

```
library(rjags)
```

```
## Warning: 'rjags' R 4.4.2
```

```
## coda
```

```
## Warning: 'coda' R 4.4.2
```

```
## Linked to JAGS 4.3.1
```

```
## Loaded modules: basemod,bugs
```

```
library(coda)
```

```
#
```

```
file_path <- "D:/Stat431/ObesityDataSet_raw_and_data_synthetic.csv"
```

```
data <- read.csv(file_path, stringsAsFactors = FALSE)
```

```
#map ordered categories to 0..6
```

```
data$NObeyesdad <- as.numeric(factor(data$NObeyesdad,  
                                     levels = sort(unique(data$NObeyesdad)))) - 1
```

```
# make sure the needed columns exist and are numeric/binary as intended
```

```
# FAVC: "yes"/"no" -> 1/0
```

```
if (!is.numeric(data$FAVC)) {  
  data$FAVC <- as.numeric(tolower(trimws(data$FAVC)) == "yes")  
}
```

```
# ordered categorical drinking frequency -> 0..3
```

```
calc_levels <- c("no", "Sometimes", "Frequently", "Always")
```

```
data$CALC <- as.numeric(factor(data$CALC, levels = calc_levels)) - 1
```

```
# physical activity frequency should be numeric; coerce if it came in as character
```

```
if (!is.numeric(data$FAF)) {  
  data$FAF <- as.numeric(data$FAF)  
}
```

```
data$AgeGroup <- cut(  
  data$Age,  
  breaks = c(0, 21, 30, 45, 61),  
  labels = FALSE,  
  include.lowest = TRUE,  
  right = TRUE  
)
```

```
# Drop rows with any NAs in required fields to keep JAGS happy
```

```
needed <- c("NObeyesdad", "AgeGroup", "FAVC", "FAF", "CALC")
```

```
data <- data[stats::complete.cases(data[, needed]), ]
```

```
#JAGS Model
```

```
jags_model <- "  
model {
```

```

# Hyperpriors
mu_0 ~ dnorm(0, 0.01)
tau_0 ~ dgamma(1, 0.01)

# Group-level means for 4 age groups
for (j in 1:4) {
  mu_age[j] ~ dnorm(mu_0, tau_0)
}

# Priors for coefficients
beta_favc ~ dnorm(0, 0.01)
beta_faf ~ dnorm(0, 0.01)
beta_calc ~ dnorm(0, 0.01)

# Residual precision
tau ~ dgamma(1, 0.01)
sigma <- 1 / sqrt(tau)

# Likelihood
for (i in 1:N) {
  mu[i] <- mu_age[AgeGroup[i]] +
    beta_favc * FAVC[i] +
    beta_faf * FAF[i] +
    beta_calc * CALC[i]
  NObeyesdad[i] ~ dnorm(mu[i], tau)
}
}
"

#Data for JAGS
jags_data <- list(
  NObeyesdad = data$NObeyesdad,
  AgeGroup = as.integer(data$AgeGroup),
  FAVC = as.numeric(data$FAVC),
  FAF = as.numeric(data$FAF),
  CALC = as.numeric(data$CALC),
  N = nrow(data)
)

# Initial Values
inits <- list(
  list(mu_0 = 0, tau_0 = 1, beta_favc = 0, beta_faf = 0, beta_calc = 0, tau = 1),
  list(mu_0 = 1, tau_0 = 2, beta_favc = 0.1, beta_faf = 0.1, beta_calc = 0.1, tau = 2)
)

#Run JAGS
jags <- jags.model(
  textConnection(jags_model),
  data = jags_data,
  inits = inits,
  n.chains = 2,
  quiet = TRUE
)

```

```

update(jags, 2000, progress.bar = "none")

samples <- coda.samples(
  jags,
  variable.names = c("mu_0", "mu_age", "beta_favc", "beta_faf", "beta_calc", "sigma"),
  n.iter = 5000,
  progress.bar = "none"
)

print(summary(samples))

```

```

##
## Iterations = 2001:7000
## Thinning interval = 1
## Number of chains = 2
## 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
## beta_calc  0.453173 0.08007 0.0008007      0.0018560
## beta_faf   -0.186449 0.04861 0.0004861      0.0010471
## beta_favc  -0.009855 0.13086 0.0013086      0.0054562
## mu_0        3.242142 0.44017 0.0044017      0.0084175
## mu_age[1]   2.282476 0.15220 0.0015220      0.0064634
## mu_age[2]   3.120374 0.15350 0.0015350      0.0067727
## mu_age[3]   3.493595 0.17234 0.0017234      0.0068251
## mu_age[4]   4.093415 0.49783 0.0049783      0.0085299
## sigma       1.860075 0.02876 0.0002876      0.0002876
##
## 2. Quantiles for each variable:
##
##              2.5%       25%       50%       75%      97.5%
## beta_calc  0.2987  0.39938  0.4520  0.50628  0.61379
## beta_faf   -0.2819 -0.21894 -0.1863 -0.15401 -0.09215
## beta_favc  -0.2704 -0.09837 -0.0111  0.07922  0.24738
## mu_0        2.4145  2.97736  3.2262  3.48622  4.13625
## mu_age[1]   1.9767  2.18075  2.2844  2.38434  2.57637
## mu_age[2]   2.8133  3.01813  3.1204  3.22710  3.41605
## mu_age[3]   3.1530  3.37725  3.4955  3.61203  3.82745
## mu_age[4]   3.1622  3.74927  4.0793  4.42158  5.10120
## sigma       1.8046  1.84055  1.8599  1.87880  1.91860

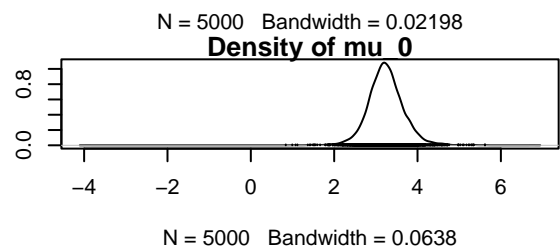
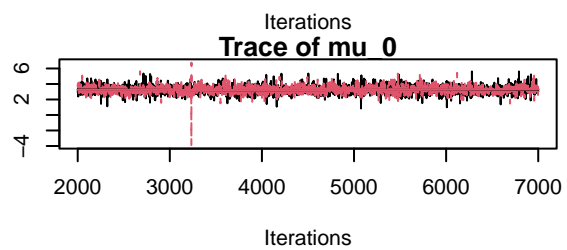
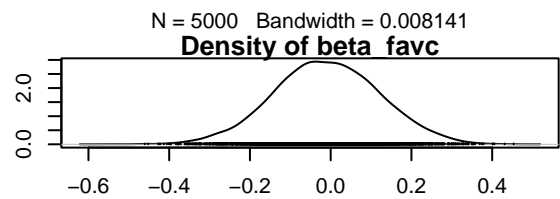
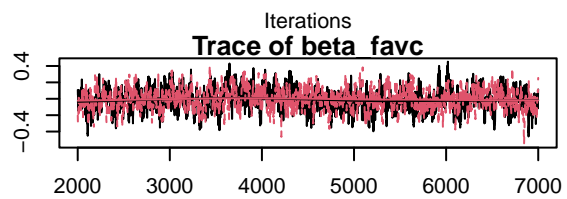
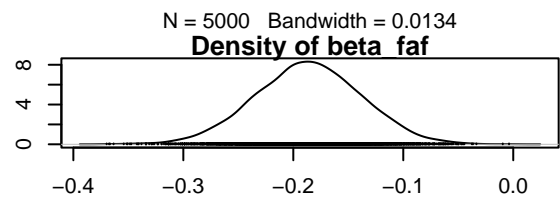
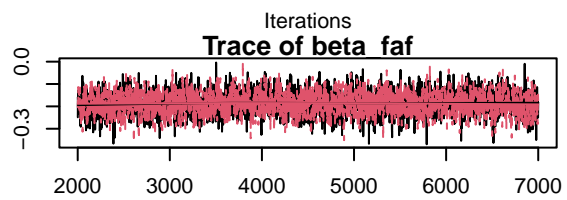
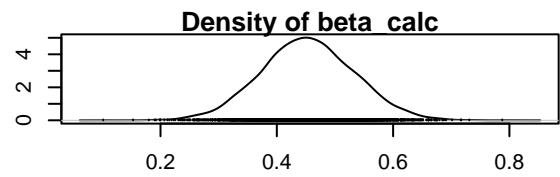
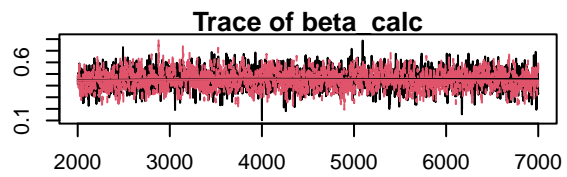
```

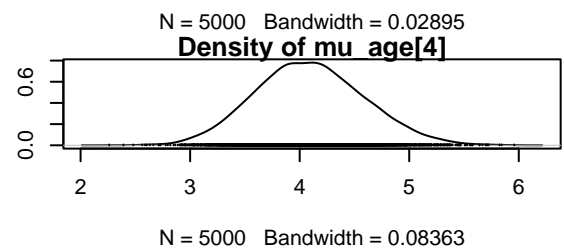
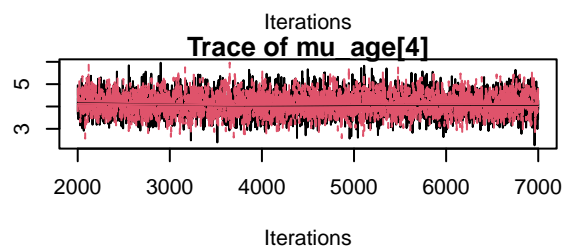
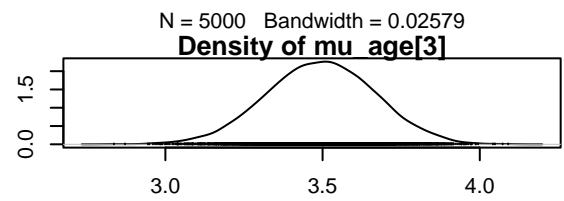
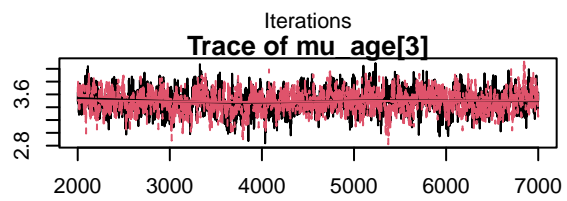
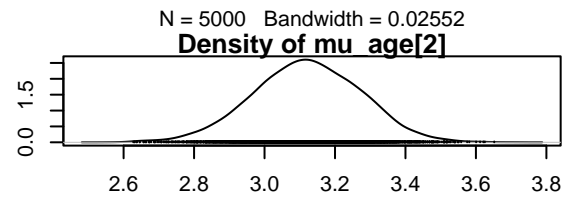
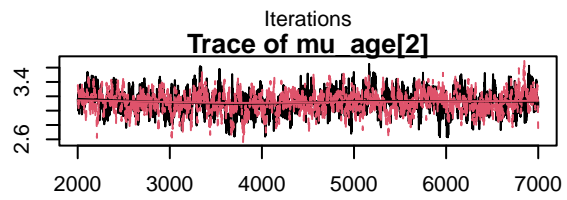
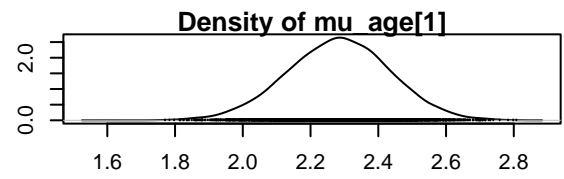
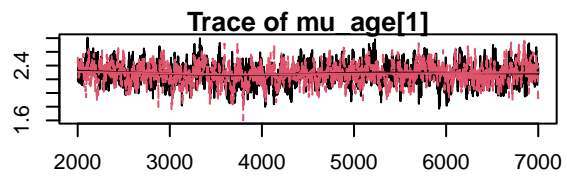
```

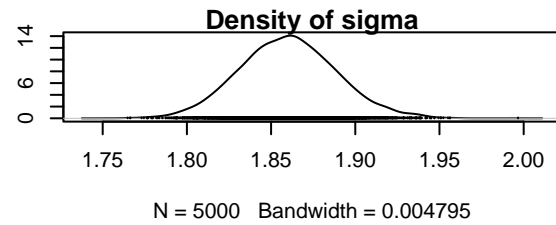
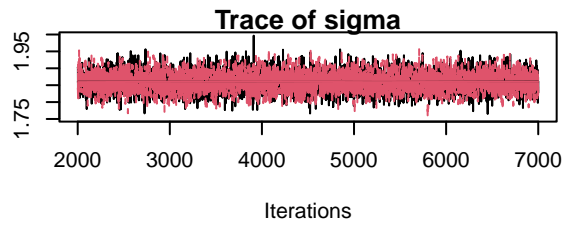
# Ensure margins are small enough even if the device is tight
oldpar <- par(no.readonly = TRUE)
on.exit(par(oldpar), add = TRUE)
par(mar = c(4, 4, 1, 1))

plot(samples)

```







```
gelman.diag(samples, autoburnin = FALSE)
```

```
## Potential scale reduction factors:
##
##          Point est. Upper C.I.
## beta_calc      1      1.01
## beta_faf       1      1.00
## beta_favc      1      1.01
## mu_0           1      1.00
## mu_age[1]      1      1.00
## mu_age[2]      1      1.00
## mu_age[3]      1      1.00
## mu_age[4]      1      1.00
## sigma          1      1.00
##
## Multivariate psrf
##
## 1
```

```
effectiveSize(samples)
```

```
## beta_calc beta_faf beta_favc mu_0 mu_age[1] mu_age[2] mu_age[3]
## 1866.5110 2197.4618 577.6421 2994.6264 562.2971 519.4396 646.1388
## mu_age[4] sigma
## 3450.7150 10000.0000
```

```
#Save Posterior  
graphics.off()  
png("posterior_trace_plot_updated.png", width = 1200, height = 800)  
par(mar = c(4, 4, 1, 1))  
plot(samples)  
dev.off()
```

```
## pdf  
## 2
```

```
cat("Saved: posterior_trace_plot_updated.png\n")
```

```
## Saved: posterior_trace_plot_updated.png
```