Gibbs sampler example (solution)

Math 315, Fall 2019

Do Distracting Colors Influence the Time to Complete a Game?

The Stroop effect demonstrates that automatized behaviors[1] can interfere with other desired behaviors. John Stroop tested the reaction time of college undergraduates in identifying colors. Students took a longer time identifying colors of ink when the ink was used to spell a different color. For example, if the word "red" was printed in blue ink, students took longer to identify the color blue because they automatically read the word "red." Even though students were told only to identify the ink color, the automatized behavior of reading interfered with the task and slowed their reaction time.

Statistics students developed a study to explore the impact of distracters. Specifically, they wanted to determine whether students at their college would perform differently when a distracting color was incorporated into a computerized game. The game challenged people to place an assortment of shaped pegs into the appropriate spaces as quickly as possible. Below are additional details of the study:

- 40 students were randomly selected from the college
- 20 students were randomly assigned to the standard game, the remaining 20 students were assigned to a game with a color distracter
- Subjects saw a picture of the game and had the rules clearly explained to them before they played
- Subjects played the game in the same area with similar background noise
- The research group collected the the time, in seconds, required to complete the game

```
stroop <- read.csv("http://aloy.rbind.io/data/stroop_game.csv")
```

A noninformative two-sample Bayesian model for this situations can be written as

$$
\begin{aligned}
Y_i &\overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2),\ i = 1, \ldots, 20 && \text{(standard group)}\\
Y_j &\overset{\text{iid}}{\sim} \mathcal{N}(\mu + \delta, \sigma^2),\ j = 21, \ldots, 40 && \text{(color group)}\\
\mu &\sim \mathcal{N}(0, 100^2)\\
\delta &\sim \mathcal{N}(0, 100^2)\\
\sigma^2 &\sim \text{InvGamma}(0.01, 0.01)
\end{aligned}
$$

where μ, δ, and σ^2 are assumed to be mutually independent.

1. The full conditional posterior distributions can be shown to be:

$$
\begin{aligned}
\mu|\text{rest} &\sim \mathcal{N}\left(\frac{\tau\sum_{i=1}^{n} Y_i + \tau\sum_{j=n+1}^{n+m}(Y_j - \delta)}{\tau(n+m) + 1/100^2},\ \frac{1}{\tau(n+m) + 1/100^2}\right)\\
\delta|\text{rest} &\sim \mathcal{N}\left(\frac{\tau\sum_{i=n+1}^{n+m}(Y_i - \mu)}{\tau m + 1/100^2},\ \frac{1}{\tau m + 1/100^2}\right)\\
\sigma^2|\text{rest} &\sim \text{InvGamma}\left(0.01 + \frac{n+m}{2},\ 0.01 + \frac{\sum_{i=1}^{n}(Y_i - \mu)^2}{2} + \frac{\sum_{j=n+1}^{n+m}(Y_j - \mu - \delta)^2}{2}\right)
\end{aligned}
$$

where $\tau = 1/\sigma^2$. Write a Gibbs sampler to fit the model above.

```
library(MCMCpack) # for rinvgamma()

# Data
y <- stroop$Time
```

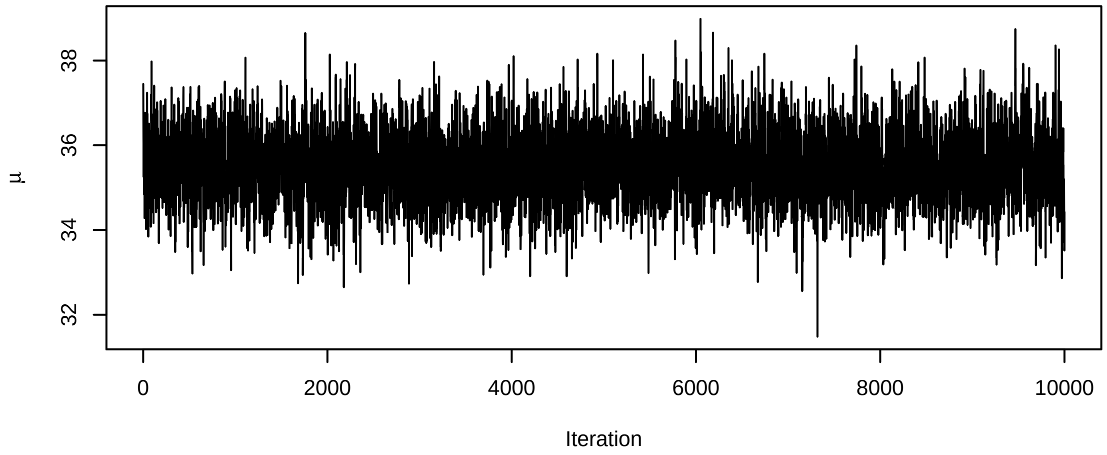
[1] Many psychologists would call this procedural knowledge instead of automatized behavior.

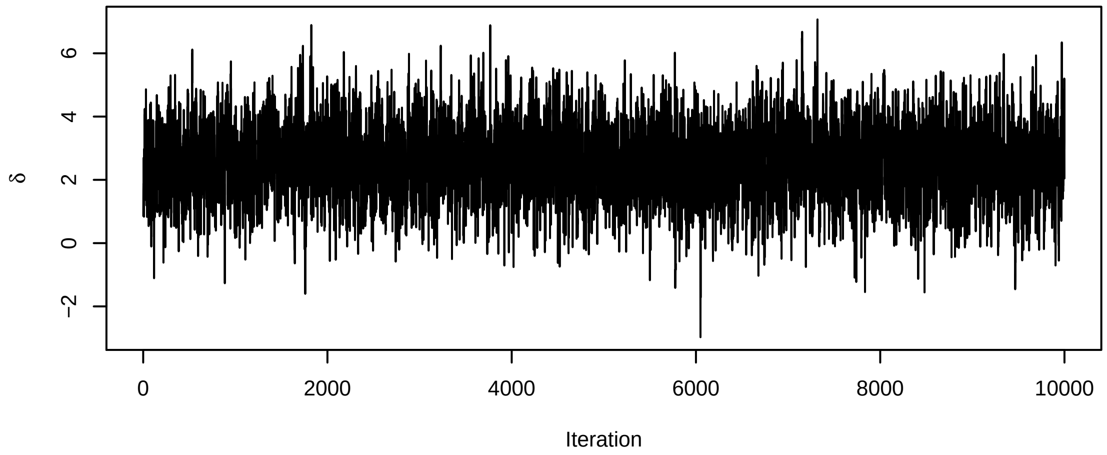
```r
y1 <- y[stroop$Type == "Standard"]
y2 <- y[stroop$Type == "Color"]

# Summary statistics
n <- length(y1)
m <- length(y2)

# Initial parameter values
mu     <- mean(y)
delta  <- 0
sigma2 <- var(y)


# Sampling details and storage
S                    <- 10000
mcmc.draws           <- matrix(nrow = S, ncol = 3)
colnames(mcmc.draws) <- c("mu", "delta", "sig2")

# Running the Gibbs sampler
for(i in 1:S) {
  # Sample mu
  mu.num  <- sum(y1) / sigma2 + sum(y2 - delta) / sigma2
  mu.prec <- (n + m) / sigma2 + 1/100^2
  mu      <- rnorm(1, mean = mu.num / mu.prec, sd = 1 / sqrt(mu.prec))

  # Sample delta
  delta.num  <- sum(y2 - mu) / sigma2
  delta.prec <- m / sigma2 + 1/100^2
  delta      <- rnorm(1, mean = delta.num / delta.prec, sd = 1 / sqrt(delta.prec))

  # sample sigma2
  A <- 0.01 + (n + m) / 2
  B <- 0.01 + sum((y1 - mu)^2) / 2 + sum((y2 - mu - delta)^2) / 2
  sigma2 <- rinvgamma(1, A, B)

  # store the draws
  mcmc.draws[i,] <- c(mu, delta, sigma2)
}
```

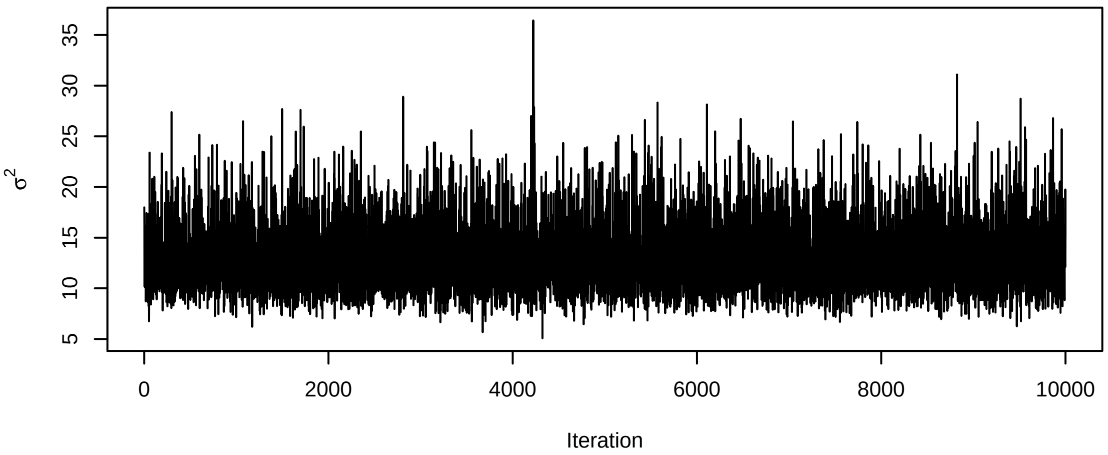
2. Examine the convergence of your chain for each parameter. Did the chain converge? About how long did it take?

   All three chains clearly converge quickly (within 100 samples).

```r
par(mfrow = c(3, 1))
plot(1:S, mcmc.draws[,"mu"], type = "l", ylab = expression(mu), xlab = "Iteration")
plot(1:S, mcmc.draws[,"delta"], type = "l", ylab = expression(delta), xlab = "Iteration")
plot(1:S, mcmc.draws[,"sig2"], type = "l", ylab = expression(sigma^2), xlab = "Iteration")
```



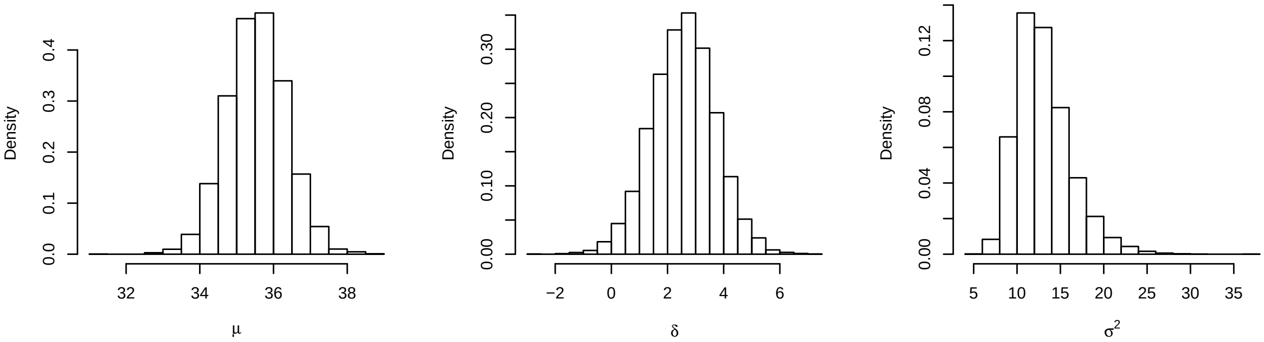




3. Plot the marginal posterior density for each parameter after discarding the burn-in period.

```r
# The chains converged quickly, so let's only toss out 100
mcmc.draws <- mcmc.draws[-c(1:100),]

par(mfrow = c(1, 3))
```

```r
hist( mcmc.draws[,"mu"], freq = FALSE, xlab = expression(mu), main = NULL)
hist( mcmc.draws[,"delta"], freq = FALSE, xlab = expression(delta), main = NULL)
hist( mcmc.draws[,"sig2"], freq = FALSE, xlab = expression(sigma^2), main = NULL)
```



4. Use your results to determine whether a color distracter slowed student reaction time.

   You could either use a posterior CI or the posterior probability of the null/alternative hypothesis to answer the research question.

```r
# CI approach
mean(mcmc.draws[,"delta"])
```

```
## [1] 2.569593
```

```r
quantile(mcmc.draws[,"delta"], probs = c(0.05, 0.95))
```

```
##        5%       95%
## 0.6849986 4.3991460
```

```r
# Probability of Ha: delta > 0
mean(mcmc.draws[,"delta"] > 0)
```

```
## [1] 0.9864646
```