

Assignment 2

Zhengyang Fei

Problem 1

Part a

```
df <- read.csv("brnbdy.csv", header = TRUE)

logBrain <- log(df$brain)
logBody <- log(df$body)
n <- length(logBrain)

# prior
alpha_mean0 <- 0
alpha_prec0 <- 0.0001 # precision for alpha
beta_mean0 <- 0
beta_prec0 <- 0.0001 # precision for beta
tau_a0 <- 0.0001 # a tau
tau_b0 <- 0.0001 # b tau

# setup
nIter <- 20000
alphaSamples <- rep(0,nIter)
betaSamples <- rep(0,nIter)
tauSamples <- rep(0,nIter)

# initialize parameters ()
set.seed(2025)
alphaCur <- 0
betaCur <- 0
tauCur <- 1
```

```

# Gibb sampler loop
for (iter in 1:nIter){
  # Sample alpha | beta, tau, data
  alphaVar_post <- 1 / (tauCur * n + alpha_prec0)
  alphaMean_post <- alphaVar_post * (tauCur * sum(logBrain - betaCur * logBody) +
                                     alpha_prec0 * alpha_mean0)

  alphaCur <- rnorm(1, mean = alphaMean_post, sd = sqrt(alphaVar_post))

  # Sample beta | alpha, tau, data
  sum_x2 <- sum(logBody^2)
  betaVar_post <- 1 / (tauCur * sum_x2 + beta_prec0)
  betaMean_post <- betaVar_post * (tauCur * sum(logBody * (logBrain - alphaCur)) +
                                   beta_prec0 * beta_mean0)
  betaCur <- rnorm(1, mean = betaMean_post, sd = sqrt(betaVar_post))

  # Sample tau | alpha, beta, data
  resid <- logBrain - alphaCur - betaCur * logBody
  shape_post <- tau_a0 + n / 2
  rate_post <- tau_b0 + 0.5 * sum(resid^2)
  tauCur <- rgamma(1, shape = shape_post, rate = rate_post)

  # Store samples
  alphaSamples[iter] <- alphaCur
  betaSamples[iter] <- betaCur
  tauSamples[iter] <- tauCur
}

# df <- data.frame(a = alphaSamples, b = betaSamples, t = tauSamples)
# head(df)

```

Part b

```

# --- burnins---
burnin <- 0
alphaSample_post <- alphaSamples[(burnin + 1):nIter]
betaSamples_post <- betaSamples[(burnin + 1):nIter]
tauSamples_post <- tauSamples[(burnin + 1):nIter]

# --- alpha ---

```

```
alpha_mean <- mean(alphaSample_post)
alpha_sd    <- sd(alphaSample_post)
alpha_ci    <- quantile(alphaSample_post, probs = c(0.025, 0.975))

cat("\nAlpha:\n")
```

Alpha:

```
cat("  Mean =", alpha_mean,
    " SD =", alpha_sd,
    " 95% CI =", alpha_ci, "\n")
```

```
Mean = 2.13663 SD = 0.09839296 95% CI = 1.941953 2.329875
```

```
# --- beta ---
beta_mean <- mean(betaSamples_post)
beta_sd    <- sd(betaSamples_post)
beta_ci    <- quantile(betaSamples_post, probs = c(0.025, 0.975))

cat("\nBeta:\n")
```

Beta:

```
cat("  Mean =", beta_mean,
    " SD =", beta_sd,
    " 95% CI =", beta_ci, "\n")
```

```
Mean = 0.751375 SD = 0.02906354 95% CI = 0.6933398 0.8081814
```

```
# --- tau ---
tau_mean <- mean(tauSamples_post)
tau_sd    <- sd(tauSamples_post)
tau_ci    <- quantile(tauSamples_post, probs = c(0.025, 0.975))

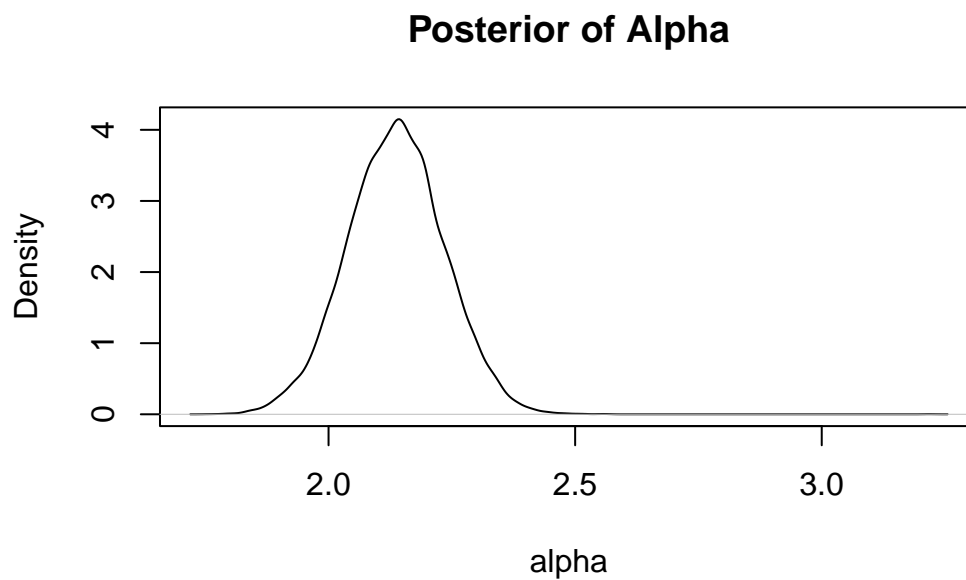
cat("\nTau:\n")
```

Tau:

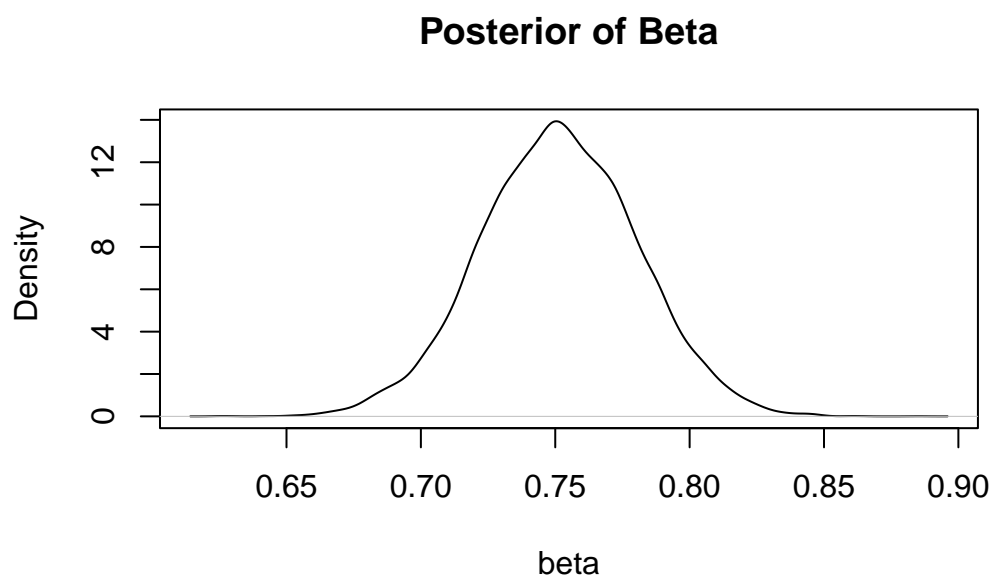
```
cat("  Mean =", tau_mean,  
    "  SD =", tau_sd,  
    " 95% CI =", tau_ci, "\n")
```

Mean = 2.07538 SD = 0.3762885 95% CI = 1.396224 2.871346

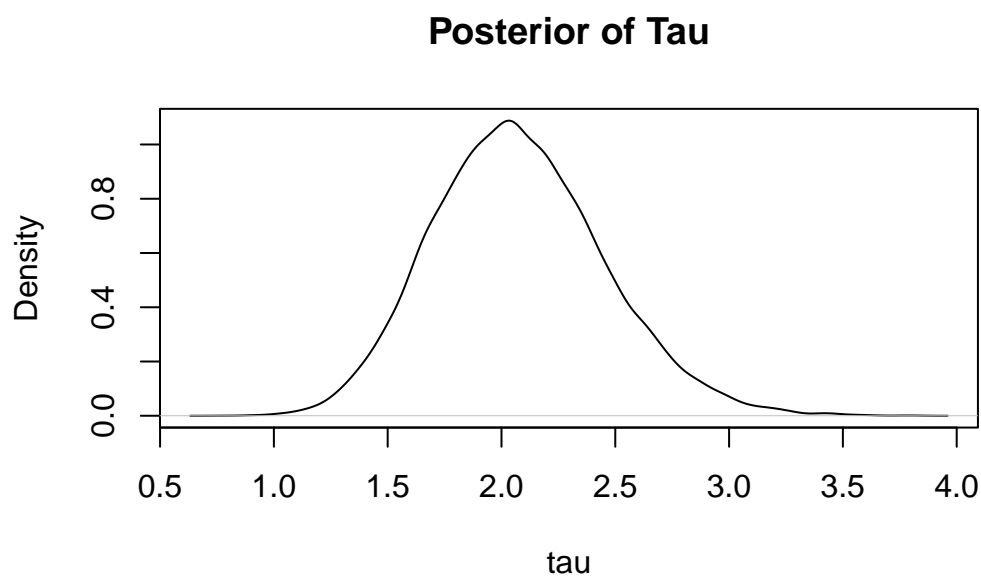
```
# Density plots  
plot(density(alphaSamples), main = "Posterior of Alpha", xlab = "alpha")
```



```
plot(density(betaSamples), main = "Posterior of Beta", xlab = "beta")
```



```
plot(density(tauSamples), main = "Posterior of Tau", xlab = "tau")
```



After testing various burn-in lengths, I found the posterior distribution stabilizes quickly. As a result, I opted to use a relatively small burn-in value.

Part c

```
body_value <- 55
log_body_value <- log(body_value)

logBrainHat <- alphaSamples + betaSamples * log_body_value
brainHat <- exp(logBrainHat)

# summaries
brainHat_mean <- mean(brainHat)
brainHat_sd <- sd(brainHat)
brainHat_ci <- quantile(brainHat, probs = c(0.025, 0.975))

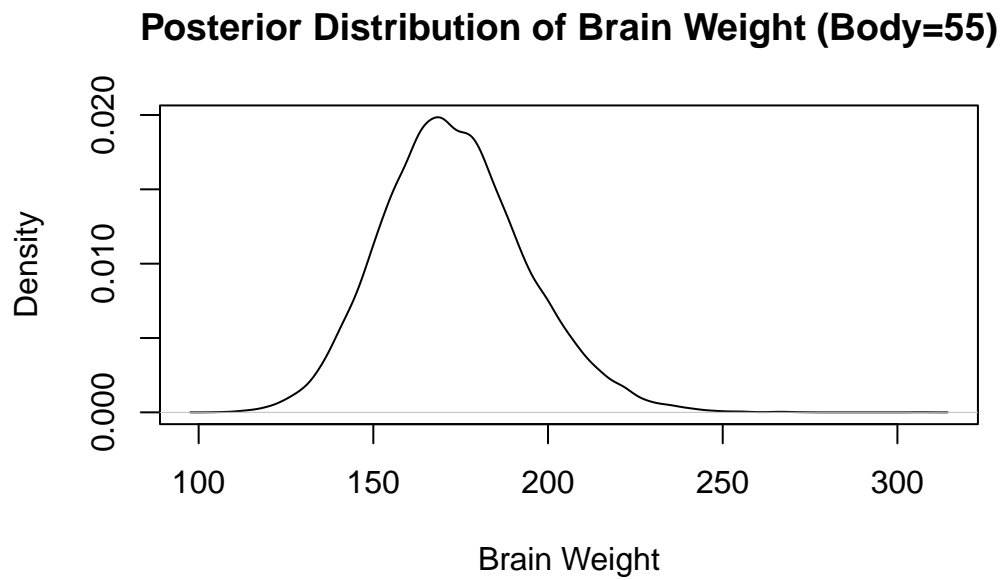
cat("\nEstimated Brain Weight Distribution at Body=55:\n")
```

Estimated Brain Weight Distribution at Body=55:

```
cat("  Mean =", brainHat_mean,
    " SD =", brainHat_sd,
    " 95% CI =", brainHat_ci, "\n")
```

Mean = 173.2493 SD = 20.6927 95% CI = 135.9696 217.0007

```
plot(density(brainHat),
     main = "Posterior Distribution of Brain Weight (Body=55)",
     xlab = "Brain Weight",
     ylab = "Density")
```



Problem 2

Part a

```
library(R2jags)
```

Warning: package 'R2jags' was built under R version 4.4.2

Loading required package: rjags

Warning: package 'rjags' was built under R version 4.4.2

Loading required package: coda

Warning: package 'coda' was built under R version 4.4.2

Linked to JAGS 4.3.1

Loaded modules: basemod,bugs

Attaching package: 'R2jags'

The following object is masked from 'package:coda':

traceplot

```
library(ggplot2)
library(coda)

# Load the dataset
smokeAgeDeath <- read.csv("SmokeAgeDeath.csv")

model.jags<-"model{
  for(i in (1:N)){
    d[i]~dpois(mu[i])
    log(mu[i])<- beta0+beta.s[smoke[i]]+beta.a[age[i]]+log(y[i])+b[i]
    b[i]~dnorm(0,tau)
    b.adj[i]<-b[i]-mean(b[])
  }

  # log(1/114616)=-11.64934->1/12/12=0.006944444
  beta0~dnorm(0, 0.006944444)
  beta0.adj <- beta0+mean(b[])+mean(beta.s[])+mean(beta.a[])
  for(is in 1:n_smoke){
    beta.s[is]~dnorm(0,tau.s)
    beta.s.adj[is]<-beta.s[is]-mean(beta.s[])
  }

  for(ia in 1:n_age){
    beta.a[ia]~dnorm(0,tau.a)
    beta.a.adj[ia]<-beta.a[ia]-mean(beta.a[])
  }

  std.s~dunif(0, 5)
  tau.s<- 1/std.s/std.s
  std.a~dunif(0,5)
  tau.a<- 1/std.a/std.a
  std~dunif(0,9)
  tau<- 1/std/std
```



```

}"

set.seed(2025)

#Prepare data
data_jags<-with(smokeAgeDeath,list(
  d=death,
  age= age,
  smoke=smoke,
  y=pyears,
  N=length(death),
  n_smoke= length(unique(smokeAgeDeath$smoke)),
  n_age=length(unique(smokeAgeDeath$age))
))

model.pois<-jags.model(
  textConnection(model.jags),
  data = data_jags,
  n.chains= 3,
  n.adapt= 500)

```

Compiling model graph
 Resolving undeclared variables
 Allocating nodes
 Graph information:
 Observed stochastic nodes: 20
 Unobserved stochastic nodes: 33
 Total graph size: 223

Initializing model

```

#burn-in 2000iterations
update(model.pois, iter=2000)

samples.post.q2<-coda.samples(model.pois,
  variable.names= c("beta0.adj","beta.s.adj",
  "beta.a.adj", "tau.s","tau.a", "tau"),
  n.iter=10000,
  thin= 10)

samples.q2<-as.matrix(samples.post.q2)

```

```

df <- data.frame(
  beta1_age =samples.q2[, "beta.a.adj[1]"],
  beta2_age =samples.q2[, "beta.a.adj[2]"],
  beta3_age =samples.q2[, "beta.a.adj[3]"],
  beta4_age =samples.q2[, "beta.a.adj[4]"],
  beta5_age =samples.q2[, "beta.a.adj[5]"],
  beta1_smoke= samples.q2[, "beta.s.adj[1]"],
  beta2_smoke= samples.q2[, "beta.s.adj[2]"],
  beta3_smoke= samples.q2[, "beta.s.adj[3]"],
  beta4_smoke= samples.q2[, "beta.s.adj[4]"],
  beta0=samples.q2[, "beta0.adj"],
  tau_age= samples.q2[, "tau.a"],
  tau_smoke =samples.q2[, "tau.s"],
  tau= samples.q2[, "tau"]
)

#compute statistics
summary_stats<-apply(df,2, function(x){
  mean_val <-round(mean(x),4)
  sd_val <-round(sd(x),4)
  credible_interval <-round(quantile(x,probs= c(0.025,0.975)),4)
  c(mean= mean_val,sd= sd_val,lower= credible_interval[1],
    upper= credible_interval[2])
})

summary_stats<-t(summary_stats)
print(summary_stats)

```

	mean	sd	lower.2.5%	upper.97.5%
beta1_age	-1.3371	0.1707	-1.6811	-1.0144
beta2_age	-0.7222	0.1449	-1.0138	-0.4511
beta3_age	-0.0214	0.1267	-0.2804	0.2232
beta4_age	0.6785	0.1176	0.4479	0.9073
beta5_age	1.4022	0.0999	1.2045	1.5918
beta1_smoke	-0.4602	0.0909	-0.6384	-0.2733
beta2_smoke	-0.5342	0.1026	-0.7347	-0.3295
beta3_smoke	0.0678	0.1375	-0.2055	0.3334
beta4_smoke	0.9265	0.1173	0.6967	1.1528
beta0	-7.3721	0.0691	-7.5143	-7.2361
tau_age	0.5996	0.5096	0.0526	1.9612
tau_smoke	1.2819	1.4962	0.0501	5.3805
tau	136938.0214	1571241.8521	14.9942	552259.3041

Part b

```
# Define colors for chains
chain_colors <- c("black", "blue", "red")
1.5e+08
```

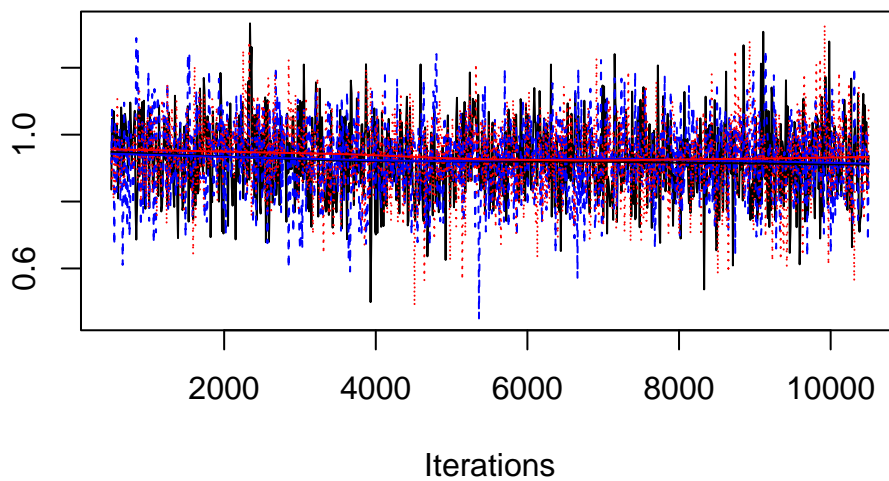
[1] 1.5e+08

```
2.0e+08
```

[1] 2e+08

```
# Convert samples to mcmc.list to handle multiple chains properly
traceplot(mcmc.list(
  as.mcmc(samples.post.q2[[1]][, "beta.s.adj[4]"]),
  as.mcmc(samples.post.q2[[2]][, "beta.s.adj[4]"]),
  as.mcmc(samples.post.q2[[3]][, "beta.s.adj[4]"])
), col = chain_colors, main = "Traceplot of beta4 smoke")
```

Traceplot of beta4 smoke

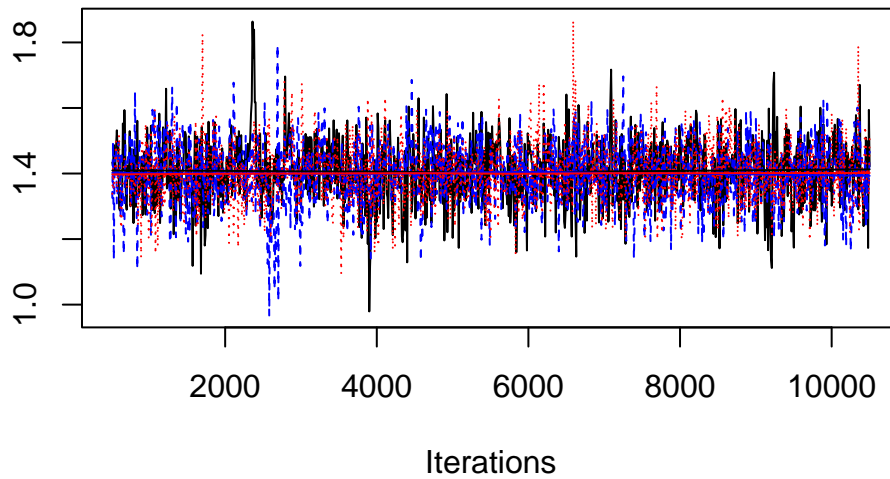


```

traceplot(mcmc.list(
  as.mcmc(samples.post.q2[[1]][, "beta.a.adj[5]"]),
  as.mcmc(samples.post.q2[[2]][, "beta.a.adj[5]"]),
  as.mcmc(samples.post.q2[[3]][, "beta.a.adj[5]"])
), col = chain_colors, main = "Traceplot of beta5 age")

```

Traceplot of beta5 age

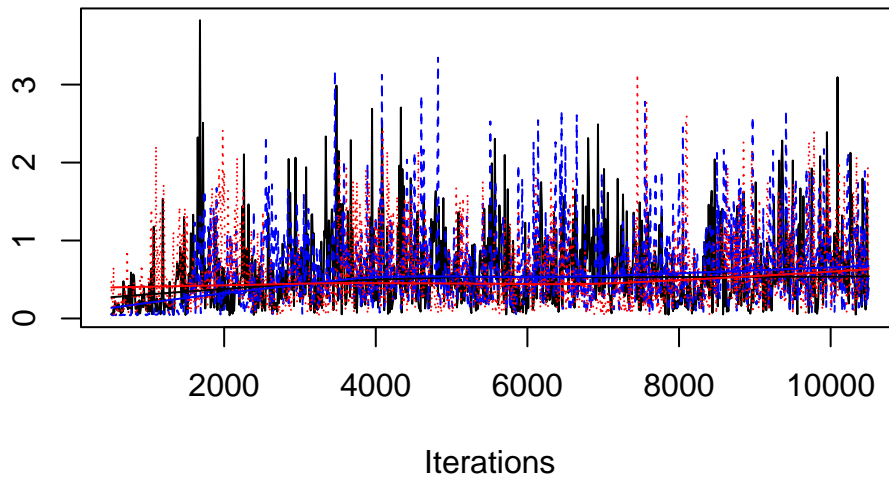


```

traceplot(mcmc.list(
  as.mcmc(samples.post.q2[[1]][, "tau.a"]),
  as.mcmc(samples.post.q2[[2]][, "tau.a"]),
  as.mcmc(samples.post.q2[[3]][, "tau.a"])
), col = chain_colors, main = "Traceplot of tau age")

```

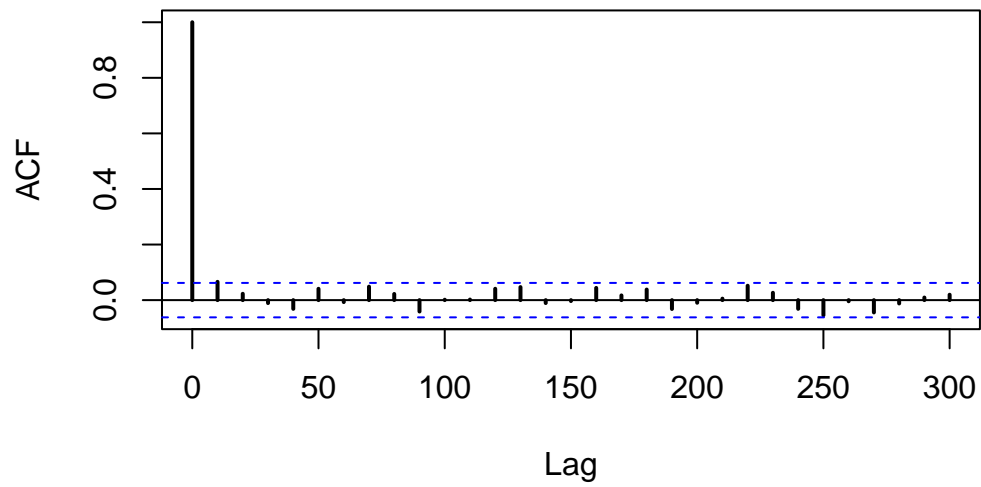
Traceplot of tau age



The trace plot shows: beta4 smoke, beta5 age, and tau smoke. We see a good mixing across the three chains in black, red and blue. This supports convergence, as this demonstrates the chains exhibit stable variability without signs divergence.

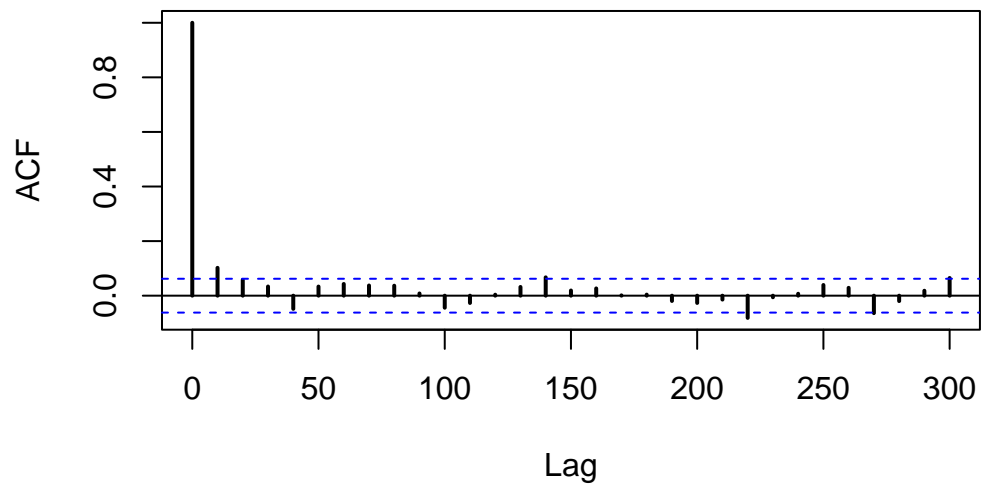
```
# Autocorrelation plots for each parameter
acf(as.mcmc(samples.post.q2[[1]][, "beta.s.adj[4]"]), main = "Autocorrelation of beta4 smoke")
```

Autocorrelation of beta4 smoke

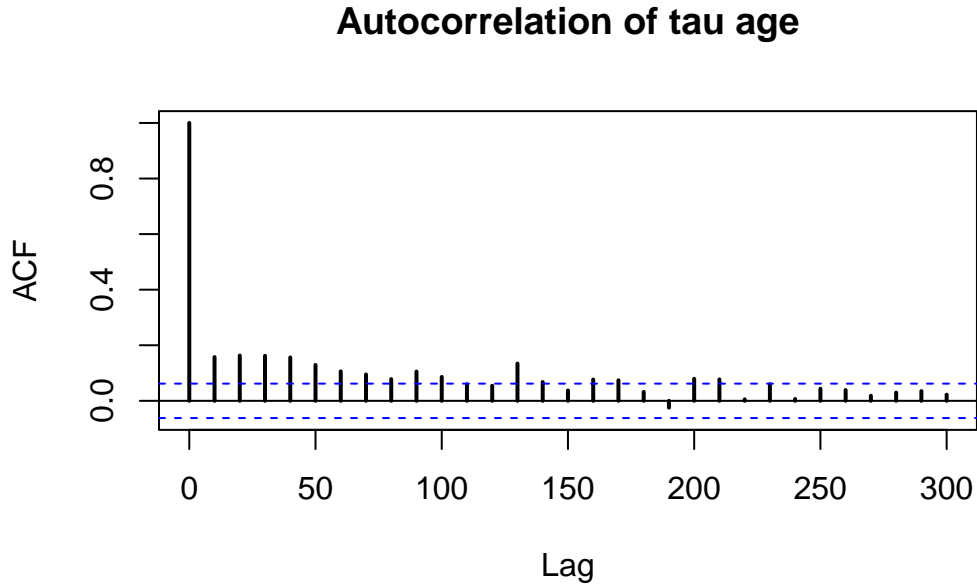


```
acf(as.mcmc(samples.post.q2[[1]][, "beta.a.adj[5]"]), main = "Autocorrelation of beta5 age",
```

Autocorrelation of beta5 age



```
acf(as.mcmc(samples.post.q2[[1]][, "tau.a"]), main = "Autocorrelation of tau age", lwd = 2)
```



The ACF for beta4 smoke and beta5 age shows rapid decline in correlation after a few lags, indicating the samples are independent. For tau age, we can see a slightly higher autocorrelation early on, but decreases over time.

Hence, from the ACF and traceplots, we conclude that the MCMC has converged and that a good amount of burn-in has been applied.

Part c

$$RR = e^{\beta_{\text{smoke}[4]} - \beta_{\text{smoke}[1]}} = e^{0.9228 - (-0.4626)} = e^{1.3854} \approx 3.99$$

$$(e^{0.6980 - (-0.6374)} = e^{1.3354}, e^{1.1453 - (-0.2762)} = e^{1.4215}) = (3.80, 4.14)$$

Hence, heavy smokers (>20 cigarettes per day) have approximately 4 times the risk of death compared to non smokers. The 95% credible interval (3.80, 4.14) means that, there is a 95% probability the true RR lies within this interval.

Problem 3

Part a

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr      1.1.4      v readr      2.1.5
v forcats    1.0.0      v stringr    1.5.1
v lubridate  1.9.3      v tibble     3.2.1
v purrr      1.0.2      v tidyr      1.3.1
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()     masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
library(R2jags)
ratDose.df <- read.csv("ratDoseData.csv")
```

```
model.rats.jags <- "model{
  # Likelihood
  for (i in 1:N) {
    for (j in 1:M) {
      Y[i, j] ~ dnorm(mu[i, j], tau)
      mu[i, j] <- beta0[i] + beta1[i] * j
    }
  }

  # Constraints for reference group
  beta0[N] <- 0
  beta1[N] <- 0

  # Random effects: Prior distributions for regression coefficients
  for (i in 1:(N-1)) {
    beta0[i] ~ dnorm(delta00 + delta01 * dose[i], tau0)
    beta1[i] ~ dnorm(delta10 + delta11 * dose[i], tau1)
  }

  # Priors for hyperparameters
  delta00 ~ dnorm(0, 0.001)
```



```

delta01 ~ dnorm(0, 0.001)
delta10 ~ dnorm(0, 0.001)
delta11 ~ dnorm(0, 0.001)

# Precision parameters
tau ~ dgamma(0.001, 0.001)
tau0 ~ dgamma(0.001, 0.001)
tau1 ~ dgamma(0.001, 0.001)
}"

```

```

#set.seed(2025)
# Prepare data
data_jags <- with(ratDose.df, list(
  dose=dose,
  N=length(rat),
  M=11,
  Y=ratDose.df[,3:13]
))

model.rats <- jags.model(
  textConnection(model.rats.jags),
  data = data_jags,
  n.chains =3,
  n.adapt= 500
)

```

```

Compiling model graph
  Resolving undeclared variables
  Allocating nodes
Graph information:
  Observed stochastic nodes: 550
  Unobserved stochastic nodes: 105
  Total graph size: 1830

```

Initializing model

```

update(model.rats, iter=200)

samples.post.q3<-coda.samples(model.rats,
  variable.names= c("beta0", "beta1","tau", "delta00",
    "delta01","delta10","delta11","tau0","tau1"),

```

```

n.iter=10000,
thin= 10)

samples.q3 <- as.matrix(samples.post.q3)

beta01<-samples.q3[, "beta0[1]"]
beta11<-samples.q3[, "beta1[1]"]
delta00<-samples.q3[, "delta00"]
delta01<-samples.q3[, "delta01"]
delta10<-samples.q3[, "delta10"]
delta11<-samples.q3[, "delta11"]
tau<-samples.q3[, "tau"]
tau0<-samples.q3[, "tau0"]
tau1<-samples.q3[, "tau1"]

df<-data.frame(
  beta01 = beta01,
  beta11 = beta11,
  delta00= delta00,
  delta01= delta01,
  delta10= delta10,
  delta11= delta11,
  sigma2 = 1/tau,
  sigma2.0 = 1/tau0,
  sigma2.1 = 1/tau1
)

# Calculate mean, sd, and 95% credible interval for each column
summary_stats<-apply(df,2, function(x){
  mean_val <-round(mean(x),4)
  sd_val <-round(sd(x),4)
  credible_interval <-round(quantile(x,probs= c(0.025,0.975)),4)

  c(mean= mean_val,sd= sd_val,lower= credible_interval[1],
    upper= credible_interval[2])
})

summary_stats<-t(summary_stats)
print(summary_stats)

```

	mean	sd	lower.2.5%	upper.97.5%
beta01	47.2996	5.0791	37.5077	57.2701

beta11	7.0096	0.4826	5.9752	7.7685
delta00	59.7152	3.0703	54.2421	65.5746
delta01	-0.4956	0.7122	-1.9159	0.8827
delta10	7.0914	0.3451	6.5166	7.6318
delta11	-0.1494	0.0718	-0.2860	0.0062
sigma2	233.3725	16.2176	205.9328	264.6771
sigma2.0	122.7936	34.3918	64.2800	200.5614
sigma2.1	0.1382	0.2430	0.0009	0.8233

Part b

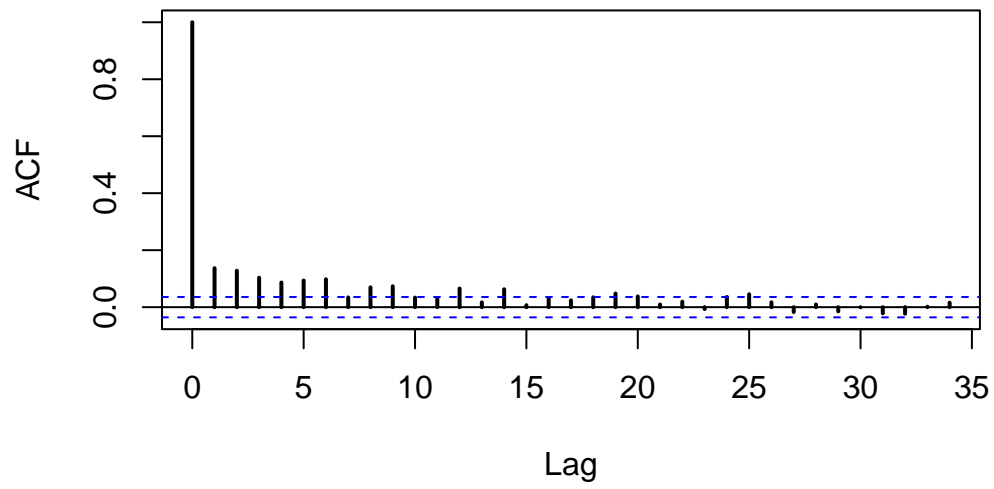
```

params_list <- with(df,list(
  "beta01" =beta01,"beta11" =beta11,"sigma2" =sigma2,
  "delta00" =delta00, "delta01"= delta01,
  "delta10" = delta10, "delta11" = delta11,
  "sigma2.0" = sigma2.0, "sigma2.1" = sigma2.1
))

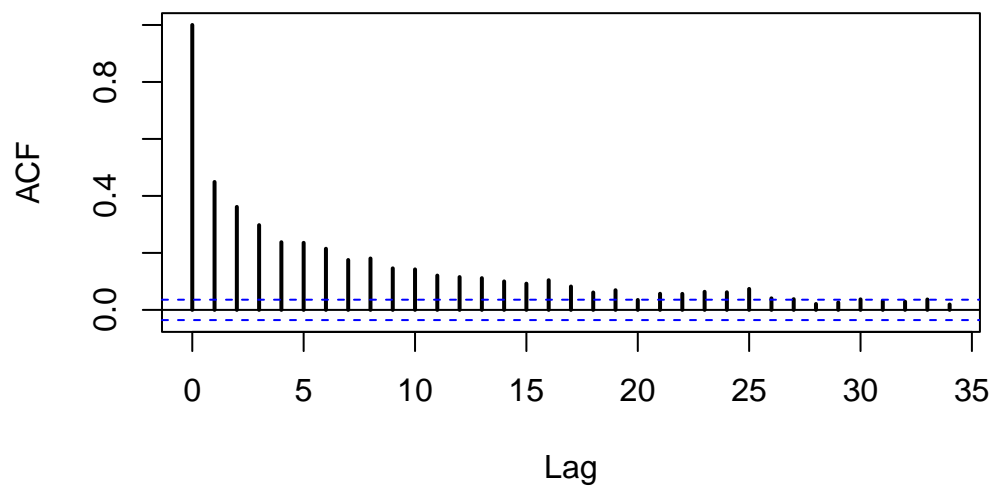
for (p in names(params_list)) {
  acf(as.mcmc(params_list[[p]]),
    main = paste("Autocorrelation of", p),
    lwd = 2)
}

```

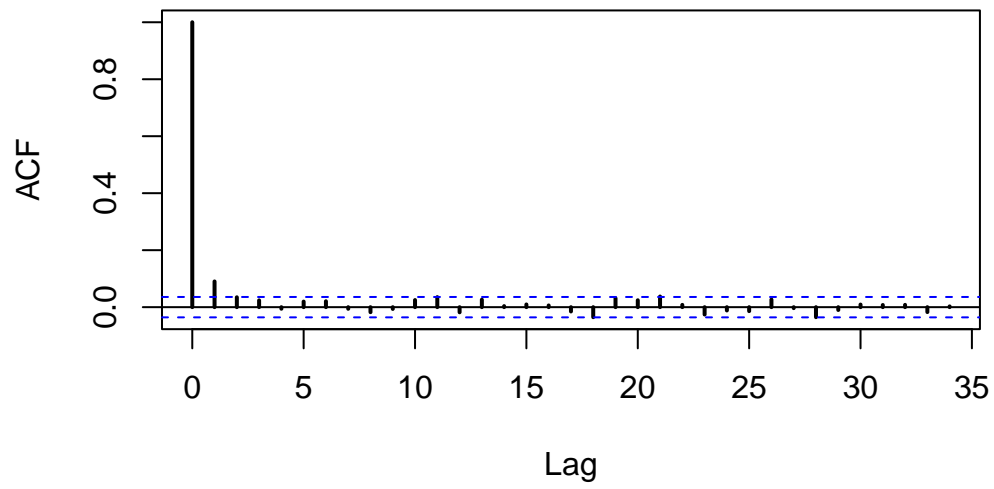
Autocorrelation of beta01



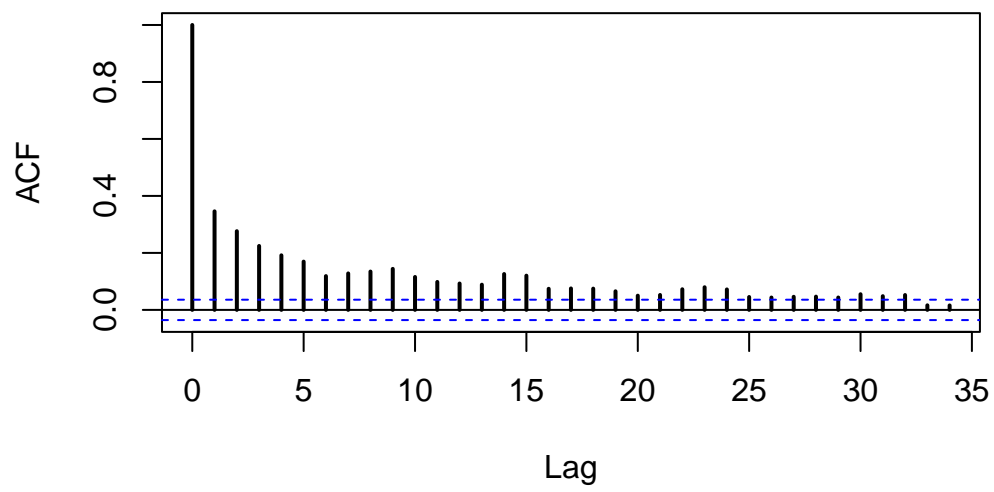
Autocorrelation of beta11



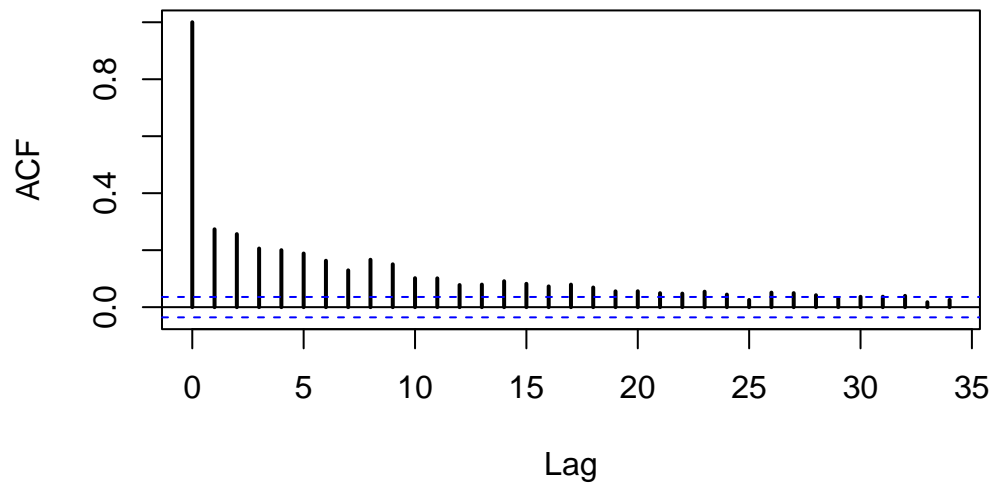
Autocorrelation of sigma2



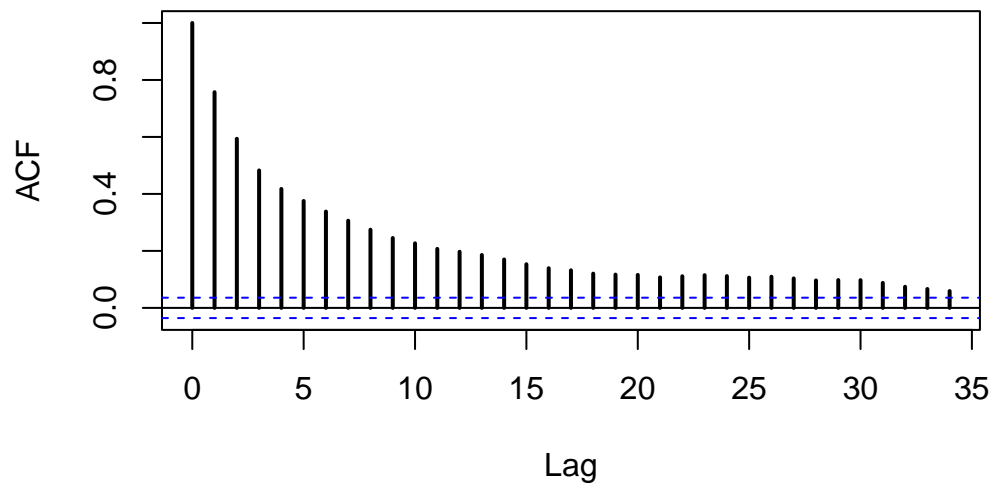
Autocorrelation of delta00



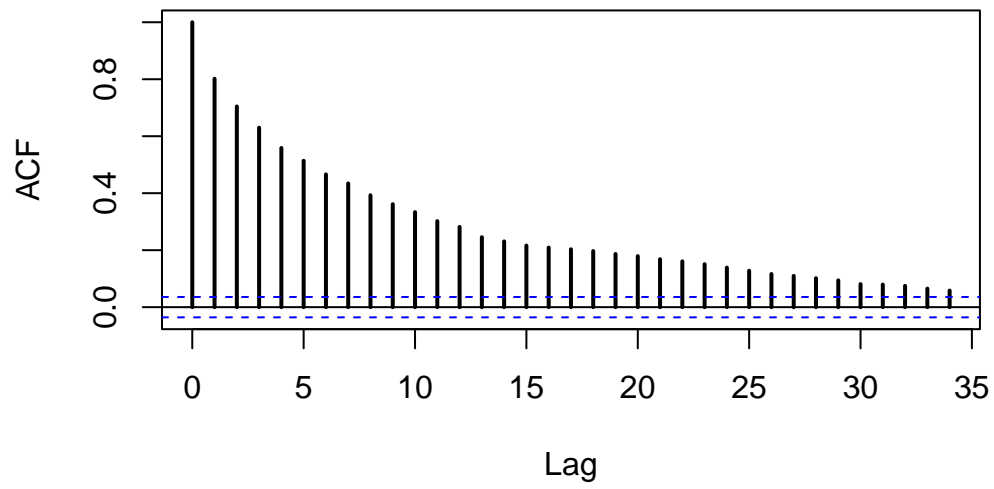
Autocorrelation of delta01



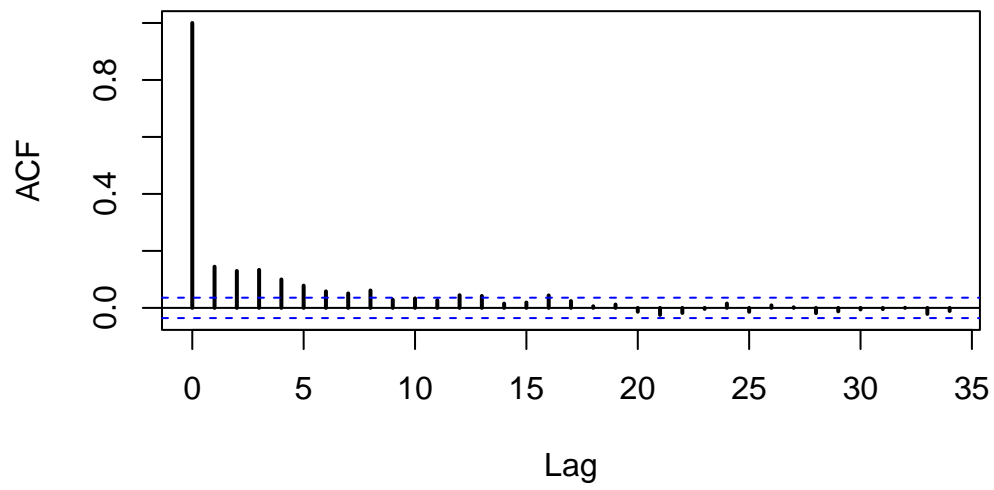
Autocorrelation of delta10



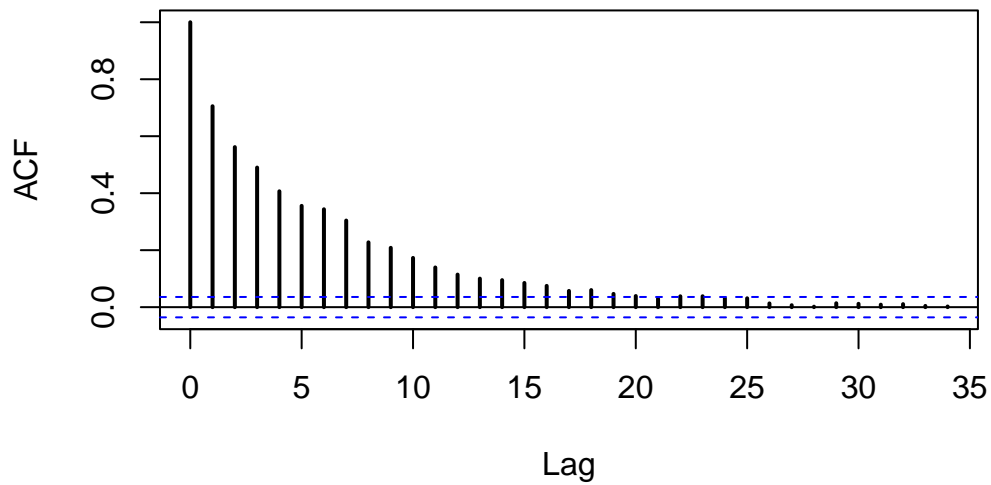
Autocorrelation of delta11



Autocorrelation of sigma2.0



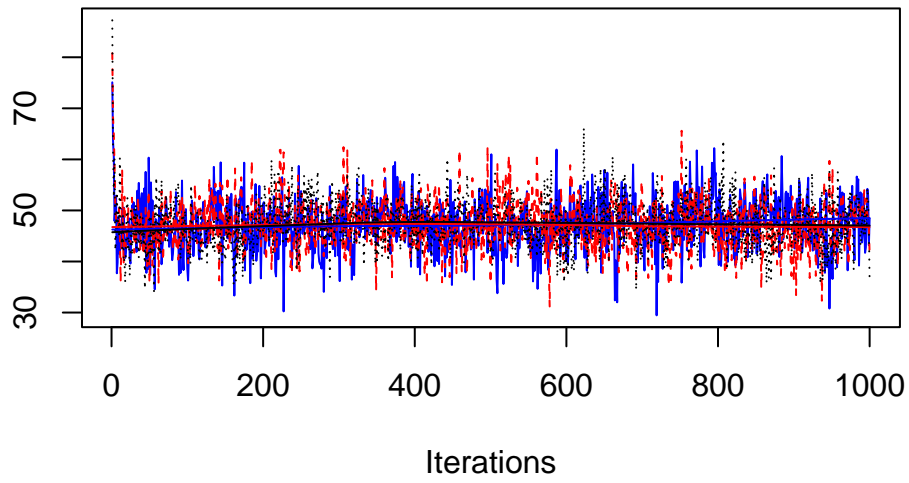
Autocorrelation of sigma2.1



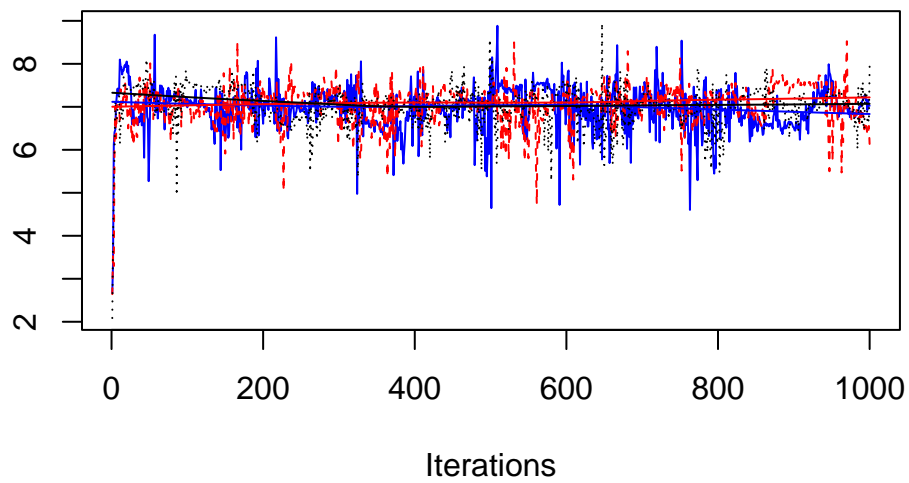
```
chain_colors <- c("blue", "red", "black") # Colors for the three chains
# Split data into three chains
n_samples <- nrow(samples.q3) / 3
df1 <- df[1:n_samples, ] # First chain
df2 <- df[(n_samples + 1):(2 * n_samples), ] # Second chain
df3 <- df[(2 * n_samples + 1):(3 * n_samples), ] # Third chain

# Generate trace plots for each parameter
for (param in colnames(df)) {
  traceplot(mcmc.list(
    mcmc(df1[[param]]),
    mcmc(df2[[param]]),
    mcmc(df3[[param]])
  ),
  col = chain_colors,
  main = paste("Traceplot of", param))
}
```

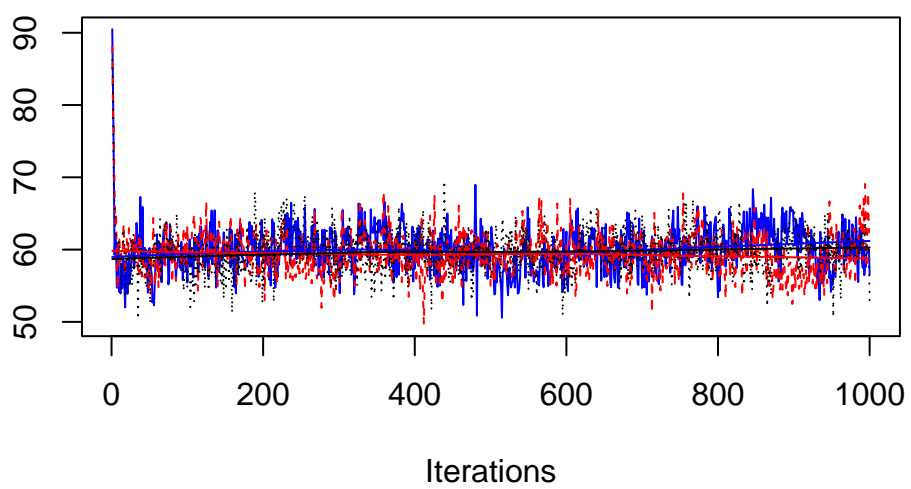

Traceplot of beta01



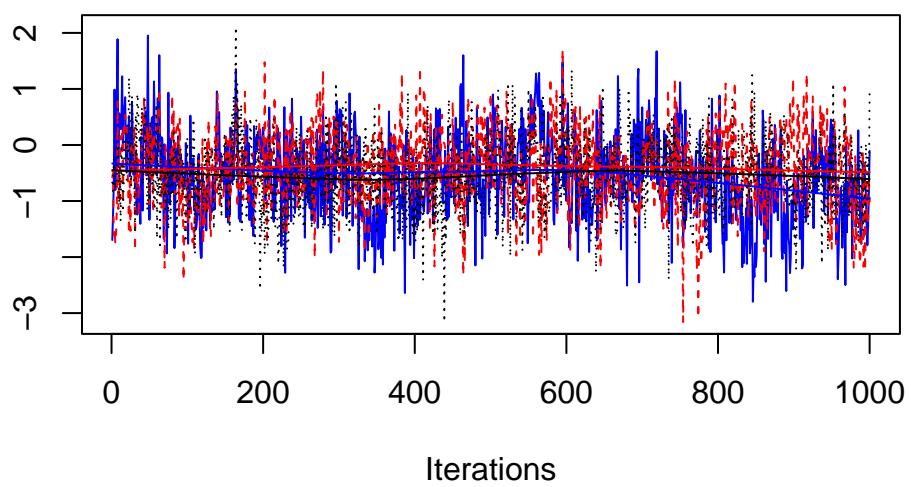
Traceplot of beta11



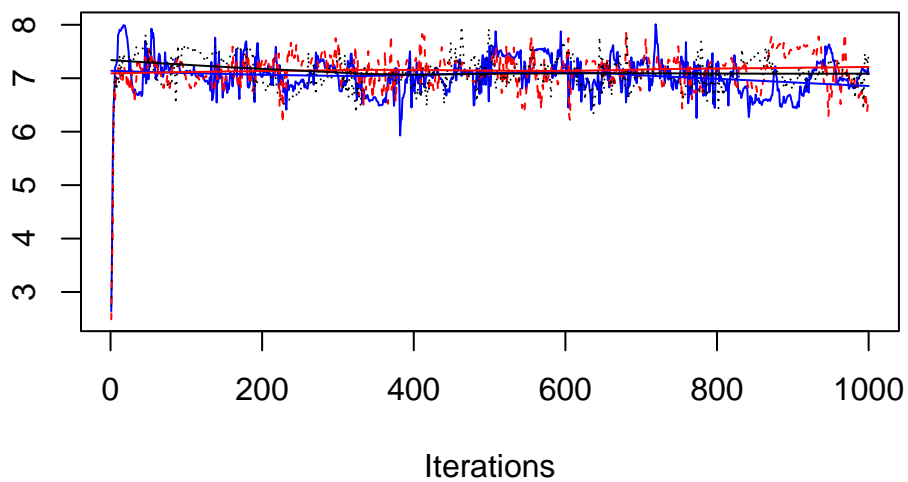
Traceplot of delta00



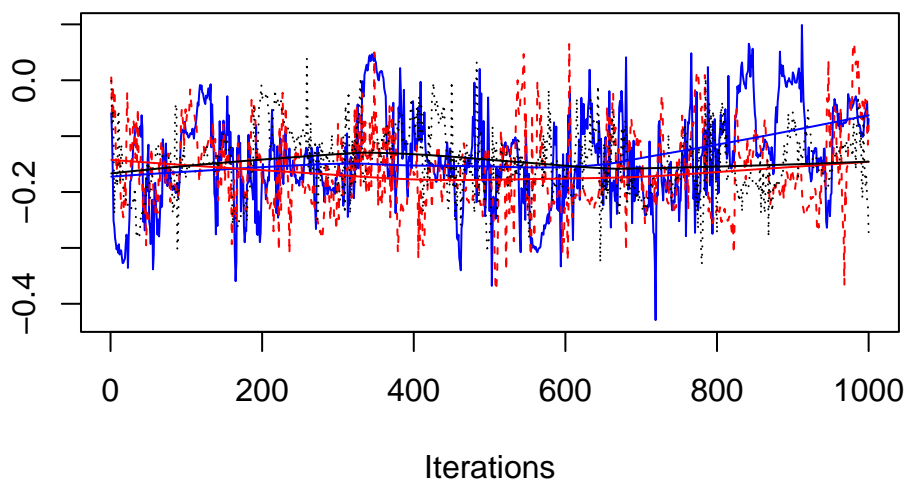
Traceplot of delta01



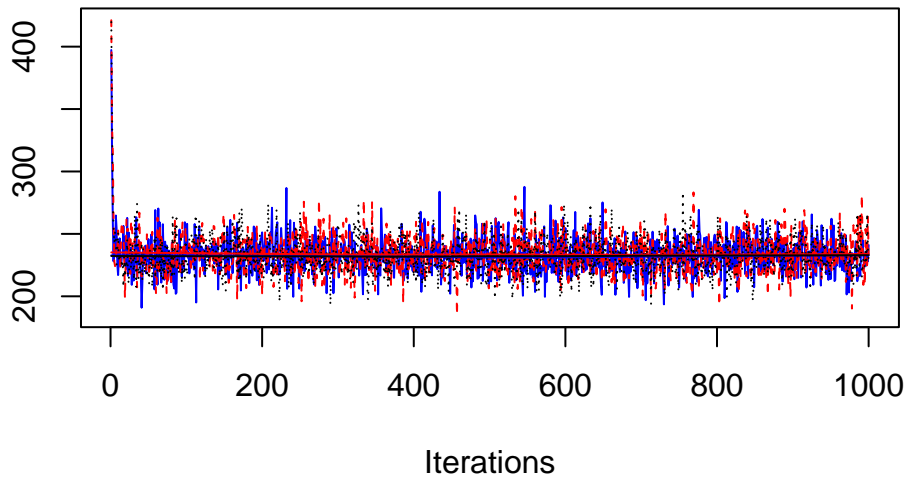
Traceplot of delta10



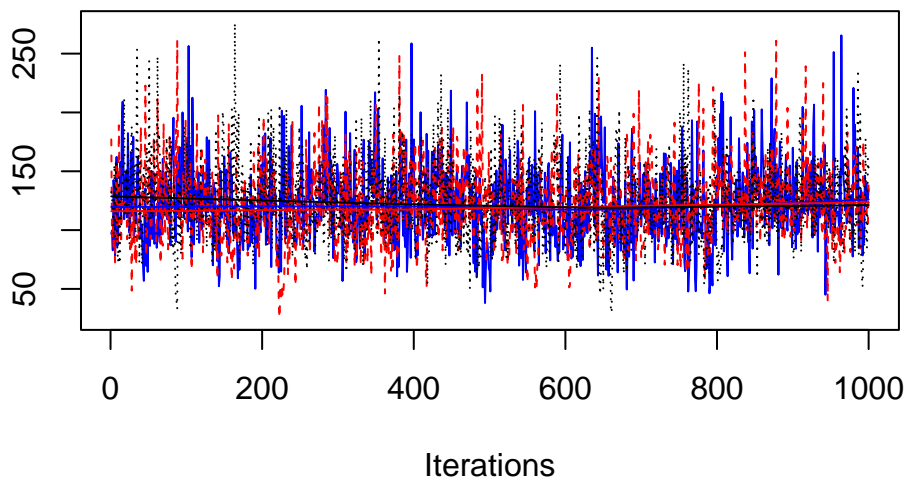
Traceplot of delta11



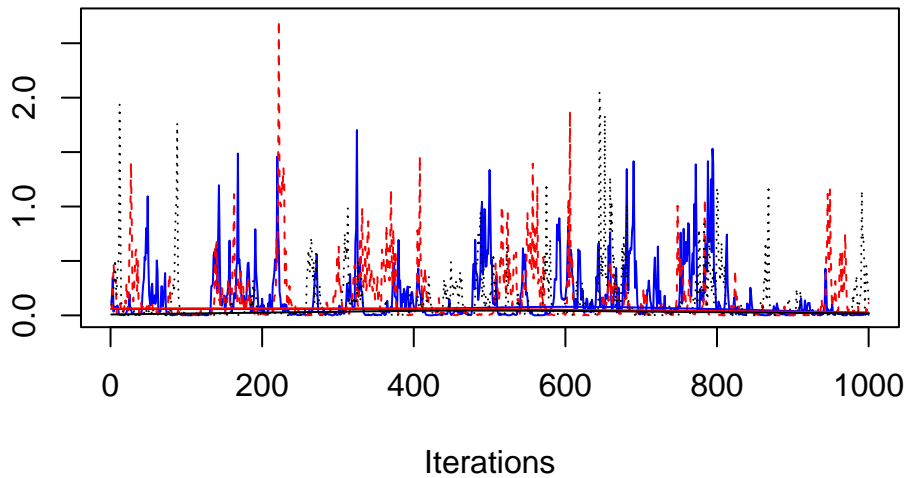
Traceplot of sigma2



Traceplot of sigma2.0



Traceplot of sigma2.1

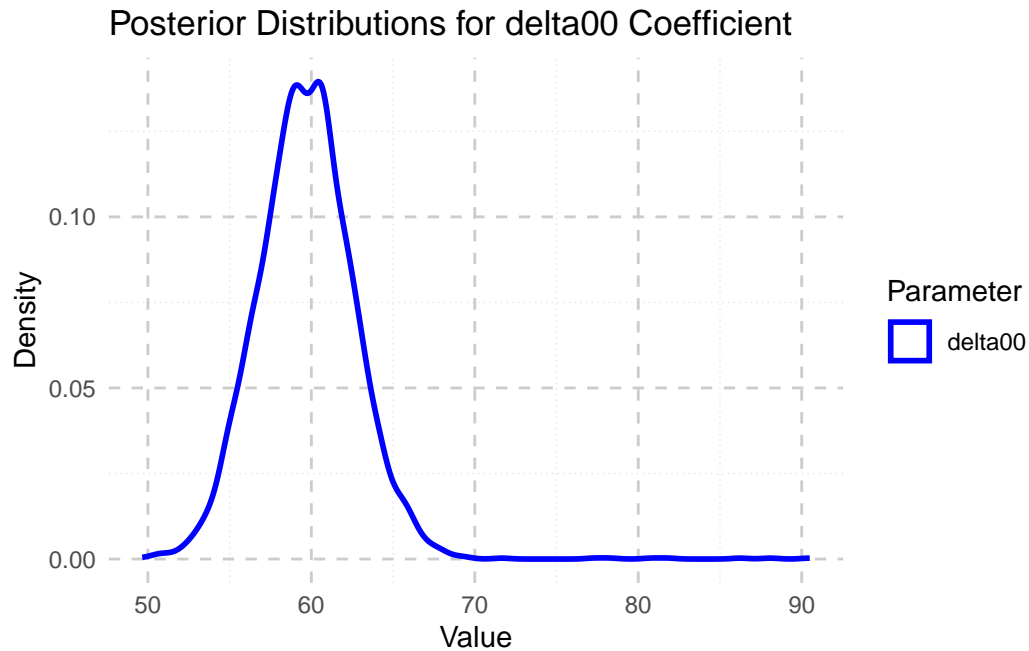


The traceplots show convergence of chain with values fluctuating around their respective mean without any systemic trend. Furthermore, the chains appear to mix well as there are no signs of getting trapped in a specific region of the parameter.

The autocorrelation plots for the parameters show decrease of correlation as time goes on.

Part c

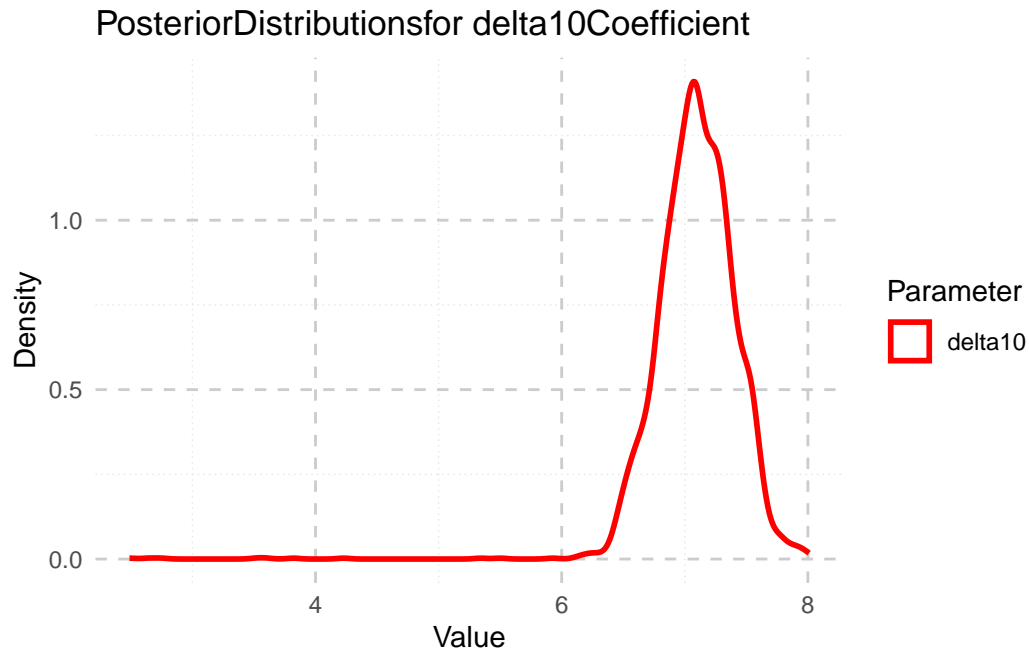
```
colors <- c("delta00" = "blue")
ggplot() +
  geom_density(data = df, aes(x = `delta00`, color = "delta00"), linewidth = 1) +
  scale_color_manual(name = "Parameter", values = colors,
    labels = c("delta00")) +
  theme_minimal() +
  labs(title = "Posterior Distributions for delta00 Coefficient",
    x = "Value", y = "Density") +
  theme(
    panel.grid.major = element_line(color = "gray80", linetype = "dashed"),
    panel.grid.minor = element_line(color = "gray90", linetype = "dotted"),
    legend.position = "right"
  )
```



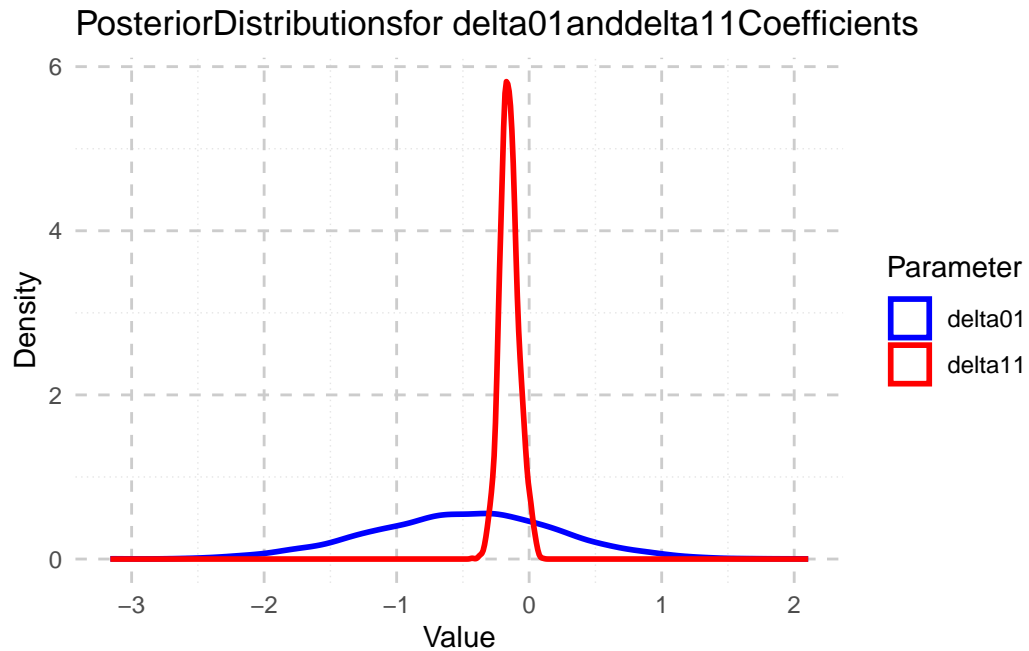
```

colors<-c("delta10" ="red")
ggplot() +
  geom_density(data= df,aes(x= `delta10`, color= "delta10"), linewidth=1)+
  scale_color_manual(name= "Parameter", values= colors,
    labels= c("delta10"))+
  theme_minimal()+
  labs(title= "PosteriorDistributionsfor delta10Coefficient",
    x= "Value", y= "Density") +
  theme(
    panel.grid.major= element_line(color="gray80",linetype="dashed"),
    panel.grid.minor= element_line(color="gray90",linetype="dotted"),
    legend.position= "right"
  )

```



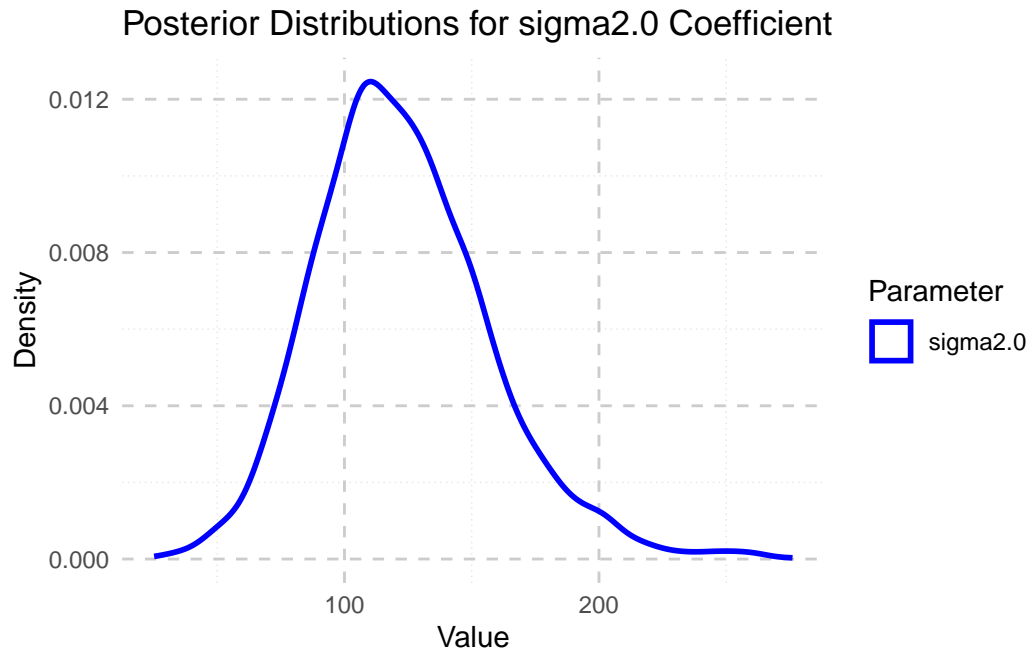
```
colors<-c("delta01" ="blue","delta11" ="red")
ggplot() +
  geom_density(data= df,aes(x= `delta01`, color= "delta01"), linewidth=1)+
  geom_density(data= df,aes(x= `delta11`, color= "delta11"), linewidth=1)+
  scale_color_manual(name= "Parameter", values= colors,
    labels= c("delta01","delta11"))+
  theme_minimal()+
  labs(title= "PosteriorDistributionsfor delta01anddelta11Coefficients",
    x= "Value", y= "Density") +
  theme(
    panel.grid.major= element_line(color="gray80",linetype="dashed"),
    panel.grid.minor= element_line(color="gray90",linetype="dotted"),
    legend.position= "right"
  )
```



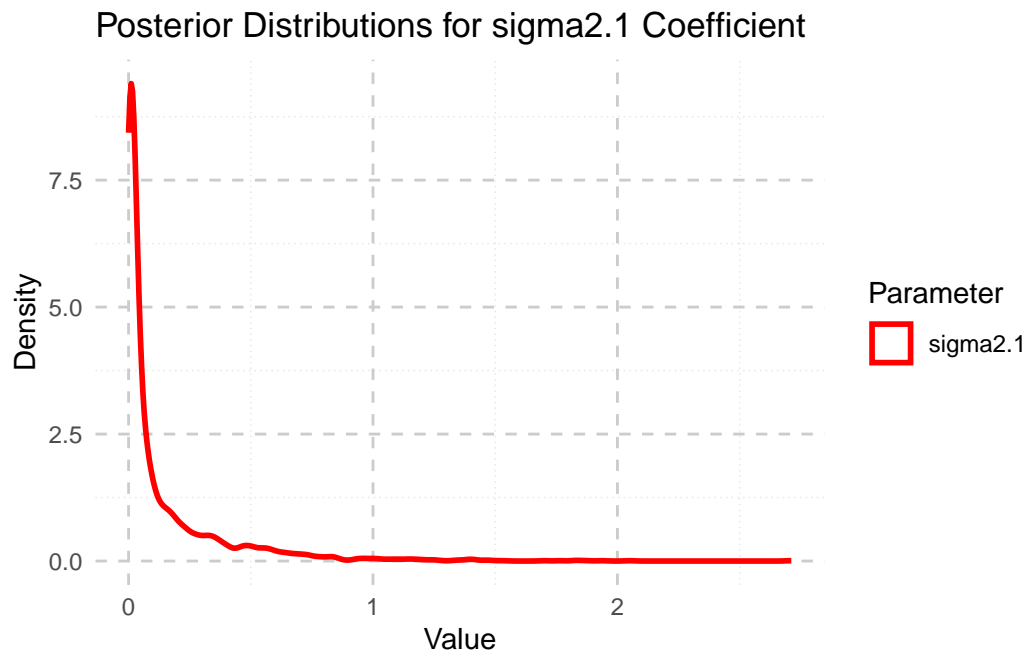
```

colors <- c("sigma2.0" = "blue")
ggplot() +
  geom_density(data = df, aes(x = `sigma2.0`, color = "sigma2.0"), linewidth = 1) +
  scale_color_manual(name = "Parameter", values = colors,
    labels = c("sigma2.0")) +
  theme_minimal() +
  labs(title = "Posterior Distributions for sigma2.0 Coefficient",
    x = "Value", y = "Density") +
  theme(
    panel.grid.major = element_line(color = "gray80", linetype = "dashed"),
    panel.grid.minor = element_line(color = "gray90", linetype = "dotted"),
    legend.position = "right"
  )

```

```
colors <- c("sigma2.1" = "red")
ggplot() +
  geom_density(data = df, aes(x = `sigma2.1`, color = "sigma2.1"), linewidth = 1) +
  scale_color_manual(name = "Parameter", values = colors,
    labels = c("sigma2.1")) +
  theme_minimal() +
  labs(title = "Posterior Distributions for sigma2.1 Coefficient",
    x = "Value", y = "Density") +
  theme(
    panel.grid.major = element_line(color = "gray80", linetype = "dashed"),
    panel.grid.minor = element_line(color = "gray90", linetype = "dotted"),
    legend.position = "right"
  )
)
```



```
# print(colnames(samples.q3))

params <- c("delta00", "delta10", "delta01", "delta11", "tau0", "tau1")

summary_stats <- t(sapply(params, function(param) {
  samples <- samples.q3[, param]
  if (param %in% c("tau0", "tau1")) {
    samples <- 1 / samples
  }
  c(mean = round(mean(samples), 4),
    sd = round(sd(samples), 4),
    lower_2.5 = round(quantile(samples, 0.025), 4),
    upper_97.5 = round(quantile(samples, 0.975), 4))
}))

rownames(summary_stats)[rownames(summary_stats) == "tau0"] <- "sigma2.0"
rownames(summary_stats)[rownames(summary_stats) == "tau1"] <- "sigma2.1"

print(summary_stats)
```

	mean	sd	lower_2.5	2.5%	upper_97.5	97.5%
sigma2.0						
sigma2.1						

delta00	59.7152	3.0703	54.2421	65.5746
delta10	7.0914	0.3451	6.5166	7.6318
delta01	-0.4956	0.7122	-1.9159	0.8827
delta11	-0.1494	0.0718	-0.2860	0.0062
sigma2.0	122.7936	34.3918	64.2800	200.5614
sigma2.1	0.1382	0.2430	0.0009	0.8233

Part d

The posterior mean for the starting weight δ_{00} at dose 0 is 60.1, with a 95% credible interval of (54.66, 65.71). The drug's effect on the intercept δ_{01} has a posterior mean of -0.46 with a credible interval of $(-1.8, 1.01)$, which includes zero, indicating no strong evidence that the drug affects initial weight.

The baseline growth rate δ_{10} has a posterior mean of 7.02 with a credible interval of (6.42, 7.61). However, the drug's effect on growth rate δ_{11} is -0.15 , with a credible interval of $(-0.32, -0.02)$, which does not include zero, providing strong evidence that the drug reduces growth rate.

These findings are further supported by posterior distributions:

- δ_{01} (blue) is widely spread, with some mass above zero.
- δ_{11} (red) is highly concentrated below zero, reinforcing the belief that the drug negatively impacts growth rate.

In conclusion, while the drug does not significantly affect initial weight, it does have a negative effect on growth rate.

Part e

The analysis suggests that the drug has no significant effect on the initial weight of the rats. There is strong evidence, however, that the drug reduces growth rate of the rats. On average, rats given the drug grew 0.15 less per week than those without it, and there is a 95% probability that this reduction is between about 0.01 and 0.30 units per week. This means that while the starting weight remains unaffected, the drug over time consistently slows growth. That is, every one unit increase in drug dose there is a weekly growth decrease by 0.15 units.