

Computer Lab 1

Ravinder Reddy Atla

4/6/2021

Daniel Bernoulli

```
# Data distribution parameters
n <- 24
s <- 8
f <- n-s
# Prior distribution parameters
alpha_0 <- 3
beta_0 <- 3
```

a. Generating random numbers from posterior distribution and verifying graphically

```
alpha_n <- alpha_0 + s
beta_n <- beta_0 + f

posterior_mean <- alpha_n/(alpha_n + beta_n)
posterior_variance <- (alpha_n * beta_n)/((alpha_n + beta_n)^2 *
                                           (alpha_n + beta_n + 1))
posterior_sd <- sqrt(posterior_variance)

posteriorDrawParams <- function(num_draws,alpha_n,beta_n){
  set.seed(1234)
  theta_n <- rbeta(num_draws, shape1 = alpha_n, shape2 = beta_n)
  sample_mean <- mean(theta_n)
  sample_sd <- sd(theta_n)
  return(c(sample_mean,sample_sd))
}

num_samples <- 1
mu_vector <- c()
sample_mu <- 0.3
sample_dev <- 1
#plot(num_samples,sample_mu)

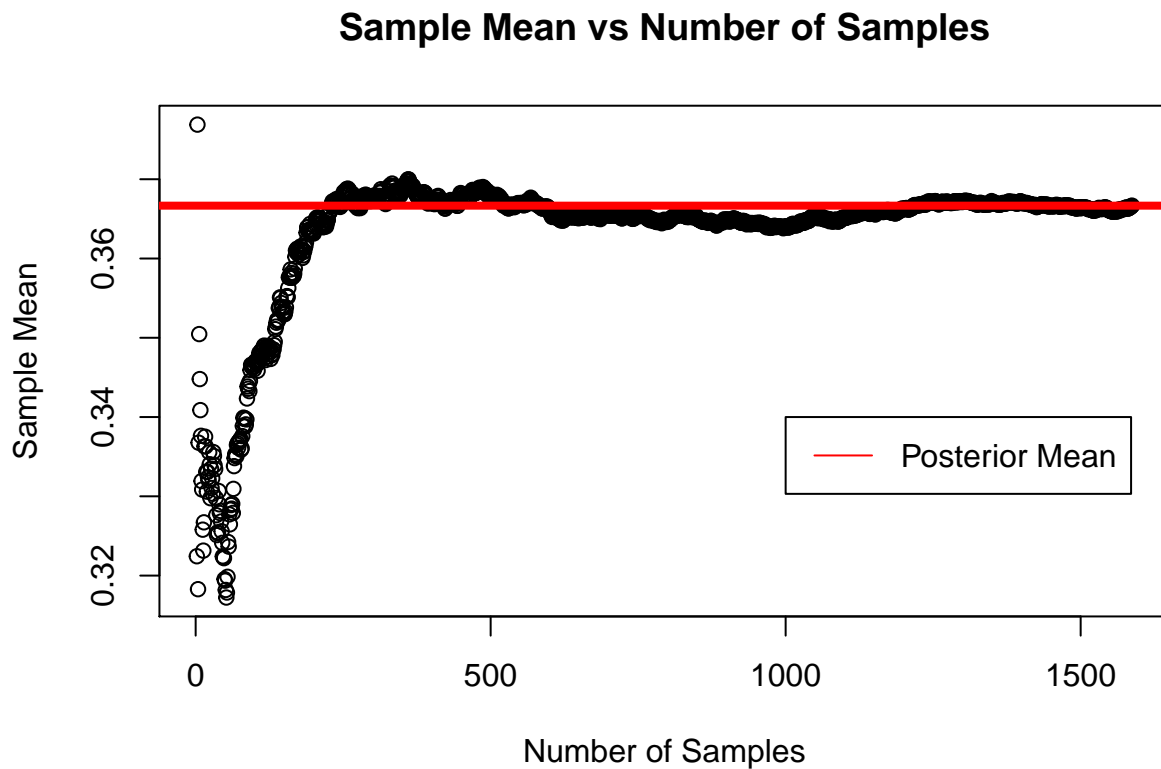
while(abs(posterior_mean - sample_mu) > 1e-7){
  num_samples <- num_samples + 1
  sample_mu <- posteriorDrawParams(num_samples,alpha_n,beta_n)[1]
  mu_vector <- c(mu_vector,sample_mu)
  #points(num_samples,sample_mu)
}

plot(c(2:num_samples),mu_vector,
     xlab = 'Number of Samples',ylab = 'Sample Mean',
```

```

main = 'Sample Mean vs Number of Samples'
legend(1000,0.34,legend = c('Posterior Mean'),col = c('red'),lty=1)
abline(h = posterior_mean,lwd=4,col = 'red')

```



```

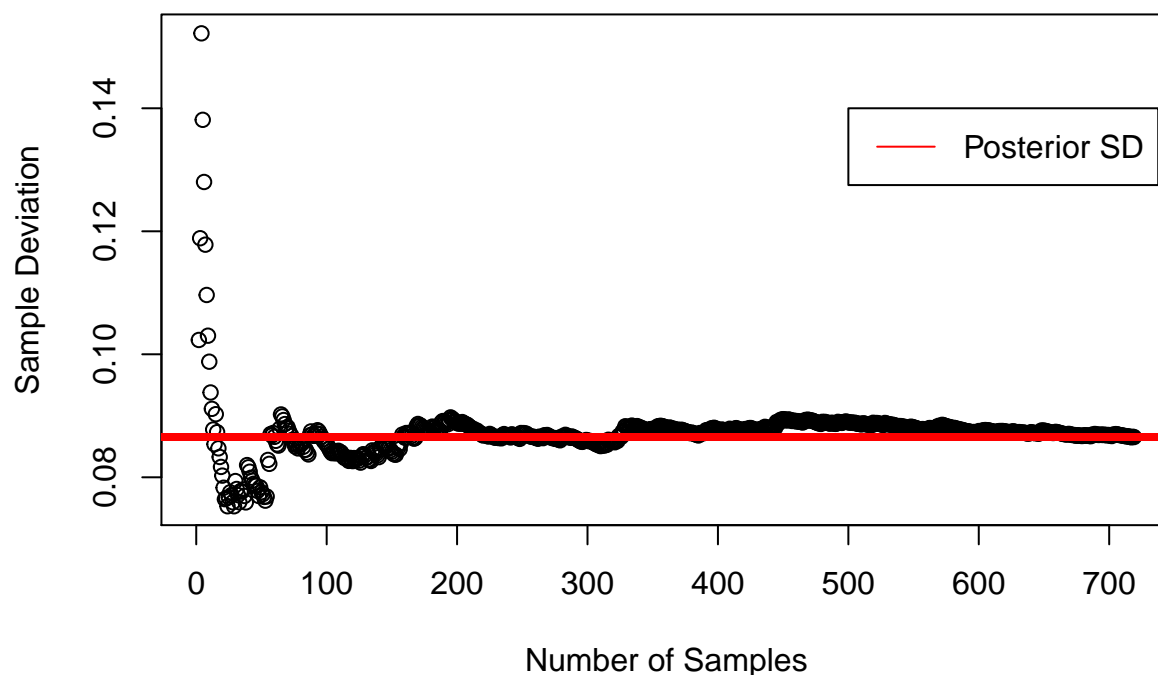
num_samples <- 1
sd_vector <- c()
sample_dev <- 1

while(abs(posterior_sd - sample_dev) > 1e-5){
  num_samples <- num_samples + 1
  sample_dev <- posteriorDrawParams(num_samples,alpha_n,beta_n)[2]
  sd_vector <- c(sd_vector,sample_dev)
}

plot(c(2:num_samples),sd_vector,
     xlab = 'Number of Samples',ylab = 'Sample Deviation',
     main = 'Sample SD vs Number of Samples')
legend(500,0.14,legend = c('Posterior SD'),col = c('red'),lty=1)
abline(h = posterior_sd,lwd=4,col = 'red')

```

Sample SD vs Number of Samples



b.

```
posteriorDraw <- function(num_draws,alpha_n,beta_n){
  set.seed(1234)
  theta_n <- rbeta(num_draws, shape1 = alpha_n, shape2 = beta_n)
  sample_mean <- mean(theta_n)
  sample_sd <- sd(theta_n)
  return(theta_n)
}

# P(theta > 0.4|y) from sampled data
simulated_data <- posteriorDraw(num_draws = 10000,alpha_n,beta_n)
posterior_prob <- length(simulated_data[simulated_data > 0.4])/length(simulated_data)
print(posterior_prob)
```

```
## [1] 0.3473
```

```
# Actual value of P(theta > 0.4|y)
x <- 0.4
actual_prob <- pbeta(x,alpha_n,beta_n,lower.tail = FALSE)
print(actual_prob)
```

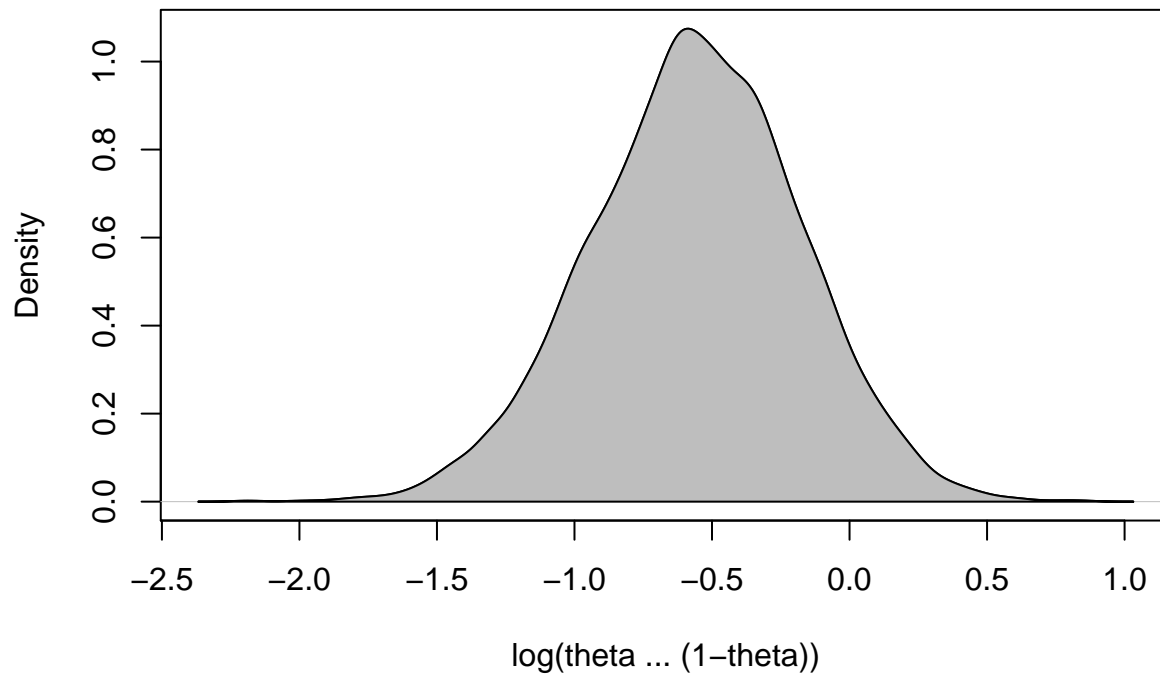
```
## [1] 0.3426654
```

c.

```
theta_post <- posteriorDraw(num_draws = 10000,alpha_n,beta_n)
phi <- log(theta_post) - log(1 - theta_post)
plot(density(phi),
     main = 'Posterior distribution of log odds of theta',
```

```
xlab = 'log(theta / (1-theta))')
polygon(density(phi), col = 'grey')
```

Posterior distribution of log odds of theta



Log-normal distribution and Gini coefficient

a.

```
y <- c(38, 20, 49, 58, 31, 70, 18, 56, 25, 78)
mu <- 3.8
n <- 10
tau_sq <- sum((log(y)-mu)^2)/n

rinvchisquare <- function(num_draws, n, tau_sq){
  x <- rchisq(num_draws, df = n-1)
  x_inv <- ((n-1)*tau_sq)/x
  return(x_inv)
}
```

```
post_sample <- rinvchisquare(10000, n, tau_sq)
```