Concepts of Bayesian Data Analysis

	Raïsa Carmen *s0204278*	**Barouyr Demirjian** *r0916873*
Group 25:	**Vancesca Dinh** *r0830510*	**Chaojie Huang** *r0816192*
	Arthur Terlinden	

1 Task A

Question 1

Here, we write question 1

2 Task B: Dose-response model

(1) Assume that the likelihood of the experiment is specified by

$$y \sim binomial(N, \pi)$$

$$logit(\pi) = \alpha + \beta d.$$

Here β is the parameter of interest. Take vague priors for α and β. Write jags, OpenBugs or Nimble code for this problem. Take 2 MCMC chains with different starting values, and check convergence with the appropriate techniques.

To denote the uncertainty of parameters α and β, we specify two vague priors as a massive variance (extremely small precision) for these two parameters. The first vague prior is:

$$\alpha \sim N(0, 10000)$$

$$\beta \sim N(0, 10000)$$

And the second vague prior is:

$$\alpha \sim t(0, 0.0001, 5)$$

$$\beta \sim t(0, 0.0001, 5)$$

Our parameter settings include running 2 MCMC chains with different starting values for α and β. The first chain starts with ($\alpha = 0$, $\beta = 0$), while the second chain starts with ($\alpha = -0.5$,$\beta = 0.1$). We run each chain for a total of 10000 iterations, with the first 5000 iterations used as a burn-in period.

To assess the convergence of the model with a prior normal distribution, we employed several diagnostic tools including trace plots, the Gelman-Rubin diagnostic test, and autocorrelation plots. We observed from the trace plots of α and β (refer to Figure 1) that the estimates from each chain quickly stabilized around a steady state. Additionally, both chains were found to converge around the same conclusion. Furthermore, we conducted the Gelman-Rubin diagnostic test, which showed that the estimated potential scale reduction factors of α and β were both 1. These results suggest that our model has converged well. Moreover, the Gelman-Rubin diagnostic plots (refer to Figure 1) support this conclusion, as both the potential scale reduction factors of α and β were found to decrease quickly and remain stable as the number of iterations increased. We also examined the autocorrelation plots (refer to Figure 2), which indicated low autocorrelation. The autocorrelation decreased and remained around zero as the lag number increased, indicating that the chains have mixed well. Overall, these diagnostic tools suggest that our model that the prior is normal distribution has converged well, and the inference based on the Markov chain Monte Carlo simulation is reliable.

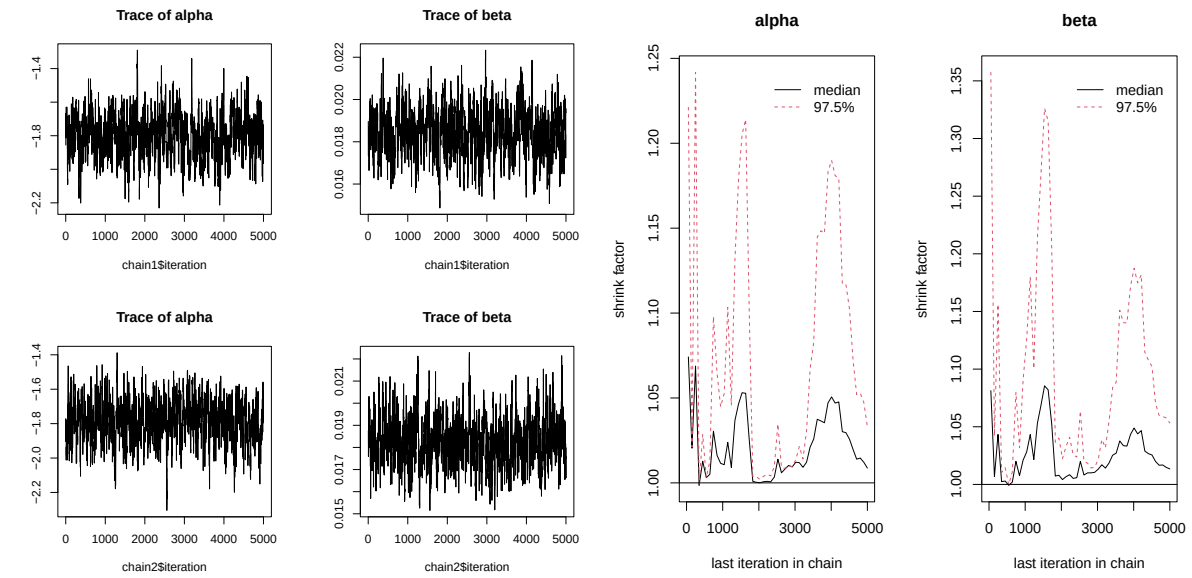

Figure 1: Trace and Gelman-Rubin diagnostic plots of α and β (prior: normal distribution)

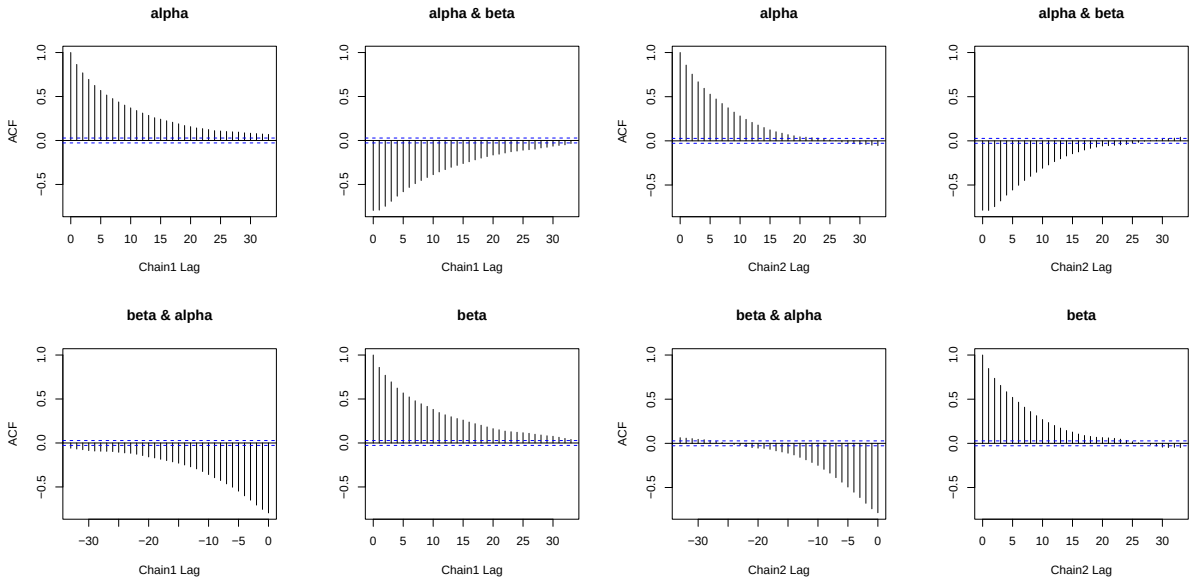

Figure 2: Autocorrelation plots of α and β (prior: normal distribution)

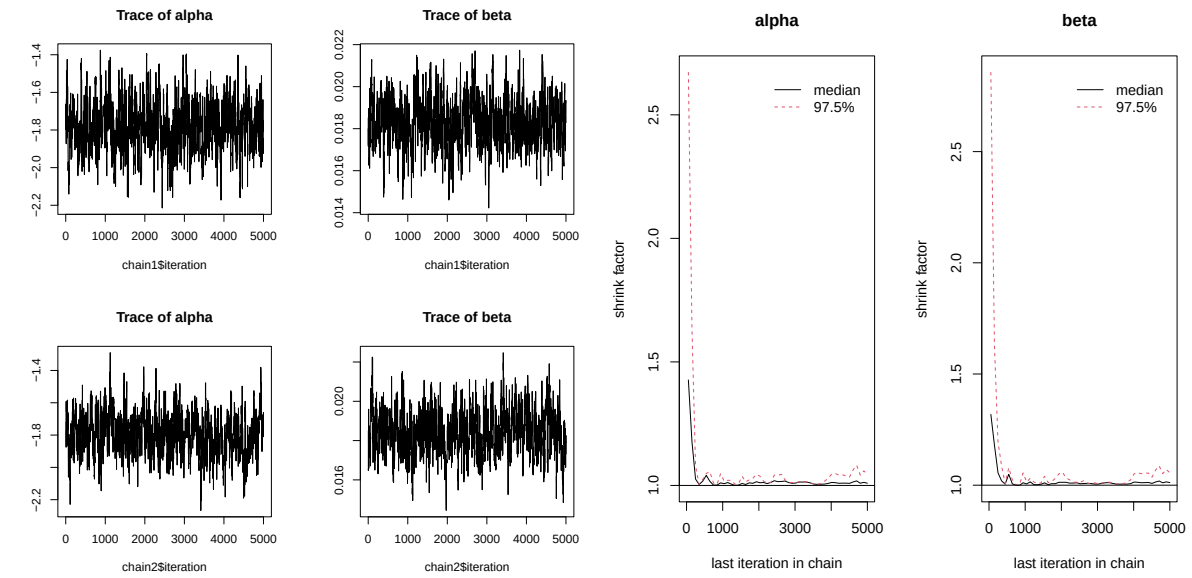

Figure 3: Trace and Gelman-Rubin diagnostic plots of α and β (prior: t-distribution)

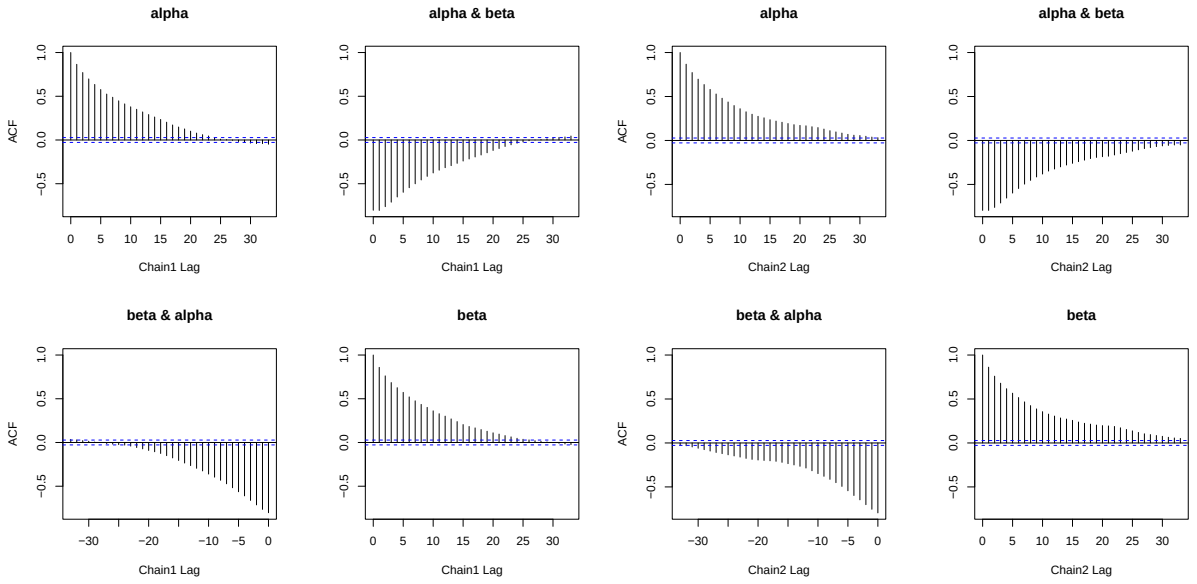

Figure 4: Autocorrelation plots of α and β (prior: t-distribution)

Table 1: Bayesian posterior measures of α and β

	Mean	Median	St.Dev	HPD Interval
N(0, 10000)				
alpha	-1.7931153	-1.7912688	0.1253469	[-2.052, -1.568]
beta	0.0183311	0.0182931	0.0010990	[0.016, 0.02]
t(0, 0.0001, 5)				
alpha.1	-1.7945744	-1.7922911	0.1326002	[-2.059, -1.537]
beta.1	0.0183608	0.0183749	0.0011560	[0.016, 0.02]

We analyzed the convergence of the model with a prior t-distribution using trace plots, autocorrelation plots, and Gelman-Rubin diagnostic plots, which are presented in Figure 3 and Figure 4. We observed that all the trends remained consistent with the plots obtained using a normal distribution for α and β. Moreover, the estimated potential scale reduction factors of α and β were found to be equal to 1, indicating good convergence of the model. These results provide compelling evidence that the model with a prior t-distribution has achieved good convergence and that the inference derived from the Markov chain Monte Carlo simulation is reliable.

(2) Summarize all results graphically and summarize with the usual Bayesian posterior measures. What do you conclude from these?

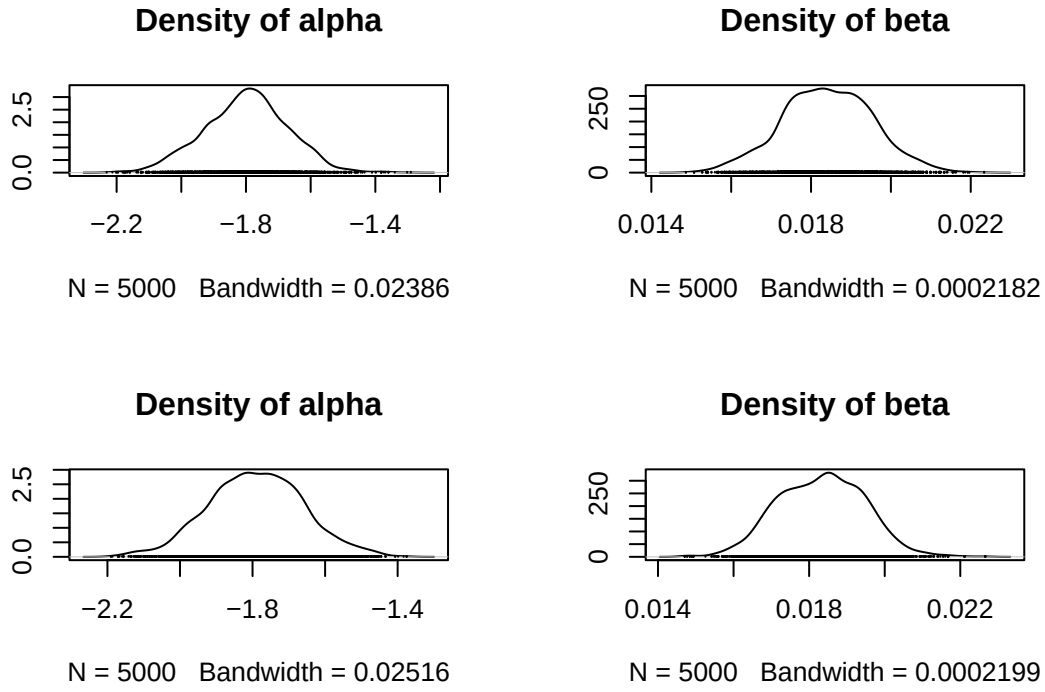

Figure 5: Density plots of α and β with normal distribution (top) and t distribution (bottom).

The density plots with a normal distributed prior and a t distributed prior, as presented in Figure 5. The plots for both α and β display smooth distributions with similar mean values, suggesting that both priors lead to similar approximations of the true posterior distribution.

Based on the Bayesian posterior measures of α and β (Table 1), it can be concluded that the probability of malformations is ?????0.143????? in the absence of any administered dose. As the dose increases, the probability of malformations also increases, indicating a positive dose-response relationship. Additionally, the 95% highest

Table 2: Posterior measures of BMD.

BMD (Mean)	HPD interval	Prior
Normal distribution	17.191	[15.25, 19.153]
t-distribution	17.186	[15.279, 19.26]

posterior density (HPD) interval, which captures the 95% most plausible parameter values, does not include the value of 0, providing evidence for the existence of a dose effect.

(3) Plot the posterior dose-response relationship together with the observed probabilities of a malformation per dose.

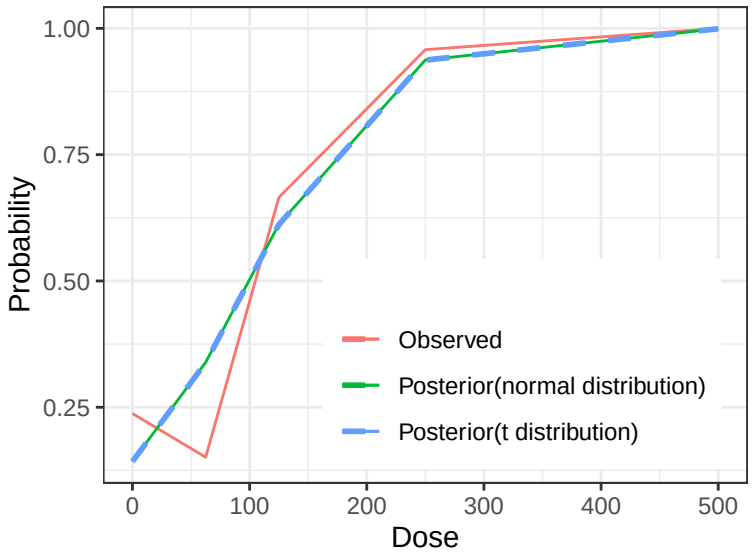

Figure 6: Posterior dose-response relationship and observed probabilities.

Figure 6 indicates that the trends in the posteriors exhibit a dose-response relationship that is similar to the observed probabilities, with the exception of the dose range of 50-70. Moreover, the posterior with t-distributed priors closely aligns with the posterior with normal distributed priors. In general, all three lines demonstrate an increasing probability of malformations as the dose increases, with this increase leveling off as the dose approaches approximately 400.

(4) A safe level of exposure can be defined as a dose corresponding to a very small increase in excess risk of q, e.g. q = 0.05. This is called the Benchmark dose (BMD) d^* and can be obtained by solving the equation

$$r(d^*) = \frac{P(d^*) - P(0)}{1 - P(0)} = q$$

with $P(d)$ the probability of an adverse effect at dose level d. For a logistic regression with a linear dose model, the BMD is given by

$$BMD = \frac{logit(q^*) - \alpha}{\beta}$$

with $q^* = q(1 - P(0)) + P(0)$. Determine the posterior estimate of the safe level of exposure for DYME corresponding with an excess risk of q = 0.05.

The posterior mean values for BMD and the corresponding HPD interval are shown in Table 2. The lower bounds

of the HPD interval for the prior normal distribution (19.153 for Normal distribution and 19.26 for t-distribution) should be considered as the Benchmark does.

(5) As an alternative, a safe level of exposure can be obtained from a threshold model, defined as

$$y \sim binomial(N, \pi)$$

$$logit(\pi) = \alpha + \beta(d - \tau)I(d > \tau)$$

, with τ the threshold dose below which there is no excess risk. Write code for this model, and summarize the results. How do these results compare with previous results?

In this threshold model, we essentially fit a piecewise linear regression for $logit(\pi)$:

- if the dose is smaller than τ, $I(d < \tau) = 0$ meaning the intercept is α and the slope is zero.
- if the dose is larger than τ, $I(d < \tau) = 1$ meaning the intercept is $\alpha - \beta\tau$ and the slope is β

We will use the same vague priors from a Normal distribution:

$$\alpha \sim N(0, 10000)$$

$$\beta \sim N(0, 10000)$$

All other tuning parameter such as initial values and burn-in are chosen the same as in the previous model. We will test several models with $\tau \in \{0, 62.5, 125, 250, 500\}$. We will select the model that minimises the WAIC.

Figure 7 shows that the lowest WAIC is reached when $\tau = 62.5$. This means that, for doses ≤ 62.5, $logit(\pi) = \alpha$, independent of the dose. For doses > 62.5, $logit(\pi) = \alpha + \beta(d - 62.5)$. Figure 8 indeed shows that the fit of the posterior dose-response relationship with the observed probabilities improved, especially in the range with lower doses.

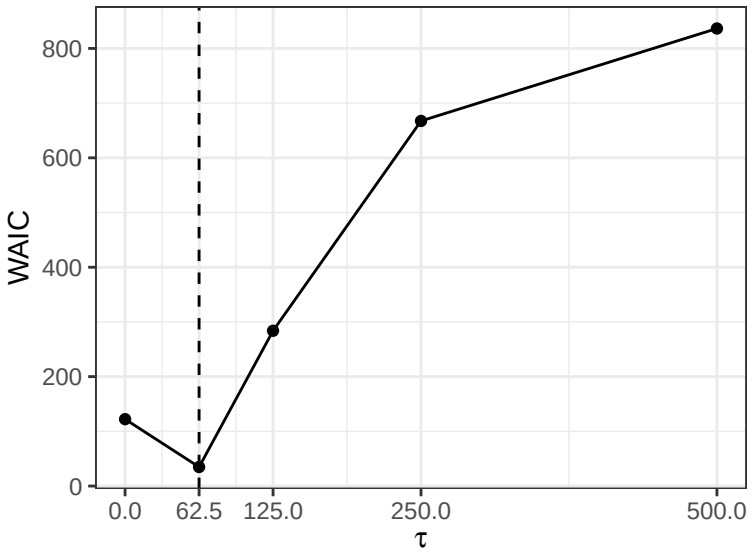

Figure 7: WAIC for each value of τ.

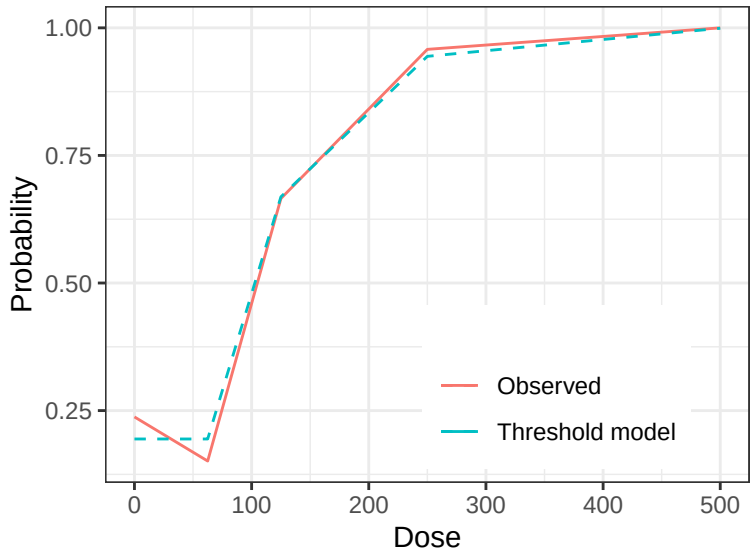

Figure 8: Posterior dose-response relationship and observed probabilities for the model with the lowest WAIC.

A Appendix

A.1 All code for the report

```
#---------------------------load all packages----------------------------------#
library(tidyverse)
library(rprojroot)
library(kableExtra)
library('nimble')
library(coda)
library(cowplot)
library(latex2exp)

knitr::opts_chunk$set(echo = TRUE)
knitr::opts_knit$set(
  root.dir = find_root(criterion = has_file("BayesianInference.Rproj")))
#-----------------------------setup task B-----------------------------------#
n <- 5
# Dose level
dose <- c(0,62.5,125,250,500)
# Number of fetus
N <- c(282,225,290,261,141)
# Number of malformations
y <- c(67,34,193,250,141)
#-------------------------------Question B1----------------------------------#
init1 <- list(alpha = 0, beta = 0)
init2 <- list(alpha = -0.5, beta = 0.1)
initial.values <- list(init1, init2)

# MCMC settings
n.iter <- 10000  # iterations
n.burnin <- 5000 # burn-in
n.chains <- 2    # chains

# Model settings
model.data <- list('dose' = dose, 'N' = N,'y' = y)
model.constant <- list('n' = n)

# Model 1
# A prior with Normal Distribution
model_1 <- nimbleCode({
  # Specify a vague prior with normal distribution
  alpha ~ dnorm(0, sd = 10000)
  beta ~ dnorm(0, sd = 10000)
  # likelihood
  for (i in 1:n) {
    logit(p[i]) <-  alpha + beta * dose[i]
    y[i] ~ dbin(p[i],N[i])
  }})

# Output of Model 1
mcmc.output1 <- nimbleMCMC(code = model_1,
                           data = model.data,
                           constants = model.constant,
```

```
                          inits = initial.values,
                          niter = n.iter,
                          nburnin = n.burnin,
                          summary = TRUE,
                          nchains = n.chains
)
# Trace plots
pdf("images/trace_normal.pdf")
par(mfrow = c(2,2))
traceplot(as.mcmc(mcmc.output1$samples$chain1), xlab = "chain1$iteration")
traceplot(as.mcmc(mcmc.output1$samples$chain2), xlab = "chain2$iteration")
dev.off()
# Gelman-Rubin diagnostic plots
combinedchains1 = mcmc.list(as.mcmc(mcmc.output1$samples$chain1), as.mcmc(mcmc.output1$samples$chain2))
gelman.diag(combinedchains1)
pdf("images/gelman_normal.pdf")
gelman.plot(combinedchains1,xlim = c(0,5000))
dev.off()
# Autocorrelation plots
par(mfrow = c(2, 2))
#densplot(as.mcmc(mcmc.output1$samples$chain1), main = "Chain1 Density of alpha")
#densplot(as.mcmc(mcmc.output1$samples$chain2), main = "Chain2 Density of alpha")
pdf("images/ac1_normal.pdf")
acf(as.mcmc(mcmc.output1$samples$chain1), xlab = 'Chain1 Lag')
dev.off()
pdf("images/ac2_normal.pdf")
acf(as.mcmc(mcmc.output1$samples$chain2), xlab = 'Chain2 Lag')
dev.off()
# Model 2
# A prior with t Distribution
model_2 <- nimbleCode({
  # Specify a vague prior with t distribution
  alpha ~ dt(0, 0.0001, 5)
  beta ~ dt(0, 0.0001, 5)
  # likelihood
  for (i in 1:n) {
    logit(p[i]) <-  alpha + beta * dose[i]
    y[i] ~ dbin(p[i],N[i])
  }})

# Output of Model 1
mcmc.output2 <- nimbleMCMC(code = model_2,
                           data = model.data,
                           constants = model.constant,
                           inits = initial.values,
                           niter = n.iter,
                           nburnin = n.burnin,
                           summary = TRUE,
                           nchains = n.chains
)

# Trace plots
pdf("images/trace_t.pdf")
par(mfrow = c(2,2))
```

```r
traceplot(as.mcmc(mcmc.output2$samples$chain1), xlab = "chain1$iteration")
traceplot(as.mcmc(mcmc.output2$samples$chain2), xlab = "chain2$iteration")
dev.off()
# Autocorrelation plots
par(mfrow = c(2,2))
densplot(as.mcmc(mcmc.output2$samples$chain1), main = "Chain1 Density of alpha")
densplot(as.mcmc(mcmc.output2$samples$chain2), main = "Chain2 Density of alpha")

pdf("images/ac1_t.pdf")
acf(as.mcmc(mcmc.output2$samples$chain1), xlab = 'Chain1 Lag')
dev.off()
pdf("images/ac2_t.pdf")
acf(as.mcmc(mcmc.output2$samples$chain2), xlab = 'Chain2 Lag')
dev.off()
# Gelman-Rubin diagnostic plots
combinedchains2 = mcmc.list(as.mcmc(mcmc.output2$samples$chain1), as.mcmc(mcmc.output2$samples$chain2))
gelman.diag(combinedchains2)
pdf("images/gelman_t.pdf")
gelman.plot(combinedchains2,xlim = c(0,5000))
dev.off()
#-------------------------------Question B2-----------------------------------#
par(mfrow = c(2,2))
densplot(as.mcmc(mcmc.output1$samples$chain1,mcmc.output1$samples$chain2))
densplot(as.mcmc(mcmc.output2$samples$chain1,mcmc.output2$samples$chain2))
# Normal distribution
#mcmc.output1$summary
samples_n <- rbind(mcmc.output1$samples$chain1,mcmc.output1$samples$chain2)
HPD1 <- as.data.frame(round(HPDinterval(as.mcmc(samples_n)),3)) %>%
  mutate(interval = paste0("[", lower, ", ", upper, "]"))

# t distribution
#mcmc.output2$summary
samples_t <- rbind(mcmc.output2$samples$chain1,mcmc.output2$samples$chain2)
HPD2 <- as.data.frame(round(HPDinterval(as.mcmc(samples_t)),3)) %>%
  mutate(interval = paste0("[", lower, ", ", upper, "]"))

as.data.frame(rbind(mcmc.output1$summary$all.chains[, -c(4, 5)],
                    mcmc.output2$summary$all.chains[, -c(4, 5)])) %>%
  mutate(HPD = c(HPD1$interval, HPD2$interval)) %>%
  kable(booktabs = TRUE,
        caption = "Bayesian posterior measures of  and ",
        col.names = c("Mean", "Median", "St.Dev", "HPD Interval")) %>%
  kableExtra::group_rows(group_label = "N(0, 10000)",
                         start_row = 1, end_row = 2) %>%
  kableExtra::group_rows(group_label = "t(0, 0.0001, 5)",
                         start_row = 3, end_row = 4)
#-------------------------------Question B3-----------------------------------#
# posterior with normal distribution
chains_output1 <- data.frame(mcmc.output1[[1]])
chain1_output1 <- chains_output1[, 1:2] %>%
  rename("alpha" = "chain1.alpha", "beta" = "chain1.beta")
chain2_output1 <- chains_output1[, 3:4] %>%
  rename("alpha" = "chain2.alpha", "beta" = "chain2.beta")
df_output1 <- rbind(chain1_output1, chain2_output1)
```

```r
# get the mean value of alpha and beta
alpha_n <- round(mean(df_output1$alpha), 3)
beta_n <- round(mean(df_output1$beta), 3)

# posterior with t distribution
chains_output2 <- data.frame(mcmc.output2[[1]])
chain1_output2 <- chains_output2[, 1:2] %>%
  rename("alpha" = "chain1.alpha", "beta" = "chain1.beta")
chain2_output2 <- chains_output2[,3:4] %>%
  rename("alpha" = "chain2.alpha", "beta" = "chain2.beta")
df_output2 <- rbind(chain1_output2, chain2_output2)
# get the mean value of alpha and beta
alpha_t <- round(mean(df_output2$alpha), 3)
beta_t <- round(mean(df_output2$beta), 3)

# Create a dataframe for plotting
df_plotting <- data.frame(cbind(dose,N,y)) %>%
  mutate(prob_observed = y/N,# Observed
         prob_n = expit(alpha_n + beta_n * dose),# Posterior with normal distr
         prob_t = expit(alpha_t + beta_t * dose)# Posterior with t distr
         )

ggplot(df_plotting, aes(x = dose)) +
  geom_line(aes(y = prob_observed, color = "Observed")) +
  geom_line(aes(y = prob_n, color = "Posterior(normal distribution)")) +
  geom_line(aes(y = prob_t, color = "Posterior(t distribution)"),
            linetype = "dashed", linewidth = 1) +
  labs(x = "Dose", y = "Probability", color = "") +
  theme_bw() +
  theme(legend.position = c(0.65, 0.25))
#--------------------------------Question B4-----------------------------------#
# Excess risk q=0.05
# prior Normal distribution
df_output1_bmd <- df_output1 %>%
  # Caculate BMD
  mutate(P0 = exp(alpha) / (1 + exp(alpha))) %>%
  mutate(q.star = (0.05 * (1 - P0)) + P0) %>%
  mutate(bmd = (logit(q.star) - alpha) / beta)

# HPD Interval
hpd_n <- as.data.frame(round(HPDinterval(as.mcmc(df_output1_bmd)), 3)) %>%
  mutate(interval = paste0("[", lower, ", ", upper, "]"))

# prior t distribution
df_output2_bmd <- df_output2 %>%
  # Caculate BMD
  mutate(P0 = exp(alpha) / (1 + exp(alpha))) %>%
  mutate(q.star = (0.05 * (1 - P0)) + P0) %>%
  mutate(bmd = (logit(q.star) - alpha) / beta)
# HPD Interval
hpd_t <- as.data.frame(round(HPDinterval(as.mcmc(df_output2_bmd)), 3)) %>%
  mutate(interval = paste0("[", lower, ", ", upper, "]"))

data.frame(Prior = c("Normal distribution", "t-distribution"),
```

```
          mean = c(round(mean(df_output1_bmd$bmd), 3),
                   round(mean(df_output2_bmd$bmd), 3)),
          hpd = c(hpd_n[5, "interval"], hpd_t[5, "interval"])) %>%
  kable(booktabs = TRUE,
        col.names = c("BMD (Mean)", "HPD interval", "Prior"),
        caption = "Posterior measures of BMD.") %>%
  kableExtra::kable_styling()
#--------------------------Question B---------------------------------------#
#possibly interesting reference for piecewise linear regression in R with nimble
#https://gkonstantinoudis.github.io/nimble/PiecewiseLinear.html
init1 <- list(alpha = 0, beta = 0)
init2 <- list(alpha = -0.5, beta = 0.1)
initial.values <- list(init1, init2)

# MCMC settings
n.iter <- 10000  # iterations
n.burnin <- 5000 # burn-in
n.chains <- 2    # chains


testmodel <- function(tau = 0){
  # Model settings
  indicator <- 1*(dose > tau)
  model.data <- list('dose' = dose, 'N' = N,'y' = y, 'indicator' = indicator)
  model.constant <- list('n' = n)

  # Model 1
  # A prior with Normal Distribution
  model <- nimbleCode({
    # Specify a vague prior with normal distribution
    alpha ~ dnorm(0, sd = 10000)
    beta ~ dnorm(0, sd = 10000)
    # likelihood
    for (i in 1:n) {
      logit(p[i]) <-  alpha + beta * dose[i] * indicator[i]
      y[i] ~ dbin(p[i],N[i])
    }})

  # Output of Model 1
  mcmc.output <- nimbleMCMC(code = model,
                            data = model.data,
                            constants = model.constant,
                            inits = initial.values,
                            niter = n.iter,
                            nburnin = n.burnin,
                            summary = TRUE,
                            nchains = n.chains,
                            WAIC = TRUE
  )
  return(mcmc.output)
}
tau_values <- c(0,62.5,125,250,500)
results <- map(tau_values, testmodel)
save(results, file = "output/results_threshold_model.Rdata")
```

```r
waic <- sapply(X = seq(length(tau_values)),
      FUN = function(x) {results[[x]]$WAIC$WAIC})
#plot the relationship between WAIC and tau
data.frame(waic = waic,
           tau = tau_values) %>%
  ggplot(aes(x = tau, y = waic)) +
  geom_point() +
  geom_line() +
  theme_bw() +
  ylab("WAIC") + xlab(TeX("$\\tau$")) +
  geom_vline(aes(xintercept = tau_values[which(waic == min(waic))]),
             linetype = "dashed") +
  scale_x_continuous(breaks = tau_values)
#plot posterior vs observed probabilities
chosen_model <- results[[which(waic == min(waic))]]
chains_output1 <- data.frame(chosen_model[[1]])
chain1_output1 <- chains_output1[, 1:2] %>%
  rename("alpha" = "chain1.alpha", "beta" = "chain1.beta")
chain2_output1 <- chains_output1[, 3:4] %>%
  rename("alpha" = "chain2.alpha", "beta" = "chain2.beta")
df_output1 <- rbind(chain1_output1, chain2_output1)
# get the mean value of alpha and beta
alpha_n <- round(mean(df_output1$alpha), 3)
beta_n <- round(mean(df_output1$beta), 3)

# Create a dataframe for plotting
df_plotting <- data.frame(cbind(dose, N, y)) %>%
  mutate(indicator = 1*(dose > tau_values[which(waic == min(waic))]),
         prob_observed = y/N,
         threshold_model = expit(alpha_n + beta_n * dose * indicator)
         )
ggplot(df_plotting, aes(x = dose)) +
  geom_line(aes(y = prob_observed, color = "Observed")) +
  geom_line(aes(y = threshold_model, color = "Threshold model"), linetype = "dashed") +
  theme_bw() +
  theme(legend.position = c(0.7, 0.2)) +
  scale_color_discrete("") +
  ylab("Probability") +
  xlab("Dose")
```