Project Assignment Bayesian Inference
2023-2024
Group 14

1st year Master of Statistics
Hasselt University

Group members:
Joseph Kaunda (2364031)
Luong Vuong (2365900)
Mane Minasyan (2364084)
Melissa Bovijn (2366593)
Henry Hockliffe (2366021)

Submission Date: 01 June, 2024

Lecturer:
Prof. Christel Faes

1 Introduction

The Centers for Disease Control and Prevention (CDC) monitors the vaccination rates for Varicella, also known as chickenpox, among young children. Chickenpox is a highly contagious disease caused by the varicella-zoster virus (VZV). The introduction of the chickenpox vaccine has significantly reduced the frequency and severity of the disease. In the United States, the chickenpox vaccine has been included in the standard childhood immunization schedule since the mid-1990s. Since its implementation, there has been a substantial decline in chickenpox cases, hospitalizations, and fatalities. The CDC aims for a vaccination coverage rate of approximately 90% or higher among children to achieve herd immunity and prevent chickenpox outbreaks. The accompanying table summarizes data from a survey conducted in 2020, showing the number of children who received at least one dose of the Varicella vaccine. The data includes the number of vaccinated children compared to the total number of children surveyed, categorized by age and region across five U.S. regions.

2 Questions and Answers

2.1 Question 1

Investigate the trend in the vaccination coverage with age using the model
$Y_{ij} \sim Binom(\pi_{ij}, N_{ij})$ with

$$\text{logit}(\pi_{ij}) = \beta_0 + \beta_1 \cdot \text{Age}_j, \tag{1}$$

where i is the location, j is age and π_{ij} is the vaccination coverage. Assume non-informative priors for the parameters to be estimated. Write the code in BUGS language.
Answer

```
vaccinemodel <- function() {
  for (i in 1:J) {
    for (j in 1:M) {
      Y[i,j] ~ dbin(pi[i,j], N[i,j])
      logit(pi[i, j]) <- beta0 + beta1 * Age[j]
    }
  }

  # Non-informative priors
  beta0 ~ dnorm(0, 1.0E-2)
  beta1 ~ dnorm(0, 1.0E-2)
}
```

2.2 Question 2

Question: Run the MCMC method and check convergence of the MCMC chains. Give the details on how you checked convergence.
Answer
To investigate the trend in the vaccination coverage with age using the model formulated above (in question 1), normal non-informative priors were used. The MCMC method used three Markov chains with both parameters set to 0.1, 0.3 and 0.5 for each chain. All the chains used 2000000 iterations and left out 1000000 burn-in iterations. The analysis was performed in JAGS 4.3.1.
Convergence of the MCMC chains was checked graphically by analysing the trace plots, autocorrelation plots, and running mean plots for all parameters (β_0 and β_1). A formal diagnostic test was also conducted with Gelman-Rubin plots and Heidelberger and Welch's convergence diagnostic.

Trace plots: Monitoring the Markov chains jointly, the trace plots in figure 1 exhibited rapid mixing for all parameters and therefore indicated that the posterior distribution was rapidly explored. The stable pattern shown for all parameters indicated good chain mixing and therefore stationarity.

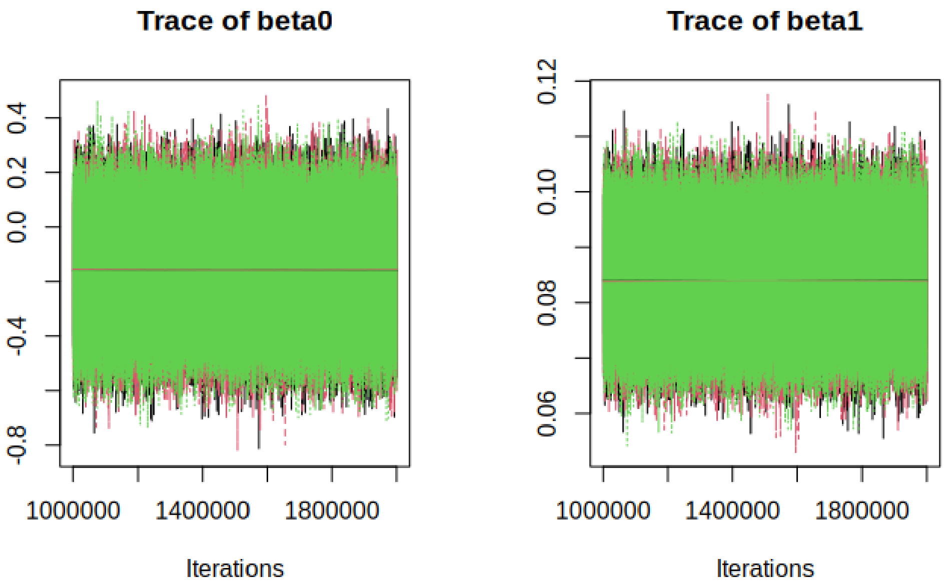

Figure 1: Traceplots for β_0 and β_1

Autocorrelation plots: The joint autocorrelation plots in figure 2 for the Markov chains showed a slow mixing for both β_0 and β_1 but the autocorrelation got to around zero as the number of iterations increased. Higher iterations showed stationarity which indicated independent sampling of the regression parameters.

Running-mean plots: A stable behavior for all the parameters was shown by the running-mean plots (figure 3) which suggested stationarity.

Geweke diagnostic: The Geweke diagnostic compared the mean and variance of two separate segments of the chain to detect any significant differences. With convergence, the mean and variance should be similar across different segments. The diagnostic computed a z-score by comparing the means of the two segments, standardized by the standard deviation of their difference. Almost all the Z-values for β_0 and β_1 were inside the $[-1.96, 1.96]$ interval indicating stationarity (figures 4, 5 and 6) . This confirmed the behavior seen in the trace plot as being stable enough to conclude stationarity.

Heidelberger–Welch (HW) diagnostic: We formally assessed convergence of β_0, β_1 using the Heidelberg and Welch diagnostic (HW) tests. HW tests on a single chain were carried out in two steps. First, stationarity was checked to assess whether the trace of the simulated values were stochastically generated from a stationary process. Secondly, accuracy was determined to assesses the adequacy of the remaining part of the chain, estimating the posterior mean with desired precision. The HW diagnostic was applied with the R program heidel.diag with default settings yielding a 95% confidence interval. The results in table

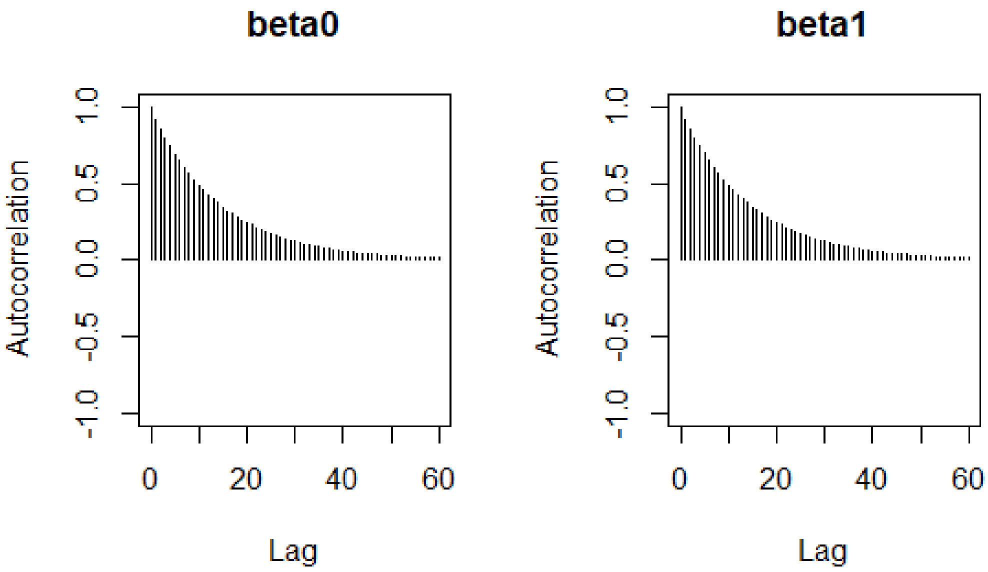

Figure 2: Autocorrelation plot of β_0 and β_1

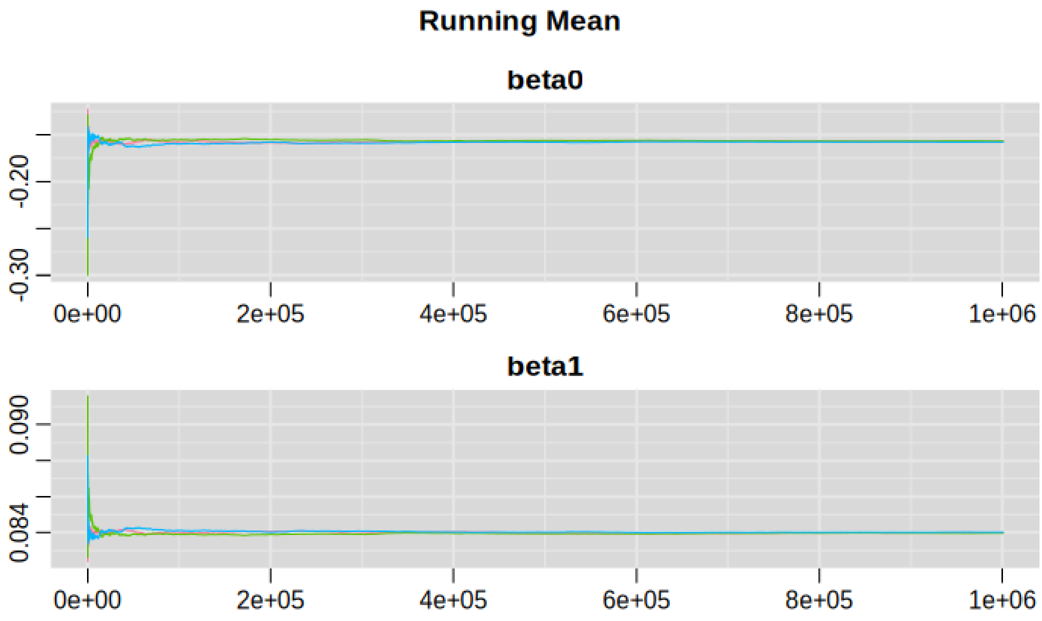

Figure 3: Running-mean plots for β_0 and β_1

1 showed that stationary and Halfwidth tests were passed for β_0 (p=0.597, p=0.614 and, p=0.754 for chain 1,2 and 3 respectively) and β_1 (p=0.599, p=0.620, and p=0.756 for chain 1,2 and 3 respectively) indicating that there was convergence. Therefore, the length of the

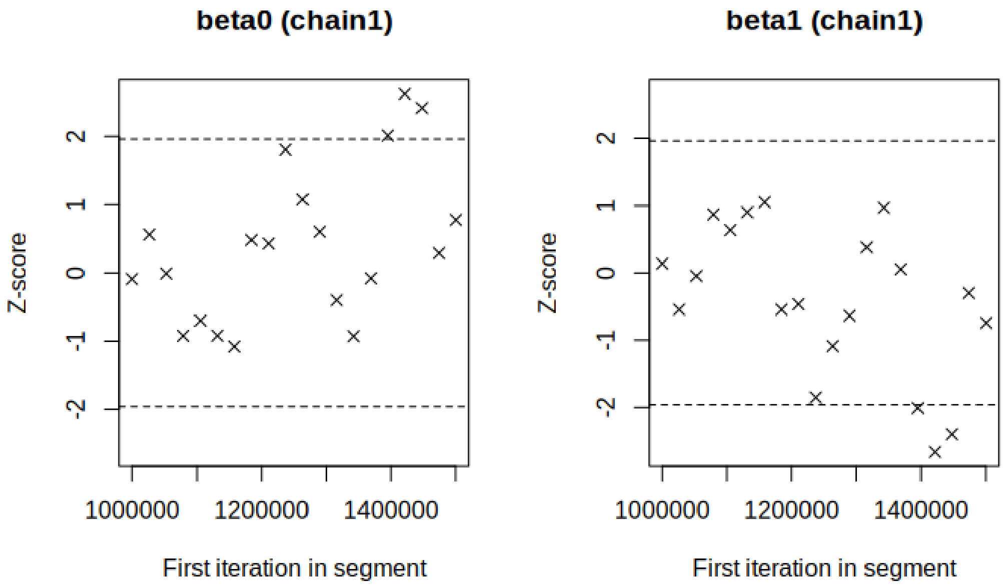

Figure 4: Geweke diagnostic plots for β_0 and β_1 in chain 1

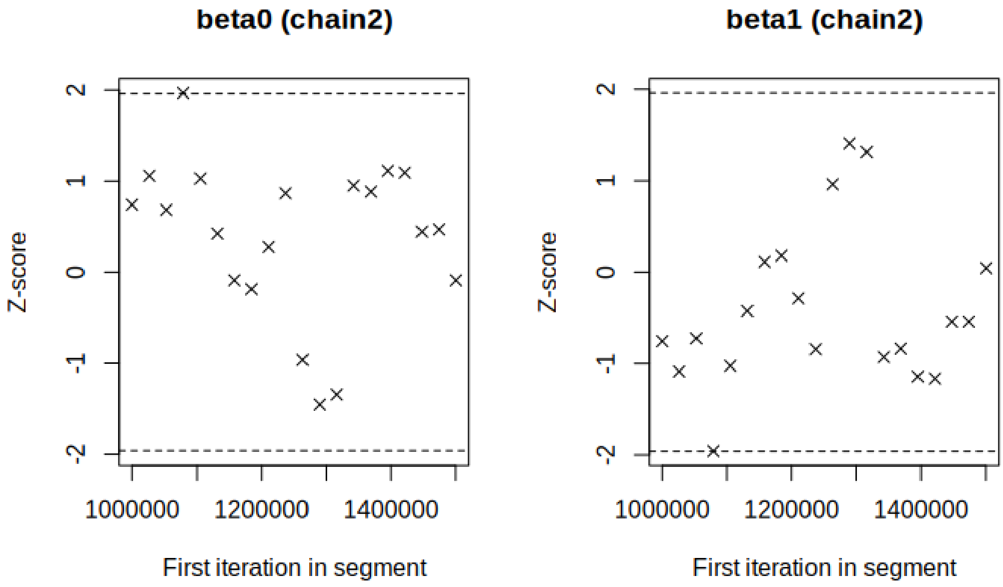

Figure 5: Geweke diagnostic plots for β_0 and β_1 in chain 2

chains and the number of iterations were sufficient for the stationary and halfwidth tests respectively.

Brooks–Gelman–Rubin (BGR) diagnostic: The Gelman-Rubin diagnostic formally tests for convergence by comparing the variances of multiple chains of MCMC simulations to

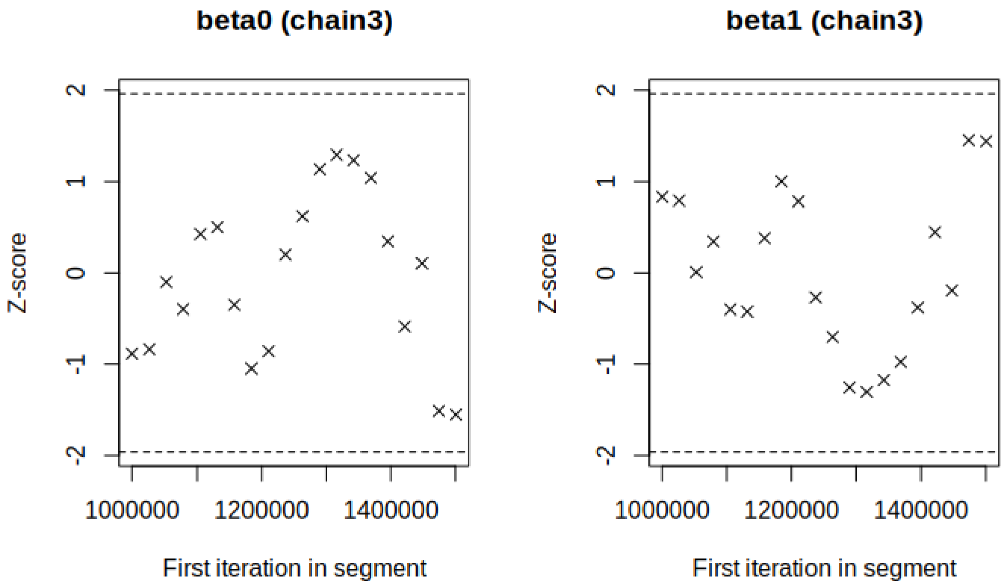

Figure 6: Geweke diagnostic plots for β_0 and β_1 in chain 3

the variance within each chain. Convergence was achieved when the variances are similar such that the ratio of these variances was 1 or close to 1. For both β_0 and β_1, $\hat{R}_c = 1.00$ with 97.5% upper bound equal to 1. This showed good mixing for both regression parameters. The estimated potential scale reduction factor (PSRF) in the study was 1 indicating that convergence is achieved. The graphical plot of BGR (figure 7) showed instant stabilization for β_0 and β_1. No further iterations were necessary to achieve stationarity.

Table 1: Heidelberger-Welch test for analyzing vaccine coverage with age

Stationary test					Halfwidth test		
Chain	Parameter	Test	Iteration	p-value	test	Mean	Halfwidth
chain1							
	β_0	passed	1	0.597	passed	-0.157	1.38e-03
	β_1	passed	1	0.599	passed	0.084	6.67e-05
chain2							
	β_0	passed	1	0.614	passed	-0.157	1.13e-03
	β_1	passed	1	0.620	passed	0.084	6.5e-05
chain3							
	β_0	passed	1	0.754	passed	-0.158	1.34e-03
	β_1	passed	1	0.756	passed	0.084	6.54e-05

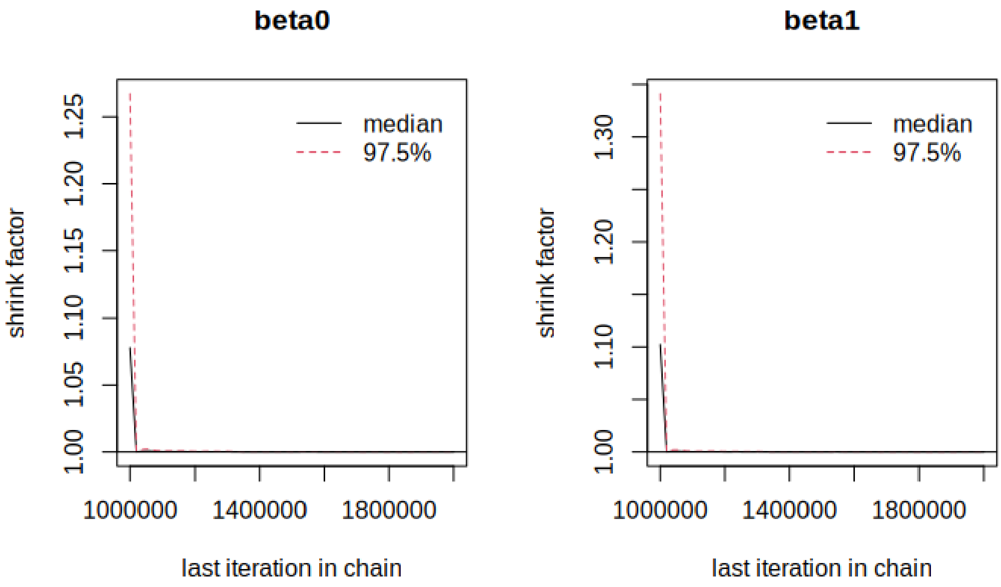

Figure 7: Brooks–Gelman–Rubin (BGR) diagnostic plot of β_0 and β_1

2.3 Question 3

Make a plot of the posterior densities and give summary measures of the posterior distributions of the parameters of interest. Interpret the results.

Answer

The summary statistics of the posterior measured as displayed in table 2 showed the mean, median, standard deviation and the HPD interval. There was a positive association between vaccination coverage and age (β_1=0.0840). With a probability of 95%, we expected the parameter estimate of β_0 and β_1 to have a 95% HPD interval of [-0.4137, 0.0965] and [0.0716, 0.0963] respectively given the evidence provided by the observed data and the prior. The time-series standard error was much smaller than the standard deviation indicating that the parameter estimates were reliable and robust. It also indicated that the data points were closely clustered around the mean, indicating a stable and predictable pattern over time. The mean and median of β_0 and β_1 were almost similar indicating symmetry and that its posterior was normally distributed. This observation was reiterated by the posterior density plots (figure 8) which showed that the posterior parameter estimates (β_0 and β_1) which result from the product of a non-informative prior and likelihood followed a normal distribution.

Table 2: Summary statistics of posterior measures

Parameter	Mean	Stdev	Median	Time-series SE	HPD interval
β_0	-0.1573	0.1302	-0.1570	3.983e-04	[-0.4137, 0.0965]
β_1	0.0840	0.0063	0.0839	1.935e-05	[0.0716, 0.0963]

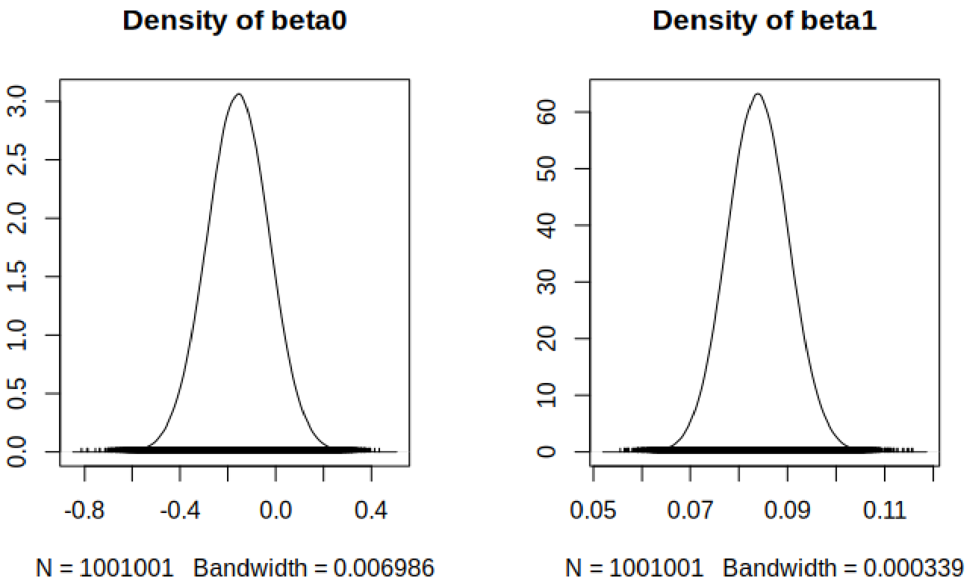

Figure 8: Density plots of posterior measures β_0 and β_1

2.4 Question 4

Question: The target vaccination coverage is 90%. Calculate, for each age group, the posterior probability that the target is reached (i.e. the posterior probability that the vaccination coverage is equal to or above 90%).

Answer :

In this section, we analyzed the varicella vaccination coverage using Bayesian statistics to determine the posterior probability that the vaccination coverage reached at least 90% for various age groups. This analysis was essential for understanding how the likelihood of achieving the target vaccination coverage changed with age.

Model Specification

We began by specifying a Bayesian logistic regression model using JAGS (Just Another Gibbs Sampler). The model assumed that the logit (log-odds) of the vaccination probability was a linear function of age. We ran 2 million iterations with 1 million burn in, generating a sample for each parameter. Note that MCMC settings were similar as those defined in question 2 and 3 above. Next, we defined the age groups of interest and calculated the posterior probability that the vaccination coverage was at least 90% for each age group.

The resulting probability vector for the four age groups was [0, 0, 0, 1] for ages of 13, 19, 24 and 35 months respectively. This indicated the posterior probability that the vaccination coverage met or exceeded 90% for each age group. Specifically, it was very unlikely for age groups of 13, 19, and 24 months old to have at least 90% vaccination coverage rate (0% probability). On the other hand, it was almost certain that the 35-month-old group would have vaccination coverage of 90% or above.

2.5 Question 5

Question: Compare the vaccination coverage at each age group with the vaccination coverage at 13 months of age. Do this by calculating the ratio of the vaccination coverage. Interpret the results.

Answer:

In this analysis, we aimed to compare vaccination coverage for different age groups with vaccination coverage for 13 months of age. The goal was to calculate the coverage ratios for each age group relative to the coverage at 13 months and interpret these ratios using a Bayesian approach.

We calculated the vaccination coverage probabilities from logit (expit) function for each age group. Then we calculated the coverage ratios for each age group compared to the coverage at 13 months by dividing the coverage probabilities at each age by the coverage probability at 13 months. We computed summary statistics (mean and 95% equal tail credible interval) for the coverage ratios. The output summary provided the mean and the 95% credible interval for each age group's coverage ratios relative to the 13-month group's.

13-month Coverage Ratio (Baseline): The ratio is exactly 1, serving as the reference for

Table 3: Ratio of vaccination coverage relative to 13 month

Summary statistics	13 months	19 months	24 months	35 months
Mean	1	1.13	1.21	1.31
2.5th percentile	1	1.10	1.17	1.26
97.5th percentile	1	1.15	1.25	1.36

other age groups.

19-month Coverage Ratio: The coverage ratio is on average 13% higher than the 13-month baseline, with a 95% credible interval ranging from 1.10 to 1.15.

24-month Coverage Ratio: The coverage ratio is on average 21% higher than the 13-month baseline, with a 95% credible interval ranging from 1.17 to 1.25.

35-month Coverage Ratio: The coverage ratio is on average 31% higher than the 13-month baseline, with a 95% credible interval ranging from 1.26 to 1.36.

2.6 Question 6

In this question, we modelled the observed vaccination converage as a function of Age using a non-linear normal regression model (logistic growth model) as follows:

$$r_{ij} = \alpha + \frac{\beta}{1 + e^{-\gamma(\text{Age}_j - \delta)}} + \varepsilon_{ij}$$

where:

- $r_{ij} = \frac{Y_{ij}}{N_{ij}}$ is the observed coverage .
- α: Baseline vaccination coverage.
- β: Maximum effect of age on increasing coverage.
- γ: Steepness of the age-related increase in coverage.
- δ: Age at which the increase in coverage is most rapid.
- $\varepsilon_{ij} \sim N(0, \sigma^2)$: Random variability in coverage not explained by age

The BUGS to model this is as follows

```
    cat("model
{
  for (i in 1:J) {
    for (j in 1:M) {
      Y[i,j] ~ dbin(p[i,j], N[i,j])
      p[i,j] <- alpha + beta / (1 + exp(-gamma * (Age[j] - delta)))
    }
  }

  # Priors
  alpha ~ dbeta(1, 1)  # Non-informative prior for base level
  beta ~ dgamma(0.01, 0.01)  # Non-informative prior for positive increment
  gamma ~ dbeta(1, 1)  # Non-informative prior for positive growth rate
  delta ~ dnorm(0.01, 0.01)  # Non-informative prior
}", file = "varicella_BUGS.txt")
}
```

We chose different non-informative priors for different parameters since we wanted our priors to be conjugate priors, thereby speeding up the converge of our MCMC chains.

Check convergence of the MCMC chains

The parameters in our model were highly correlated, therefore we had to increase the number of iterations, and increased the burn in phase so that our MCMC chains could converge. Specifically, we had 6 million iterations, with 5 million burins. We checked for our convergence using both graphical and hypothesis-testing methods.
First of all, our trace plot (figure 9) showed that for all four parameters, the 3 MCMC chains mingled, suggesting convergence was achieved.
Similarly, our running mean plot (figure 10 illustrated that the mean of each parameter across the entire 3 chains stabilized as the number of iterations increased. This also suggested that our MCMC chains have reached a stationary distribution.
Another tool for checking convergence that we looked at was the plot showing the test statistics of Geweke's Diagnostics (figure 11). We saw that all the test statistics of all parameters in all 3 chains fell within the regions between -1.96 and 1.96. This means that, for all our parameters, the means at the first 10% of every chain were not significantly different from the ones at the latter 50% of the chain. This was at the 5% significance level. This test provided us with more evidence that our convergence has been reached.
Finally, we looked at Gelman diagnostics. Specifically, we looked at the scale reduction factors for each parameter. We wanted to achieve a factor of 1, which means that between-chain variance and within-chain variance are equal, thus convergence is obtained. In our study, this was the case, as is shown by both our table 4 and figure 12
All of the diagnostics we looked at brought us to the same conclusion that our MCMC chains with 6 million iterations and 5 million burin have reached the convergence.

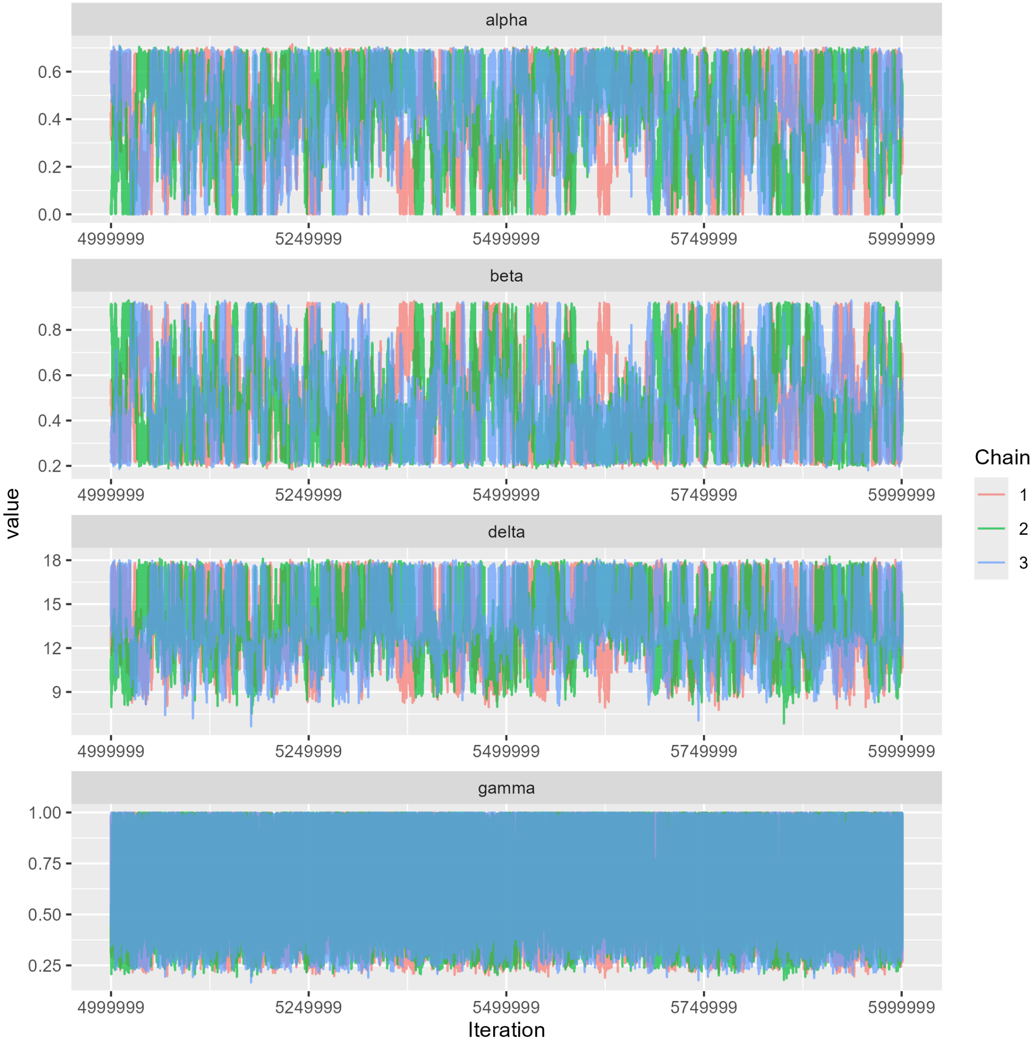

Figure 9: Trace plot log growth model

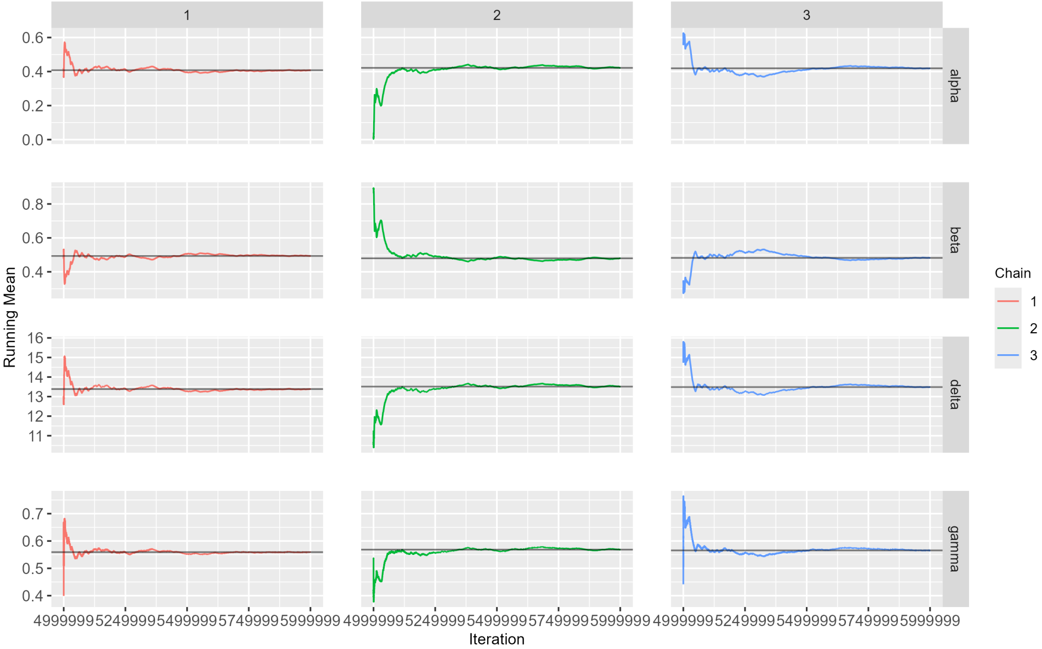

Figure 10: Running mean log growth model

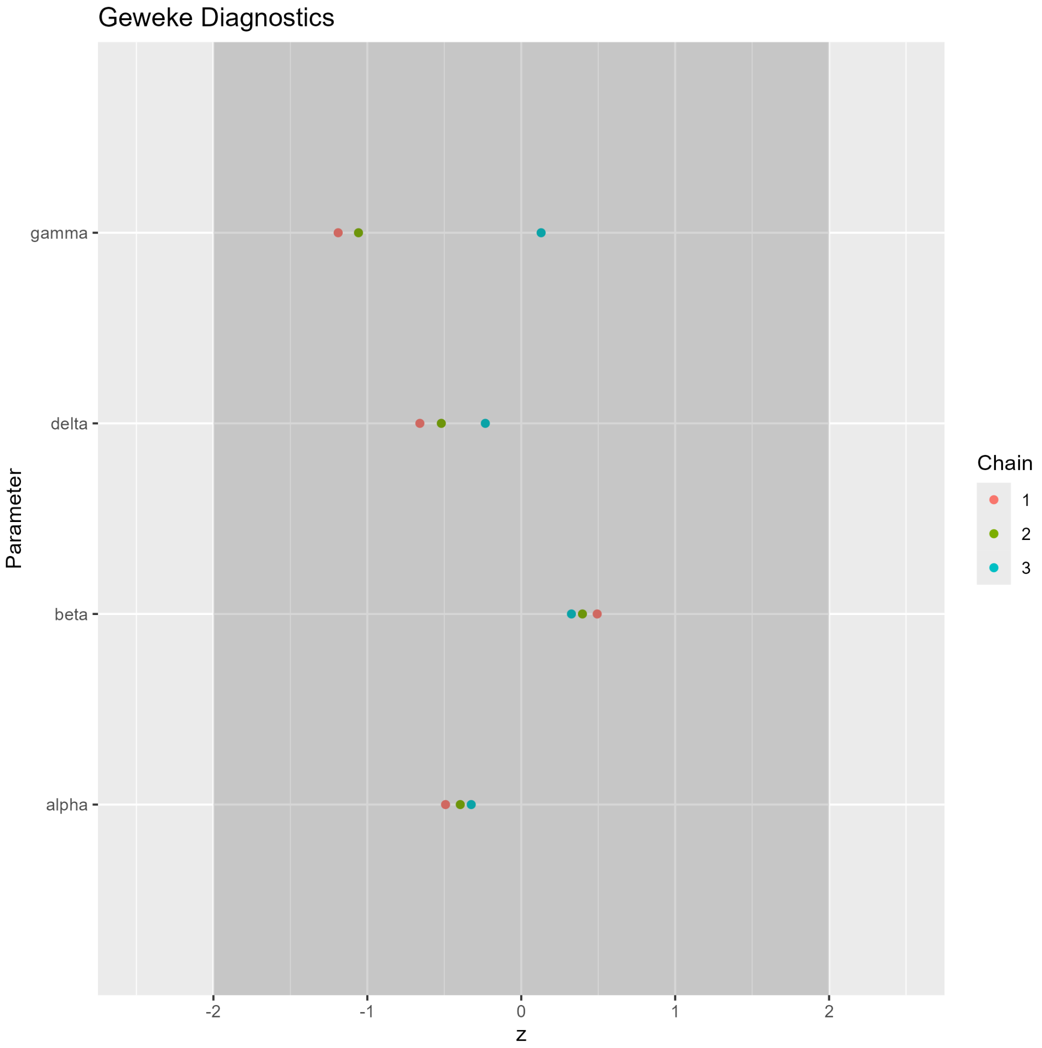

Figure 11: Geweke diagnostic plot log growth model

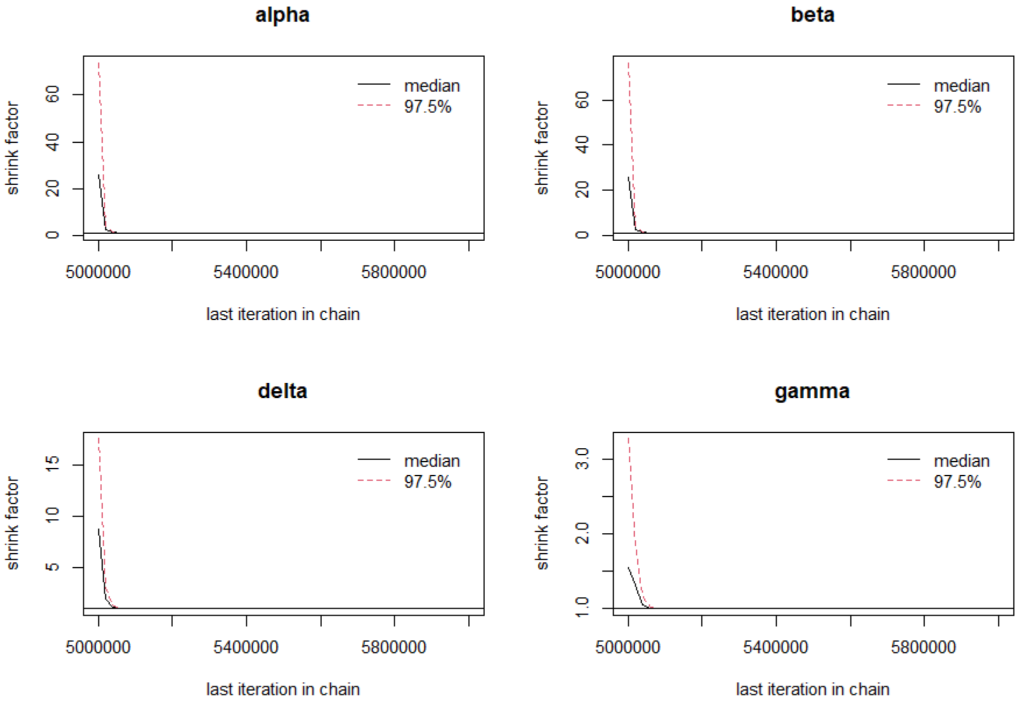

Figure 12: Gelman diagnostic plot log growth model

Table 4: Potential scale reduction factors and multivariate PSRF

Parameter	Point Est.	Upper C.I.
alpha	1	1
beta	1	1
delta	1	1
gamma	1	1
Multivariate PSRF		1

2.7 Question 7

Make a plot of the estimated coverage as a function of age, including the uncertainty on the estimates. Include also the observed vaccination proportion in the plot.

Answer:

From question 6, we had the estimates of all our parameters across all the 3 MCMC chains. These estimates allowed us to calculate the prediction of vaccination coverage at four different age groups. These included 13 months, 19 months, 24 months, and 35 months. We plotted the vaccination coverage prediction in each age group together with the observed coverage rate per location in figure 13. Our prediction included a solid line demonstrating the median coverage, while the shaded green area illustrated the 95% equal-tailed prediction interval.

The observed coverage rate per age at different regions was presented with points of different colors.

Overall, our model predicted the median observed vaccination coverage well, whereas our 95% prediction interval could not capture all the observed vaccination coverage.

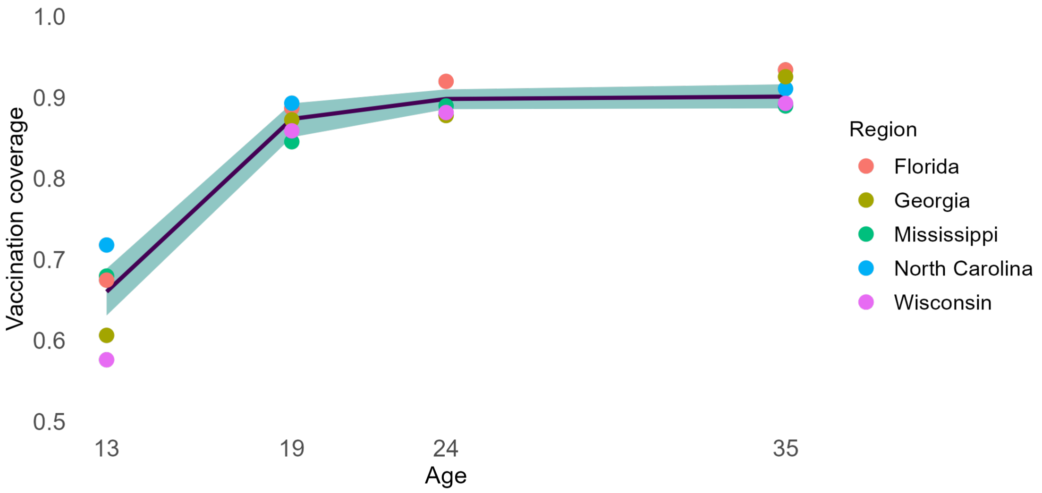

Figure 13: Vaccination coverage by age group

2.8 Question 8

Predict the number of vaccinated children at the age of 15 months (per location).
Answer:
Similarly to question 7, we used our MCMC chains from question 6 to predict vaccination coverage at the age of 15 months. We then derived the uncertainty estimates of the coverage, which allowed us to predict the number of vaccinated children at 15 months of age per location. Our results are shown in table 5.
Overall, North Carolina had the highest coverage prediction while the lowest numbers were at Mississippi and Wisconsin.

Table 5: Predicted number of vaccinated children by location

Location	Median	95% Prediction Interval
Mississippi	137	125–148
Florida	161	146–173
North Carolina	212	193–229
Georgia	143	130–154
Wisconsin	134	122–145

3 Appendix - R/BUGS code

3.0.1 Question 1:

```
vaccinemodel <- function() {
  for (i in 1:J) {
    for (j in 1:M) {
      Y[i,j] ~ dbin(pi[i,j], N[i,j])
      logit(pi[i, j]) <- beta0 + beta1 * Age[j]
    }
  }

  # Non-informative priors
  beta0 ~ dnorm(0, 1.0E-2)
  beta1 ~ dnorm(0, 1.0E-2)
}
```

3.0.2 Questions 2 and 3:

```
library("R2OpenBUGS")
library("coda")
library("readr")
library("mcmcplots")

varicella_vaccine_coverage <- read_csv("varicella_vaccine_coverage.csv")

# convert first two columns to factors
varicella_vaccine_coverage$Geography <-
    as.factor(varicella_vaccine_coverage$Geography)
varicella_vaccine_coverage$Age <- as.factor(varicella_vaccine_coverage$Age)
# change Age to Age_former
varicella_vaccine_coverage$Age_former <- varicella_vaccine_coverage$Age

# Add Age column
Age <- rep(c(13, 19, 24, 35), 5)
varicella_vaccine_coverage$Age <- Age

## write the BUGS program and put it in a txt file
cat("model
{
  for (i in 1:J) {
    for (j in 1:M) {
      Y[i,j] ~ dbin(pi[i,j], N[i,j])
      logit(pi[i, j]) <- beta0 + beta1 * Age[j]
    }
  }

  # Priors
  beta0 ~ dnorm(0, 1.0E-2) # Non-informative prior for base level
  beta1 ~ dnorm(0, 1.0E-2) # Non-informative prior for
}", file = "varicella_BUGS.txt")

## prepare the data and collect them into the object ‘my.data’
# Prepare the data
Y <- matrix(varicella_vaccine_coverage$Vaccinated, nrow = 5, ncol = 4,
    byrow = TRUE)
N <- matrix(varicella_vaccine_coverage$Sample_Size, nrow = 5, ncol = 4,
    byrow = TRUE)
Age <- unique(varicella_vaccine_coverage$Age)

my_data <- list(
  J = nrow(Y),
  M = ncol(Y),
  Y = Y,
  N = N,
  Age = Age
)

my.inits <- list(
  list(beta0 = 0.5,
      beta1 = 0.5,
      .RNG.name = "base::Wichmann-Hill",
       .RNG.seed = 1),
  list(beta0 = 0.3,
      beta1 = 0.3,
      .RNG.name = "base::Wichmann-Hill",
       .RNG.seed = 2),
  list(beta0 = 0.1,
      beta1 = 0.1,
      .RNG.name = "base::Wichmann-Hill",
       .RNG.seed = 3))

# Parameters to monitor
parameters <- c("beta0", "beta1")

# Run the MCMC
library(rjags)
jags_model <- jags.model(file = 'varicella_BUGS.txt',
                      data = my_data,
                      inits = my.inits,
                      n.chains = 3)

coverage.sim <- coda.samples(jags_model,
                         parameters,
                         n.iter = 2000000,
                         thin = 1)



# account for burnin
burnin <- 1000000
coverage.sim <- window(coverage.sim, start = burnin)

# Posterior summary statistics
summary(coverage.sim)

## Produce general summary of obtained MCMC sampling
#print(coverage.sim)
plot(coverage.sim)
## Convert coverage.mcmc into mcmc.list for processing with CODA

coverage.mcmc <- as.mcmc.list(coverage.sim)

# summary and HPDinterval of mcmc
summary(coverage.mcmc)
HPDinterval(coverage.mcmc) # HPD intervals of all parameters

# Check convergence - autocorrelation, running-mean plots

# trace plots
plot(coverage.mcmc, trace = TRUE, density = FALSE)

# autocorrelation
acfplot(coverage.mcmc[,1:2])
autocorr.plot(coverage.mcmc)

traceplot(coverage.mcmc) # trace plots

# running-mean
rmeanplot(coverage.mcmc, plot.title = "Running Mean")

# Check convergence - Gelman-rubin plots
gelman.diag(coverage.mcmc)
gelman.plot(coverage.mcmc)

## Heidelberger and Welch’s convergence diagnostic
```

```r
heidel.diag(coverage.mcmc, eps=0.1, pvalue=0.05)

# Brooks--Gelman--Rubin (BGR) diagnostic

gelman.diag(coverage.mcmc, confidence = 0.95, transform=FALSE,
    autoburnin=TRUE,
                   multivariate=TRUE)
gelman.plot(coverage.mcmc)

# History plot & posterior distributions
plot(coverage.mcmc, trace = TRUE, density = FALSE)
plot(coverage.mcmc, trace = FALSE, density = TRUE)

# combined HPD interval for 3 chains
library(MCMCvis)

MCMCsummary(coverage.mcmc,
           params = 'all', ISB = FALSE, exact = TRUE, Rhat = TRUE, n.eff =
               TRUE,
           HPD = TRUE, hpd_prob = 0.95, round = 4)
HPDinterval(coverage.mcmc) # HPD intervals of all parameters

# posterior densities
plot(coverage.mcmc, trace = FALSE, density = TRUE)
summary(coverage.mcmc) # summary statistics
```

3.0.3 Questions 4

```r
library(coda)
library(ggmcmc)
library(R2OpenBUGS)
# Load the MCMC samples from a JAGS model
coverage.sim <- coda.samples(jags_model,
                          parameters,
                          n.iter = 2000000,
                          thin = 1)

# Define the burn-in period
burnin <- 1000000
coverage.sim <- window(coverage.sim, start = burnin)

# Create a copy of the MCMC samples for further processing
coverage.sim_04 <- coverage.sim

# Select the samples after the burn-in period
```

```r
coverage.sim_04_1_2_mil <- window(coverage.sim_04, start = burnin)

# Convert the MCMC samples to a ggs object for easier manipulation
out.ggs_1_2_mil <- ggs(coverage.sim_04_1_2_mil)

# Extract samples for the intercept parameter (beta0)
beta0_samples <- out.ggs_1_2_mil[out.ggs_1_2_mil$Parameter == "beta0", "value"]

# Extract samples for the slope parameter (beta1)
beta1_samples <- out.ggs_1_2_mil[out.ggs_1_2_mil$Parameter == "beta1", "value"]

# Check if the number of samples for beta0 and beta1 match
n_samples <- length(beta0_samples)
if (length(beta1_samples) != n_samples) {
  stop("Mismatch in number of samples for beta0 and beta1")
}

# Define the age groups for which coverage probabilities will be calculated
Age <- c(13, 19, 24, 35)

# Define the expit function (logistic function) to transform log-odds to probabilities
expit <- function(x) {
  return(exp(x) / (1 + exp(x)))
}

# Initialize a data frame to store the coverage probabilities
coverage_probabilities <- data.frame(Age = Age, Probability = numeric(length(Age)))

# Calculate the coverage probabilities for each age group
for (i in 1:length(Age)) {
  age <- Age[i]
  coverage <- expit(beta0_samples + beta1_samples * age)
  prob_gt_09 <- mean(coverage >= 0.9)
  coverage_probabilities$Probability[i] <- prob_gt_09
}

# Print the coverage probabilities for each age group
print(coverage_probabilities)
```

3.0.4 Question 5

```
coverage.sim <- coda.samples(jags_model,
                             parameters,
                             n.iter = 2000000,
                             thin = 1)
burnin <- 1000000
coverage.sim <- window(coverage.sim, start = burnin)
coverage.sim_04 <- coverage.sim
# Set the starting point for the simulation window (after burn-in period)
coverage.sim_04_2_4_mil <- window(coverage.sim_04, start = burnin)

# Convert the MCMC simulation object to a ggs object for easier manipulation
out.ggs_2_4_mil <- ggs(coverage.sim_04_2_4_mil)

# Extract samples for the intercept parameter (beta0) from the ggs object
beta0_samples <- out.ggs_2_4_mil[out.ggs_2_4_mil$Parameter == "beta0", "value"]

# Extract samples for the slope parameter (beta1) from the ggs object
beta1_samples <- out.ggs_2_4_mil[out.ggs_2_4_mil$Parameter == "beta1", "value"]

# Define the ages at which coverage is to be calculated
Age <- c(13, 19, 24, 35)

# Define the expit function (logistic function) to transform log-odds to probabilities
expit <- function(x){
  return (exp(x) / (1 + exp(x)))
}

# Calculate coverage for each age using the samples of beta0 and beta1
coverage <- sapply(Age, function(age) {
  expit(beta0_samples + beta1_samples * age)
})

# Convert the list of coverage values to a matrix, where each column corresponds to an age
coverage <- matrix(unlist(coverage), ncol = length(Age), byrow = FALSE)

# Extract coverage values at 13 months (the first column of the matrix)
coverage_13 <- coverage[, 1]

# Calculate the ratios of coverage at each age to the coverage at 13 months
ratios <- sweep(coverage, 1, coverage_13, "/")
```

```r

# Summarize the ratios by calculating the mean and 95% confidence interval for each age
summary_ratios <- apply(ratios, 2, function(x) {
  c(mean = mean(x), quantile(x, c(0.025, 0.975)))
})

# Print the summary of the ratios
print(summary_ratios)
```

3.0.5 Question 6:

```r
## load the data
varicella_vaccine_coverage <- read_csv("varicella_vaccine_coverage.csv")

# convert first two columns to factors
varicella_vaccine_coverage$Geography <- as.factor(varicella_vaccine_coverage$Geography)
varicella_vaccine_coverage$Age <- as.factor(varicella_vaccine_coverage$Age)
# change Age to Age_former
varicella_vaccine_coverage$Age_former <- varicella_vaccine_coverage$Age

# Add Age column
Age <- rep(c(13, 19, 24, 35), 5)
varicella_vaccine_coverage$Age <- Age

## write the BUGS program and put it in a txt file
cat("model
{
  for (i in 1:J) {
    for (j in 1:M) {
      Y[i,j] ~ dbin(p[i,j], N[i,j])
      p[i,j] <- alpha + beta / (1 + exp(-gamma * (Age[j] - delta)))
    }
  }

  # Priors
  alpha ~ dbeta(1, 1) # Non-informative prior for base level
  beta ~ dgamma(0.01, 0.01) # Non-informative prior for positive increment
  gamma ~ dbeta(1, 1) # Non-informative prior for positive growth rate
  delta ~ dnorm(0.01, 0.01) # Non-informative prior
}", file = "varicella_BUGS.txt")

## prepare the data and collect them into the object ‘my.data’
# Prepare the data
```

```
Y <- matrix(varicella_vaccine_coverage$Vaccinated, nrow = 5, ncol = 4,
    byrow = TRUE)
N <- matrix(varicella_vaccine_coverage$Sample_Size, nrow = 5, ncol = 4,
    byrow = TRUE)
Age <- unique(varicella_vaccine_coverage$Age)

my_data <- list(
  J = nrow(Y),
  M = ncol(Y),
  Y = Y,
  N = N,
  Age = Age
)

## set the initial values

# Initial values
my.inits <- list(
  list(alpha = 0.5,
      beta = 0.5,
      gamma = 0.5,
      delta = 20,
      .RNG.name = "base::Wichmann-Hill",
      .RNG.seed = 1),
  list(alpha = 0.3,
      beta = 0.3,
      gamma = 0.3,
      delta = 30,
      .RNG.name = "base::Marsaglia-Multicarry",
      .RNG.seed = 2),
  list(alpha = 0.4,
      beta = 0.4,
      gamma = 0.4,
      delta = 25,
      .RNG.name = "base::Super-Duper",
      .RNG.seed = 3))

## collect the parameters to be monitored
# Parameters to monitor
parameters <- c("alpha", "beta", "gamma", "delta")

## run the MCMC chain
# Run the MCMC
jags_model <- jags.model(file = 'varicella_BUGS.txt',
                      data = my_data,
                      inits = my.inits,
                      n.chains = 3)

coverage.sim_02 <- coda.samples(jags_model,
                             parameters,
                             n.iter = 6000000,
                             thin = 1)

coverage.sim_02_5_6_mil <- window(coverage.sim_02, start = 5000000)

# Posterior summary statistics
summary(coverage.sim_02_5_6_mil)

### check with ggmcmc

# convert from mcmc.list to a dataset
out.ggs_5_6_mil_thin1 <- ggs(coverage.sim_02_5_6_mil)

# create traceplot object
trace_plot_5_6_mil_thin1 <- ggs_traceplot(out.ggs_5_6_mil_thin1)

# make a running mean plot
running_mean_5_6_mil_thin1 <- ggs_running(out.ggs_5_6_mil_thin1) +
  theme(axis.text = element_text(size = 10),
       axis.title = element_text(size = 10),
       legend.text = element_text(size = 9),
       legend.title = element_text(size = 9),
       strip.text = element_text(size = 9),
       panel.spacing = unit(1, "lines"))

geweke.plot_5_6_mil_thin1 <- ggs_geweke(out.ggs_5_6_mil_thin1)

### Convergence test
# shrunken factor
gelman.diag(coverage.sim_02_5_6_mil, autoburnin = FALSE, transform = TRUE)

# plot this diagnostic
gelman.plot(coverage.sim_02_5_6_mil, autoburnin = FALSE)
```

3.0.6 Question 7:

```
# use out.ggs_5_6_mil_thin1 for this question (see also question 6)
# convert into a wide format
out.ggs_5_6_mil_thin1_wide <- out.ggs_5_6_mil_thin1 |>
  pivot_wider(names_from = "Parameter", values_from = "value")
```

```r

# rep out.ggs_5_6_mil_thin1_wide 4 times
out.ggs_5_6_mil_thin1_wide <- rbind(out.ggs_5_6_mil_thin1_wide) |>
  rbind(out.ggs_5_6_mil_thin1_wide) |>
  rbind(out.ggs_5_6_mil_thin1_wide) |>
  rbind(out.ggs_5_6_mil_thin1_wide)

# add Age, which has 4 values (13, 19, 24, 35), each is repeated nrow times
out.ggs_5_6_mil_thin1_wide <- out.ggs_5_6_mil_thin1_wide |>
  mutate(Age = rep(c(13, 19, 24, 35), nrow(out.ggs_5_6_mil_thin1_wide)/4))

# add coverage column, which equals: alpha + beta / (1 + exp(-gamma * (Age - delta)))
out.ggs_5_6_mil_thin1_wide <- out.ggs_5_6_mil_thin1_wide |>
  mutate(coverage = alpha + beta / (1 + exp(-gamma * (Age - delta))))

# summary statistics
coverage_predict <- out.ggs_5_6_mil_thin1_wide |>
  group_by(Age) %>%
  summarize(median = median(coverage),
           min = quantile(coverage, 0.025),
           max = quantile(coverage, 0.975),
           .groups = ’drop’)

# add coverage column to varicella_vaccine_coverage dataset, which is Vaccinated/Sample_size
varicella_vaccine_coverage <- varicella_vaccine_coverage |>
  mutate(coverage = Vaccinated/Sample_Size)

# add coverage and Geography columns from varicella_vaccine_coverage to coverage_predict dataset matching by Age
coverage_predict_observed <- coverage_predict |>
  left_join(varicella_vaccine_coverage |> select(Age, Geography, coverage), by = "Age")

# plot the observed and predicted coverage
coverage_by_age <- ggplot(data = coverage_predict_observed, aes(x = Age, y = median)) +
  geom_ribbon(aes(ymin = min, ymax = max), fill = "#21918c", alpha = 0.5) +
  geom_line(color = "#440154", size = 0.8) +
  geom_point(aes(y = coverage, color = Geography)) +
  ylab("Vaccination coverage") +
  scale_x_continuous(limits = c(12, 36), breaks = c(13, 19, 24, 35), expand = c(0, 0.01)) +
  scale_y_continuous(limits = c(0.5, 1), breaks = seq(0.5, 1, by =
      0.1),expand = c(0, 0.01)) +
  theme_minimal() +
  theme(axis.title = element_text(size = 8, family = "sans"),
       axis.text = element_text(size = 8, family = "sans"),
       legend.title = element_text(size = 7, family = "sans"),
       legend.text = element_text(size = 7, family = "sans"),
       panel.grid.major = element_blank(),
       panel.grid.minor = element_blank()) +
  labs(color = "Region")
coverage_by_age
```

3.0.7 Question 8:

```
# prediction at age 15
predict_15 <- out.ggs_5_6_mil_thin1_wide |>
  filter(Age == 13) |>
  mutate(Age = 15) |>
  mutate(coverage = alpha + beta / (1 + exp(-gamma * (Age - delta))))

predict_15_summary <- predict_15 |>
  summarize(median = median(coverage),
           min = quantile(coverage, 0.025),
           max = quantile(coverage, 0.975))

# add Geography with Sample_Size
predict_15_summary_geo <- rbind(predict_15_summary,
predict_15_summary,
predict_15_summary,
predict_15_summary,
predict_15_summary) |>
  mutate(Geography = unique(varicella_vaccine_coverage$Geography),
         Sample_Size = unique(varicella_vaccine_coverage$Sample_Size))

predict_15_summary_geo <- predict_15_summary_geo |>
  mutate(Median = round(median * Sample_Size, 0),
        Min = round(min * Sample_Size, 0),
        Max = round(max * Sample_Size, 0))

# Extract Geography, Min, Median, Max, and export to latex
predict_15_summary_geo |>
  select(Geography, Min, Median, Max)
```