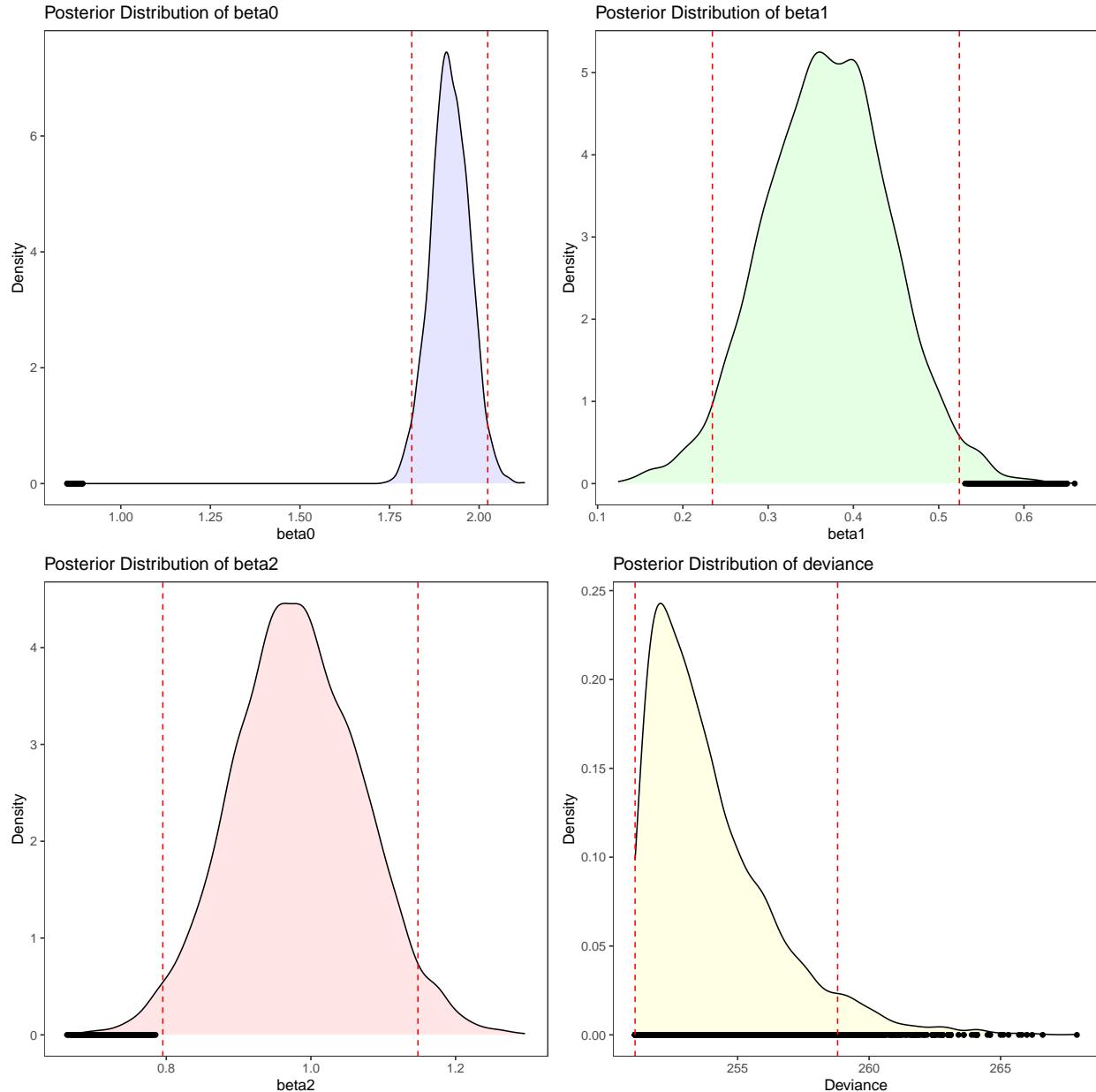
```

Question 3

Density plots

Consider using densities from rjags model

```
# [1] 0.2346
```



```

##
## Iterations = 5001:10000
## Thinning interval = 1

```

```

## Number of chains = 1
## Sample size per chain = 5000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean      SD  Naive SE Time-series SE
## beta0     1.9187 0.05379 0.0007607     0.0008041
## beta1     0.3698 0.07410 0.0010480     0.0011437
## beta2     0.9811 0.08931 0.0012630     0.0014289
## deviance 253.9800 2.39063 0.0338086     0.0383106
##
## 2. Quantiles for each variable:
##
##           2.5%     25%     50%     75%   97.5%
## beta0     1.8120  1.8830  1.9170  1.9560  2.0240
## beta1     0.2239  0.3197  0.3700  0.4184  0.5164
## beta2     0.8058  0.9216  0.9797  1.0430  1.1620
## deviance 251.3000 252.2000 253.3000 255.1000 260.1000
##
## Iterations = 2001:12000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean      SD  Naive SE Time-series SE
## beta0  1.9198 0.05457 0.0003151     0.000842
## beta1  0.3679 0.07547 0.0004357     0.001071
## beta2  0.9789 0.08983 0.0005186     0.001140
##
## 2. Quantiles for each variable:
##
##           2.5%     25%     50%     75%   97.5%
## beta0  1.8137 1.8831 1.9199 1.9566 2.0263
## beta1  0.2206 0.3172 0.3673 0.4183 0.5192
## beta2  0.8044 0.9179 0.9781 1.0384 1.1563

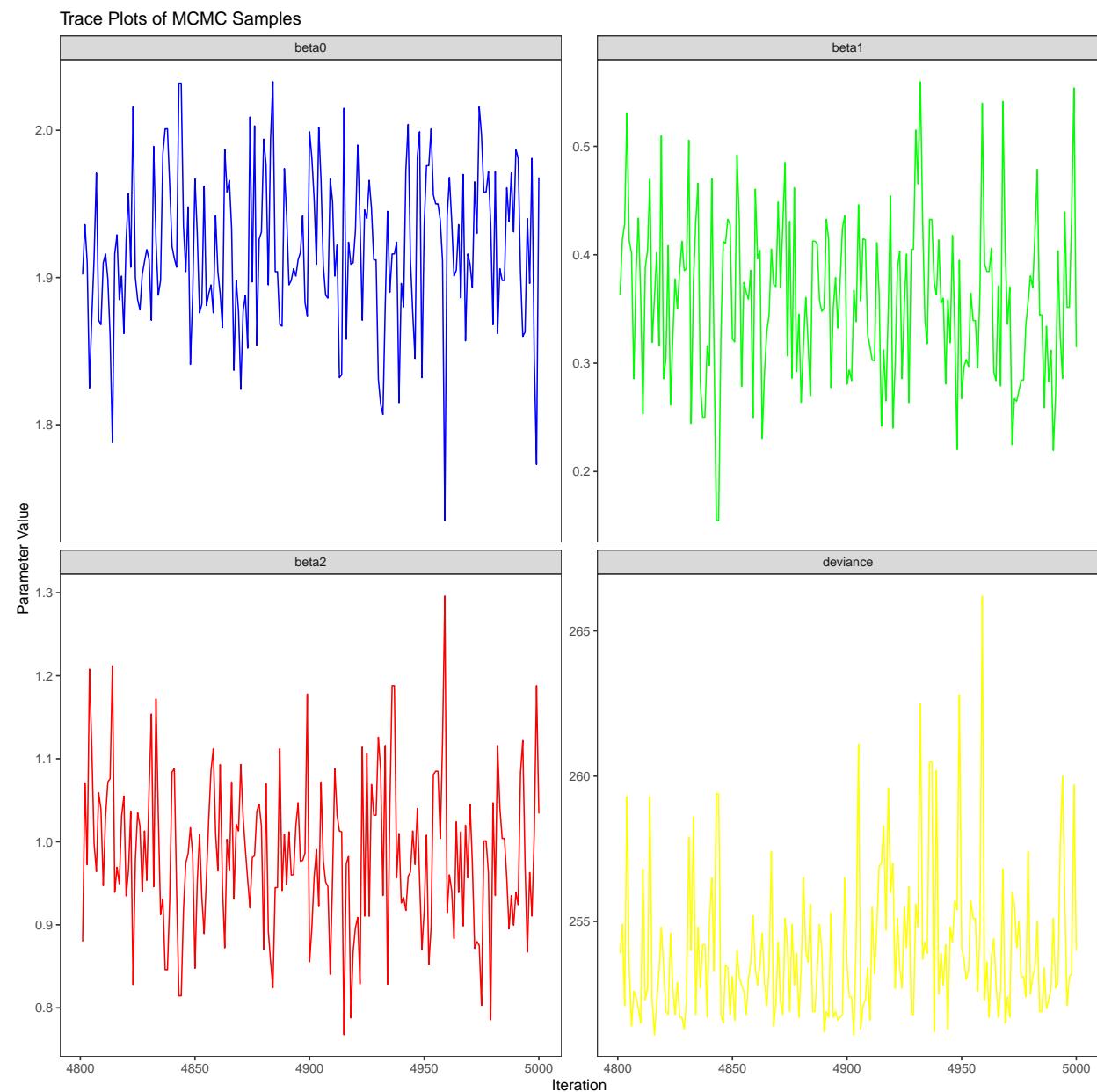
```

Just the densities are enough to answer question 3

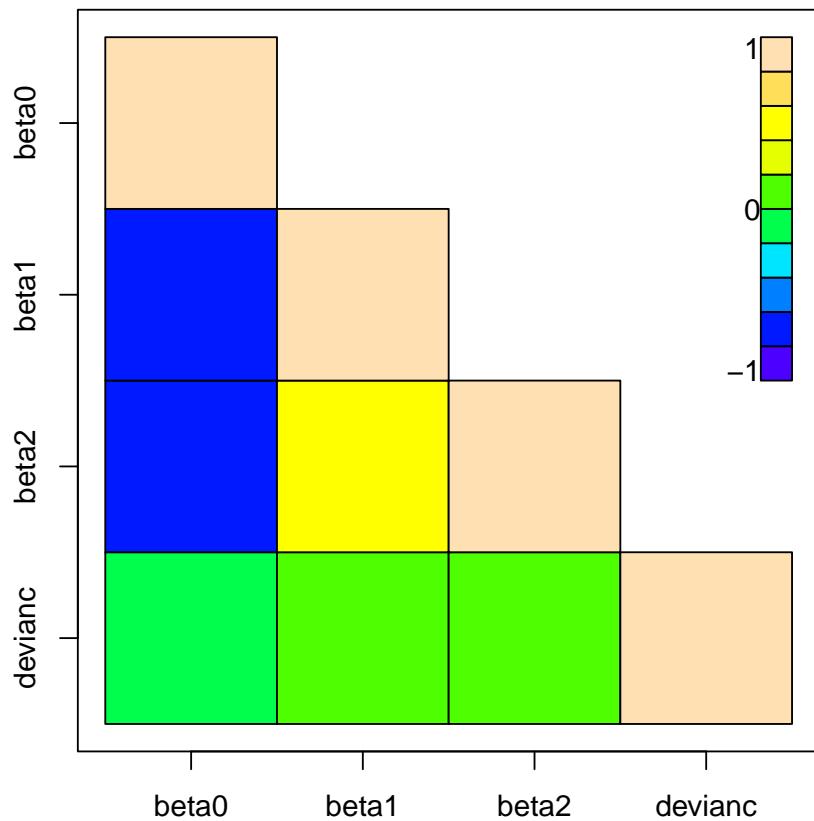
Summary of posterior distribution

- The Posterior of β_0 has a mean of **1.9186524** which is the central tendency of the intercept term, with a 95% credible HPD interval of **(1.812, 2.024)** in which the intercept term lies with 95% probability.
- The Posterior of β_1 has a mean of **0.3697874** which is the central tendency of the Poverty = 133% to <400% FPL term, with a 95% HPD interval of **(0.2346, 0.5242)** in which this Poverty term lies with 95% probability.
- The Posterior of β_2 has a mean of **0.9811075** which is the central tendency of the Poverty = >400% FPL term, with a 95% HPD interval of **(0.7952, 1.148)** in which this Poverty term lies with 95% probability.
- The Posterior of the **Deviance** has a mean of **253.98** which is the central tendency of the Deviance, with a 95% HPD interval of **(251.1, 258.8)**.

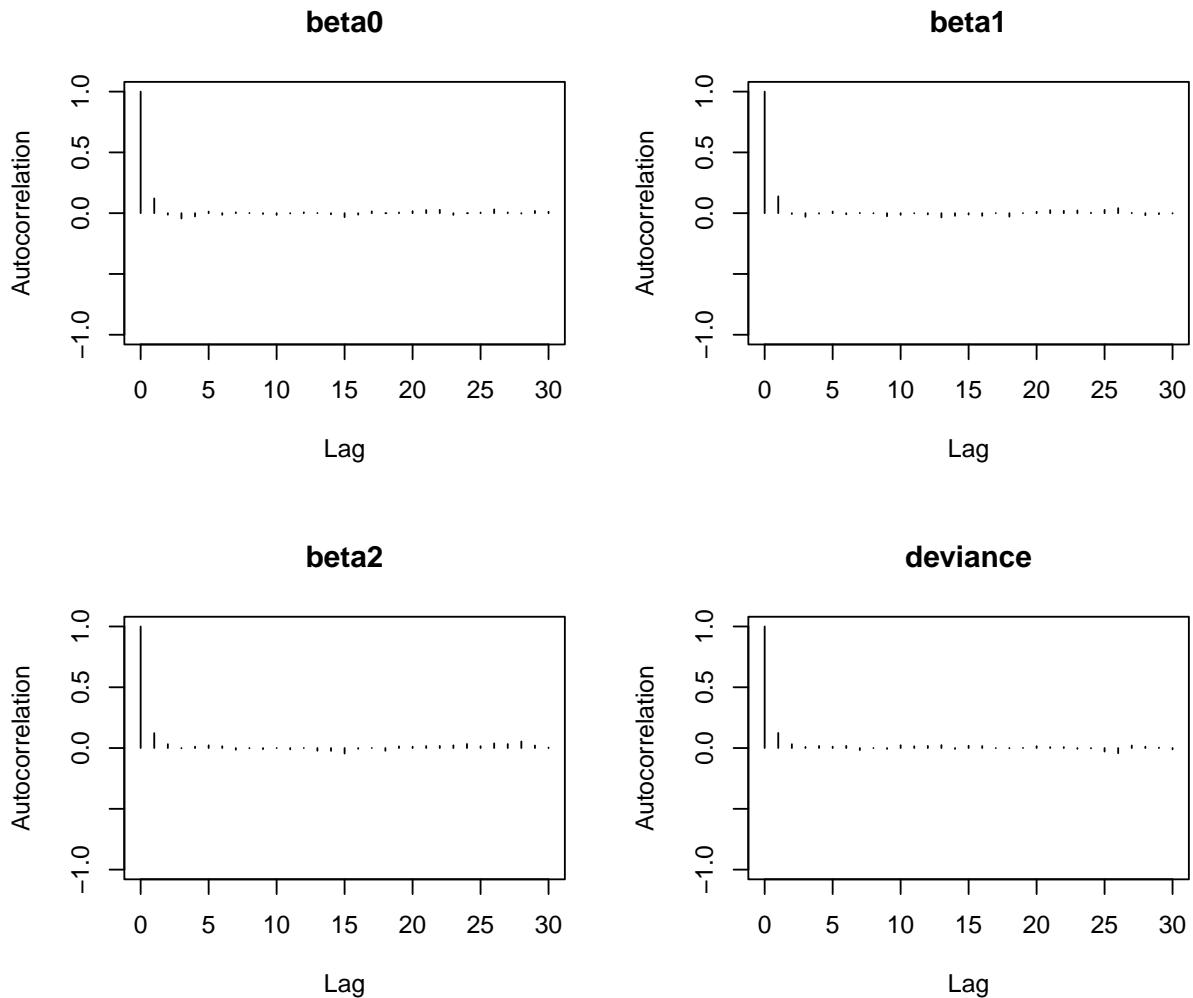
Trace Plots



Correlation plot



Autocorrelation plot

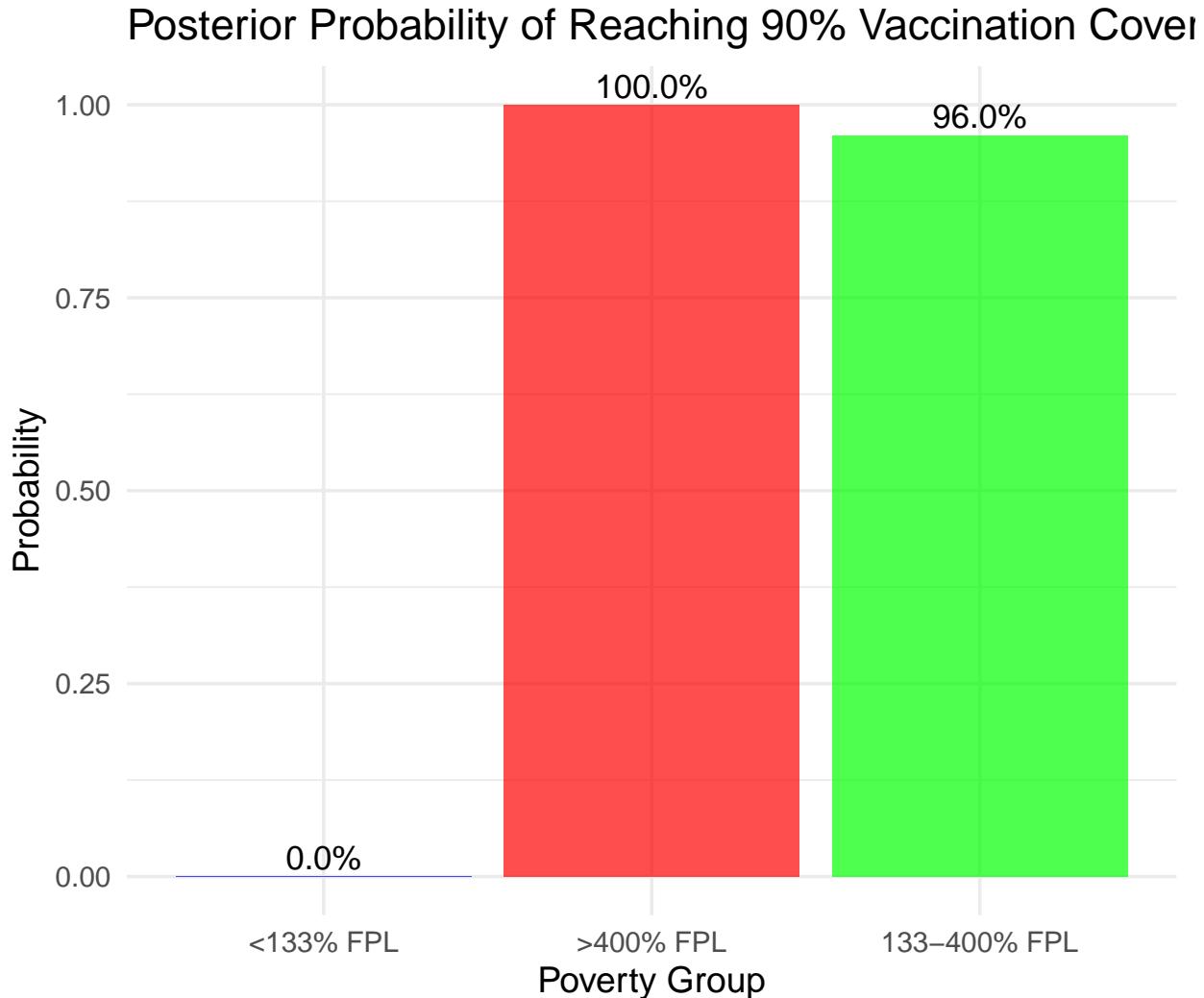


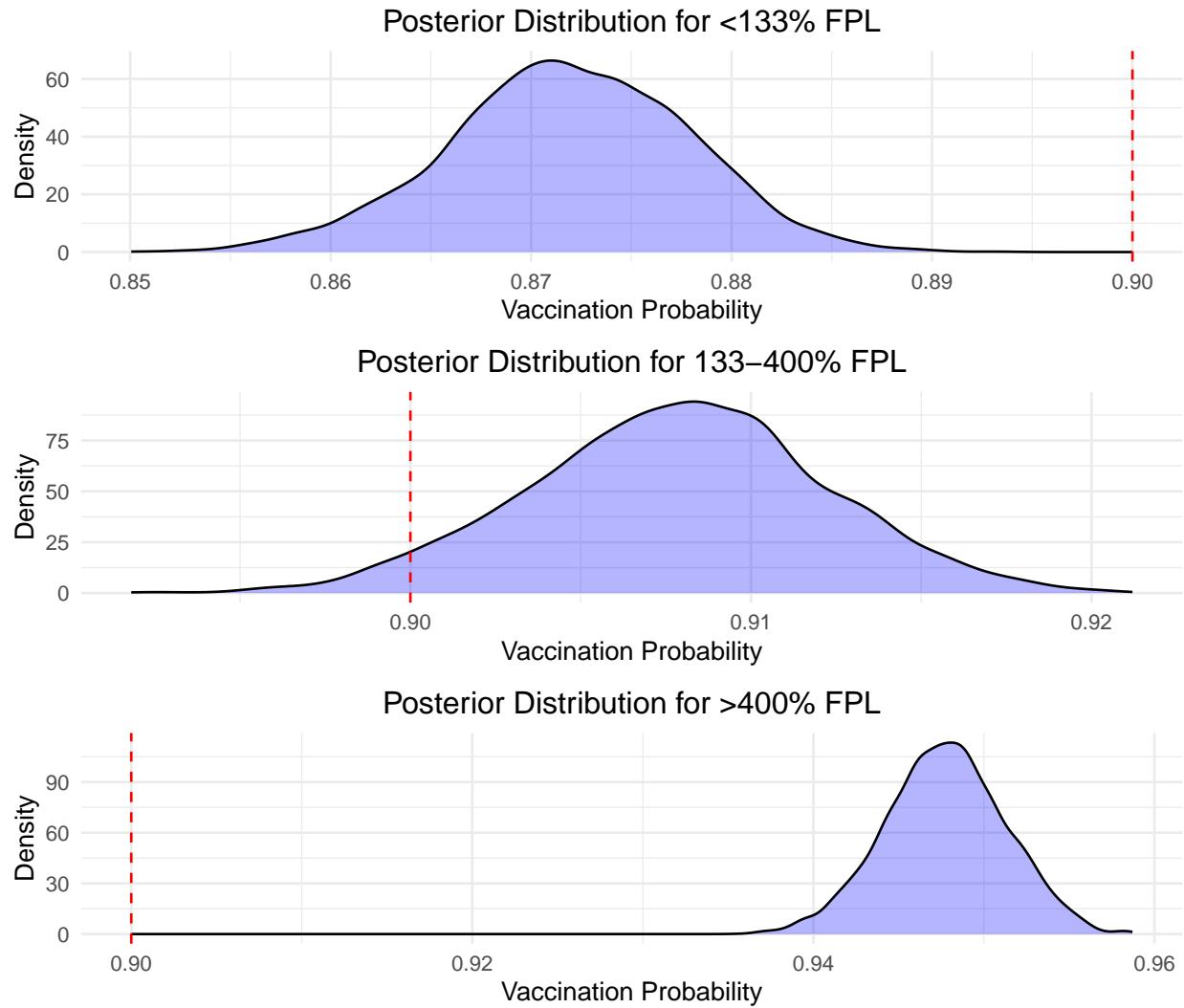
Interpretation of the results is done in the main text.

QUESTION 4: Calculate the posterior probability that the vaccination coverage target (90%) is reached for each poverty group.

Question 4

```
## Posterior probability that vaccination coverage target (90%) is reached:  
## <133% FPL: 0  
## 133–400% FPL: 0.9604  
## >400% FPL: 1
```





Question 5

```
##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.
## -0.21298 -0.14849 -0.13559 -0.13568 -0.12280 -0.06633
```

Question 6

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

For this, beta0 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

```
## Abstracting beta1 ... 5000 valid values
## Abstracting beta2 ... 5000 valid values
## Abstracting deviance ... 5000 valid values

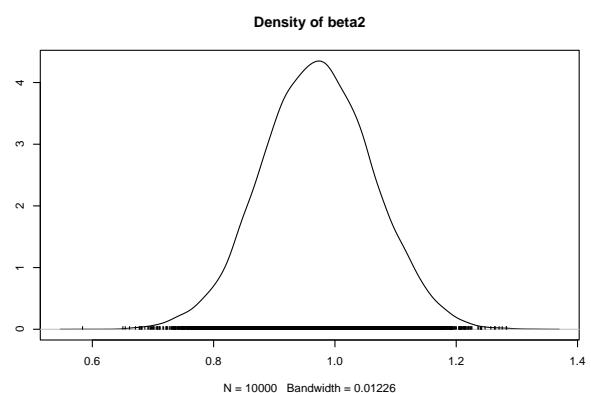
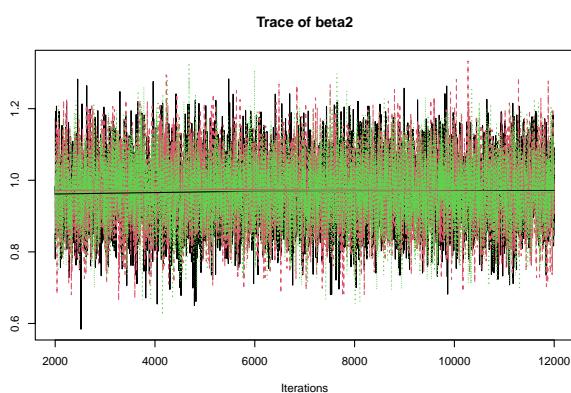
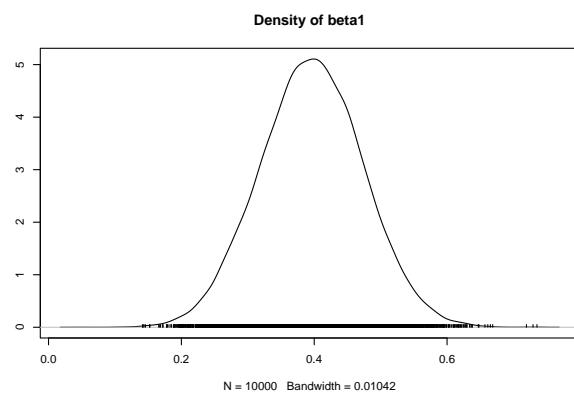
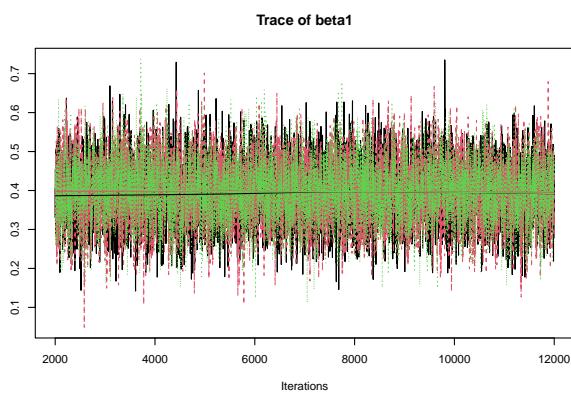
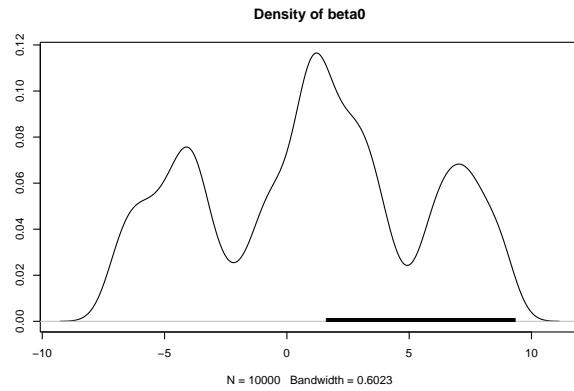
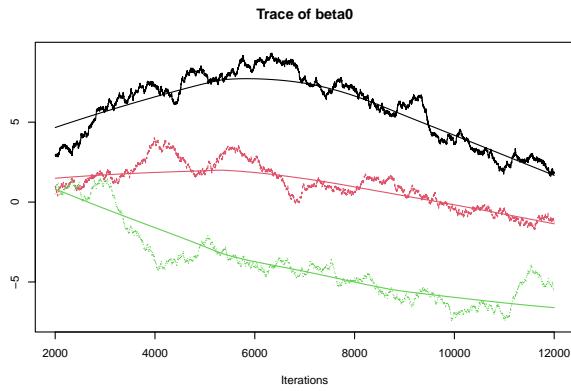
## The mean probability of vaccination coverage for each poverty group is as follows:
## Giving a value to beta0 means that the proportion for the group less than 133% FPL is the logit of b
## <133% FPL: 0.8655125
```

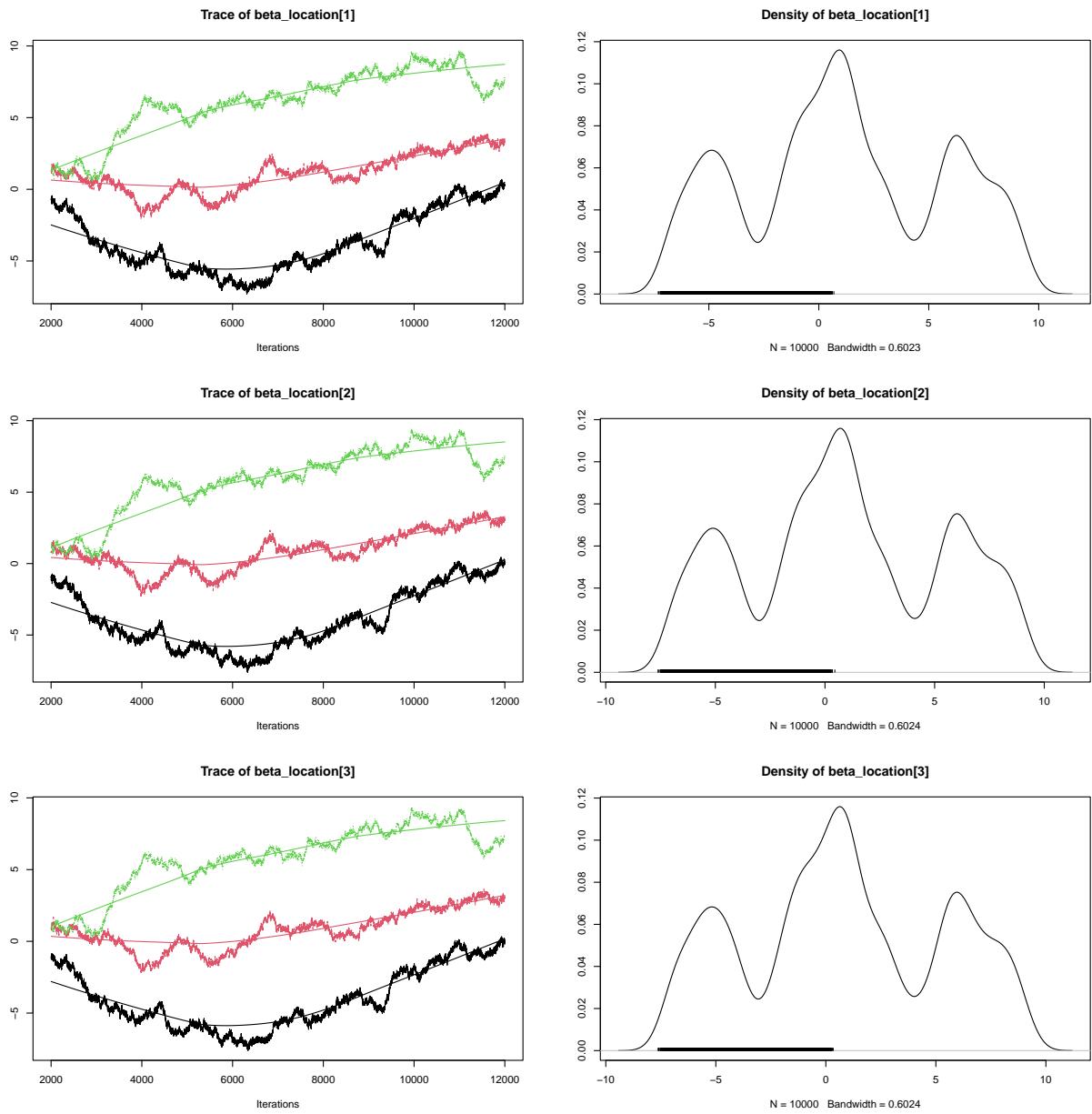
```

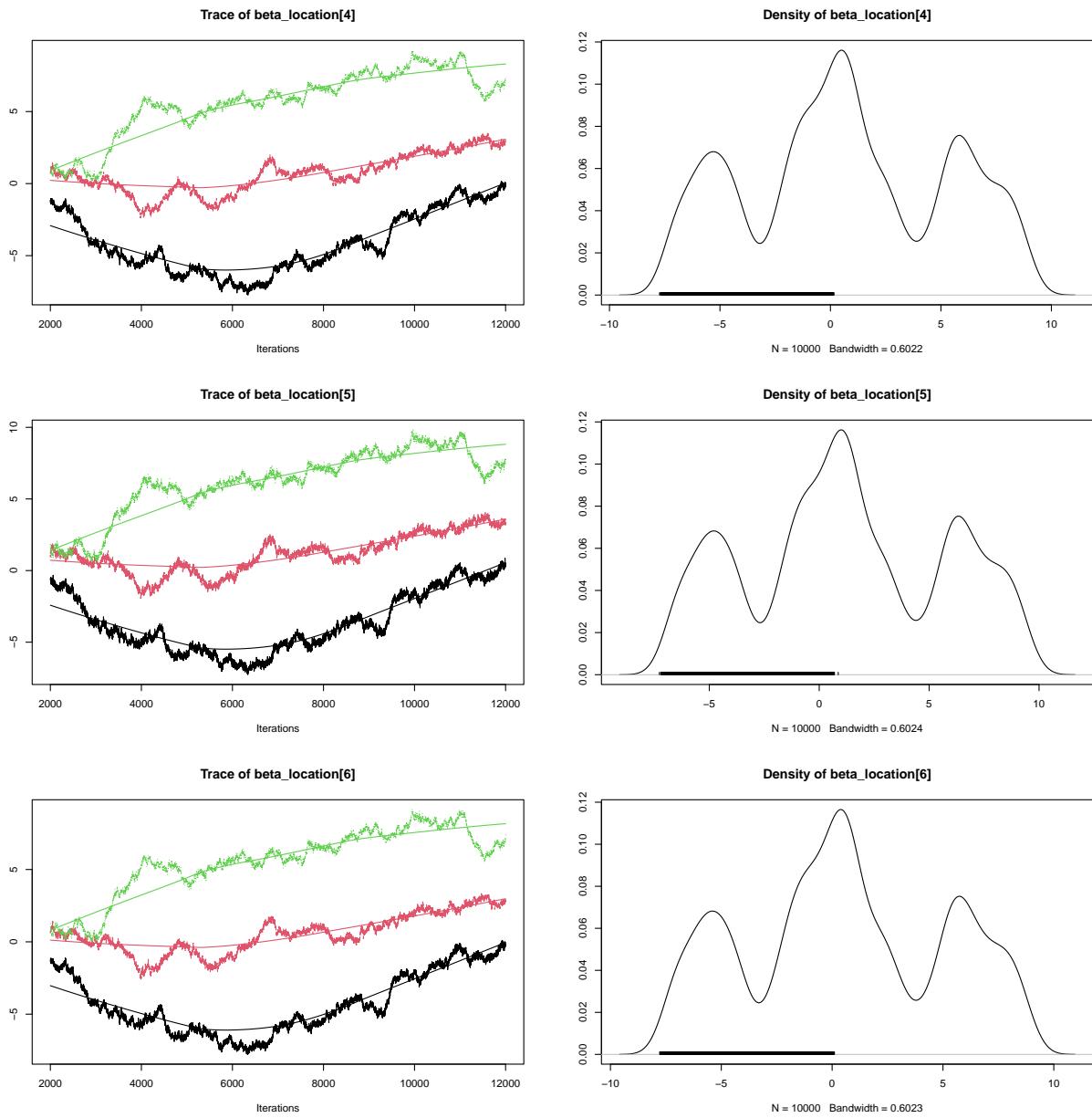
## 133% to <400% FPL: 0.9102155
## >400% FPL: 0.9468115

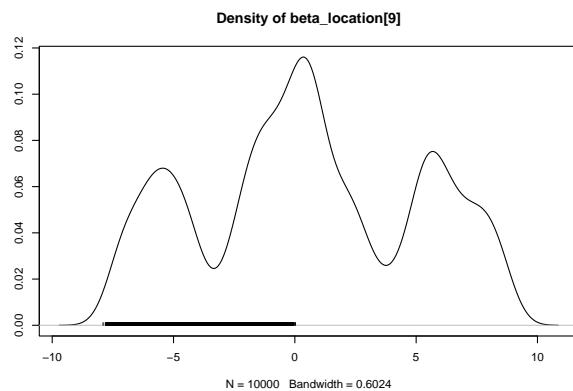
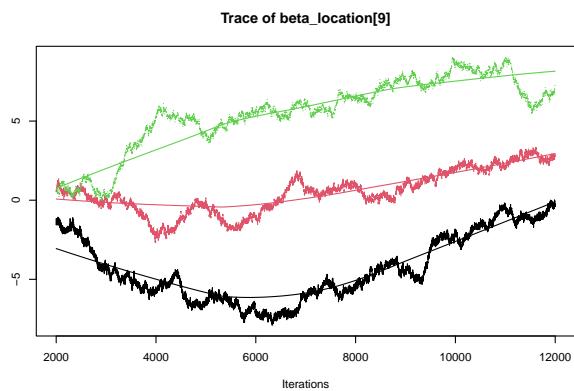
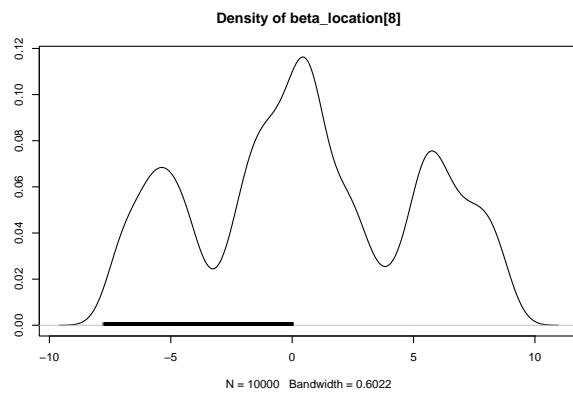
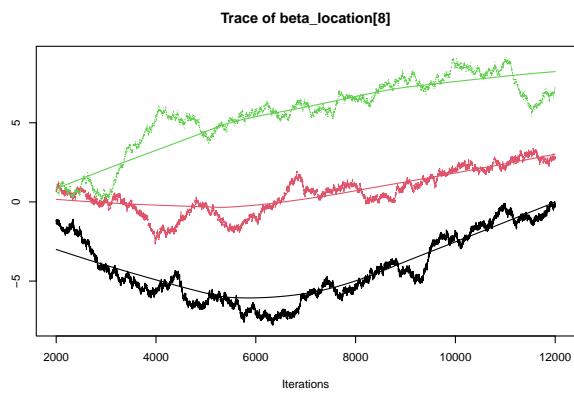
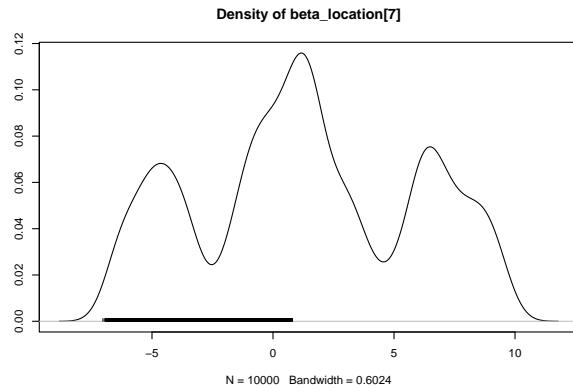
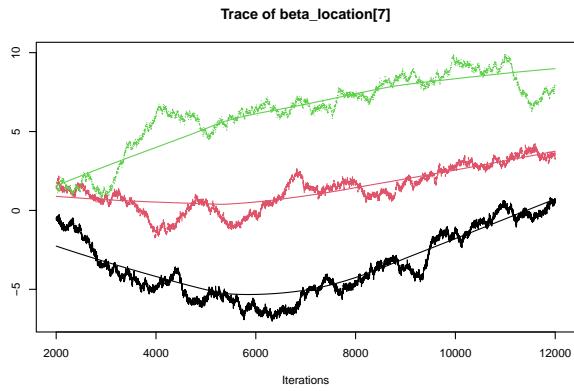
## Compiling model graph
## Resolving undeclared variables
## Allocating nodes
## Graph information:
## Observed stochastic nodes: 27
## Unobserved stochastic nodes: 12
## Total graph size: 209
##
## Initializing model
##
## Iterations = 2001:12000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
## plus standard error of the mean:
##
##           Mean        SD   Naive SE Time-series SE
## beta0      1.0193    4.46580  0.0257833     0.755589
## beta1      0.3961    0.07729  0.0004462     0.001040
## beta2      0.9709    0.09087  0.0005247     0.001089
## beta_location[1] 1.1321    4.46587  0.0257837     0.752344
## beta_location[2] 0.9047    4.46673  0.0257887     0.762519
## beta_location[3] 0.8282    4.46660  0.0257880     0.753364
## beta_location[4] 0.6999    4.46533  0.0257806     0.759187
## beta_location[5] 1.2102    4.46684  0.0257893     0.750266
## beta_location[6] 0.5994    4.46634  0.0257864     0.753408
## beta_location[7] 1.3706    4.46714  0.0257910     0.754171
## beta_location[8] 0.6358    4.46505  0.0257790     0.747852
## beta_location[9] 0.5575    4.46712  0.0257909     0.738430
##
## 2. Quantiles for each variable:
##
##           2.5%     25%     50%     75%   97.5%
## beta0      -6.7201  -3.2529  1.2004  3.9309  8.6998
## beta1       0.2452   0.3442  0.3964  0.4483  0.5475
## beta2       0.7916   0.9093  0.9704  1.0322  1.1492
## beta_location[1] -6.5397  -1.7897  0.9541  5.3749  8.8439
## beta_location[2] -6.7686  -2.0076  0.7276  5.1639  8.6289
## beta_location[3] -6.8403  -2.0810  0.6594  5.0771  8.5484
## beta_location[4] -6.9692  -2.2039  0.5137  4.9478  8.4188
## beta_location[5] -6.4460  -1.7004  1.0219  5.4594  8.9337
## beta_location[6] -7.0655  -2.3337  0.4198  4.8495  8.3296
## beta_location[7] -6.3121  -1.5415  1.1928  5.6314  9.0918
## beta_location[8] -7.0349  -2.2758  0.4590  4.8911  8.3430
## beta_location[9] -7.1071  -2.3565  0.3748  4.8052  8.2685

```









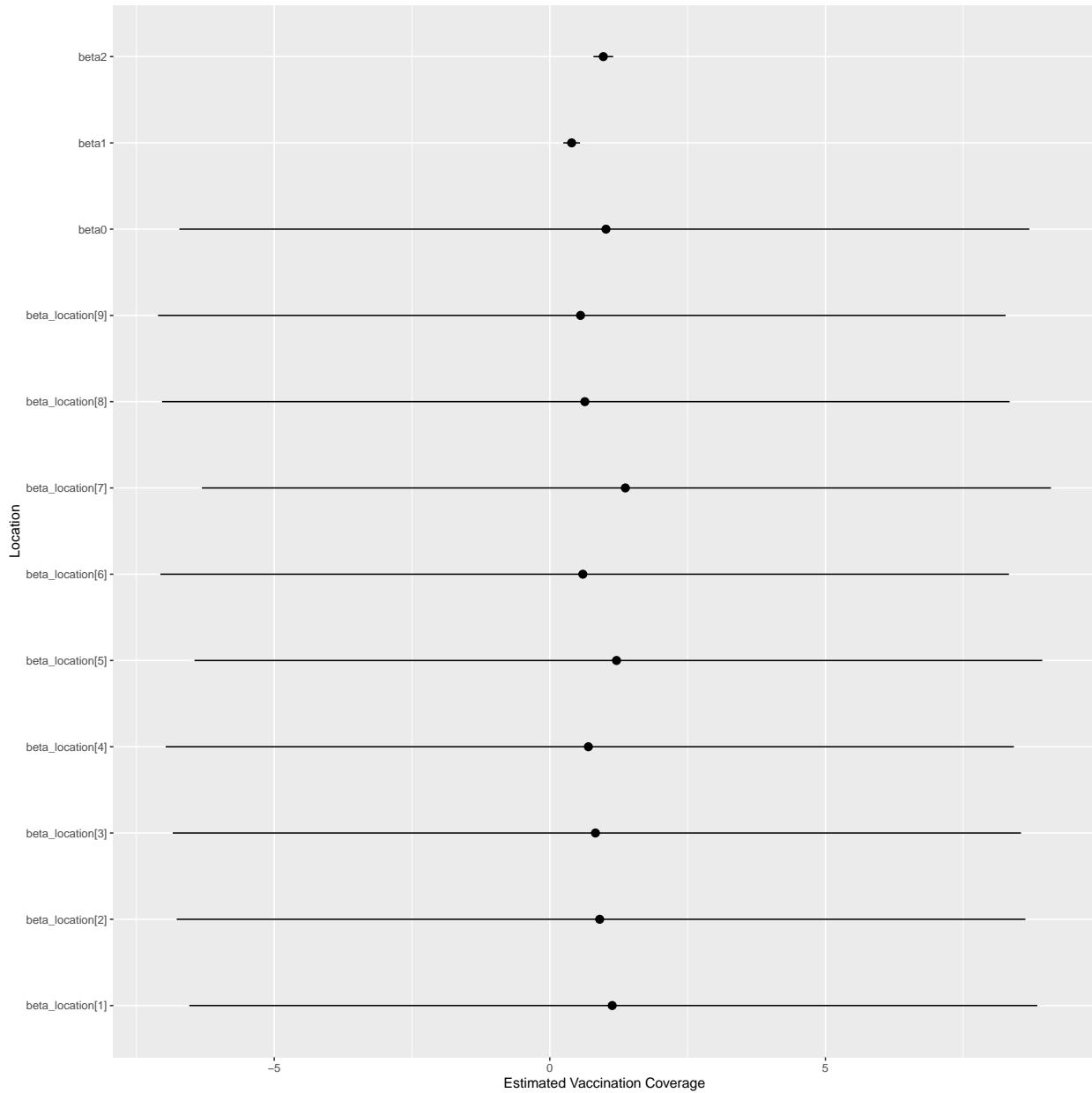
```
## Potential scale reduction factors:
##
##          Point est. Upper C.I.
## beta0      5.32    13.7
## beta1      1.00    1.0
## beta2      1.00    1.0
## beta_location[1] 5.31    13.6
## beta_location[2] 5.30    13.6
## beta_location[3] 5.31    13.6
## beta_location[4] 5.31    13.6
## beta_location[5] 5.31    13.6
## beta_location[6] 5.31    13.6
## beta_location[7] 5.31    13.6
## beta_location[8] 5.31    13.6
```

```
## beta_location[9]      5.31      13.6
##
## Multivariate psrf
##
## 3.79

##          beta0          beta1          beta2 beta_location[1]
##       6.987624     5518.894872    6984.060728     7.108588
## beta_location[2] beta_location[3] beta_location[4] beta_location[5]
##       6.745531      7.067408     6.998928     6.757167
## beta_location[6] beta_location[7] beta_location[8] beta_location[9]
##       7.051708     6.985748     7.119151     7.103143
```

Question 7

My implementation is not correct!!



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.itер = 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)

```